UNIVERSITY OF WEST ATTICA

SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROGRAM OF DOCTORAL STUDIES

PhD THESIS

Development of optimization and data-driven model predictive control methods using computational intelligence techniques: Design and applications with emphasis on the economic operation of engineering systems

Myron Papadimitrakis

ATHENS-EGALEO

September 2023

The implementation of the doctoral thesis was co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the Act "Enhancing Human Resources Research Potential by undertaking a Doctoral Research" Sub-action 2: "IKY Scholarship Programme for PhD candidates in the Greek Universities"

Ευρωπαϊκή Ένωση
European Social Fund

Operational Programme
Human Resources Development,
Education and Lifelong Learning
Co-financed by Greece and the European Union

ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΠΡΟΓΡΑΜΜΑ ΔΙΔΑΚΤΟΡΙΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ανάπτυξη μεθόδων βελτιστοποίησης και βασισμένου-σε-δεδομένα προβλεπτικού αυτομάτου ελέγχου με χρήση τεχνικών υπολογιστικής νοημοσύνης: Σχεδιασμός και εφαρμογές με έμφαση στην οικονομική λειτουργία συστημάτων μηχανικής

Μύρων Παπαδημητράκης

ΑΘΗΝΑ - ΑΙΓΑΛΕΩ

Σεπτέμβριος 2023

Η υλοποίηση της διδακτορικής διατριβής συγχρηματοδοτήθηκε από την Ελλάδα και την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο-ESF) μέσω του Επιχειρησιακού Προγράμματος "Ανάπτυξη Ανθρώπινου Δυναμικού, Εκπαίδευση και Δια Βίου Μάθηση" στο πλαίσιο της Πράξης "Ενίσχυση του Ερευνητικού Δυναμικού του Ανθρώπινου Δυναμικού με την ανάληψη διδακτορικής έρευνας" Υποδράση 2: "Πρόγραμμα Υποτροφιών ΙΚΥ για υποψήφιους διδάκτορες στα Ελληνικά Πανεπιστήμια

Ευρωπαϊκή Ένωση
European Social Fund

Operational Programme
Human Resources Development,
Education and Lifelong Learning
Co-financed by Greece and the European Union

ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

PhD THESIS

Development of optimization and data-driven model predictive control methods using computational intelligence techniques: Design and Applications with emphasis on the economic operation of engineering systems

Myron Papadimitrakis

SUPERVISOR: Alex Alexandridis, Professor UniWA

THREE-MEMBER ADVISORY COMMITTEE:

Alex Alexandridis, Professor UniWA
Greg Koulouras, Associate Professor UniWA
Elias Zois, Associate Professor UniWA

SEVEN-MEMBER EXAMINATION COMMITTEE

Digitally signed by
Alexandros
Alexandridis
Date: 2023.10.25
11:40:22 +03'00'

Alex Alexandridis,
Professor UniWA

Digitally signed by Grigorios Koulouras
DN: c=GR, l=Athens, o=University of West Attica, ou=Class A - Private Key created and stored in hardware CSP, sn=Koulouras, givenName=Grigorios, serialNumber=6090016933, cn=Grigorios Koulouras, email=gregkoul@uniwa.gr
Date: 2023.10.25 01:45:29 +03'00'

Greg Koulouras,
Associate Professor UniWA

Digitally signed
by Ilias Zois
Date:
2023.10.25
01:24:50 +03'00'

Elias Zois,
Associate Professor UniWA

Digitally signed by
Konstantinos
Psomopoulos
Date: 2023.10.25
11:10:23 +03'00'

Konstantinos Psomopoulos,
Professor UniWA

Haralambos Sarimveis,
Professor, National Technical University of Athens

Kyriakos Vamvoudakis,
Professor, Georgia Institute of Technology

P. Patrinos

Panos Patrinos,
Associate Professor, KU Leuven

Examination Date 12/10/2023

Διδακτορική Διατριβή

Ανάπτυξη μεθόδων βελτιστοποίησης και βασισμένου-σε-δεδομένα προβλεπτικού αυτομάτου ελέγχου με χρήση τεχνικών υπολογιστικής νοημοσύνης: Σχεδιασμός και εφαρμογές με έμφαση στην οικονομική λειτουργία συστημάτων μηχανικής

Μύρων Παπαδημητράκης

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Αλέξανδρος Αλεξανδρίδης, Καθηγητής ΠΑΔΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:
Αλέξανδρος Αλεξανδρίδης, Καθηγητής ΠΑΔΑ
Γρηγόρης Κουλούρας, Αναπληρωτής Καθηγητής ΠΑΔΑ
Ηλίας Ζώης, Αναπληρωτής Καθηγητής ΠΑΔΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Αλέξανδρος Αλεξανδρίδης,
Καθηγητής ΠΑΔΑ

Γρηγόρης Κουλούρας,
Αναπληρωτής Καθηγητής ΠΑΔΑ

Ηλίας Ζώης,
Αναπληρωτής Καθηγητής ΠΑΔΑ

Κωνσταντίνος Ψωμόπουλος,
Καθηγητής ΠΑΔΑ

Χαράλαμπος Σαρίμβεης,
Καθηγητής, Εθνικό Μετσόβειο Πολυτεχνείο

Κυριάκος Βαμβουδάκης,
Καθηγητής, Georgia Institute of Technology

Πάνος Πατρινός,
Αναπληρωτής Καθηγητής, KU Leuven

Ημερομηνία Εξέτασης 12/10/2023

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

Ο/η κάτωθι υπογεγραμμένος Μύρων Παπαδημητράκης του Εμμανουήλ, υποψήφιος διδάκτορας του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:
«Είμαι συγγραφέας και δικαιούχος των πνευματικών δικαιωμάτων επί της διατριβής και δεν προσβάλω τα πνευματικά δικαιώματα τρίτων. Για τη συγγραφή της διδακτορικής μου διατριβής δεν χρησιμοποίησα ολόκληρο ή μέρος έργου άλλου δημιουργού ή τις ιδέες και αντιλήψεις άλλου δημιουργού χωρίς να γίνεται αναφορά στην πηγή προέλευσης (βιβλίο, άρθρο από εφημερίδα ή περιοδικό, ιστοσελίδα κ.λπ.). Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών
Μύρων Παπαδημητράκης

ΠΕΡΙΛΗΨΗ

Η παρούσα διατριβή ασχολείται με τη δημιουργία συστημάτων ελέγχου με προβλεπτικά μοντέλα (Model predictive control - MPC) βάσει δεδομένων και μεθόδων βελτιστοποίησης με χρήση εργαλείων υπολογιστικής νοημοσύνης (computational intelligence - CI) και μηχανικής μάθησης (machine learning - ML). Λαμβάνονται υπόψη τόσο οι θεωρητικές όσο και οι πρακτικές πτυχές του MPC με βάση την υπολογιστική νοημοσύνη καθώς και της μεταευρετικής βελτιστοποίησης, και παρουσιάζονται τα οικονομικά πλεονεκτήματα των προτεινόμενων αλγορίθμων σε σχέση με τη βελτιστοποίηση & τον προβλεπτικό έλεγχο ενός ποικίλου φάσματος εφαρμογών μηχανικής.

Πρώτον, όσον αφορά τη μεταευρετική βελτιστοποίηση, ένας σημαντικός στόχος της παρούσας διατριβής είναι η αντιμετώπιση προβλημάτων υψηλής διαστατικότητας, μη κυρτών προβλημάτων με αποδεκτή ακρίβεια επίλυσης. Για το λόγο αυτό, επινοείται ένας συνεργατικός αλγόριθμος σμήνους σωματιδίων, ικανός να χρησιμοποιεί συνεργατικά σμήνη σωματιδίων σε ομαδοποιημένες μεταβλητές σχεδιασμού. Η ομαδοποίηση πραγματοποιείται με την εφαρμογή ενός αλγορίθμου εντοπισμού κοινότητας στον πίνακα ευαισθησίας του εξεταζόμενου συστήματος, εντοπίζοντας έτσι μεταβλητές σχεδιασμού που είναι δομικά ή τοπολογικά αλληλένδετες. Η προτεινόμενη μέθοδος προσομοιώνεται σε ένα δοκιμαστικό σύστημα της IEEE και, σε συνδυασμό με ένα μοντέλο πρόβλεψης φορτίου με μηχανική μάθηση που αναπτύσσεται επίσης στην παρούσα διατριβή, συνθέτει μια αποτελεσματική πρόταση για αποδοτικό & οικονομικό έλεγχο έξυπνων δικτύων.

Δεύτερον, σχεδιάζεται ένας μη γραμμικός ελεγκτής MPC με χρήση δεδομένων που βασίζεται σε νευρωνικά δίκτυα συναρτήσεων ακτινικής βάσης για παρακολούθηση τροχιάς. Η τυπική απόδοση του MPC εξαρτάται σε μεγάλο βαθμό από την ποιότητα του μοντέλου πρόβλεψης- αν αυτό είναι ανακριβές, τότε οι κινήσεις ελέγχου που προκύπτουν από τη λύση του προβλήματος βέλτιστου ελέγχου θα είναι μη βέλτιστες για το πραγματικό σύστημα. Αυτό σημαίνει ότι ένα γραμμικοποιημένο μοντέλο ενός συστήματος υψηλής διαστατικότητας με σημαντικές μη γραμμικότητες θα είναι ακατάλληλο για χρήση στο πλαίσιο του MPC, ενώ η αντίστοιχη ολοκληρωμένη μορφή των διαφορικών εξισώσεων του κρίνεται υπολογιστικά δαπανηρή. Μάλιστα, σε ορισμένες περιπτώσεις, ένα τέτοιο φυσικό μοντέλο διαφορικών εξισώσεων μπορεί να είναι εξαιρετικά δύσκολο να δημιουργηθεί για ορισμένες περιπτώσεις, επιβάλλοντας μια προσέγγιση βάσει δεδομένων. Ως εκ τούτου, η παρούσα διατριβή προτείνει τη χρήση ενός μοντέλου πρόβλεψης MPC με δίκτυα συναρτήσεων ακτινικής βάσης όπου είναι απαραίτητο, χρησιμοποιώντας καταγεγραμμένα δεδομένα του συστήματος. Η ικανότητα του προτεινόμενου σχήματος MPC στο χειρισμό των δύο προαναφερθέντων ζητημάτων μοντελοποίησης παρουσιάζεται για την περίπτωση ενός συστήματος ενεργής ανάρτησης με υψηλή διαστατικότητα, καθώς και για τη δημιουργία μοντέλων κινούμενων εμποδίων βάσει δεδομένων για τον έλεγχο πλοήγησης πλοίων για την αποφυγή σύγκρουσης με χρήση MPC.

Ως φυσική συνέχεια της εργασίας σχετικά με το MPC παρακολούθησης τροχιάς, η τρίτη συνεισφορά αυτής της διατριβής είναι η δημιουργία ενός σχήματος οικονομικού MPC βάσει δεδομένων για τον αποτελεσματικό και οικονομικό έλεγχο ενός συστήματος πρόωσης πλοίου. Η συγκεκριμένη επιλογή της μελέτης περίπτωσης είναι ιδιαίτερα αιτιολογημένη, καθώς πρόκειται για ένα αντικείμενο υψηλής οικονομικής σημασίας για τον ναυτιλιακό τομέα της ελληνικής οικονομίας. Αρχικά, κατασκευάζεται ένας σταθεροποιητικός νόμος ελέγχου EMPC για το πρόβλημα της οικονομικής πρόωσης πλοίου και συγκρίνεται με το κλασσικό MPC τύπου παρακολούθησης τροχιάς, επιβεβαιώνοντας μια σημαντική διαφορά στην αποδοτικότητα καυσίμου. Χρησιμεύοντας ως αποδεικτικό σκαλοπάτι, τα αποτελέσματα αυτά εμπνέουν την ανάπτυξη ενός EMPC ελεγκτή βασισμένου σε δεδομένα για την πρόωση πλοίων που βασίζεται στην ενισχυτική μάθηση. Ο αλγόριθμος μάθησης που αναπτύσσεται & εφαρμόζεται είναι σε θέση να χειριστεί τις δομικές αποκλίσεις μοντελοποίησης μεταξύ πραγματικού συστήματος και μοντέλου, επιτυγχάνοντας έτσι υψηλότερες επιδόσεις κλειστού βρόχου και, τελικά, απτά οικονομικά οφέλη.

Τέλος, προκειμένου να αξιοποιηθούν τόσο τα αποτελέσματα του MPC για την παρακολούθηση τροχιάς & της αποφυγής σύγκρουσης όσο και τα αποτελέσματα του EMPC για την οικονομική πρόωση των πλοίων, προτείνεται ένας νόμος ελέγχου για τον έλεγχο πλοήγησης και οικονομικής πρόωσης των πλοίων βάσει δεδομένων και τίθενται τα θεωρητικά θεμέλια για περαιτέρω έρευνα & ανάπτυξη. Τέλος, η γνώμη του συγγραφέα είναι ότι οι εργασίες που παρουσιάζονται στην παρούσα διατριβή μπορούν να επεκταθούν και σε άλλους τομείς της μηχανικής & πρακτικές εφαρμογές.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα αυτομάτου ελέγχου & Υπολογιστική νοημοσύνη

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: έλεγχος βάσει δεδομένων, οικονομικός έλεγχος με προβλεπτικά μοντέλα, δίκτυα ακτινικών συναρτήσεων βάσης, υπολογιστική νοημοσύνη, μεταευρετική βελτιστοποίηση, βελτιστοποίηση σμήνους σωματιδίων, έξυπνα δίκτυα, ενεργή ανάρτηση, έλεγχος πρόωσης πλοίων, έλεγχος πλοήγησης πλοίων

ABSTRACT

This thesis addresses the creation of data-driven model predictive control (MPC) schemes and optimization methods utilizing computational intelligence (CI) & machine learning (ML) tools. Both theoretical and practical aspects of CI-based MPC as well as metaheuristic optimization are taken into account, and the economic merits of the proposed algorithms are showcased over the optimization & predictive control of a diverse range of engineering applications.

First, regarding metaheuristic optimization, a significant objective of this thesis is to address high-dimensional, non-convex problems with reasonable solution accuracy. For this reason, a cooperative particle swarm algorithm is devised, capable of using cooperative particle sets on grouped design variables. The grouping occurs by applying a community-detection algorithm over the sensitivity matrix of the system at hand, thus identifying design variables that are structurally or topologically interrelated. The proposed method is tested on an IEEE benchmark system, and, together with a machine-learning ensemble load prediction model that is also developed in this thesis, an effective proposition for efficient & economic smart grid dispatch is made.

Second, a data-driven tracking nonlinear model predictive controller is devised based on radial basis function neural networks. Standard MPC performance heavily relies on the quality of the prediction model; if it is inaccurate, then the control actions yielded by the solution of the optimal control problem will be suboptimal for the real plant. This means that a linearized model of a high-dimensional system with significant nonlinearities will be unfit for usage within MPC, while its respective ODE-integrated form will be too computationally expensive. Such a first-principles ODE model may be extremely hard to yield for some cases, mandating a data-driven approach. Therefore, this thesis proposes complementing an MPC prediction model with radial basis function networks whenever necessary, using recorded plant data. The ability of the proposed MPC scheme in handling the two aforementioned modelling drawbacks is showcased for the case of a high-dimensional active suspension plant, as well as for the data-driven vessel trajectory inference for collision avoidance using MPC.

As a natural continuation of the work on tracking MPC, the third contribution of this thesis is the creation of a data-driven economic MPC scheme for the efficient & economic control of a vessel propulsion system. This specific choice of case study is highly motivated, since it is an item of significant economic importance for the maritime sector of the Greek economy. Initially, a stabilizing EMPC control law is constructed for the vessel propulsion problem and compared to standard tracking MPC, confirming a significant difference in fuel-efficiency. Serving as proof of concept, these results inspire the development of a data-driven EMPC for vessel propulsion based on reinforcement learning. This learning scheme is able to handle structural modelling discrepancies between plant and model, therefore achieving higher closed loop performance and tangible economic benefit.

Lastly, in order to leverage both the collision avoidance tracking MPC and the economic vessel propulsion EMPC results, a control law for the data-driven navigation & economic propulsion control of vessels is proposed and its theoretical foundation for further development is laid. Also, it is the author's opinion that the work presented in this thesis is extendable to other engineering domains and practical applications.

SUBJECT AREA: Control Systems & Computational Intelligence

KEYWORDS: data-driven control, economic model predictive control, radial basis function networks, computational intelligence, metaheuristic search, particle swarm optimization, smart grids, active suspension, vessel propulsion control, vessel trajectory tracking

To my family.

Acknowledgements

19th September, 2023

I owe my greatest thanks for the completion of this dissertation to my advisor, Alex Alexandridis, for mainly three reasons: First, for the flexibility he granted me in terms of choosing my own research directions, while also taking care to gently “nudge” me towards the more fruitful ones. Second, for his availability as an advisor both at the long, high-level conceptual design discussions in front of the whiteboard, and during the laborious implementation phases. Prof. Alexandridis was always eager to step in at times of stagnation and general hopelessness, offering much-needed perspectives that not only proved valuable in tackling the problem at hand, but more importantly demonstrated the value of perseverance in research. The third reason pertains to the beneficial conditions that he secured for my PhD; these included tangible items, such as computing resources & continuous funding, as well as intangible provisions, e.g. assistance for conference attendances. In addition, Prof. Alexandridis spent much of his precious time to shield me from the tedious administrative tasks that regularly arose from the projects I worked on, thus enabling an overall joyful & undistracted research experience for which I am thankful.

Next, I would like to thank my supervising committee members, Prof. Greg Koulouras & Prof. Elias Zois, for always being of assistance during my PhD studies. I would also like to thank the examination committee members, Prof. Constantinos Psomopoulos, Prof. Haralambos Sarimveis, Prof. Kyriakos Vamvoudakis & Prof. Panos Patrinos, for the laborious task of reading this thesis.

I consider myself very lucky to have worked in the TelSiP lab, adorned by such interesting characters that constantly contributed to a warm and friendly working environment. Many thanks must first go to the fellow PhD students who joined the lab roughly the same time as me: Nikos, you “travelling soul”, thank you for our fruitful collaboration during the first years of our PhD. Despina, thank you for all our interesting discussions over all these years; I always admired your work ethic. Aristotelis, it was a great pleasure to share my passion for learning with such an interesting and competent person; I cherish all our lunch-time talks on academic and non-academic topics. Marios S., thank you for the fruitful collaboration over the course of our PhD, and for the commitment you demonstrated on multiple instances. The newest members of the lab are next: Teo & Giannis, thank you for our great talks and the brief but fruitful collaboration over the “biocontrol” project. In addition, thank you Giorgos, Savvas, & Stelios who made my life easier on many occasions. Lastly, I extend my thanks to the rest of the people affiliated with TelSiP who also contributed to the overall atmosphere; I always enjoyed the yearly “tsiknopempti” festivities, as well as the rigorous debates over the best “gourounopoula” restaurant in west Attica.

I am indebted to Mr. Nikolaos Livanos for hosting me at EMTech during the iReact project. It was a great experience working in such a relaxed and, at the same time, competent environment, made possible by the fantastic people working there. Special thanks to Vagelis A. and the rest of the team; I learned a lot through our collaboration, especially regarding best practices on software development, which had significant carry-over to the development work I did for my PhD.

Of equal importance is life outside research: A huge thanks to my lifelong friends Thanasis G., George K., Marios M. & Damianos is in order, for listening to all my academic rants without ever showing a hint of boredom; your perspective always helped me focus on the things that truly mattered. Thank you, Thanasis K. and Vagelis D., for the great movie recommendations and those insightful life-coaching sessions; they were very much appreciated. Thank you, Katerina, Christine, Anastasia, for all the motivating discussions we had. Thank you, Gogo, Mary, Afroditi and Marianna; you always had the sweetest camping trip ideas for August. Many thanks to the rest of the Volos gang (*Β.Ψ.Μ* and others - you know who you are) who remain united regardless of distance. Finally, my deepest thanks go to my "bestie", Iro, who has been with me for the better part of this journey; you always inspired me through your hard work ethic, always cared to understand me, and always supported me beyond what I could ever expect. Hopefully you are not getting bored of my self-absorbed engineering monologues any time soon.

Last but not least, I owe my most profound gratitude to my parents Maria & Manolis, and my sister Pinelopi, for their unconditional love and support. My whole academic career would not have been possible without their material & immaterial assistance, and, more importantly, neither would I be the person I am now if I was not endowed with their values and principles. Thank you, mom, dad and sis.

List of Publications

[1] M. Papadimitrakis, N. Giamarelos, M. Stogiannos, E.N. Zois, N.A. Livanos, A. Alexandridis, "*Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications*", Renewable and Sustainable Energy Reviews, 145 (2021), 111072

[2] M. Papadimitrakis, M. Stogiannos, H. Sarimveis, A. Alexandridis, "*Multi-ship control and collision avoidance using MPC and RBF-based trajectory predictions*", Sensors, 21 (2021) 6959.

[3] N. Giamarelos, E. N. Zois, M. Papadimitrakis, M. Stogiannos, N.-A. I. Livanos and A. Alexandridis, "*Short-Term Electric Load Forecasting with Sparse Coding Methods*", IEEE Access, 9 (2021), pp. 102847-102861.

[4] M. Papadimitrakis, A. Alexandridis "*Active vehicle suspension control using road preview model predictive control and radial basis function networks*", Applied Soft Computing, 120 (2022), pp. 108646

[5] M. Papadimitrakis, A. Kapnopoulos, S. Tsavartzidis, A. Alexandridis "*A cooperative PSO algorithm for Volt-VAR optimization in smart distribution grids*", Electric Power Systems Research, 212 (2022), pp. 108618

[6] M. Stogiannos, M. Papadimitrakis, H. Sarimveis and A. Alexandridis, "*Vessel Trajectory Prediction Using Radial Basis Function Neural Networks*", IEEE EUROCON - 19th International Conference on Smart Technologies, Lviv, Ukraine (2021).

[7] N. Giamarelos, M. Papadimitrakis, M. Stogiannos, Elias N. Zois, Nikolaos A.-I. Livanos, Alex Alexandridis, *Mixed power-load forecasting using a machine learning-based multi-model scheme and an online decision mechanism,* Sensors, 23, 5436 (2023)

[8] M. Papadimitrakis, A. Alexandridis, "*A vessel propulsion controller based on economic model predictive control*", American Control Conference (ACC), San Diego, CA, USA, (2023)

[9] M. Papadimitrakis, A. Alexandridis, *"A Data-driven Vessel Propulsion Controller based on Reinforcement Learning and Economic Model Predictive Control"* (2023) (submitted)

Contents

List of Figures

List of Tables

Abbreviations

ACO	Ant Colony Optimization	**MPC**	Model Predictive Control
AIS	Automated Identification System	**MV**	Manipulated Variable
AP	Active Power	**NLP**	Non-linear Program
APC	Active Power Curtailment	**NN**	Neural Network
CA	Collision Avoidance	**OCP**	Optimal Control Network
CI	Computational Intelligence	**ODE**	Ordinary Differential Equation
COLREG	Collision Regulations	**OPF**	Optimal Power Flow
CPSO	Cooperative Particle Swarm Optimization	**PID**	Proportional - Integral - Derivative
DAE	Differential-Algebraic Equations	**PSO**	Particle Swarm Optimization
DCPA	Distance to Closest Point of Approach	**PV**	Photovoltaic
DN	Distribution Network	**QCQP**	Quadratically-constrained Quadratic Program
dPSO	decentralized Particle Swarm Optimization	**QP**	Quadratic Program
EMPC	Economic Model Predictive Control	**RBF**	Radial Basis Function Network
FFT	Fast Fourier Transform	**RES**	Renewable Energy Source
FM	Fuzzy Means	**RF**	Random Forest
GA	Genetic Algorithms	**RHS**	Right Hand Side
GAS	Globally Asymptotically Stable	**RK4**	Runge-Kutta 4
GNC	Guidance, Navigation & Control	**RL**	Reinforcement Learning
IPM	Interior Point Method	**RMS**	Root Mean Square
KKT	Karush-Kuhn-Tucker	**RMSE**	Relative Mean Squared Error
LiDAR	Light Detection And Ranging	**RNN**	Recurrent Neural Networks
LOS	Line of Sight	**RPC**	Reactive Power Curtailment
LP	Linear Program	**RPLM**	Real Power Losses Minimization
LQR	Linear Quadratic Regulator	**SDP**	Semi-Definite Program
LSTM	Long Short-Term Memory	**SFC**	Specific Fuel Consumption
MAE	Mean Absolute Error	**SQP**	Sequential Quadratic Program
MAPE	Mean Absolute Percentage Error	**SR**	Sparse Regression
MDP	Markov Decision Process	**TCPA**	Time to Closest Point of Approach
MIMO	Multiple Input - Multiple Output	**TD**	Temporal Difference
ML	Machine Learning	**TMPC**	Tracking MPC
MLP	Multi Layer Perceptron	**VDM**	Voltage Deviation Minimization

Part I: Introduction

Chapter 1:

Introduction

The extensive arsenal of mathematical programming tools has been put to use against some of the most challenging engineering optimization problems; indeed, standard deterministic optimization methods have been implemented successfully, including gradient-based methods, Newton-Raphson methods, interior point methods [9]–[11] and others. Since deterministic optimization approaches usually enjoy a solid theoretical foundation and rigorous proofs of convergence, they can be implemented within sophisticated algorithms and solve safety-critical optimization problems [12], [13]. Deterministic optimization methods generally exhibit low computational complexity, which, historically, was a significant concern to be addressed for real-life implementation. Today, the impressive developments regarding hardware architecture and embedded computing within a range of consumer devices and industrial applications, have led to the ever-increasing dimensionality of engineering & decision-making optimization problems to be solved. Today's state-of-the-art commercial & academic deterministic-based solvers that can address large problems utilize the latest advancements in numerical computing and mathematical optimization; notable examples include the sparsity-handling CVXGEN solver [14] or the FORCESpro interior point solver oriented for multistage problems [15]. Still, most of the deterministic-based optimization approaches come with three inherent disadvantages, namely (a) poor performance on non-convex optimization problems due to entrapment in local minima, (b) inability to handle mixed design variables and (c) unsuitability for multi-objective problems, as they cannot easily deal with discontinuous or concave Pareto fronts [16]. Given that a significant portion of the modern engineering optimization problems are non-convex, multi-objective in nature and contain mixed integer and continuous design variables, it is easily understood that nowadays, these methods may not be the best choice at hand.

Still, the aforementioned deterministic solvers remain the powerhouse of constrained finite-time optimal control, otherwise called model predictive control. The MPC control framework uses a dynamic model of the controlled plant in order to predict its response to a control input; then, based on this model, a constrained optimization problem is formulated which is solved at each control sampling instance. Since its original inception [17], [18], the MPC community has developed a mature theoretical foundation in order to address distributed [19], robust [20], and economic control [21] of systems. This theoretical foundation will grant control stability guarantees to a properly formulated MPC scheme and, if the aforementioned deterministic solvers are employed as well, then numerical stability will also be afforded. For this reason, MPC has found a number of applications within the process [22] & automotive [4], [23] industries, as well as vehicle guidance, navigation & control (GNC) [2], [13], [24]. Notwithstanding its attractive stability properties, standard MPC performance heavily relies on the quality of the controller model; if it is inaccurate, then the control actions yielded by the solution of the MPC optimization problem will not be optimal for the real plant. Typical adaptive control techniques for MPC [25] could address modeling inaccuracies if they are merely parametric – however, if there also exist structural differences between the real plant and the controller model, (e.g. the plant is of higher order) then such techniques may not deliver the optimal control law. In addition, a large plant model with significant nonlinearities would render a standard integrated model computationally prohibitive to evaluate in real time, while its linearized counterpart would also fail to adequately capture the underlying dynamics. Moreover, the complexity of the actual process to be controlled could be so high, that a mathematical representation in ODE form would be impossible, thus rendering the MPC approach ineffective.

The nature of the MPC modelling shortcomings & deterministic optimization flaws that were previously discussed, hints at the idea of employing computational intelligence (CI) & machine learning (ML) techniques in order to alleviate them. The domain of CI refers to the theory, design and development of biologically-inspired algorithmic paradigms, encompassing the respective scientific disciplines of neural networks (NNs), fuzzy theory and evolutionary computation. A NN can be used to model black-box dynamics, since it constitutes a massively parallel network that can learn & generalize through a set of training examples; notable NN categories include feedforward NNs, convolutional NNs, recurrent NNs, etc [26]. In turn, as a branch of CI, metaheuristic optimization is motivated by biological processes such as genetic evolution & swarm intelligence in order to construct optimization algorithms that overcome the drawbacks associated with deterministic optimization. These algorithms usually evolve a population of candidate solutions in a stochastic manner, achieving superior exploration capabilities for optimization problems. Moreover, they can handle non continuous and concave Pareto fronts, generating several elements of the optimal set in a single run. Notable swarm intelligence methods include particle swarm optimization (PSO) & ant colony optimization (ACO), while evolutionary algorithms include genetic algorithms (GAs) and artificial immune systems. In general, NN methods have found numerous applications for modeling nonlinear processes in the industry [27]–[29] while metaheuristic optimization algorithms have been successfully implemented in parameter estimation and non-convex optimization of large-scale systems [4], [30]–[33].

It appears then, that the capabilities of CI-based methods can be leveraged for the design of novel data-driven MPC schemes & optimization algorithms. Indeed, NN-based models are an attractive option for substituting - or complementing - first-principle models. Radial basis function networks (RBFN) in particular are widely considered for modeling nonlinear dynamics; consisting of a single middle layer, they can be easily trained using the fuzzy means (FM) algorithm, resulting in models of increased accuracy [34] and low computational complexity during real-time evaluation [35]. As these advantages are of significance in the context of predictive control, RBFNs constitute a popular choice in conjunction with MPC [36]. Another data-driven control approach that can amend modelling inaccuracy is learning-based MPC [37], which can be constructed by employing ML tools such as reinforcement learning (RL) [38]. RL is a sequential decision-making algorithm that adapts a parametric representation of the process using a trial-and-error procedure. In the case of MPC, the RL component utilizes recorded closed-loop data in order to learn new controller parametrizations, which are iteratively applied in real-time in order to achieve improved control performance. In turn, CI methods (particularly, metaheuristic optimization algorithms), can also be used to address large-scale, non-convex, multi-modal optimization problems that frequently arise in modern decision-making schemes. Since metaheuristic optimization algorithms rely on stochastic search, they are less prone to entrapment in local minima, contrary to standard deterministic solvers. In addition, they can be extended in order to exploit the structural topology of the optimization problem, by grouping the design variables accordingly in cooperation sets, thus creating so-called “cooperative” algorithms; these have been very effective in tackling high dimensional problems [39], [40].

Motivated by the aforementioned discussion, this thesis approaches data-driven MPC and optimization from a computational intelligence & machine learning perspective. Therefore, its main objectives are as follows:

- Regarding MPC, RBF models are employed in order to model highly nonlinear and/or unknown process dynamics which would be otherwise prohibitive to address using standard ODE integrated models. In addition, an economic MPC control law is paired with an RL component, thus producing a data-driven MPC scheme capable of handling significant structural & parametric plant-model mismatches and disturbances in real time.

- Regarding metaheuristic optimization, special interest is placed in creating a particle-swarm-based algorithm that can handle large problems. For this reason, a cooperative algorithm together with a design-variable-partitioning scheme is devised and applied to a high-dimensional decision-making problem.

It should be noted that in the context of this thesis, the case studies used to evaluate these methods are chosen with two qualities in mind: The first is their ability to highlight the challenges that CI-based approaches intend to solve, therefore enhancing its academic value. The second is relevance to the current objectives of the Greek industrial sector, thus highlighting the practical merit of this research. To this end, the applications mainly revolve around efficient smart grid operation, as well as economic navigation and propulsion of freight vessels.

1.1 Outline & Contribution

This dissertation is organized into three main parts. The first and second part pertain to introductory material and required preliminaries, spanning Chapters 1-4. Here, basic notions of mathematical programming and convex optimization are presented, together with some notable algorithmic concepts found within MPC solvers and elsewhere. Next, a brief introduction is given to the CI-based methods that are used in this work: These include the class of radial basis function NNs together with the fuzzy-means training algorithm, as well as some notable metaheuristic algorithm categories, such as particle swarm. Lastly, a short presentation of optimal control & MPC concepts is given in Chapter 4, including basic stability notions for both tracking and economic MPC that are employed in this thesis.

The third part of this dissertation contains Chapters 5-7 and presents the design & the application of the proposed methods and algorithms that were developed in the context of this PhD:

- **Chapter 5: Computational Intelligence Methods for efficient Smart Grid Dispatch**
  This chapter presents the CI-based methods that were developed in order to tackle the two main problems related to smart grid dispatch, namely the accurate load forecasting task and the efficient power dispatch of distributed renewable sources. The proposed load prediction algorithm makes use of a pool of several machine-learning and CI-based models, which are evaluated on-line using a novel dynamic selection algorithm. This leads to high prediction performance for a number of horizons, as tested on data from a Greek distribution network substation. Next, a cooperative particle-swarm algorithm is designed for the optimal power flow problem of electrical distribution networks with high penetration of photovoltaic sources. The proposed method makes use of a community-detection algorithm in order to exploit the topology of the problem and thus assign effective cooperation sets based on the notion of electrical distance. The economic effectiveness of the method is demonstrated on an IEEE benchmark system with various scenarios. Lastly, it is noted that both the load prediction and the optimal power flow methods proposed here went on to be incorporated in a real smart grid decision support system.

- **Chapter 6: Data-driven Tracking Nonlinear Model Predictive Control**

This chapter presents the design of data-driven tracking MPC methods based on radial basis function NNs in order to adequately capture nonlinear or otherwise hard-to-model plant dynamics. Two representative case studies are thus chosen: The first refers to the control of a vehicle's active suspension, which, due to its high-dimensionality and nonlinearity, prohibits the usage of standard linear or ODE-integrated MPC models. The proposed scheme is evaluated on a detailed full car model for various road and excitation scenarios. The second case study of this chapter pertains to the vessel collision avoidance control using obstacle trajectory models trained on real automatic identification system (AIS) data. Here, the motivation of using a data-driven MPC approach is that no first-principles model could address the inference of obstacle vessel trajectories. The proposed control method utilizes RBF models trained on historic AIS data, leading to safer and more economical vessel trajectories than using an MPC controller with naïve, straight-line obstacle trajectory predictions.

- **Chapter 7: Data-driven Economic Nonlinear Model Predictive Control**

  In this chapter, a data-driven economic nonlinear MPC scheme is presented for the economic control of a vessel propulsion system. This specific choice of a case study is highly motivated, both as a practical continuation of the vessel-control-related work presented in Chapter 6, and as an item of significant economic importance for the marine sector of the Greek economy. First, a stabilizing EMPC control law is constructed for the vessel propulsion case, showing significant fuel-efficiency improvement compared to a standard MPC. This control law served as a proof of concept for the development of a data-driven EMPC based on reinforcement learning, capable of handling structural modelling discrepancies that are ever-present between real vessel propulsion systems and the respective controller models. In addition, this RL-based economic MPC scheme incorporates an economic stage cost that accurately describes the real high-level chartering economics of freight shipping, rather than an ad-hoc tracking stage cost, leading to tangible economic improvements, as simulated on various scenarios. Lastly, in order to combine the aforementioned economic propulsion results & the vessel trajectory tracking controller from chapter 6, a control law for the data-driven autonomous navigation & economic propulsion control of vessels is designed and presented.

Finally, this dissertation is concluded in Chapter 8.

Part II: Preliminaries

Chapter 2:

Deterministic and Metaheuristic Optimization

Mathematical optimization, or mathematical programming, is a process that optimizes (maximizes or minimizes) the value of an objective function with respect to a set of constraints. Optimization problems arise in all engineering disciplines, and especially in control systems and control theory in general.

There are two different types of optimization methods that are widely used today. Deterministic methods use specific rules for moving from one candidate solution to another, and provide theoretical certificates for the optimality of the reported solution. Non-deterministic (stochastic or metaheuristic) methods employ probabilistic processes to evolve candidate solutions. Due to these processes, metaheuristic methods usually do not provide optimality certificates for their solution, still, they possess certain properties that deterministic algorithms do not exhibit, such as the ability to converge to better solutions when the program is non-convex and multi-modal.

In the following chapter, notable methodologies of both categories are briefly presented in detail from a theoretical point of view, and their relevance as applied solvers for the optimization problems encountered in this work is highlighted.

2.1 Deterministic Optimization

This section assumes standard optimization knowledge, namely, types of optimization problems, characteristics of convex functions, as well as Lagrangian functions and duality theory. Some general concepts regarding quadratic programs and optimality conditions are introduced, before presenting notable deterministic optimization methods that are used in this thesis.

2.1.1 Introduction to Quadratic Programs

Consider the general non-linear program formulation:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.1.1a}$$

$$\text{subject to } g(\mathrm{x}) = 0 \tag{2.1.1b}$$

$$\mathrm{h}(\mathrm{x}) \geq 0 \tag{2.1.1c}$$

where f: $\mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, g: $\mathbb{R}^n \rightarrow \mathbb{R}^p$ is the equality constraint function, and h: $\mathbb{R}^n \rightarrow \mathbb{R}^q$ is the inequality constraint function. Functions f, g, h are assumed to be continuously differentiable up to two times or more. Setting these functions as

$$f(x) = q^T x + \frac{1}{2} x^T H x \tag{2.1.2a}$$

$$g(\mathrm{x}) = Ax - b \tag{2.1.2b}$$

$$\mathrm{h}(\mathrm{x}) = \mathrm{Cx} - \mathrm{d} \tag{2.1.2c}$$

where $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^p$, $C \in \mathbb{R}^{m\times n}$, $d \in \mathbb{R}^m$, $H \in \mathbb{R}^{n\times n}$ means that Problem (24) becomes a Quadratic Program (QP). Note that H is usually called "Hessian matrix" of the problem since $H = \nabla^2$ f(x). The Hessian matrix also encapsulates an important property of QPs: if $H \succ 0$, the QP is convex, meaning that solutions are significantly easier to find, while non-convex QPs with $H \prec 0$ may have multiple local minima and therefore are very hard to solve.

2.1.2 Optimality Conditions

Consider the following unconstrainted optimization problem:

$$\min_{x \in D} f(x) \tag{2.1.3}$$

Where $f: D \rightarrow \mathbb{R}$, and D is the feasible domain of f, where f a continuous and twice differentiable function.

Theorem 2.1.1 (Second order Necessary Conditions): *Given a local minimizer* x^* *of a twice-differentiable function* f, *then*

$$\nabla^2 f(\mathrm{x}^\star) \succcurlyeq 0 \tag{2.1.4}$$

Proof: If $\nabla^2 \mathrm{f}(\mathrm{x}^\star) \nsucceq 0$ then a descent direction p exists so that $p^T \nabla^2 \mathrm{f}(\mathrm{x}^\star)\, p < 0$. This means that once again, there exists a feasible descent direction in which $f(x)$ can be further minimized. A step size $k > 0$ small enough so that $\forall \kappa \in [0, k]$ we have $\mathrm{x}^\star + \kappa\mathrm{p} \in D$ and $\nabla f(\mathrm{x}^\star + \kappa p)^T\, p < 0$. Employing the 2nd order Taylor expansion yields:

$$f(\mathrm{x}^\star + kp) = f(\mathrm{x}^\star) + k\,\nabla f(\mathrm{x}^\star + \kappa p)^T\, p + \frac{1}{2} k^2 p^T\, \nabla^2 f(\mathrm{x}^\star)\, p \qquad (2.1.5)$$

Since the third term of the RHS is < 0, it means that $f(\mathrm{x}^\star + kp) < f(\mathrm{x}^\star)$, i.e. $\mathrm{x}^\star$ is not a minimizer. ■

Next, for the case of continuous convex functions [41], sufficient optimality conditions can be constructed.

Theorem 2.1.2 (Convex First Order Sufficient Conditions): *Given a stationary point* $\bar{\mathrm{x}}$ *of a convex function* f*, then* $\bar{\mathrm{x}}$ *is indeed the global minimizer* $\mathrm{x}^\star$ *of* f.

Proof: We know that since f is convex, we have for any feasible y:

$$f(\mathrm{y}) \geq f(\mathrm{x}^\star) + \nabla f(\mathrm{x}^\star)^T\, (y - \mathrm{x}^\star) \geq f(\mathrm{x}^\star) \qquad (2.1.6)$$

Assuming the above doesn't hold, then there exists a y with $\nabla f(\mathrm{x}^*)^T\, (\mathrm{y} - \mathrm{x}^\star) < 0$. Using a Taylor expansion

$$f(x^* + t(y - x^\star)) = f(x^\star) + t\nabla f(x^\star)^T\, (y - x^\star) + O(t) \qquad (2.1.7)$$

Here, the 2nd term is < 0 and the third term can be omitted as t becomes small enough. This means that $\mathrm{f}(\mathrm{x}^\star + \mathrm{t}(\mathrm{y} - \mathrm{x}^\star)) < f(\mathrm{x}^\star)$, i.e. $\mathrm{x}^\star$ is not a minimizer. ■

The above results can be generalized in order to include smooth nonlinear functions f.

Theorem 2.1.3 (Second Order Sufficient Conditions): *Given a stationary point* $\bar{\mathrm{x}}$ *of a smooth & twice-differentiable function* f *and*

$$\nabla^2 f(\bar{\mathrm{x}}) \succ 0 \qquad (2.1.8)$$

then $\bar{x}$ *is a strict local minimizer of* f.

Proof: The idea is to consider a small closed "ball" B around $\bar{\mathrm{x}}$ so that $\forall x \in B$ we have $\nabla^2 f(\mathrm{x}) \succ 0$, i.e. f can be considered convex in that region. Then, Theorem 2.1.2 can be leveraged and the Theorem is proved. ■

Consider now the Lagrangian of the general NLP (2.1.1), which is not assumed to be convex. We write its Lagrangian as

$$\mathcal{L}(x,\lambda,\nu) = f(x) + \sum_{i=1}^{p} \lambda_i \, h_i(x) + \sum_{i=1}^{m} \nu_i \, g_i(x) \tag{2.1.9}$$

with $\mathbf{dom}\, \mathcal{L} = D \times \mathbb{R}^p \times \mathbb{R}^m$, λ_i the Lagrange multiplier of inequality constraint h_i and ν_i the Lagrange multiplier of equality constraint g_i. Vectors λ, ν are otherwise called dual variables of the problem. The Lagrange dual function can be subsequently defined as the infimum of the Lagrangian over x:

$$L(\lambda,\nu) = \inf_{x \in D} \mathcal{L}(x,\lambda,\nu) \tag{2.1.10}$$

Note that the Lagrange dual function $L(\lambda,\nu)$ can be used to generate a lower bound on the optimal value $x^\star$ of Problem (2.1.1):

$$L(\lambda,\nu) \leq x^\star, \ \forall \lambda \succcurlyeq 0, \nu \tag{2.1.11}$$

This lower bound is nontrivial when it also holds $(\lambda,\nu) \epsilon$ $\mathbf{dom}\, L$. When we also have $\lambda \succcurlyeq 0$, then the respective (λ,ν) pair is called dual feasible. It is practical to see which (λ,ν) pair will yield the best lower bound. For this reason, the Lagrange dual problem is formulated:

$$\max L(\lambda,\nu) \tag{2.1.12a}$$
$$\text{subject to } \lambda \succcurlyeq 0 \tag{2.1.12b}$$

The $(\lambda^\star, \nu^\star)$ pair is referred to as the optimal Lagrange multipliers. Note that this is a convex optimization problem, regardless of whether the original Problem (2.1.1) is convex. For its optimal value, denoted as $L^\star$, it holds that:

$$L^\star \leq x^\star \tag{2.1.13}$$

Which is true even when the original problem is not convex. This property is called weak duality, and the difference $x^\star - L^\star$ is called the duality gap. If inequality (2.1.13) is strong, that is $x^\star = L^\star$, then strong duality holds, meaning that the duality gap is zero. Strong duality of the Lagrange dual does not hold in general; is a given only if the primal Problem (2.1.1) is convex, otherwise special conditions must be established, called constraint qualifications. One such constraint qualification is *Slater's condition* [41].

Another important concept that arises from strong duality is the following; let $x^\star$ be a primal optimal and $(\lambda^\star, \nu^\star)$ the dual optimal points. Then,

$$L(\lambda^\star,\nu^\star) \leq f(x^\star) + \sum_{i=1}^{p} \lambda_i^\star \, h_i(x^\star) + \sum_{i=1}^{m} \nu_i^\star \, g_i(x^\star) \tag{2.1.14}$$

This follows because $\inf_{x \in D} \mathcal{L}(x, \lambda^\star, v^\star)$ is less than or equal to $\mathcal{L}(x, \lambda, v)$ at $\mathrm{x}^\star$. Since $\lambda_\mathrm{i}^\star \geq 0$, $\mathrm{h}_\mathrm{i}(\mathrm{x}^\star) \geq 0$, and $\mathrm{g}_\mathrm{i}(\mathrm{x}^\star) = 0$, it is concluded that

$$\sum_{i=1}^{p} \lambda_i^\star \, h_i(\mathrm{x}^\star) = 0 \tag{2.1.15}$$

Which also yields:

$$\lambda_i^\star \, h_i(\mathrm{x}^\star) = 0, \;\; i = 1, \dots, p \tag{2.1.16}$$

This is called the *complementary slackness condition*, and is equivalently expressed as

$$\lambda_i^\star > 0 \rightarrow h_i(\mathrm{x}^\star) = 0 \;\; or \;\; h_i(\mathrm{x}^\star) < 0 \rightarrow \lambda_i^\star = 0 \tag{2.1.17}$$

meaning that $\lambda_\mathrm{i}^\star$ is zero unless $\mathrm{h}_\mathrm{i}(\mathrm{x}^\star)$ is active (=0) at the optimum.

Using the complementary slackness condition, the Karush-Kuhn-Tucker conditions can be expressed. Since $\mathrm{x}^\star$ minimizes $\mathcal{L}(x, \lambda^\star, v^\star)$, this means that its gradient goes to zero [41]:

$$\nabla \mathcal{L}(\mathrm{x}^\star, \lambda^\star, v^\star) = \nabla f(\mathrm{x}^\star) + \sum_{i=1}^{p} \lambda_i^\star \, \nabla h_i(\mathrm{x}^\star) + \sum_{i=1}^{m} v_i^\star \, \nabla g_i(\mathrm{x}^\star) = 0 \tag{2.1.18}$$

This results in the following optimality conditions:

Theorem 2.1.4 (Karush-Kuhn-Tucker conditions / First-order necessary conditions for optimality (FONC)): *The primal-dual variable set* $\mathrm{x}^\star, \lambda^\star, \mathrm{v}^\star$ *is guaranteed to be optimal when the following conditions are met:*

$$h_i(\mathrm{x}^\star) \geq 0 \tag{2.1.19a}$$

$$\lambda_i^\star \geq 0 \tag{2.1.19b}$$

$$\lambda_i^\star h_i(\mathrm{x}^\star) = 0 \tag{2.1.19c}$$

$$\nabla f(\mathrm{x}^\star) + \sum_{i=1}^{p} \lambda_i^\star \, \nabla h_i(\mathrm{x}^\star) = 0 \tag{2.1.19d}$$

Proof: See [41]. ■

In the above, it is of course assumed that the linear independence constraint qualification (LICQ) holds. In the case where the problem at hand is convex, then we also have that the duality gap is zero. Note that the KKT conditions are equivalent to $\nabla \mathrm{f}(\mathrm{x}^\star) = 0$ condition for unconstrained optimization.

2.1.3 Newton-based Methods

Newton's method as an optimization technique is applied to the first derivative f' of a double differentiable function f to find the roots of the derivative (i.e., the solutions of $f'(x) = 0$), also known as the stationary points of f. These solutions can be minimum, maximum, or saddle points.

The idea behind Newton's method is that the function f under consideration is locally approximated by a quadratic function, which is minimized in the case where the minimum of f is sought [42]. This is achieved by constructing a sequence x_k, which converges to some x^*, for which $f'(x^*) = 0$. That is, x^* is a stationary point of f. Thus, f is approximated by the second-order Taylor expansion:

$$f(x) \simeq f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k). \tag{2.1.20}$$

The RHS of 3.11 is minimized by the sequence:

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)^T. \tag{2.1.21}$$

Taking into account the second-order sufficiency conditions (Theorem 2.1.3), we assume that near x^*, the Hessian matrix $\nabla^2 f(x_k)$ is positive definite. Given continuous second order derivatives for f, it indeed holds that $\nabla^2 f(x_k) \succ 0$ near x^*, meaning that the method is well defined in its neighbourhood [43]. Then, the following hold:

1. If the initial point x_0 is sufficiently close to x^*, the sequence (2.1.21) converges to x^* with quadratic convergence.
2. The norm of the gradients $\|\nabla f_k\|$ also quadratically converges to 0.

2.1.4 Active Set Methods

Active set algorithms are widely used for cases of parametric quadratic programs, significantly boosting the solution speed. The idea is to identify the active constraints within the set of inequality constraints $h(x)$ before solution. These constraints are subsequently expressed as equality constraints, resulting in an equality-constrained QP subproblem which is simpler to solve.

Consider the following convex QP ($H \succcurlyeq 0$):

$$\min_{x \in \mathbb{R}^n} q^T x + \frac{1}{2} x^T H x \tag{2.1.22a}$$

$$\text{s.t. } Ax + b \geq 0 \tag{2.1.22b}$$

Applying the KKT conditions (2.1.19) yields:

$$H\mathrm{x}^\star + \mathrm{q} - \mathrm{A}^\mathrm{T}\lambda^\star = 0 \tag{2.1.23a}$$

$$\mathrm{Ax}^\star + b \geq 0 \tag{2.1.23b}$$

$$\lambda^\star \geq 0 \tag{2.1.23c}$$

$$\lambda_i^\star(\mathrm{Ax}^\star + b)_i = 0, \text{ for i} = 1, \dots, \mathrm{p} \tag{2.1.23d}$$

Before applying the active set method to calculate the primal-dual pair $(\mathrm{x}^\star, \lambda^\star)$ as well as the corresponding active set $\mathcal{A}(\mathrm{x}^\star) \subset \{1, \dots, \mathrm{p}\}$ that satisfy the KKT conditions, the following index set notation is used:

$$\text{Active inequality index set } \mathbb{A} \subset \{1, \ldots, \mathrm{p}\} \tag{2.1.24a}$$
$$\text{Inactive inequality index set } \mathbb{I} = \{1, \ldots, \mathrm{p}\} \backslash \mathbb{A} \tag{2.1.24b}$$

Meaning that the following expressions hold:

$$b = \begin{pmatrix} b_{\mathbb{A}} \\ \overline{b_{\mathbb{I}}} \end{pmatrix}, A = \begin{pmatrix} A_{\mathbb{A}} \\ \overline{A_{\mathbb{I}}} \end{pmatrix}, Ax + b \geq 0 \Leftrightarrow A_{\mathbb{A}}x + b_{\mathbb{A}} \geq 0 \text{ AND } A_{\mathbb{I}}x + b_{\mathbb{I}} \geq 0 \tag{2.1.25}$$

It can be shown that $\mathrm{x}^{\star}$ is a global minimizer of Problem (2.1.22) iff there exist index sets $\mathbb{A}$ & $\mathbb{I}$ as well as a dual variable $\lambda_{\mathbb{A}}^{\star}$ so that the following (modified) KKT conditions hold:

$$H\mathrm{x}^{\star} + \mathrm{q} - A_{\mathbb{A}}{}^{\mathrm{T}}\lambda^{\star} = 0 \tag{2.1.26a}$$
$$A_{\mathbb{A}}\mathrm{x}^{\star} + b_{\mathbb{A}} = 0 \tag{2.1.26b}$$
$$A_{\mathbb{I}}\mathrm{x}^{\star} + b_{\mathbb{I}} \geq 0 \tag{2.1.26c}$$
$$\lambda_{\mathbb{A}}^{\star} \geq 0 \tag{2.1.26d}$$
$$\lambda^{\star} = \begin{pmatrix} \lambda_{\mathbb{A}}^{\star} \\ \lambda_{\mathbb{I}}^{\star} \end{pmatrix}, \text{with } \lambda_{\mathbb{I}}^{\star} = 0 \tag{2.1.26e}$$

The dual variable $\lambda_{\mathbb{I}}^{\star}$ is by definition zero since it is the Lagrange multiplier of an inactive inequality constraint. The idea is to solve (2.1.26a) and (2.1.26b) for $\mathrm{x}^{\star}$ and $\lambda^{\star}$ iteratively by changing set $\mathbb{A}$ accordingly, until (2.1.26c) and (2.1.26d) are satisfied. The interested reader can find a detailed active-set algorithm in **source.** It is briefly noted that these algorithms are commonly paired with sequential quadratic programming (SQP) solvers, where the general NLP (2.1.1) is treated by iteratively solving the following QP:

$$\min_{x \in \mathbb{R}^n} \nabla f(x_k)^T p + \frac{1}{2} p^T H p \tag{2.1.27a}$$
$$\text{s.t. } h(x_k) + \left.\frac{\vartheta h}{\vartheta x}\right|_{x_k} p \geq 0 \tag{2.1.27b}$$

It can be shown that this QP has the same active set as the original NLP.

2.1.5 Interior Point Methods

Interior point methods (IPMs) are widely used for solving large convex optimization problems that include inequality constraints, assuming that they are strictly feasible, so that an optimal primal-dual variable set $\mathrm{x}^{\star}, \lambda^{\star}, \nu^{\star}$ exists. IPMs employ Newton's Method as described in subsection 2.1.3 in order to equivalently solve either sequential equality-constrained problems, or modified versions of the KKT conditions (2.1.19). The problems that can be solved span linear programs (LPs), QPs, quadratically-constrained QPs (QCQPs), and others, as long as f(x), $h(x)$ and $g(x)$ are twice differentiable. The idea is to replace the non-smooth KKT conditions (2.1.19a-c) with a slacked approximation. Condition $\lambda_i^{\star} h_i(\mathrm{x}^{\star}) = 0$ becomes $\lambda_i^{\star} h_i(\mathrm{x}^{\star}) = \sigma$ where $\sigma > 0$ is a small slack variable, and the smooth KKT-based problem is formulated:

$$\nabla f(x) - \sum_{i=1}^{q} \nabla h_i(x)\lambda_i = 0 \tag{2.1.28a}$$

$$\lambda_i h_i(x) - \sigma = 0, \qquad \mathrm{i} = 1, \dots, \mathrm{p} \tag{2.1.28b}$$

This problem is solvable using Newton's method, and is shown that as the relaxation diminishes, i.e. $\sigma \to 0$, then the slacked solutions $\tilde{x}(\sigma)$, $\tilde{\lambda}(\tau)$ approach the optimal solutions of the NLP $x^\star$, $\lambda^\star$. IPMs have exhibited powerful results for the case of QP and other convex programming problems, as well as general nonlinear programs (2.1.1).

2.1.6 Semi-Definite Programming

A special class of convex programs utilize linear matrix inequalities (LMIs) as constraints:

$$B_0 + \sum_{i=1}^{n} B_i x_i \succcurlyeq 0 \tag{2.1.29}$$

Here, $B_{0,\dots m} \in \mathbb{S}^k$, where $\mathbb{S}^k$ is the space of symmetric matrices with dimensions $\mathbb{R}^{k \times k}$. When a convex program involves the constraint of positive-definitiveness of a design matrix, then it is called a semi-definite program (SDP):

$$\min_{x \in \mathbb{R}^n} c^T x \tag{2.1.30a}$$

$$\text{s.t. } B_0 + \textstyle\sum_{i=1}^{n} B_i x_i \succcurlyeq 0 \tag{2.1.30b}$$

$$Ax - b = 0 \tag{2.1.30c}$$

It is noted that the SDP form generalizes over LPs, QPs, as well as QCQPs (e.g. if the B_i matrices are set as diagonal, then generalized inequality (2.1.30b) is equivalent to a set of linear inequalities, thus representing an LP). Multiple applications of SDPs can be found, pertaining to matrix eigenvalue optimization, matrix norm minimization, etc. Apart from the specialized SDP solvers that exist, e.g. [44], a useful approach to solving SDPs is using the barrier method, coupled with an IPM formulation. The SDP (2.1.30) is transformed as:

$$\min_{x \in \mathbb{R}^n} c^T x + \mu J(B_0 + \sum_{i=1}^{n} B_i x_i) \tag{2.1.31a}$$

$$\text{s.t. } Ax - b = 0 \tag{2.1.31b}$$

Where $J(\cdot)$ is called the log barrier function and is denoted as:

$$J(X) = -\ln \det (X) \tag{2.1.32}$$

The benefit of employing this approach is that IPM methods can readily be used, enjoying polynomial complexity [45].

2.2 Metaheuristic Optimization

The typical deterministic optimization methods that were presented in Section 2.1 come with certain optimality guarantees, which is an important feature. However, they also come with three inherent disadvantages, namely (a) poor performance on non-convex optimization problems, (b) inability to handle discrete design variables and (c) unsuitability for multi-objective problems, as they cannot easily deal with discontinuous or concave Pareto fronts [16]. Given that a significant portion of engineering optimization problems are non-convex, multi-objective in nature and contain mixed integer and continuous design variables, it is easily understood that nowadays, deterministic methods may not be the best choice for every application. Metaheuristic methods on the other hand, constitute a class of optimization algorithms, or algorithmic frameworks, that are in principle better equipped to overcome the difficulties commonly encountered in optimization, and provide better quality solutions than deterministic approaches.

This section briefly discusses the main categories of metaheuristic methods, and presents the particle swarm optimization that is used in the context of this thesis in detail.

2.2.1 Brief Literature Review

Evolutionary Computation: Evolutionary computation (EC) is a widely used computer science discipline, comprising methods that simulate the evolution of members of a population which are regarded as possible solutions to the optimization problem. Each individual receives a measure of fitness, and then a selection procedure uses biologically inspired techniques to stimulate the solutions with high levels of fitness. Genetic algorithms (GAs) [46] and differential evolution [47] comprise the two most distinctive representatives of EC, albeit a number of other EC techniques have also been proposed [48]. Genetic algorithms are search algorithms which iteratively evolve a population of candidate solutions encoded as chromosomes, through genetically inspired operations like crossover and mutation; a selection process picks the solutions that will be passed onto the next iteration. GA techniques are able to perform reliably and can easily collaborate with existing models and systems [49], as well as integrate into hybrid approaches [50]. Additionally, they are easily scalable with parallel implementation abilities [51], [52] and they impose no restrictions on the functions they process.

Swarm intelligence: Swarm intelligence methods exhibit characteristics seen in decentralized, self-organized groups of biological organisms. SI systems typically consist of a population of simple agents that interact locally with each other and with their environment; these interactions, albeit local, aim to lead to the emergence of a "smart" global behavior. A notable SI method is the ant colony optimization. The main idea is the simultaneous development of multiple threads (possible solutions) based on local data and a dynamic memory structure containing information on the quality of previous results. ACO has proven to be effective in solving combinatorial optimization problems and has found application in a number of fields in industry [53], but presents some limitations in dealing with continuous design variables. Various modifications of the original algorithm have been reported in the literature [54]–[56].

Non-population-based methods: Unlike the previously mentioned categories, which evolve a population or swarm of solutions, some of the early metaheuristic methods were based on modifying a single solution, the most prominent being simulated annealing (SA) [57] and tabu search (TS) [58]. SA draws inspiration from the search of a minimum energy state which occurs during the process of annealing in metallurgy. Its distinct characteristic is that it allows for temporarily accepting a worse solution with a probability, which becomes smaller as the iterations progress. SA tuning involves only a few parameters while an obvious advantage involves the significantly reduced computational cost as a result of operating on a single solution; on the other hand, accuracy is usually inferior compared to population-based methods. Recent advances and SA modifications are reported in [59], [60]. TS owes its name to the so-called tabu lists, that record the search history in order to avoid cycling, i.e. revisiting previously found solutions. The basic idea behind TS has been subjected to a number of modifications, which improve the algorithm's efficiency [61]. TS is suitable for large-scale optimization problems as it combines the significant advantage of reduced computational complexity with reasonable performance in terms of accuracy, albeit it cannot compete with population-based methods in that respect.

Artificial immune systems: Artificial immune systems (AIS) are inspired by theoretical immunology, simulating the processes used by the biological immune system to respond to external threats. AIS follow a distributed model with an absence of any point of total control, using exclusively local information. Due to their inherent decentralized nature, AIS require minimal CPU and memory resources, in contrast to population-based techniques. On the other hand, AIS may need customization in order to solve optimization problems unlike the rest of the methods presented in this section. AIS-based techniques have been extensively studied and applied to many engineering fields [62], including power grid applications [63], [64]. They provide inspiration for hybrid methodologies with remarkable characteristics [65].

2.2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) [66] comprises one of the most important swarm intelligence methodologies. The algorithm encodes a population of possible solutions (particles), which are driven towards optimality by exchanging local and global information. Each particle updates its position by considering the best position it had occupied in the past, the best position of the swarm and stochastic parameters averting the algorithm from becoming trapped in local minima. PSO requires no special encoding, thus enjoying an advantage over other metaheuristic methods. Its simplicity, combined with effectiveness and speed, make PSO ideal for use in applications where computational cost is a critical parameter. Due to these merits, PSO has been widely used, while various modifications have been proposed [67], [68], aiming to cure its defects, which are mainly associated with premature convergence.

The PSO algorithm is described as follows:

```
Algorithm 2.2.1 Simple particle swarm algorithm
Initialize for every particle i:
    Randomize Position $x^0_{id}$, Set best position $p^0_{id} \leftarrow x^0_{id}$, Calculate fitness value $f(x^0_{id})$
    Find best fitness value among particles, set it as current global best $p^0_{gd}$.
while termination condition false
    for particle i:
        Calculate new particle speed as $v^{t+1}_{id} \leftarrow v^t_{id} + c_1 \cdot rand(0,1) \cdot (p^t_{id} - x^t_{id}) + c_2 \cdot rand(0,1) \cdot (p^t_{gd} - x^t_{id})$
        Calculate new particle position $x^{t+1}_{id} \leftarrow x^t_{id} + v^{t+1}_{id}$
        Calculate new fitness value $f(x^{t+1}_{id})$
        If $f(x^{t+1}_{id}) \geq f(p^t_{id})$:  $p^t_{id} \leftarrow x^{t+1}_{id}$;
        If $f(x^{t+1}_{id}) \geq f(p^t_{gd})$:  $p^t_{gd} \leftarrow x^{t+1}_{id}$;
        $k \leftarrow k + 1$
```

Multiple variants of this basic algorithm exist, pertaining to the addition of inertial weights [69], constriction factors [70], and cooperative particle partitions [71].

Chapter 3:
Radial Basis Function Neural Networks

Simple machine learning models (such as linear models) cannot account for the underlying nonlinearities of a complex engineering system. Methods based on computational intelligence, e.g. neural networks (NNs), seem to be an attractive alternative, as they are very effective in modeling nonlinear plants [26] and can be configured to accommodate for changes in plant parameters. Radial basis function (RBF) NNs constitute a popular neural network architecture with numerous applications in nonlinear system identification and control, both in industry and academia [72]. They are widely considered for modeling nonlinear dynamics, mainly because of their simple structure and increased accuracy [34]. They are comprised of a single hidden layer which is linearly attached to the output layer of the network, meaning that the employed training algorithms are faster and more efficient than their more complicated multilayer perceptron (MLP) counterparts [34].

In this section, the basic formulation of RBF NNs is presented, as well as the fuzzy-means training algorithm that is used in the context of this thesis.

3.1 Basic Formulation

In RBFN, the training procedure aims to find the multidimensional surface that best approximates a set of training examples. This surface is constructed as a sum of simpler surfaces exhibiting radial basis symmetry around centers specifically placed in the input space. Training an RBFN corresponds to finding the number of RBF centers, their coordinates in the input space and the weights connecting the hidden layer to the output layer.

A typical RBF NN structure can be found in Fig. 3.1.1. The input layer distributes *N* input variables to *L* nodes of the hidden layer. Each node of the hidden layer is comprised of a center with *N* dimensions. The hidden layer performs a nonlinear transformation that maps the input space on a new, higher dimensional space. The first step in calculating the output is to compute the activity $\mu_l(\boldsymbol{u}(k))$for every node *l* and every datapoint *k*; this is calculated as the Euclidean norm of the difference between the *k*th input vector $\boldsymbol{u}(k)$ and the *l*th node center $\widehat{\boldsymbol{u}}_l$:

$$\mu_l(\boldsymbol{u}(k)) = \|\boldsymbol{u}(k) - \widehat{\boldsymbol{u}}_l\| = \sqrt{\sum_{i=1}^{N}(\boldsymbol{u}(k) - \widehat{\boldsymbol{u}}_l)^2}\,,\ \ k = 1, \dots, K \tag{3.1.1}$$

Using the activity $\mu_l(\boldsymbol{u}(k))$, the node activation function can be computed. In this work, a typical activation function is used.

Then, for each datapoint and each node, an activation function value is computed. The hidden node responses for the *k*-th datapoint are written as:

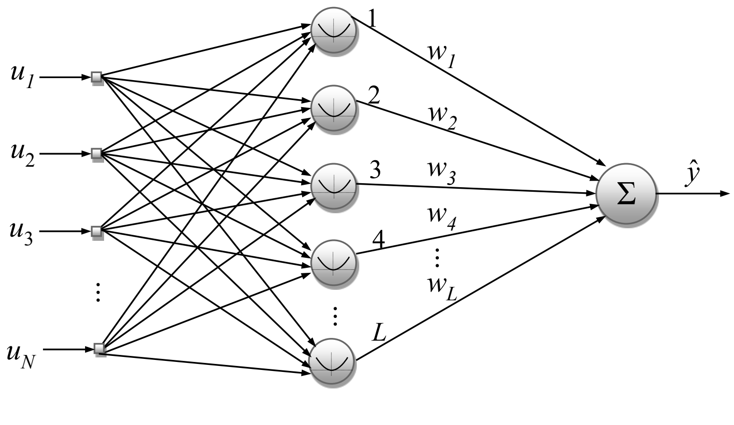


Figure 3.1.1: Schematic of an RBF network: The input layer distributes *N* input variables to *L* nodes of the hidden layer. Each node of the hidden layer is comprised of a center with *N* dimensions. The hidden layer performs a nonlinear transformation that maps the input space on a new, higher dimensional space. The weighted sum of the nodes constitutes the output.

$$\boldsymbol{z}(k) = \left[g\left(\mu_1(\boldsymbol{u}(k))\right), g\left(\mu_2(\boldsymbol{u}(k))\right), \dots, g\left(\mu_L(\boldsymbol{u}(k))\right)\right] \tag{3.1.2}$$

The final output value $\hat{y}$ of the RBF NN is then calculated as a linear combination of the hidden note responses,

$$\hat{y}(k) = \boldsymbol{z}(k)\,\boldsymbol{w} \tag{3.1.3}$$

where $\boldsymbol{w}$ is a vector containing synaptic weights. For a given real output vector $\boldsymbol{Y}$, after formulating the vector of hidden node responses $\boldsymbol{Z}$, the weight vector $\boldsymbol{w}$ can be calculated trivially by least squares in matrix form:

$$\boldsymbol{w}^T = \boldsymbol{Y}^T\boldsymbol{Z}\,(\boldsymbol{Z}^T\boldsymbol{Z})^{-1} \tag{3.1.4}$$

Thus, it can be concluded that the most important stage of creating the RBF NN is the establishment of the hidden node centers $\hat{\boldsymbol{u}}_l$.

3.2 Fuzzy Means RBF

A standard approach to determine the coordinates of the RBF centers involves the *k*-means algorithm [73], for a given number of centers; however, as this number is not a priori known, a tedious trial-and-error procedure is required to determine it. The fuzzy means algorithm is well suited for this task [34], because it follows a fuzzy clustering approach in order to determine the node centers. Consider a system with N normalized input variables u_i. In order to create a fuzzy partitioning of the input space, the domain of each input variable u_i must be segmented into *S* one-dimensional fuzzy subspaces (FS). Each subspace $\boldsymbol{A}^l$ where $l = 1, \dots, S$ is created by combining *N* fuzzy sets for each input direction. It is possible to define the fuzzy subspaces through the center vector $\boldsymbol{a}^l$ containing the centers and the side vector $\delta\boldsymbol{a}$.

The resulting subspaces form a grid in the *N*-dimensional input space, where each node of the grid is a candidate RBF center. The FM algorithm aims to determine which candidate nodes will be finally selected as centers. This is accomplished through the use of a membership function μ_{A^l} formulated as:

$$\mu_{A^l}(\boldsymbol{u}(k)) = \begin{cases} 1 - d_r^l(\boldsymbol{u}(k)), & if\ d_r^l(\boldsymbol{u}(k)) \leq 1 \\ 0, & if\ otherwise \end{cases} \tag{3.2.1}$$

The distance $d_r^l(\boldsymbol{u}(k))$ is computed as

$$d_r^l(\boldsymbol{u}(k)) = \sqrt{\sum_{i=1}^{N} \left(a_{i,ji}^l - u_i(k)\right)^2 \Big/ \left(\sqrt{N}\delta\boldsymbol{a}\right)} \tag{3.2.2}$$

where $\boldsymbol{u}(k)$is the *k*th input vector, $a_{i,ji}^l$ is the center of the fuzzy subspace $\mathbf{A}^l$, δa is half the width which is the same for each input, and *N* is the dimensionality of the input space. Eq. (3.2.2) defines a surface in the input space that bounds the input vectors that will be covered by fuzzy subspace $\mathbf{A}^l$, or in other words, that will receive nonzero membership degree in the membership function. The FM algorithm uses a fast-non-iterative procedure to find a subset of the subspaces, so that all input datapoints are covered by at least one fuzzy subspace. Therefore, the resulting RBF NN relies only on the number of fuzzy sets *s*, which can be yielded through exhaustive search of a short range.

Chapter 4:
Model Predictive Control

The model predictive control class of control methods originate from the intersection of mathematical optimization and control theory [74]. The MPC algorithm makes use of a dynamic model of the plant in order to calculate an approximation of the plant's future response to the control inputs. Based on this model, a constrained optimization problem is formulated online to obtain the optimal sequence of control moves for a given time horizon. Due to being able to accommodate for plant nonlinearities, and its abilities to handle MIMO formulations, various physical constraints [75], and multiple objectives [76], MPC has emerged as a highly successful control scheme with applications in various fields during the last decades [74].

Consider the following discrete-time dynamical system:

$$x_{k+1} = f(x_k, u_k) \tag{4.0.1}$$

Here, $u_k \in \mathbb{R}^{n_u}$ is denoted as the input vector and $x_k \in \mathbb{R}^{n_x}$ the state vector. For generality's sake, (4.0.1) can also be written in form $x^+ = f(x, u)$. Given an initial state x_0 and a control sequence $u_0, \dots, u_{N-1}$ up to timestep $N - 1$, this discrete time system could be sequentially simulated to obtain $x_1, \dots, x_N$.

Next, the foundation of MPC controllers can be laid out in this section.

4.1 Linear Quadratic Regulator

As a precursor to MPC, it serves to initially present the linear quadratic regulator (LQR). First, using simple calculations of the system's (4.0.1) Jacobian around a specific origin, a linearized discrete-time system is created as follows:

$$x_{k+1} = Ax_k + Bu_k \tag{4.1.1}$$

Where $A \in \mathbb{R}^{n_x \times n_x}$ is the state transition matrix and $B \in \mathbb{R}^{n_u \times n_u}$ the input matrix. For generality in model representation, the output equation can be included,

$$y_{k+1} = Cx_{k+1} + Du_k \tag{4.1.2}$$

however, in this analysis it is set that $C = I$ (i.e. the states are fully measured) and $D = 0$. The origin of linearization is considered as the desired optimal control setpoint, which for simplicity is set to 0. Then, an objective function $J(\cdot)$ is defined measuring the deviation of sequences $\boldsymbol{u} = u_0, \dots, u_{N-1}$ and $x_1, \dots, x_N$ from the setpoint using the sum of squares:

$$J(x_0, u) = \frac{1}{2}\sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \tag{4.1.3a}$$

$$\ell(x_k, u_k) = Q^T x_k Q + R^T u_k R \tag{4.1.3b}$$

$$V_f(x_N) = {Q_f}^T x_N Q_f \tag{4.1.3c}$$

Here, $\ell(\cdot)$ is called the stage cost and $V_f(\cdot)$ the terminal penalty, while matrices Q, R represent tuning parameters. Large values of Q incentivize the quicker return of the state to the setpoint, while large values of R penalize large control actions. Matrix Q_f weighs the terminal state penalty. Using (4.1.3a), the optimal finite-horizon linear-quadratic (LQ) problem can be formulated:

$$\min_u J(x_0, u) \tag{4.1.4}$$

In order to ensure tractability of this optimal control problem (OCP), the Q, R and Q_f matrices must be symmetric and positive-semidefinite. Since x_0 is known, this problem can be formulated to be solved with backward dynamic programming (DP). A detailed presentation is beyond the scope of this thesis, however the main results are stated; the optimal control policy resulting from the finite-horizon Problem (4.1.4) at each stage k is:

$$u_k^\star = K(k)x \quad k = N-1, N-2, \dots 0 \tag{4.1.5}$$

Where $K(k)$ is the Riccati matrix,

$$K(k) = -(B^T \Pi(k+1)B + R)^{-1} B^T \Pi(k+1)A \quad k = N-1, N-2, \dots, 0 \tag{4.1.6}$$

$\Pi(\cdot)$ is the Riccati iteration defined as

$$\Pi(k-1) = Q + A^T \Pi(k)A - A^T \Pi(k)B(B^T \Pi(k)B + R)^{-1} B^T \Pi(k)A \tag{4.1.7}$$

$$k = N, N-1, \dots, 0$$

With $\Pi(N) = Q_f$, as the iterates are calculated backward. Lastly, the optimal cost-to-go is calculated as follows for up to time N:

$$J_k^0 = \frac{1}{2} x^T \Pi(k) x, \;\; k = N, N-1, \dots, 0 \tag{4.1.8}$$

The above notions can be extended for the infinite horizon LQ case, with objective function:

$$J(x_0, u) = \frac{1}{2} \sum_{k=0}^{\infty} \ell(x_k, u_k) + V_f(x_k, u_k) \tag{4.1.9}$$

By again iterating the Riccati equation, the so-called optimal infinite horizon control law together with the optimal cost-to-go are yielded:

$$u_\infty^0 = Kx \quad V_\infty^0(x) = \frac{1}{2} x^T \Pi(k) x \tag{4.1.10}$$

where the gain and Riccati iteration are defined as:

$$K = -(B^T \Pi B + R)^{-1} \, B^T \Pi A \tag{4.1.11a}$$

$$\Pi = Q + A^T \Pi A - A^T \Pi B (B^T \Pi B + R)^{-1} B^T \Pi A \tag{4.1.11b}$$

It is noted that control law (4.1.10) has guaranteed nominal stability given controllability for the (A, B) pair and $Q \succcurlyeq 0, R \succcurlyeq 0$, because then a positive-definite solution of (4.1.11b) exists and the eigenvalues of $(A + BK)$ are less than unity.

4.2 Nonlinear Tracking MPC

Even though the infinite-horizon control law u_∞^0 generated by DP has guaranteed nominal stability under mild assumptions, there are multiple practical impediments to its use. These pertain to the desirability of feedback due to uncertainty, the handling of nonlinear dynamical systems and the incorporation of constraints in the OCP. The idea of MPC is to generate the optimal control sequence $u_k^\star$ online for the current timestep given the current state x_0 and a finite time horizon, while honoring constraints on states and inputs. Consider the constraints of the nonlinear discrete-time system (4.0.1):

$$(x,u) \in \mathbb{Z} \tag{4.2.1}$$

Where $\mathbb{Z}$ denotes the admissible set of states and inputs $\mathbb{X} \times \mathbb{U}$, where $\mathbb{X}$ and $\mathbb{U}$ are the admissible state and input spaces, respectively. Space $\mathbb{Z}$ is generally required to be a convex polyhedron and can include inequalities, equality constraints, and rate constraints. The OCP pertaining to a constrained tracking nonlinear MPC (NMPC) is formulated as:

$$\min_{x,u} J_N(x,u) \tag{4.2.2a}$$

$$\text{s.t.}\,(x,u) \in \mathbb{Z},\;\; x_N \in \mathbb{X}_f \tag{4.2.2b}$$

$$x_0 = x_0' \tag{4.2.2c}$$

$$x^+ = f(x,u) \tag{4.2.2d}$$

Where the stage cost $\ell(\cdot)$ and the terminal penalty $V_f(\cdot)$ are denoted as (4.1.3b), (4.1.3c) and

$$J_N(x,u) = \sum_{k=0}^{N-1} \ell(x_k,u_k) + V_f(x_N) \tag{4.2.3}$$

An additional constraint set $\mathbb{X}_f \subseteq \mathbb{X}$ is also included, which represents the terminal constraint set. An alternative representation to OCP (4.2.2) can be constructed; considering space $\mathbb{X}_f$ together with $\mathbb{Z}$, the implicit u constraint $u \,\epsilon\, \mathcal{U}_N$ is formed, denoted as:

$$\mathcal{U}_N(x) \coloneqq \{u|(x,u) \in \mathbb{Z}_N\} \tag{4.2.4}$$

Where $\mathbb{Z}_N \coloneqq \{(x,u)|(\varphi(k;x,u),u(k)) \in \mathbb{Z}, \forall k \in \mathbb{I}_{0:N-1}, \varphi(N;x,u) \in \mathbb{X}_f\}$, and φ is some continuous function. Then, (4.2.2) can be written

$$J_N^0(x) \coloneqq \min_u\{J_N(x,u)|u \in \mathcal{U}_N(x)\} \tag{4.2.5}$$

Note that this optimization problem with a design variable of u is parametrized by x, inserted in both the cost $J_N(x,u)$ and the constraint set $\mathcal{U}_N(x)$. Next, $\mathcal{X}_N$ is defined, which represents the subset of $x \in \mathbb{X}$ for which (4.2.5) has a solution:

$$\mathcal{X}_N \coloneqq \{x \in \mathbb{R} | \mathcal{U}_N(x) \neq \emptyset\} \tag{4.2.6}$$

Then, the following Lemma can be stated.

Lemma 4.2.1 (A solution exists for OCP (4.2.5)): *Suppose the following assumptions hold:*

A. The functions $f\colon \mathbb{Z} \to \mathbb{R}^n$, $\ell\colon \mathbb{Z} \to \mathbb{R}_{\geq 0}$, $V_f\colon \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ are continuous and, without violating generality, a zero origin is assumed so that $f(0,0) = 0$, $\ell(0,0) = 0$, $V_f(0) = 0$.

B. The state-input admissible set $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ is closed, $\mathbb{U}$ and $\mathbb{X}_f \subseteq \mathbb{X}$ are compact, and all sets contain the origin.

Then:

1) Cost function $J_N(x,u)$ is continuous in $\mathbb{Z}_N$.

2) The control constraint set $\mathcal{U}_N(x)$ is compact.

3) $\forall x \in \mathcal{X}_N$ a solution to Problem (4.2.5) exists.

Proof: See *Proposition 2.4* [20] ■

Finally, the MPC control law for timestep k can be formulated,

$$\kappa_N(x) \coloneqq u_0^\star \tag{4.2.7}$$

where $u_0^\star$ is the first element of the optimal control sequence yielded by the solution of (4.2.5):

$$u^\star = \arg\min_u \{J_N(x,u) | u \in \mathcal{U}_N(x)\} \tag{4.2.8}$$

4.3 Stability for MPC

Classically, the stability of the MPC control law presented in the previous section employed Lyapunov-based tools and theorems. Some definitions regarding stability theory are in order; First, the origin $(0,0) \in \mathbb{R}^n$ is called *globally asymptotically stable* (GAS) for the system $x_{k+1} = f(x_k, u_k)$ if the origin is *locally stable* and if the origin is also *globally attractive*. The definition of local stability is: if $\forall \varepsilon \in \mathbb{R}_+$ a $\delta \in \mathbb{R}_+$ exists so that $|x| > \delta$ implies $\{|\varphi(k;x)| < \varepsilon \mid \forall k \in \mathbb{I}_+\}$, which alternatively is stated as: for small state perturbations from the origin, subsequent perturbations are also small. Similarly, the origin is called *globally attractive* if $\{|\varphi(k;x)| \to 0 \mid k \to \infty,\ \forall x \in \mathbb{R}^n\}$. Next, some useful function classes must be defined: a function is called $\mathcal{K}$-class if it is continuous, zero at the origin, and strictly increasing; next, a $\mathcal{K}_\infty$-class function is an unbounded $\mathcal{K}$-class function; and lastly, a continuous function $\beta(\cdot)$ is a $\mathcal{KL}$-class function if $\forall k \in \mathbb{I}_+$, $\beta(\cdot,k)$ remains a $\mathcal{K}$-class function and $\{\beta(s,i) \to 0 \mid i \to \infty,\ \forall s \in \mathbb{R}_+\}$, while $\beta(s,\cdot)$ remains nonincreasing.

Finally, the definition of a Lyapunov function can be presented: Suppose that the admissible state space $\mathbb{X} \subseteq \mathbb{R}^n$ of dynamical system $x_{k+1} = f(x_k, u_k)$ is positive invariant (i.e. if $x(t) \in \mathbb{X} \Rightarrow x(\tau) \in \mathbb{X}\ \forall \tau > t$). Then, a function $J: \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a Lyapunov function in $\mathbb{X}$ if there exist $a_1, a_2 \in \mathcal{K}_\infty$ and a positive-definite continuous a_3 function so that $\forall x \in \mathbb{X}$ it holds:

$$J(x) \geq a_1(|x|) \tag{4.3.1a}$$

$$J(x) \leq a_2(|x|) \tag{4.3.1b}$$

$$J\big(f(x)\big) - f(x) \leq -a_3(|x|) \tag{4.3.1c}$$

The basic Lyapunov stability theorem can now be stated.

Theorem 4.3.1 (Lyapunov Stability Theorem): *Suppose a system $x^+ = f(x,u)$, $x \in \mathbb{X}$ with $\mathbb{X}$ positive invariant. If a Lyapunov function exists in $\mathbb{X}$ for this system, then the origin is asymptotically stable in $\mathbb{X}$, and if $\mathbb{X} = \mathbb{R}^n$, then the origin is GAS. Lastly, if there also exists $a_i(|x|) = c_i|x|^a$ with a, $c_i \in \mathbb{R}_{>0}\ \forall i \in \{1,2,3\}$, then the origin is called exponentially stable.*

Proof: See *Theorem 2.13* [20]. ■

Therefore, the idea of MPC stability is to show that the value function $J_N^0\left(f\big(x, \kappa_N(x)\big)\right)$ of the respective OCP is a valid Lyapunov function. It can be shown that when the MPC constituents ℓ, V_f, and $\mathbb{X}_f$ are appropriately chosen, then it holds that $J_N^0\left(f\big(x, \kappa_N(x)\big)\right) \leq J_N^0(x) - \ell(x, \kappa_N(x))$.

Theorem 4.3.2 (Asymptotic stability of the origin): *Suppose assumptions of Lemma 4.2.1 (A-C) are satisfied as well as*

A. $f(x,u) \in \mathbb{X}_f$, *and* $V_f\big(f(x,u)\big) - V_f(x) \leq -\ell(x,u)\ r$.

B. *There exist* $\mathcal{K}_\infty$ *functions* a_1, a_f *so that* $\{\ell(x,u) \geq a_1(|x|) \mid \forall(x,u) \in \mathbb{Z}\}$ *and* $\{V_f(x) \leq a_f(|x|) \mid \forall x \in \mathbb{X}_f\}$.

C. *Weak controllability exists, namely for α* $\mathcal{K}_\infty$*-class* $a(\cdot)$ *function it holds* $\{J_N^0(x) \leq a(|x|) \mid \forall x \in \mathcal{X}_N\}$ *(Alternatively* $\mathbb{X}_f = \{0\}$*, i.e., there exists a terminal equality constraint to the origin).*

D. $\mathcal{K}_\infty$*-class* $a_1(\cdot)$ *and* $a_2(\cdot)$ *functions exist such that* $\{a_1(|x|) \leq J_N^0(x) \leq a_2(|x|)\}$ *and* $\left\{J_N^0\left(f\big(x,\kappa_N(x)\big)\right) - J_N^0(x) \leq -a_1(x,\kappa_N(x)\right\}$

Then, the origin of $x_{k+1} = f(x_k,\kappa_N(x_k))$ *is asymptotically stable in* $\mathcal{X}_N$.

Proof: This proof is beyond the scope of this presentation and can be found in [20], see *Proposition 2.15* and *2.16*; however, it is noted that the lower bound property for $J_N^0(x)$ is satisfied by opting for a positive definite stage cost $\ell(x,u)$, with Q and R also being positive definite. ■

A natural question to ask is whether the existence of a terminal constraint set $\mathbb{X}_\mathrm{f}$ is mandatory for asymptotic stability. There is significant motivation for its omission, since an OCP without terminal constraints is easier to solve, and the performance of the respective MPC is usually higher. Consider an OCP which satisfies Assumptions A, B of Lemma 4.2.1 & Assumption A of Theorem 4.3.2 meaning that there exists a local control law $\kappa_f\colon \mathbb{X}_\mathrm{f} \rightarrow \mathbb{U}$ which is a Lyapunov function in $\mathbb{X}_\mathrm{f}$ (a quadratic and positive definite $\mathrm{V_f}$ fulfils this). The idea is to implicitly satisfy the requirement of a terminal constraint by replacing $\mathrm{V_f}$ with $\beta\mathrm{V_f}$, where $\beta \geq 1$ a sufficiently large parameter so that the optimal terminal state is nevertheless within $\mathbb{X}_\mathrm{f}$. The full result by Limon et al. [77] is stated:

Theorem 4.3.3 (Stability at the origin; terminally unconstrained MPC): *Suppose an OCP problem with* $J_N^\beta(x,u) = \sum_{k=0}^{N-1}\ell(x_k,u_k) + \beta V_f(x_N)$*, and the associated x-parametrized problem* $J_N^{0,\beta}(x) = \min\limits_u\left\{J_N^\beta(x,u) | u \in \mathcal{U}_{\beta,N}(x)\right\}$*. Then, the region of attraction* $\Gamma_N^\beta \coloneqq \left\{x | J_N^{0,\beta} \leq Nd + \beta a\right\}$ *where* $d > 0$ *such that* $\ell(\mathrm{x},\mathrm{u}) \geq d$ *is positive invariant for the system* $f(\cdot)$.

Proof: See paper [77]. ■

The importance of this theorem is that if we require that the initial state x_0' to lie in the region of attraction Γ_N^β, then the terminal constraint is implicitly satisfied and is thus not needed to be included in the OCP formulation.

4.4 Nonlinear Economic MPC

The NMPC control law presented in the previous section is well-equipped to handle tracking problems. These include vehicle navigation and guidance, disturbance rejection, setpoint tracking of low-level objectives present in industrial processes and others, with great performance. However, the high-level objectives of such processes dictate the direct, online optimization of economic goals, which may be generic – i.e., not representable by tracking cost formulations. This is the motivation behind the creation of economic MPC.

Consider once again a discrete-time dynamical system,

$$x^+ = f(x,u) \tag{4.4.1}$$

and a generic cost function $\ell_e(\mathrm{x},\mathrm{u})$ which does not measure deviation from an origin, but rather an economic objective of the process; this implies that this cost function may not be positive definite. The optimal economic steady state (x_s,u_s) of system resulting from the pair $\big(f(\cdot),\ell_e(\cdot)\big)$ can be calculated as:

$$(x_s,u_s) \ = \arg\min_{(x,u)\in\mathbb{Z}}\{\ell_e(\mathrm{x},\mathrm{u})|x^+ = f(x,u)\} \tag{4.4.2}$$

Typically for standard tracking NMPC, this steady-state (x_s,u_s) would be set as the setpoint with the stage cost formulation, requiring that (x_s,u_s) is tracked. In EMPC however, $\ell_e(\mathrm{x},\mathrm{u})$ is used directly as a stage cost. It can be shown that a solution to its OCP exists.

Theorem 4.4.1 (A solution to the economic OCP exists): *Consider the following OCP problem:*

$$J_{e,N}^0(x) \coloneqq \min_u\{J_{e,N}(x,u)|u \in \mathcal{U}_N(x)\} \tag{4.4.3a}$$

$$J_{e,N}(x,u) = \sum_{k=0}^{N-1}\ell_e(x_k,u_k) \tag{4.4.3b}$$

Together with the following assumptions

A. *The functions $f\colon \mathbb{Z}\to\mathbb{R}^n$, $\ell_e\colon\mathbb{Z}\to\mathbb{R}_{\geq 0}$ are continuous, meaning that a (x_s,u_s) exists (4.4.2).*

B. *The state-input admissible set $\mathbb{Z}=\mathbb{X}\times\mathbb{U}$ is closed. $\mathbb{U}$ is compact and uniformly bounded in $\mathbb{X}$.*

C. *There is no terminal cost function V_f, instead the terminal set contains only the optimal steady state $\mathbb{X}_f = \{x_s\}$, and ℓ_e is lower bounded.*

Then, a solution to Problem (4.4.3) exists.

Proof: See Proposition 2.4 [20]. ■

Theorem 4.4.1 implies that the EMPC control law $\kappa_{e,N}$ is well defined and the closed loop system is $x^+ = f(x, \kappa_{e,N}(x))$. Note that $J_{e,N}^0$ is not a Lyapunov function due to the non-positive definite stage cost ℓ_e. Still, it can be shown that control law $\kappa_{e,N}(x)$ can yield an average closed-loop performance that is economically better than the optimal steady-state performance (x_s, u_s) [78]; the practical effect is that control law $\kappa_{e,N}(x)$ would periodically cycle the system around the steady-state (x_s, u_s), instead of stabilizing towards it.

4.4.1 Dissipativity and Stability Analysis

The notion of dissipativity is central to the stability analysis of the EMPC control law $\kappa_{e,N}(x)$. The idea of dissipativity has roots in physical systems: Consider mechanical energy being supplied to a system by performing work at a given rate. This energy can be considered as a storage function, translating work on the system to stored energy. For a dissipative system, the rate of change in the storage function (i.e., stored energy) is strictly less than the supplied work. Simply put, the system $x^+ = f(x, u)$ is characterized as dissipative w.r.t supply rate $s: \mathbb{Z} \to \mathbb{R}$ if a storage function $\lambda: \mathbb{X} \to \mathbb{R}$ exists such that $\forall (x, u) \in \mathbb{Z}$:

$$\lambda\big(f(x,u)\big) - \lambda(x) \leq s(x,u) \tag{4.4.4}$$

In addition, strict dissipativity holds w.r.t s and a x_s, if a $a(\cdot) \in \mathcal{K}_\infty$ exists such that $\forall (x, u) \in \mathbb{Z}$:

$$\lambda\big(f(x,u)\big) - \lambda(x) \leq s(x,u) \tag{4.4.5}$$

Using the notion of a storage function and dissipativity, one can state the following:

Theorem 4.4.2 (Asymptotic stability of EMPC for dissipative systems): *Suppose the assumptions of Theorem 4.4.1 hold and*

A. *The function $\{\tilde{J}_{e,N}^0 = J_{e,N}^0(\cdot) + \lambda(\cdot)\}: \mathcal{X}_N \to \mathbb{R}$ is continuous at the steady state (x_s, u_s)*

B. *The system $f(x, u)$ is strictly dissipative with supply rate $s(x, u) = \ell_e(x, u) - \ell_e(x_s, u_s)$*

Then, the steady-state (x_s, u_s) of $x^+ = f(x, \kappa_{e,N}(x))$ is asymptotically stable in $\mathcal{X}_N$.

Proof: Function $J_{e,N}^0(\cdot)$ is no longer a Lyapunov function since $\ell_e(\cdot)$ is not positive definite. Reference [79] constructs a new, rotated stage cost of the form:

$$\tilde{\ell}(x,u) = \ell_{\mathrm{e}}(x,u) - \ell_{\mathrm{e}}(x_s,u_s) + \lambda(x) - \lambda\big(f(x,u)\big) \tag{4.4.6a}$$

Since the system is dissipative, it holds that the rotated stage cost $\tilde{\ell}(x,u)$ has the properties:

$$\tilde{\ell}(x,u) \geq a(|x - x_s|) \quad \& \quad \tilde{\ell}(x_s,u_s) = 0 \tag{4.4.6b}$$

This means that Assumption B of Theorem 4.3.2 is satisfied, and the standard procedure can be employed in order to show that the new OCP $\tilde{J}_{e,N}^{0}$ is indeed a Lyapunov function, therefore proving asymptotic stability. ■

The stability Theorem 4.4.2 includes periodic processes and time-varying systems. It should be noted, however, that the creation of the stability-enforcing storage function λ(·) is an ad-hoc procedure, meaning that it may be difficult to compute it for most practical applications.

Part III: Design & Applications

Chapter 5:

Computational Intelligence Methods for efficient Smart Grid Dispatch

Electricity generation and distribution has undergone a major development in the last decades moving from a conventional centralized generation towards a distributed, small-scale, producer-consumer (prosumer) model, connected to the distribution network [80]. This evolution has been made possible by the development of a reliable information and communication infrastructure, but it also gave rise to certain challenges, which were met through the emergence of the smart grid [81]. The technological framework defined by the smart grid enables a more reliable, more efficient, and more economical operation, capable of accommodating increased utilization of renewable energy sources (RES) and energy storage systems. On one hand, system operators now have an abundance of incoming information and available control decisions at their disposal in order to control critical network state variables. On the other hand, they must deal with modern challenges that arise in the typical operational tasks of the grid [80]. These tasks can be broadly categorized into two classes of problems, namely optimal power flow (OPF), and resource scheduling (see fig. 5.0.1). However, there also exists a ubiquitous utility task pertaining to the electric load forecasting, which encompasses the operational tasks residing in the OPF and resource scheduling categories.

From an optimization perspective, this new distributed generation paradigm offers significant potential for the application of new methods, capable of handling the aforementioned challenges. These methods must be able to cope with a larger amount of design variables of diverse nature, while taking into account a significant number of

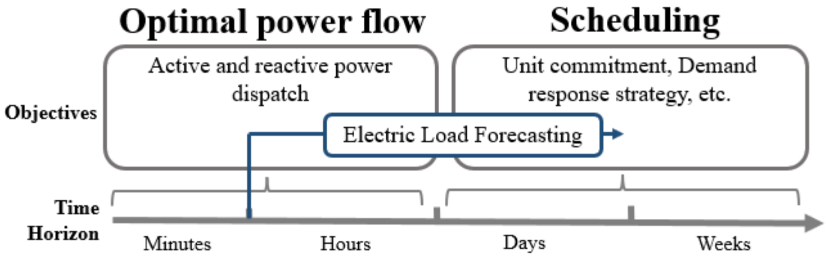

Optimal power flow
Scheduling
Objectives
Active and reactive power dispatch
Unit commitment, Demand response strategy, etc.
Electric Load Forecasting
Time Horizon
Minutes
Hours
Days
Weeks

Figure 5.0.1: The current paradigm in smart grid dispatch

incoming state measurements from the grid. In addition, they must provide satisfactory solution accuracy combined with reasonable computational complexity, even when applied to multiple optimization objectives.

This chapter employs computational intelligence methods in order to tackle the two main problems related to smart grid dispatch; the first case-study is the load forecasting task using an ensemble of multiple machine learning methods, while the second is the OPF problem using a PSO-based metaheuristic method.

5.1 Electrical Load Prediction using Machine Learning method ensembles

The increasing penetration of renewable energy sources tends to redirect the power systems community's interest from the traditional power grid model towards the smart grid framework. During this transition, load forecasting for various time horizons constitutes an essential electric utility task in network planning, operation, and management. This chapter presents a novel mixed power-load forecasting scheme for multiple prediction horizons ranging from 15 min to 24 h ahead. The proposed approach makes use of a pool of models trained by several machine-learning methods with different characteristics, namely neural networks, linear regression, support vector regression, random forests, and sparse regression. The final prediction values are calculated using an online decision mechanism based on weighting the individual models according to their past performance.

The proposed scheme is evaluated on real electrical load data sensed from a high voltage/medium voltage substation and is shown to be highly effective, as it results in R^2 coefficient values ranging from 0.99 to 0.79 for prediction horizons ranging from 15 min to 24 h ahead, respectively. The method is compared to several state-of-the-art machine-learning approaches, as well as a different ensemble method, producing highly competitive results in terms of prediction accuracy.

5.1.1 Introduction & Literature review

Generation intermittency of RES within the smart grid context has a two-fold effect: from the distribution system operator (DSO) perspective, uncertainty in distributed generation compromises the ability to effectively plan short-term power dispatch [82], while from the energy market bidder perspective, stochasticity severely constrains their bidding strategy and thus, reduces profit margins [83]. These shortcomings underline the importance of the application of effective electric load prediction models in the context of multiple operational aspects of the smart grid, such as power stability and security. Especially in the case of micro-grids, storage management is critical and cannot be accomplished without the aid of accurate short-term load forecasts for load shifting and balancing operations [84]. Moreover, the grid extension and the increasing exploitation of smart meters affect the efficient operation of the grid, leading to a complex and multifaceted framework [85]. Regarding the distribution network on the substation level, load forecasting up to one day ahead, could be a valuable asset in the grid's optimization tasks [5]. Such actions can be carried out, not only by controlling the on-load tap changer and capacitor bank movements, which is currently the industry standard, but also by operation scheduling of batteries in the near future. Load forecasting with multiple time horizons participates in different, interdependent levels of operation of a power grid and thus can make a significant contribution to addressing the aforementioned challenges. A pivotal feature of smart grid is the bidirectional power flow and communication through administrators of generation, transmission, distribution, and end-users. As a result, the corresponding energy data contain mixed power-load (hereby referred to as 'mixed load'). While the majority of the load forecasting models found in the literature predict the electric load production or consumption, the differences between them are frequently studied as well. The forecasting of the so-called net load proves useful for tackling load volatility due to increasing RES penetration [86]. These net energy load prediction models utilize historic distribution grid load data as well as measurements of weather features on a substation level in order to infer the net active power (AP) demand of the distribution grid.

To this end, the field of computational intelligence, and more specifically, the branch of machine learning [87] has proved to be an invaluable source providing a multitude of approaches to solving the aforementioned problem. ML comprises methodologies that are capable of extracting knowledge from historical data in order to develop black-box models, tackling the problem of requiring specific information about the process and avoiding altogether the computationally intensive use of first-principle equations. ML algorithms can exhibit a number of important advantages like efficiency, increased prediction accuracy, robustness, etc, but require a number of suitable data to do so. Notable ML methodologies used within load forecasting pertain to linear regression (LR) [88], sparse coding [89] & support vector regression [90], as well as more advanced neural network approaches such as feed-forward neural networks [91], RBF networks [28], and others. For a detailed literature review, the interested reader is referred to [7] as well as Chapter 3.

It is apparent from the literature that the problem of electric load forecasting has been addressed by multiple machine-learning methods, but without any of them achieving universal superiority in terms of performance. This observation is confirmed not only by studying the individual research results but also by assessing various benchmark comparisons in the literature [92]. The inability of universal prediction effectiveness of the aforementioned models is to be expected taking into account the undesirable characteristics of the load forecasting problem, which include non-linearities and high levels of noise in the associated data. Furthermore, load time series are not statistically static [93], due to the volatile, rapidly changing nature of the weather conditions that affect their power generation component. Different classes of machine-learning methods can cope better with some of these issues but usually underperform with respect to others, e.g., linear models are more robust to noise but cannot capture the non-linearities present in the load forecasting problem. To make things worse, though all of these problems are inherent to load forecasting, their mixture composition changes depending on the time horizon one tries to predict for, making it impossible to single out a unique machine-learning method that could outperform the others across different prediction horizons, e.g., linear methods are often found to perform better in short-term horizons, where data tend to be noisy, but the non-linearities can usually be adequately approximated by linear models, but mostly fail in longer time horizons, where the role of the non-linearities is dominant. It should be noted that the previous observation about the inability of a single method to beat all the others is not only tied to the context of load forecasting, but reveals a more generic concept in machine-learning and optimization, as expressed by the "no free lunch" theorem [94].

To remedy this predicament, one could resort to using a multi-model approach [95], combining multiple machine-learning methods. Unfortunately, in a real-time deployment scenario, an important practical consideration arises for multi-model schemes: How does one select the most suitable model from a pool of trained models for the next prediction timestep? One solution is to employ a rule-based decision system that uses a priori available knowledge, such as the time of day and measured weather conditions at the substation level. This presents a significant impediment. Not only are the rules of such a system difficult to conceptualize, but they also offer no guarantee of continuously optimal model selection. Doing away with a decision system altogether is also problematic since the individually generated predictions do not offer any actionable insight by themselves. A practical workaround is to discard such selection rules and instead employ a weighting system that assesses models only by using their past prediction performance [96]. The weighting of the output results of basic forecasting LSTM models in [97] is based on the similarity degree between target and identified standard values of load consumption. Two different approaches for determining the weights of multiple forecasters are followed in [98], using a novel incremental ensemble weight updating strategy and the minimum-error method, respectively. Alternatively, an extreme learning machine can be employed for combining the outputs of a pool of forecasts, as in [99]. An intelligent decision-making support scheme, including predictive performance evaluation, model properties analysis, structure and fusion strategy optimization, and optimal model preference selection, is incorporated with an evolutionary ensemble learning method proposed in [100] for short-term load forecasting (STLF) problems. Finally, an automated system is established in [101] based on hidden Markov chains for extracting similar day profiles to obtain the best model from a library of available forecasting models. Differently from the previous works, the output neural network (NN) models result from multiple training cycles based on snapshots [102] or the hidden features of a Random Vector Functional Link network [103].

It has become clear that the necessity of providing mixed load forecasts, and indeed for multiple short-term horizons, is a factor of paramount importance in the upcoming transition to smart electricity grids. Moreover, according to the preceding literature review, it is evident that in order to enhance the predictive capability of a model, it should incorporate more than one machine-learning methodologies, which of course should be able to handle the complex dynamic behavior of the mixed load. Finally, such a methodology is necessary to be applicable in an online implementation, which means that the final predictions should be provided in a reasonable amount of time and respond to the behavior of the load through a dynamic decision mechanism.

Realizing the aforementioned requirements and seeking to fill the corresponding research gaps, a novel forecasting scheme is presented here that is able to efficiently address the diverse and adverse characteristics of the load forecasting problem for various prediction horizons. The proposed method seeks to create an ensemble of prediction models based on multiple machine-learning techniques comprising different beneficial characteristics that have only been used individually for load forecasting before. Indeed, the sparse coding method introduced in the proposed model has been published very recently and used for the first time in ensemble schemes. As the participating techniques excel in different aspects of the load forecasting problem, their combined usage introduced in this work provides the ensemble with the ability to outperform each individual method in all the horizons tested. In order to efficiently combine the different machine-learning techniques, the proposed method employs an error-based metric on a rolling window of past predictions. This approach enhances the novelty of the proposed method as it does away with the adversity exhibited by complex, rule-based model selection systems. By combining the beneficial characteristics of the aforementioned techniques, the proposed scheme demonstrates superior performance in terms of prediction accuracy, compared to all the submodels, as well as a recently proposed MLP model ensemble from the literature [104], through a wide range of different prediction horizons, spanning from 15 min to 24 h-ahead. Thus, reliable forecasts can be obtained for: (a) One hour ahead or less, which are valuable for various applications at the transmission and distribution network, (b) one day ahead, contributing to the scheduling of generation sources and (c) intra-day forecasting, so as to achieve better optimization results. As a result, the introduced model ensemble can become a powerful tool for administrators and participants in the energy market, easily exploitable in both operational and managerial tasks of smart grids. It should be noted that, at least to the author's best knowledge, no machine-learning approach that is able to handle this range of prediction horizons has been proposed in the literature. Furthermore, the proposed approach expands the existing literature by using mixed power-load data, i.e., data that include renewable generation measurements. Although there is an abundance of work in forecasting the net power load, the literature on mixed-load forecasting is very scarce. It should be pointed out that the employment of mixed measurements is aligned with the requirements of modern smart grids, where the penetration of renewable resources is a key feature.

5.1.2 The load forecasting objective

The load forecasting problem that is of interest in this work spans several different time horizons: 15 min, 1-hr, 2-hr, 3-hr, 6-hr, and 24-hr. This case study makes use of real data from a high voltage/medium voltage substation located in mainland Europe, measured during the years 2017–2018. The corresponding MV distribution network contains multiple photovoltaic sites; as a result, the data measurements in question constitute mixed power-load recordings, which correspond to the mixed AP demand of the distribution grid from the transmission grid. The load measurements have been recorded every minute and contain the mixed AP demand, as well as cloud coverage, wind speed, humidity, and temperature, as measured from the substation's weather station. Due to practical concerns, individual power generation or weather data from the aforementioned photovoltaic sites should not be taken into account for the creation of the input dataset since these will normally not be available for a real-life implementation. In short, in this work, the substation's historical measurements of load and weather conditions are available for the creation of a prediction model of the mixed AP demand of the grid.

5.1.3 Methodology

5.1.3.1 The ensemble algorithm

Recognizing the individual advantages and disadvantages of the machine-learning methods, the proposed scheme seeks to create an ensemble that will successfully combine their merits in a single approach. For example, neural-network-based models such as RBF do exhibit superior prediction performance only as long as the input data point lies well within the domain of the input training dataset. On the other hand, linear and sparse prediction models, in general, show much better extrapolative performance, even though they are unable to capture more complex, non-linear dynamics. In other words, by toggling between the robust linear models and the more sensitive but also more effective non-linear ones, a superior approach to load time series prediction can be constructed.

In order to obtain the best possible performance of each sub-model, their optimal training configuration has to be determined. Starting with the simpler methods used, a linear and a sparse regression (SR) model are trained by least squares and fast iterative shrinkage thresholding algorithm, respectively, the latter being a faster implementation of the corresponding iterative shrinkage thresholding algorithm used for load forecasting [3]. In the case of the sparse coding approach, sparsity is induced by the ℓ_2 norm and the regularization parameter was set by trial and error to 0.01. Subsequently, a random forest regressor is employed, where the number of decision trees is selected to be 15 so as to keep the training time at a reasonable level without reducing its predictive ability. As regards the non-linear methods, an SVR model with Gaussian kernel function was developed [105], using sequential minimal optimization for training and Bayesian optimization to optimize the model's hyperparameters [106]. Two NN models are also introduced, featuring two different architectures. The first one is a two-layered MLP network trained by the Levenberg–Marquardt backpropagation algorithm [107], following a 10-fold cross-validation. The neurons of each layer are chosen by trial and error as 20 and 10. It is noted that, in order to compensate for the performance dependence of the MLP training methods to initialization, the training procedure was conducted 10 different times, with different randomly initialized weights of the network. The second NN uses an RBF architecture and is trained using the fuzzy means technique [108], an algorithm that has found many successful applications due to the increased accuracy it provides [4] combined with fast training times [27]. In this work, the FM algorithm has been tested for a range of fuzzy sets between 4 and 15. When deployed online, the proposed approach evaluates a MAE metric on a rolling window of past predictions coming from a pool of trained models in order to create a weight vector for the next timestep prediction. An important item of the proposed method to be specified is the length of the rolling window. It can be easily inferred that this depends not only on the prediction horizon but also on the statistical properties of the predicted variable (a more volatile, non-stationary time series would require shorter rolling window horizons). Once the model pool has been populated by trained models, the optimum length of the rolling window is calculated in an exhaustive search manner over the same validation data in the range of 3–15 regressive timesteps. The proposed method operates as follows: For each timestep k, all trained models in the pool are evaluated concurrently. Their

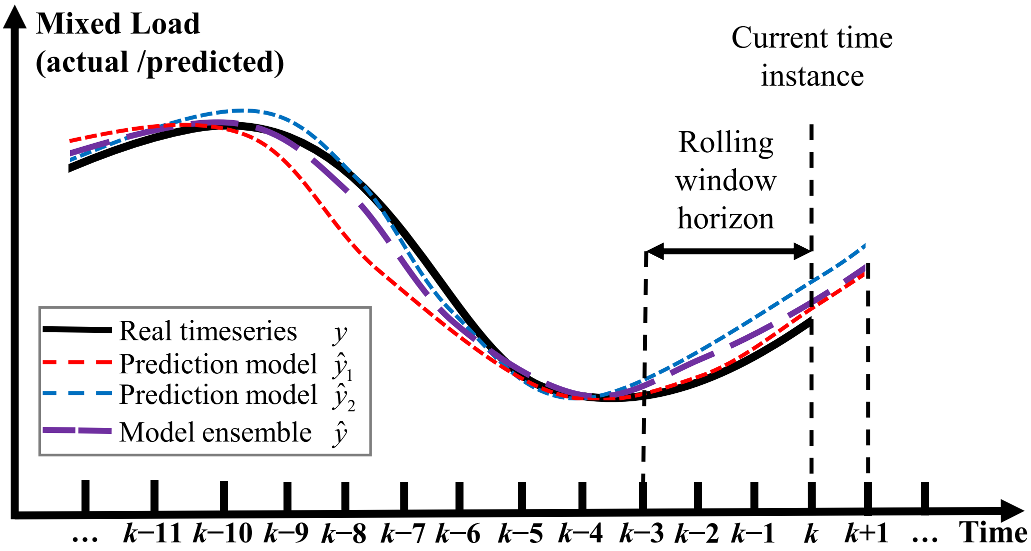


Figure 5.1.1: Schematic for a two-model version of the proposed method, where y denotes the real load, $\hat{\boldsymbol{y}}_i$ the prediction of the *i*-th model, $\hat{\boldsymbol{y}}$ the weighted prediction and *k* the current timestep. The ensemble model recognizes the superiority of $\hat{\boldsymbol{y}}_1$ over $\hat{\boldsymbol{y}}_2$ and, within the rolling window, adapts its weights accordingly, achieving highly accurate prediction for the next timestep $k + 1$.

current prediction performance is assessed by applying the MAE metric on their previous predictions up to a rolling time window of length h_w

$$MAE_i(k) = \frac{\sum_{j=0}^{h_w-1} |\hat{y}_i(k-j) - y(k-j)|}{h_w} \tag{5.1.1}$$

where $\hat{y}_i(k)$ are the predictions of the i-th model and y are the actual values of the timeseries at timestep *k*. Then, the MAE metric is used to calculate the prediction weight of each model for the next timestep $k + 1$

$$w_i(k+1) = \frac{MAE_i^{-1}(k)}{\sum_{i=1}^{N} MAE_i^{-1}(k)} \tag{5.1.2}$$

where MAE_i is the MAE of the *i*-th prediction model, *N* is the total number of models in the model pool, and w_i is the prediction weight for the next timestep. The prediction output of the proposed method is calculated as the weighted sum of the model predictions $\hat{y}_i$

$$\hat{y}(k+1) = \sum_{i=1}^{N} w_i(k+1)\, \hat{y}_i(k+1) \tag{5.1.3}$$

A snapshot of a two-model example version of the proposed method is shown in fig. 5.1.1. Note that the proposed method combines the strengths of the individual models by placing greater weight on the current better-performing model for the time window of length h_w. At first, both $\hat{y}_1$ and $\hat{y}_2$ models appear ineffective as individual predictors of the y time series. However, after closer inspection, $\hat{y}_2$ performs better for the first half of y, while $\hat{y}_1$ for the

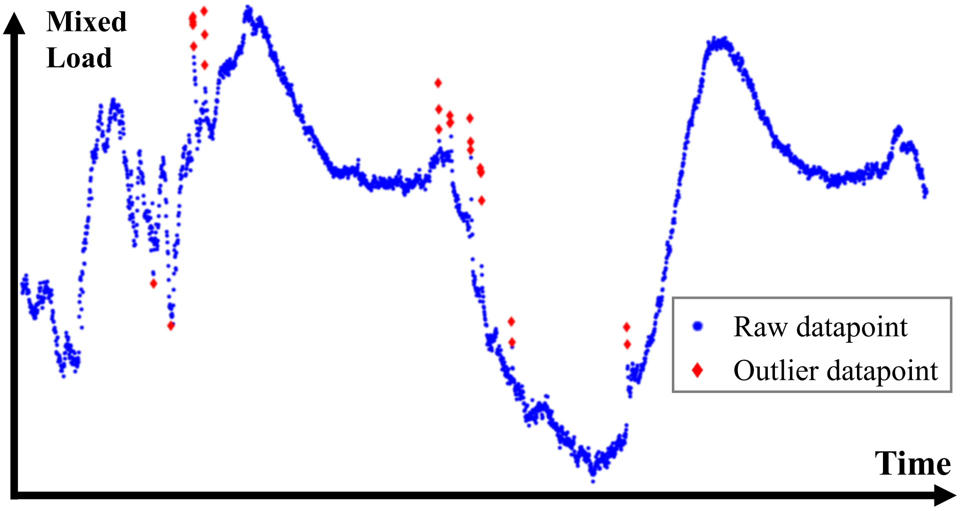


Figure 5.1.2: Operation of the rolling median threshold outlier detection algorithm. The data points marked as outliers exceed the median value of the time window multiplied by a user-specified threshold factor.

second half. By placing greater weight on the model with the best past prediction performance within the horizon h_w, the proposed method is able to toggle towards the best available model for the current circumstance. The result is an overall superior prediction performance.

5.1.3.2 Data Preprocessing and Model Training

Unavoidably, the substation measurements contain large periods of missing or corrupt data owing to sensor downtime or malfunction. For the scope of this case study, no missing data imputation has been performed-instead, corrupted data and outlier removal was the main focus of the preprocessing operation. Due to the sheer size of the dataset, manual preprocessing was impossible, mandating the creation of a bad data detection routine. Corrupted values were decidedly easy to detect since the corresponding AP signal exhibited unusually low variance around a constant value. However, outlier values on mixed load data were a challenge to successfully handle—a review of the challenges of this topic, as well as effective techniques, is available on [109]. The chosen technique must be sufficiently effective at classifying outliers in data, while avoiding false positives. In this case study, a rolling median window threshold approach is used, as it was found to compromise well between the aforementioned points. A two-day snapshot from the application of this algorithm to raw electrical load data is presented in Figure 5.1.2. The outliers usually originate from noisy sensor readings [110]. As part of data preprocessing,

Table 5.1.1. Description of training variables of the forecasting models for the different prediction horizons examined in the case study. Each row of the table refers to the different groups of input variables, whereas the last row refers to the output variable.

Prediction Horizon	15 min $(t+1)$	1 h $(t+4)$	2 h $(t+4)$	3 h $(t+12)$	6 h $(t+24)$	24 h $(t+96)$
Current and past AP measures	$p^{(t-i)}$, $i=0,95,671$	$p^{(t-i)}$, $i=0,4,92,668$	$p^{(t-i)}$, $i=0,8,88,664$	$p^{(t-i)}$, $i=0,12,84,660$	$p^{(t-i)}$, $i=0,24,72,648$	$p^{(t-i)}$, $i=0,96,576$
Average AP measures	$\frac{\sum_{n=0}^{3} p^{(t-n)}}{4}$	$\frac{\sum_{n=0}^{3} p^{(t-n)}}{4}$	$\frac{\sum_{n=0}^{7} p^{(t-n)}}{8}$	$\frac{\sum_{n=0}^{11} p^{(t-n)}}{12}$	$\frac{\sum_{n=0}^{23} p^{(t-n)}}{24}$	$\frac{\sum_{n=0}^{95} p^{(t-n)}}{96}$
Difference AP measures	$p^{(t)}-p^{(t-i)}$, $i=1$	$p^{(t)}-p^{(t-i)}$, $i=4$	$p^{(t)}-p^{(t-i)}$, $i=8$	$p^{(t)}-p^{(t-i)}$, $i=12$	$p^{(t)}-p^{(t-i)}$, $i=24$	$p^{(t)}-p^{(t-i)}$, $i=96$
Weather measures	$w^{(t+i)}$, $i=0$	$w^{(t+i)}$, $i=4$	$w^{(t+i)}$, $i=4,8$	$w^{(t+i)}$, $i=4,8,12$	$w^{(t+i)}$, $i=16,20,24$	$w^{(t+i)}$, $i=88,92,96$
Future AP forecasts (output variable)	$\hat{p}^{(t+1)}$	$\hat{p}^{(t+4)}$	$\hat{p}^{(t+8)}$	$\hat{p}^{(t+12)}$	$\hat{p}^{(t+24)}$	$\hat{p}^{(t+96)}$

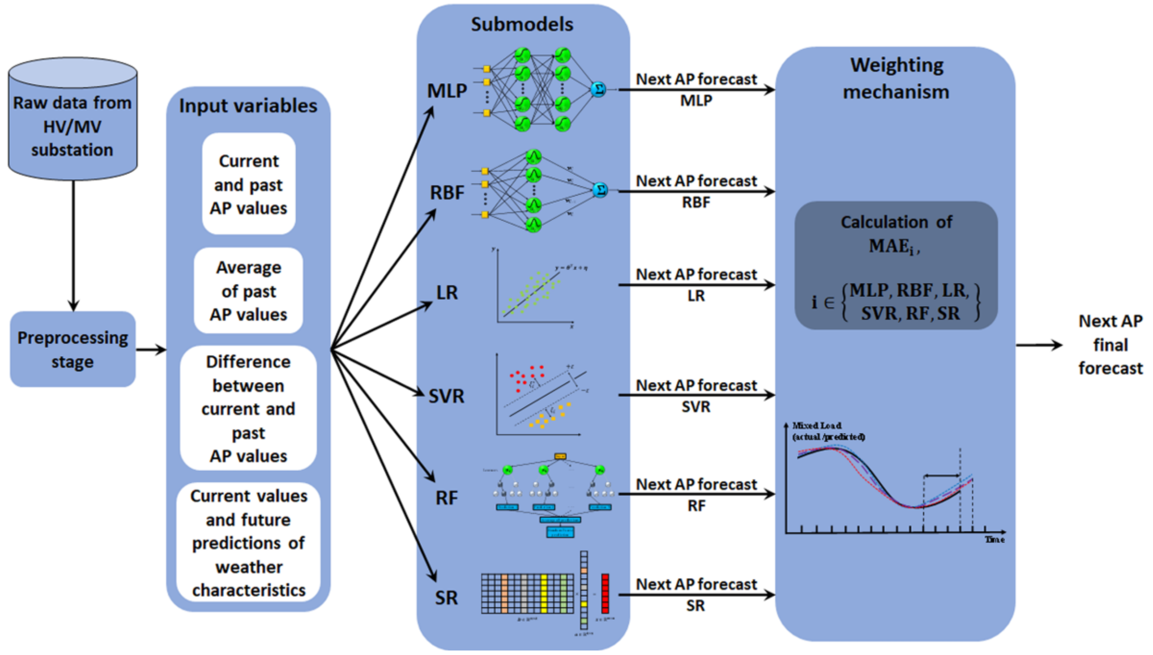

Raw data from HV/MV substation
Preprocessing stage
Input variables
Current and past AP values
Average of past AP values
Difference between current and past AP values
Current values and future predictions of weather characteristics
Submodels
MLP
RBF
LR
SVR
RF
SR
Next AP forecast MLP
Next AP forecast RBF
Next AP forecast LR
Next AP forecast SVR
Next AP forecast RF
Next AP forecast SR
Weighting mechanism
Calculation of MAE$_i$,
$i \in \{MLP, RBF, LR, SVR, RF, SR\}$
Mixed Load (actual/predicted)
Time
Next AP final forecast

Figure 5.1.3: Overview of the proposed model ensemble. Its application in mixed load forecasting comprises a series of steps, i.e. raw data acquisition, data preprocessing, collection of input variables, splitting of the dataset in a training and a testing subset, training of submodels, generation of the next AP forecast by each submodel, weighting of the individual predictions, and, lastly, calculation of the next AP final forecast.

a resampling step also took place, where each sample was defined as an average of 15 one-minute measurements.

The task of input variable selection is closely related to the prediction horizon. All models developed in the context of this study are considered autoregressive with exogenous variables, as they use inputs that consist of previous values of the output and weather data. A set of inputs was initially constructed for each prediction horizon based on the literature. Subsequently, the contribution of these variables to the prediction accuracy improvement was examined by trial and error, sometimes leading to shorter input sets for some of the horizons. Alternatively, other approaches, such as gradient boosting decision tree and Pearson correlation coefficient [111], attention mechanism [112], or Exploratory Data Analysis [113], are considered to have an effective contribution during input features reduction and selection. However, it is important to note that for each horizon, inputs remain the same for all machine-learning methods used in the present study.

The selected input variables which all models accept could be divided into 4 categories, as described in Table 5.1.1, namely (a) current and past AP values, (b) difference between current and past AP values, (c) average of past AP values, and (d) weather measurements. It has to be noted that $\boldsymbol{p}^{(t)}$ values contain the current and past, average and difference measures of the AP values, $\widehat{\boldsymbol{p}}^{(t+s)}$ is the output, i.e., the mixed power load s 15-minute intervals ahead, whereas $\boldsymbol{w}^{(t)}$ components contain the respective weather-related inputs of cloud coverage, wind speed, humidity, and temperature, respectively.

Once the preprocessing stage has been completed and input variables have been selected, the dataset was partitioned in a yearly manner in order to select the training datasets. At this point, an important consideration should be made. As mentioned in the introductory section, the load time series consists of a load and generation component. The statistical properties of both of these components are not static in relation to time, especially on a long-term scale. The network physically expands, incorporating more consumers as well as RES generators, each with different load and generation profiles, respectively. Therefore, it makes sense to select training datasets as close to the actual prediction interval as possible. Since the available data concern two successive years, the data corresponding to 2017 were selected as the training subset, and the data corresponding to 2018 were selected as the testing dataset. A point worth mentioning is that no permutation step is taking place before training. This means that the data used for testing are considered completely unseen for the proposed model, yielding a more reliable forecasting model. Due to confidentiality reasons, the real and predicted mixed load values have been normalized in order to be presented. Finally, it should be noted that models that require a validation step during training, namely models based on MLP and RBF NNs, do so using 10-fold cross-validation, while in the case of models that require multiple training runs for each training seed, the best-performing model on the validation data is kept. An overview of the implementation of the proposed model is provided in Figure 6, which illustrates, in the form of a block diagram, the entire sequence of steps that take place, starting from the acquisition of the raw AP data from the substation to the derivation of the final forecasts. It has to be highlighted that this figure is generic and does not refer to a particular prediction horizon.

At this point, it should be mentioned that in order to evaluate the accuracy of the proposed method, it was considered appropriate to compare it with a model ensemble from the literature. To be more specific, a method proposed for load forecasting based on an ensemble of multiple MLP neural networks is employed [104]. Consequently, following the experimental protocol described in this work, a number of feed-forward NNs, with a single hidden layer, were trained on 14 different random initializations of the weights. For each initialization, the number of neurons in the hidden layer ranged from 3 to 50. The hyperbolic tangent sigmoid function was selected as the transfer function among the NNs' layers, while all NNs were trained using the resilient backpropagation algorithm. The neural networks were arranged in ascending order with respect to the MAPE error on a common validation set, which, in this case, was defined as 20% of the training dataset. Then, the networks corresponding to the first 5 MAPE errors were selected, and the final forecasts were obtained by averaging the individual forecasts of these 5 models.

5.1.4 Results & discussion

In this section, the results of extensive simulations of the proposed model are presented. A set of scatterplots is shown in fig. 5.1.4, representing the actual versus the predicted values mixed load values for 1, 2, 3, 6, and 24-h-ahead horizons, respectively, through the whole testing dataset. The diagonal line implies a complete match between real values and forecasts. The axes are presented in units of normalized AP.

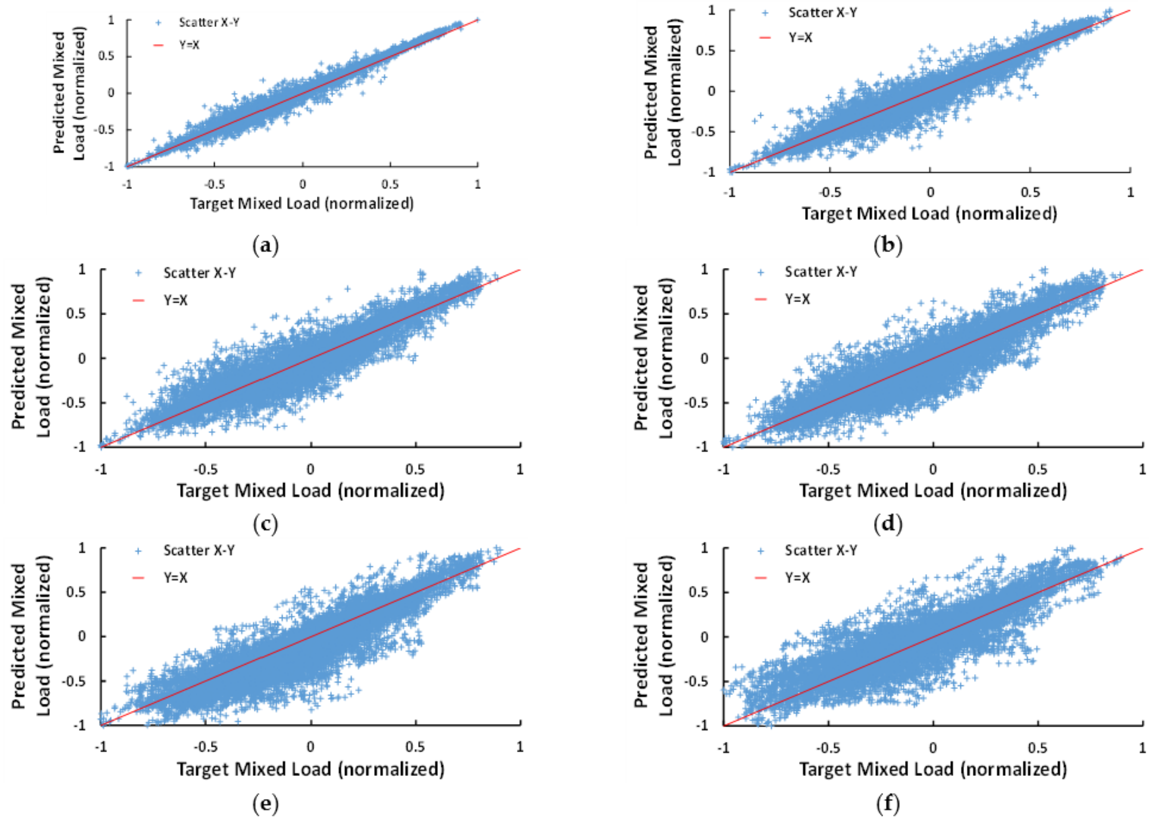


Figure 5.1.4 Scatterplots of actual versus predicted mixed load for (**a**) 15-min, (**b**) 1-h, (**c**) 2-h, (**d**) 3-I, (**e**) 6-h, and (**f**) 24-h ahead prediction. The predicted values residing on the diagonal line are identical to the actual values. Each mark refers to a data point and shows the deviation of its predicted value from its actual value.

Additional results are provided in Table 5.1.2, which contains information about the forecasting performance of the proposed method in comparison to the individual machine-learning methods comprising the model pool. In order to distinguish the results for different prediction time horizons, the table is divided into sections. The accuracy of model predictions is evaluated through the correlation coefficient (R^2), RMSE and MAE, considering them as representative and efficient criteria [114]. For comparative reasons, the table also contains the values of the indices for all submodels, as well as their percentage of ranking in the first place. This quantity, labeled as “Rank 1” in Table 5.1.2, denotes how

Table 5.1.2: Performance of the proposed multi-model scheme, the MLP model ensemble of [104], and individual machine-learning models for each prediction horizon. The values of MAE, RMSE, and R^2, achieved by each model, are presented, as well as the percentage that each submodel achieved the lowest MAE among all submodels.

Method	R^2	MAE	RMSE	Rank1
15 min				
Proposed	0.98613	0.26120	0.4703	-
MLP ensemble	0.9852	0.2760	0.4869	-
MLP	0.98568	0.26936	0.4782	19.25%
RBF	0.98574	0.27095	0.4773	19.94%
LR	0.98562	0.26700	0.4793	10.49%
SVR	0.98541	0.26931	0.4829	14.23%
RF	0.98373	0.29531	0.5071	24.27%
SR	0.98561	0.26715	0.4795	11.82%
1 h				
Proposed	0.93793	0.60224	0.9946	-
MLP ensemble	0.9344	0.6330	1.0240	-
MLP	0.91697	0.66794	1.1500	21.56%
RBF	0.93253	0.64235	1.0374	20.77%
LR	0.93168	0.64174	1.0438	8.91%
SVR	0.93008	0.65376	1.0562	10.70%
RF	0.92912	0.67311	1.0614	20.79%
SR	0.93045	0.64079	1.0532	17.27%
2 h				
Proposed	0.88147	0.88279	1.3767	-
MLP ensemble	0.8838	0.8965	1.3721	-
MLP	0.84455	0.99479	1.5854	20.32%
RBF	0.87052	0.96255	1.4472	18.32%
LR	0.87233	0.93356	1.4377	11.20%
SVR	0.86949	0.93765	1.4537	11.38%
RF	0.86653	0.96596	1.4675	22.94%
SR	0.86953	0.93189	1.4534	15.92%
3 h				
Proposed	0.84143	1.0599	1.5871	-
MLP ensemble	0.8359	1.0859	1.6192	-
MLP	0.78486	1.2504	1.8538	18.08%
RBF	0.82483	1.1367	1.6727	20.14%
LR	0.82241	1.1270	1.6843	8.74%
SVR	0.81914	1.1512	1.6997	11.82%
RF	0.81895	1.1391	1.7006	23.67%
SR	0.81893	1.1229	1.7007	17.54%
6 h				
Proposed	0.83251	1.1144	1.6462	-
MLP ensemble	0.8272	1.1951	1.6888	-
MLP	0.83036	1.1758	1.6733	20.77%
RBF	0.80289	1.2848	1.8037	21.31%
LR	0.77800	1.3308	1.9141	10.34%
SVR	0.75400	1.4300	2.0150	16.42%
RF	0.81341	1.2119	1.7549	21.08%
SR	0.77373	1.3413	1.9325	10.08%
24 h				
Proposed	0.78474	1.1835	1.8174	-
MLP ensemble	0.7827	1.2372	1.8468	-
MLP	0.78073	1.2313	1.8553	21.93%
RBF	0.73576	1.4119	2.0367	21.83%
LR	0.75712	1.3188	1.9526	11.16%
SVR	0.73669	1.3031	2.0331	16.82%
RF	0.76487	1.2694	1.9212	16.71%
SR	0.74761	1.3419	1.9905	11.56%

many times each submodel scored the 1st rank among all submodels, i.e., achieved the

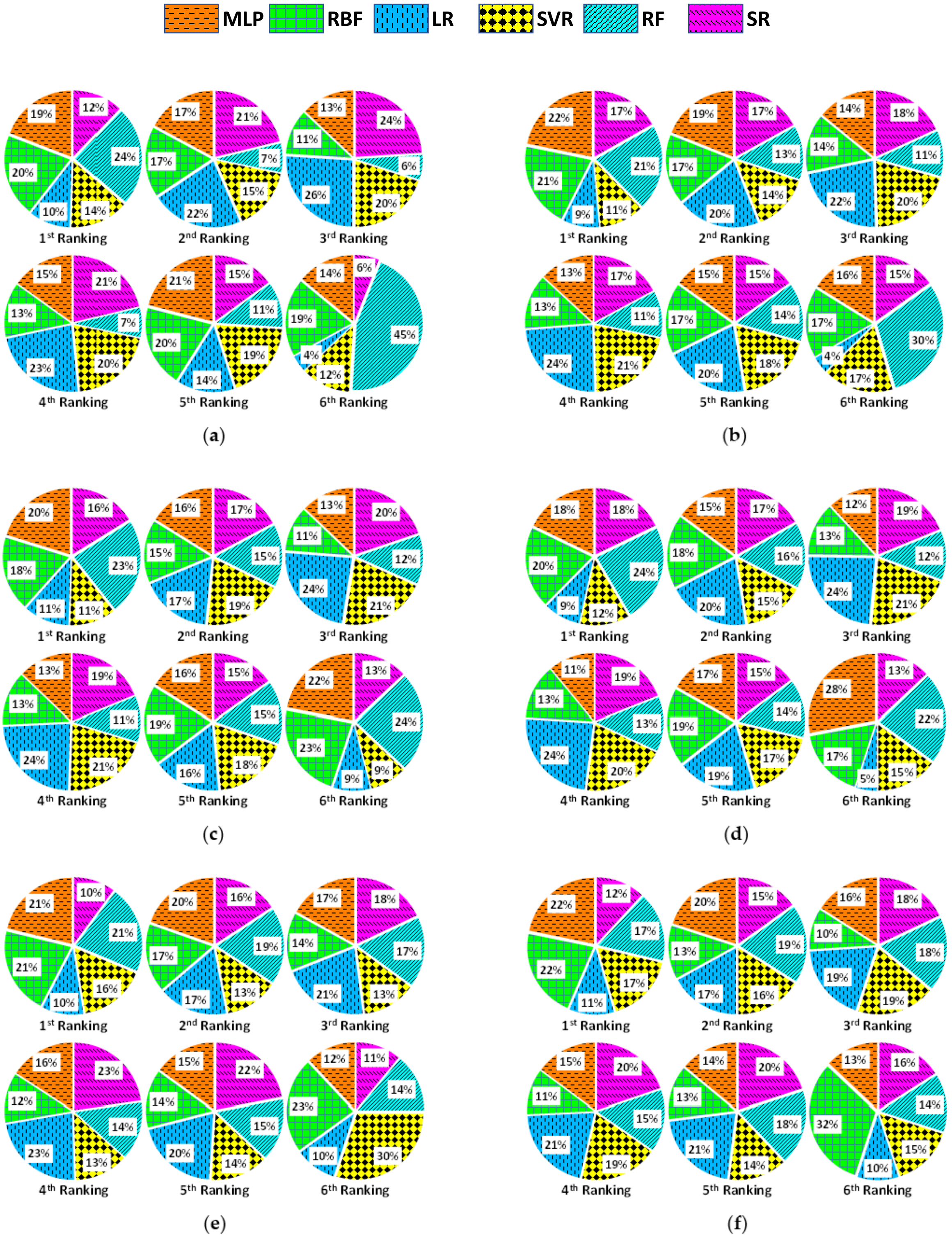
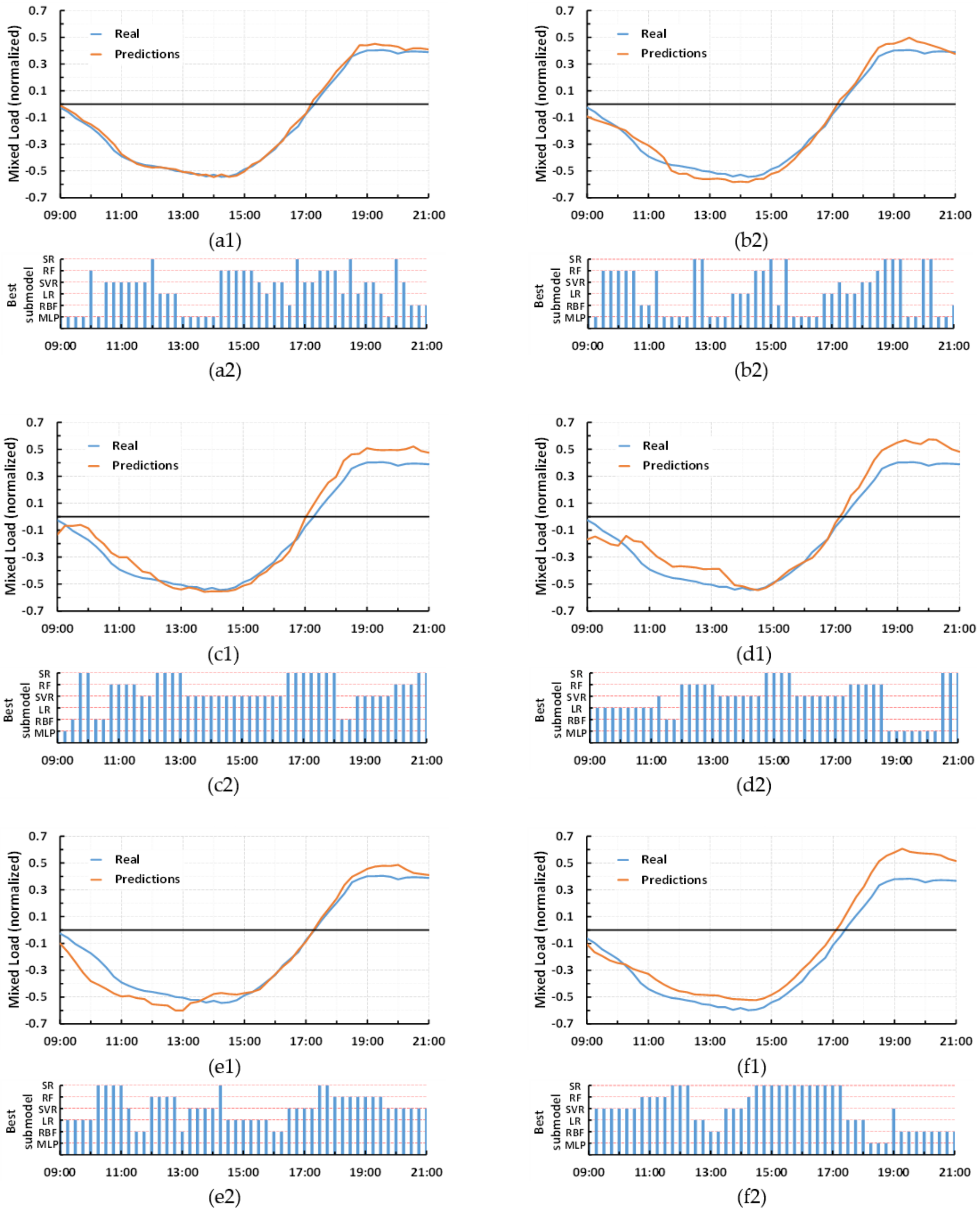


Figure 5.1.5: Pie charts depicting the ranking of the submodels included in the proposed model ensemble for (**a**) 15-min, (**b**) 1-h, (**c**) 2-h, (**d**) 3-h, (**e**) 6-h, and (**f**) 24-h ahead prediction. Each pie chart refers to a ranking position and shows the percentage that each submodel was ranked in that position. Each submodel is represented by a different color and pattern.

lowest MAE.

The aforementioned form of ranking of the submodels can be seen graphically in fig. 5.1.5. More specifically, each subfigure 5.1.5a–f refers to 15 min, 1, 2, 3, 6, and 24-h prediction horizons, respectively. Each one of these subfigures contains 6 pie charts, denoting 1st to 6th rank for the models. To be more specific, each pie chart shows the percentages corresponding to how many times each submodel ranked in the respective place, according to its weighted MAE. For example, the 2nd pie of Figure 5.1.5a implies that for 15 min-



(a1)

(b2)

(a2)

(b2)

(c1)

(d1)

(c2)

(d2)

(e1)

(f1)

(e2)

(f2)

Figure 5.1.6: Results for a randomly selected 12-h window for (**a**) 15-min, (**b**) 1-h, (**c**) 2-h, (**d**) 3-h, (**e**) 6-h, and (**f**) 24-h ahead predictions. Subgraphs labeled 1 depict actual and predicted value results, whereas subgraphs labeled 2 depict the best submodel performance results.

ahead forecasting, the MLP submodel ranked in the 2nd place among all models with a percentage of 17%, the SR submodel with a percentage of 21%, etc. Finally, analytical graphs are provided for each prediction time horizon, with Figure 5.1.6a1–f1 to depict forecasts of 15 min, 1, 2, 3, 6, and 24 h-ahead, respectively, where a randomly chosen 12-h time window (from 09:00 to 21:00) of real AP values and the respective predictions are shown for an arbitrarily chosen day belonging to the testing subset (the same day and the same window is used for all horizons). These graphs are accompanied by Figure 5.1.6a2–f2, which indicates which submodel has the largest weight for every predicted data point using a bar plot.

At this point, it should be pointed out that providing accurate predictions is indeed a challenging task due to both grid and data-related reasons. First, the system's expandability can be a limiting factor for the accuracy of future forecasts. At the same time, this is reinforced by inherent characteristics of the load time series, such as non-linearity and uncertainty. In the face of these challenges, the proposed method seems to be quite effective, providing reliable predictions. From fig. 5.1.4a–f, conclusions are drawn about the quality of predictions. When the prediction time horizon is too short (fig. 5.1.4a), the forecast error is distributed close to the diagonal line, which implies quite accurate predictions. With increasing prediction horizon, the forecasts become less accurate (fig. 5.1.4b–f), as obviously, the pairs of real and predicted values are scattered further from the ideal line.

Looking at Table 5.1.2, it is observed that the proposed model outmatches all individual submodels, and the competitive MLP model ensemble in terms of MAE, and R^2 and RMSE. Moreover, this conclusion applies to all prediction time horizons. As the prediction horizon gets longer, the forecasting error increases, which is absolutely reasonable. The only exceptions are the R^2 and RMSE values obtained by the MLP model ensemble for 2 h prediction horizon, which slightly exceeds those of the proposed model. However, these differences cannot be considered significant as they are marginal, while on the other hand, the corresponding value of the MAE index clearly favors the proposed method. A result worth mentioning is the improvement of the multi-model performance over the current best sub-model that occurs in most cases while the horizon is getting longer. More specifically, the reduction of MAE that the proposed approach achieves over the best of the individual models ranges from 0.03411 to 0.3156. Such an improvement in performance could be partly explained by the occurrence of uncertainty in the load time series. As the prediction horizon is getting longer, the level of uncertainty is also increased, which is better addressed by the ensemble model than each individual submodel alone.

Regarding the efficiency of the individual models of the pool, the results of MAE, RMSE, and R^2 show that there is not just one model to prevail over the others in all cases. For the shorter prediction horizons and, more specifically, up to 3 h, LR and SR appear to achieve marginally smaller forecasting errors than their non-linear counterparts. Although the non-linearities are an intrinsic characteristic of mixed load [115], this behavior becomes more apparent as the prediction horizon is getting longer. As a result, models which are based on LR are able to provide robust results for very short-term forecasts. On the other hand, one major advantage of neural networks is their capability of modelling non-linear systems. An important observation is that neural networks appear to perform better for longer prediction horizons, and this can be attributed to the fact that, as the prediction horizon is getting longer, the non-linear properties of the load are becoming more dominant. Therefore, when predictions for longer horizons are required, MLP neural networks take the lead. However, the same does not apply to RBF networks. As stated above, in order for RBF networks to perform well, dense and suitable data are required. Consequently, their performance is reduced for 24-h prediction horizons, where the input information is poorer due to the resampling process. Although the remaining models of the pool, SVR and RF networks, present a moderate predictive capability, they contribute positively to the overall performance of the proposed model. This conclusion confirms the need to use multiple models in order to enhance the reliability of load predictions.

Several quite interesting conclusions can also be drawn from the pie charts in Figure 5.1.5. Each percentage in the pies represents the degree to which the respective model yielded the highest weight or equivalently the lowest MAE. The highest percentages of the first rank (above 18%) belong to MLP, RBF, and RF, and this applies for all horizons except that of 24 h, where SVR takes the place of RF. RF, in particular, scores lower MAE most of the time when the prediction horizon does not exceed 3 h. Beyond that point, RBF neural networks outperform the rest of the submodels. An interesting observation is that the aforementioned models have equally high percentages in the sixth rank. Thus, these methods either achieve very good or poor performance. This observation is quite significant and strongly enhances the usefulness and effectiveness of our proposed method. The percentages of the rest of the pool models are, in most cases, divided into the intermediate rankings, with the exception of the high percentage of SVR in the sixth rank for the 6-h horizon.

5.1.5 Conclusions and Future Prospects

In this section, a multi-model ensemble prediction system was presented for the AP prediction for various horizons. It should be noted here that a limitation of the present study is that it did not involve predictions for long-term horizons. Although investigating longer prediction horizons is outside the scope of this work, the proposed model ensemble could serve as the basis for designing such a tool. On the other hand, it is quite probable that a different set of input variables, presenting higher correlation with the long-term evolution of the mixed load would be needed in this case. Another promising direction for future research towards this direction includes the integration of graph neural networks, which have been proved to be a promising candidate due to their ability to successfully interpret spatiotemporal features of the input data.

Driven by the increased performance of the proposed methodology in mixed load forecasting, its application could be extended to other critical sectors of the smart grid, such as forecasting the electricity price and the production from RES, in order to more efficiently schedule conventional sources. Arguably, the most important smart grid operational aspect that the proposed load prediction scheme can be incorporated in is the real-time power dispatch, since accurate short-term load predictions are paramount for the formulation of the grid’s optimal power flow problem.

5.2 Optimal Power Flow using Community-detection-based Cooperative Particle Swarm Optimization

The penetration level of photovoltaic (PV) systems is set to increase in the following years and already distribution networks (DNs) are straining to overcome the adverse voltage effects caused by reverse power flows and intermittent generation phenomena. Moreover, regular voltage optimization approaches that employ PV inverters as control devices, cannot deal effectively with the escalating dimensionality of the problem.

This section introduces a reactive power optimization method for PV-heavy DNs based on cooperative particle swarm optimization (PSO). The proposed approach makes use of multiple swarms, each swarm containing a group of design variables that are interrelated with respect to the optimization objective; a community detection algorithm is employed to assign the design variables to the different swarms, by identifying voltage-decoupled zones of the grid. The different swarms cooperate by exchanging information in order to better explore the search space, while still solving the optimization problem as a whole. The feasibility and effectiveness of the proposed scheme are demonstrated through comparisons with other approaches for various load and generation scenarios on the IEEE 123-bus distribution system.

5.2.1 Introduction to power dispatch in distribution grids

So far, reactive power optimization strategies using traditional voltage control devices have proven inadequate to address these concerns [116]. In the past, PV inverters were prohibited from contributing to reactive power compensation (RPC), constantly operating on a unity power factor and dealing with any overvoltage issues only by active power curtailment (APC). Indeed, even though this was the established policy for European DSOs [117], its effectiveness is challenged [118]: traditional control means, usually implemented on a substation level, could not address local voltage deteriorations caused by high PV penetration distributed over the grid [119]. Today, PV inverters have emerged as suitable control devices, as indicated by various research works [120], [121], and as reflected by the increasing confidence of DSOs in including them in the optimal reactive power flow (ORPF) problem formulation [117].

Traditionally, the ORPF problem has been addressed using standard mathematical optimization methods, such as gradient-based and interior point methods [11]. Since these methods came with certain disadvantages (the main one being their inability to handle non-convexity), research interest turned towards metaheuristic methods, which were inherently better equipped for the problem at hand: firstly, their stochastic nature allowed them to overcome local minima, and secondly, their utilization of multiple solution vectors enabled augmented search domain exploration capabilities. One of the most important metaheuristics is particle swarm optimization (PSO), which makes use of a population of potential solutions represented by particles that exchange information in the context of the problem's search space. The PSO method was first applied in the context of OPF in [68], and since then multiple of its variants have been developed that consider technical, economic and environmental optimization objectives [31], [122]. Still, the ever-increasing inclusion of PV inverters in the ORPF calculations has led to a stark increase in the number of design variables to optimize; unfortunately, this surge in the dimensionality of the optimization problem severely compromises the exploration and exploitation capabilities of metaheuristic methods, including PSO [123].

To cope with the problem of increasing dimensionality, latest research is mainly oriented towards more sophisticated, network partition-based approaches that seek to utilize the underlying topological structure of the grid [124]. The primary motivation has been the construction of distributed control schemes, requiring the network to be broken up into loosely coupled zones: [120] applies clustering to formulate an alternating direction method of multipliers algorithm for the ORPF of a DN, while use grid partitioning methods in order to identify islands in active distribution networks. A secondary motivation for partition-based approaches has been the need to reformulate the original problem into tractable and individually solvable optimization problems, for decentralized schemes [125] or two-level control strategies [126], [127]. The success of this reformulation originates from the basis of network partitioning: if the partitions are created in terms of bus voltage sensitivity with respect to control variable perturbations, then the voltage optimization objective of the resulting zone-based problems can be pursued in an independent manner. Such an example is shown in [121], where the voltage profile of each voltage-decoupled zone of the DN is successfully optimized using a PSO algorithm, by toggling the setpoints of PV inverters in the zone independently.

On the other hand, decentralized optimization schemes fail to take into account possible interactions between zones, as complete voltage decoupling is only an ideal assumption. There exist though a certain class of metaheuristic optimization methods that can exploit the same premise as decentralized schemes, i.e. grouping the design variables based on the effect they collectively have on the objective function, while still solving a unified optimization problem in a centralized way; these methods are called "cooperative" and have proven very effective in tackling high dimensional problems. The success of the aforementioned techniques in complex multimodal optimization has been highlighted in many works [39], [40] involving diverse optimization problems [128]. To the author's best knowledge though, no application of cooperative optimization methods on ORPF exists [1].

In this Section, a novel cooperative PSO (CPSO) framework is presented for the optimization and control of PV-heavy DNs. The proposed scheme employs multiple swarms to optimize different zones of the DN, where each zone contains design variables that are interrelated with respect to the optimization objective. Furthermore, in order to assign efficiently the design variables to the different swarms, a technique based on the Girvan-Newman community detection algorithm [129] is proposed. It should be noted that grouping the design variables to different zones/swarms is used only to facilitate the algorithm to better explore the search space; nevertheless, the objective function takes into account the whole grid, solving the optimization problem in a centralized way. Thus, a main advantage of the proposed scheme is that it retains its effectiveness even in networks with weakly-decoupled zones (such as networks with a moderate degree of meshing [125]), in contrast to decentralized voltage optimization schemes. Moreover, the CPSO algorithm exhibits robust characteristics, as it maintains a high degree of exploration inherent in population-based algorithms, while also employing a zone-based exploitation capability of candidate solutions that is necessary in order to overcome the challenges present in DNs. These characteristics allow it to efficiently cope with the high rate of penetration of PV systems that increases the optimization problem dimensionality. The proposed method is assessed for voltage deviation minimization, as well as the minimization of real power losses for an IEEE distribution grid, under various load and generation profiles.

5.2.2 Optimal Power Flow Problem Statement

The primary task of OPRF is to ensure that the bus voltage magnitudes stay within operational limits. This is especially needed in PV-heavy distribution grids where cloud coverage can obscure specific grid areas, thus rendering traditional control means with a grid-wide effect unsuitable. The communication infrastructure of the smart grid paradigm has allowed the inclusion of PV inverters as reactive power control devices: by tweaking the power factor of a PV inverter, an almost real-time dispatch of reactive power at the point of common coupling is allowed, resulting in increased control versatility. Fig. 5.2.1 shows the inverter capability curve; the inverter's power rating, coupled with the APC that is applied on the PV's generated power, denote its operating bounds in PQ space.

The formulation of the voltage deviation minimization problem for a PV-heavy smart distribution grid with controllable inverters is described below:

$$\min_{u \in \mathbb{R}^{n_u}} f(u) \tag{5.2.1a}$$

$$\text{s.t.} \quad P_{G_i} - P_{D_i} = V_i \sum_{j=1}^{N_{bus}} V_j \left(G_{ij}\, cos\theta_{ij} + B_{ij}\, sin\theta_{ij} \right) \tag{5.2.1b}$$

$$Q_{G_i} - Q_{D_i} = V_i \sum_{j=1}^{N_{bus}} V_j \left(G_{ij}\, sin\theta_{ij} + B_{ij}\, cos\theta_{ij} \right) \tag{5.2.1c}$$

$$V_{lower} \leq V_i \leq V_{upper} \tag{5.2.1d}$$

$$APC_k \leq 0.8 \tag{5.2.1e}$$

$$-90° \leq \varphi_{PV_k} \leq 90° \tag{5.2.1f}$$

$$Q_{k,min} \leq Q_k \leq Q_{k,max} \tag{5.2.1g}$$

where:

$$P_{G_i} = \begin{cases} P_{PV_k}\, APC_k, & i \in K \\ 0, & i \notin K \end{cases} \tag{5.2.2a}$$

$$Q_{G_i} = \begin{cases} Q_k, & i \in K \\ 0, & i \notin K \end{cases} \tag{5.2.2b}$$

$$Q_{k,min} = -\sqrt{S_k^2 - \left(P_{PV_k}\, APC_k\right)^2} \tag{5.2.2c}$$

$$Q_{k,max} = \sqrt{S_k^2 - \left(P_{PV_k}\, APC_k\right)^2} \tag{5.2.2d}$$

where N_{bus} is the number of buses in the network and P_{G_i}, Q_{G_i}, P_{D_i}, Q_{D_i} are the active and reactive power generation and demand in each bus i, respectively; B_{ij} and G_{ij} are the susceptance and conductance of the branch connecting the nodes *i* and *j*, respectively, while V_i and V_j are their corresponding voltage magnitudes; APC_k, φ_{PV_k} and S_k are the APC percentages, the power factor angle, and the nominal power rating of the k_{th} inverter, respectively; P_{PV_k} is the generated active power of the k_{th} PV panel, while Q_k is the

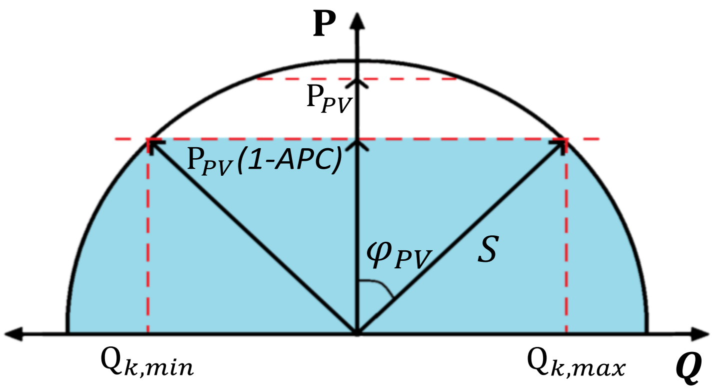


Figure 5.2.1: Reactive capability curve of a PV inverter in PQ space.

generated reactive power of the k_{th} inverter. Lastly, K is the set of bus numbers where PV installations are placed, and has a length of N_{PV}.

The objective $f(u)$ of the optimization problem is, in the case of voltage deviation minimization (VDM):

$$f(u) = \sum_{i=1}^{N_{bus}} |1 \, - V_i| \tag{5.2.3}$$

where the aim is to minimize the sum of absolute bus voltage magnitude deviations V_i from the nominal value of *1 p.u,* for every bus *i*. In the case of real power losses minimization (RPLM):

$$f(u) = \sum_{k=1}^{N_{lines}} |I_k|^2 \, R_k \tag{5.2.4}$$

where I_k is the magnitude of the electrical current through the resistance R_k of line *k*. The design variable vector u contains the APC and reactive power injection of each PV installation:

$$u = \left[Q_1, APC_1, Q_2, APC_2, \dots \, Q_{N_{PV},} \, APC_{N_{PV}}\right] \tag{5.2.5}$$

The optimization problem constraints consist of the power flow equations (5.2.1b,c), and the network operational constraints (5.2.1d-g). In this work, V_{lower} and V_{upper} are set as 0.95, 1.05 and 0.98, 1.02 for the VDM and RPLM objectives, respectively; tighter bounds must be used in the case of RPLM in order to ensure voltage quality, since the voltage deviations are not minimized explicitly. In any case, operational constraint (5.2.1d) exists in order to make sure that no single bus voltage magnitude enters an unsafe operation zone. Next, APC percent is bounded to 80% in order to promote RES penetration in the grid. It is assumed that the φ_{PV_k} of each inverter is unbounded, because the consideration of harmonic distortion effects of high-power factor angles is beyond the scope of this research. To summarize, Problem (5.2.1) is a non-convex constrained optimization problem belonging to the NP-hard class of problems. The inclusion of a realistic modeling of the inverter's RPC capability as a function of its APC (as opposed to treating it as a bounded variable) adds to the total constraint complexity, while the fairly large number of design variables (2 per PV installation) contribute to high dimensionality.

5.2.3 Methodology

5.2.3.1 Distribution network partitioning

As described earlier, there exist multiple advantages in partitioning a large distribution grid in terms of facilitating the solution of the optimization problem. Apart from practical motivations such as the decentralization of grid control, the partitioning of the optimization problem into smaller ones can alleviate the high problem dimensionality that is nowadays inherent in grids with a high penetration of distributed generation. More importantly for this work, cooperative optimization schemes can especially exploit problem partitioning in order to produce higher-quality solutions.

Since the objective of interest is to minimize the voltage deviations of the grid by optimizing the active and reactive power of distributed generators, one should seek to partition the aforementioned grid on a voltage sensitivity basis [124]. In particular, by accounting for the underlying dynamics of the network, areas that are loosely-coupled in terms of voltage fluctuation incurred by a reactive, or active power injection on a specific bus can be identified. Considering only the first-order perturbations on the original power flow equations (5.2.1b,c), a linearized set of equations occur:

$$\begin{bmatrix} \boldsymbol{\Delta\alpha} \\ \boldsymbol{\Delta V} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{\alpha P} & \boldsymbol{S}_{\alpha Q} \\ \boldsymbol{S}_{VP} & \boldsymbol{S}_{VQ} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta P} \\ \boldsymbol{\Delta Q} \end{bmatrix} \tag{5.2.6}$$

Here, $\boldsymbol{\Delta\alpha} \in \mathbb{R}^{1\times N_{bus}}$ is the vector of incremental changes of the voltage angle, $\boldsymbol{\Delta V} \in \mathbb{R}^{1\times N_{bus}}$ is the vector of incremental changes of the voltage magnitude, and $\boldsymbol{\Delta P} \in \mathbb{R}^{1\times N_{bus}}, \boldsymbol{\Delta Q} \in \mathbb{R}^{1\times N_{bus}}$ are the perturbations of the reactive and active power, respectively. The relationship between reactive/active power perturbations and bus voltage magnitudes and angles is represented by sensitivity matrices; $\boldsymbol{S}_{\alpha P} \in \mathbb{R}^{N_{bus}\times N_{bus}}$, $\boldsymbol{S}_{\alpha Q} \in \mathbb{R}^{N_{bus}\times N_{bus}}$, $\boldsymbol{S}_{VP} \in \mathbb{R}^{N_{bus}\times N_{bus}}$ and $\boldsymbol{S}_{VQ} \in \mathbb{R}^{N_{bus}\times N_{bus}}$ are the voltage angle and magnitude sensitivity matrices, with regards to active and reactive power, respectively. Keeping in mind the original motivation for grid partitioning, the $\boldsymbol{S}_{VP}$ and $\boldsymbol{S}_{VQ}$ sensitivity matrices are of interest, since their physical interpretation refers to the propagation of voltage magnitude variations through the distribution grid, due to active and reactive power injections at a bus. For example, the element i, j of the $\boldsymbol{S}_{VQ}$ sensitivity matrix represents the sensitivity of the i-th bus voltage magnitude to the injected reactive power at bus j.

In order to yield the optimal grid partitions, one needs to formulate a community detection problem [130]. Girvan and Newman proposed an algorithm for community detection in complex networks that does not require a predetermined number of communities (hereby referred to as “partitions”) [129]. The algorithm accepts a weighted adjacency matrix (or edge-weight matrix) that corresponds to the undirected graph structure of the electrical grid and generates the optimal partitions $C^{*} = \{C_1^*, C_2^*, \dots, C_{N_c}^*\}$ based on a modularity index ρ:

$$\rho = \frac{1}{2m}\sum_{i}^{N_{bus}}\left[\sum_{j}^{N_{bus}}\left(W_{ij} - \frac{k_i k_j}{2m}\right)\theta(i,j)\right] \tag{5.2.7a}$$

$$m = (1/2)\sum_{i}\sum_{j} W_{ij}, \;\; k_i = \sum_{j} A_{ij} \tag{5.2.7b}$$

Here, W_{ij} is the weighted adjacency matrix, m is the total average edge weight matrix and k_i is the average weight of all edges connected to the i node. $\theta(i,j)$ is a membership function for nodes belonging in the same partition C_k, so that:

$$\theta(i,j) = \begin{cases} 1, & j \in C_k \\ 0, & j \notin C_k \end{cases}, \;\; if\ i \in C_k \tag{5.2.8}$$

Since the optimal partitions are sought on a voltage magnitude sensitivity basis, the weighted adjacency matrix $W_{ij} \in \mathbb{R}^{N_{bus} \times N_{bus}}$ is:

$$W_{ij} = \left(\frac{\boldsymbol{S}_{VP}^{s} + \boldsymbol{S}_{VQ}^{s}}{2}\right) \circ A_{ij} \tag{5.2.9}$$

Here, ∘ denotes element-by-element multiplication, and A_{ij} is the graph’s adjacency matrix ($A_{ij} = 1$ if an edge from node i to j exists, $A_{ij} = 0$ if not). $\boldsymbol{S}_{VP}^{s}$, $\boldsymbol{S}_{VQ}^{s}$ are symmetric versions of the $\boldsymbol{S}_{VP}$ and $\boldsymbol{S}_{VQ}$ matrices, created by averaging between the original matrices and their own transposition:

$$\boldsymbol{S}_{VP}^{s} = \frac{\boldsymbol{S}_{VP} + \boldsymbol{S}_{VP}{}^{T}}{2}, \qquad \boldsymbol{S}_{VQ}^{s} = \frac{\boldsymbol{S}_{VQ} + \boldsymbol{S}_{VQ}{}^{T}}{2} \tag{5.2.10}$$

The motivation behind this step is the near-unity R/X ratio of distribution grids [131], which corresponds to equal bus voltage sensitivity to active and reactive power injections. The result is a symmetric weighted adjacency matrix W_{ij} that weighs the network buses in terms of voltage sensitivity to reactive and active power perturbations, with higher weight corresponding to a higher degree of voltage coupling [121]. In the end, the Girvan-Newman algorithm [129] can be applied to the community detection problem, in order to yield the optimal partitions. These partitions will contain buses that are highly coupled among themselves with respect to voltage fluctuations.

5.2.3.2 Cooperative PSO for partitioned electrical networks

The concept of cooperation between candidate solutions of a population has been applied in various metaheuristics, such as evolutionary algorithms and PSO [132], [133]. Specifically for the PSO category, the first cooperative instance that occurred was the CPSO-S framework [134], which splits a solution vector of *n* parts into exactly *n* 1-D particles; a generalized version of this approach is introduced in [135], where the solution vector is split into N_c groups with $N_c \leq n$, where *n* is the length of the solution vector. It is obvious that a prerequisite for the implementation of cooperative approaches is the assortment of the *n* design variables in N_c groups.

Therefore, since design variables are grouped on the basis of interrelation with respect to the optimization objective (that is, depending on whether their perturbations have a similar system-wide effect), it becomes apparent that for the smart grid voltage optimization problem, control devices that reside in a highly-coupled network zone should be grouped together. By utilizing information about the underlying electrical dynamics of the DN, the partitioning algorithm described in section 2.2 yields the voltage-decoupled zones C^*, which in turn indicate the swarms P_k, *k*=1,2,.., N_c, where N_c is the total number of swarms:

$$P_k = \left\{Q_1, APC_1, Q_2, APC_2, \dots, Q_{N_{P_k}}, APC_{N_{P_k}}\right\} \tag{5.2.11}$$

Here, N_{P_k} is the total number of PV installations that reside in network partition C_i^* (it is assumed that $N_{P_k} \geq 1$ for every C_i^*). For each swarm P_k, the particle position $P_k x_{ij}(t)$ and velocity $P_k v_{ij}(t)$ are updated according to:

$$\begin{aligned} P_k v_{ij}(t+1) = w P_k v_{ij}(t+1) + c_1 r_{1,i}(t)\left[P_k y_{ij}(t) - P_k x_{ij}(t)\right] \\ + c_2 r_{2,i}(t)\left[P_k \hat{y}_i(t) - P_k x_{ij}(t)\right] \end{aligned} \tag{5.2.12a}$$

$$P_k x_{ij}(t+1) = P_k x_{ij}(t) + P_k v_{ij}(t+1) \tag{5.2.12b}$$

Where $P_k y_{ij}(t)$ stands for the best personal position of particle *i,* in dimension *j* for swarm *k* at iteration *t* and $P_k \hat{y}_i(t)$ denotes the global best position vector of particle *i* for swarm *k* at iteration *t*; $r_{1,i}(t)$ and $r_{2,i}(t)$ are randomly sampled numbers from a uniform distribution in the range [0 1], while c_1, c_2 denote the acceleration coefficients and *w* the inertia coefficient. In order to control the exploration-exploitation trade-off, a velocity clamping constant $P_k v_{max}$ is employed to regulate the particle positions in the range $[-P_k v_{max}\ P_k v_{max}]$.

As each swarm P_k contains a distinctive part of the original $\boldsymbol{u}$ vector (14), the right cooperation between the swarms' agents is essential in order to calculate the fitness function for the overall optimization problem. This task is feasible by utilizing a context vector $\boldsymbol{u}_{P_k}$

formed by linking each of the particle positions of swarm P_k with the global best positions of the remaining swarms:

$$u_{P_k} = \left[P_1\,\hat{y}, \dots, P_{k-1}\,\hat{y}, P_k x_i, P_{k+1}\,\hat{y}, \dots, P_{N_c}\,\hat{y}\right] \tag{5.2.13}$$

Algorithm 5.2.1 Cooperative particle swarm algorithm

```
Input:    $Q_G, Q_D, P_G, P_D, V$ : Network parameters
          s: Swarm size population
          Iter: Maximum number of iterations
          c1, c2 w, $P_k$, $P_k v_{max}$: PSO configuration parameters
Output:   $\hat{\boldsymbol{u}}$ optimized PV installation control vector
Initialize the particles $P_k x_i$ for all swarms $k$ at random positions
Calculate fitness $f(\boldsymbol{u}_{P_k})$ and set global bests for all swarms $P_k\hat{y}(0)$, $k$=1,2,…, $N_c$
For =1: Iter:
    For $k$ = 1: $N_c$
    If stagnation criterion is met for kth swarm:
        reset particles $P_k x_i(t)$
    For =1: s:
        Calculate fitness $f(\boldsymbol{u}_{P_k})$ and $P_k y_i(t)$
    Calculate global best $P_k\hat{y}(t)$ for swarm $k$
    For $i$ =1: s:
        For $j$=1: $N_{P_k}$:
            Update velocity $P_k v_{ij}(t+1)$
            Update particle's position $P_k x_{ij}(t+1)$
```

This means that despite the fact that the network is partitioned into several distinct sub-swarms, the fitness function evaluation for each swarm's individual particle is estimated using the whole design vector**.** After forming the context vector $\boldsymbol{u}_{P_k}$, the fitness function evaluation $f(\boldsymbol{u}_{P_k})$ takes place by utilizing the objective function *f* with respect to swarm *k.*

A problem often encountered by PSO-based schemes is stagnation, which is related to the problem of particles being trapped in suboptimal solutions during the optimization process. This phenomenon limits the space exploration capabilities of the particles and is alleviated by implementing a resetting criterion which is described with detail in [128]. This is expected to increase the effectiveness of the method, since Problem (5.2.1) contains multiple local minima, when applied either for RPLM or VDM. The pseudocode for the proposed CPSO framework is given in Algorithm 5.2.1.

The cooperative PSO approach presents three important features with respect to the ORPF problem. First, the fitness function is evaluated after updating each part of the solution vector that corresponds to the respective swarm particles, resulting in finer-grained credit assignment. This addresses the classic “two steps forward - one step back” problem often encountered by PSO schemes, where a solution vector update improves one part of the solution vector but impairs another. This phenomenon is especially evident for the case of a network with voltage decoupled zones, where a part of a solution vector that corresponds to a specific zone may converge faster than others. The second advantage is related to the increased number of the combinations of different individuals that correspond to different swarms, boosting in this way the diversity of the solution context vector. Lastly, the third advantage refers to the robustness of the algorithm, even when applied in networks with weakly voltage-decoupled zones; CPSO consolidates the partitioned design variables in one design vector at the end of every iteration, thus taking into account any inter-zonal effects.

5.2.4 Results

5.2.4.1 Setup

The IEEE 123-bus distribution system [136] is elected as a suitable testbed for the simulation studies. Its large scale can accommodate a high number of PV installations, which warrants the application of cooperative optimization methods. Moreover, it is a well-studied case in the field of zone-based voltage control, therefore providing a reference point for discussion [120], [121]. The original IEEE 123-bus is an unbalanced system, containing multiple voltage regulators as well as a tap transformer at the slack bus. For this study, transformer and regulator taps are considered fixed; this way, only the PV inverter

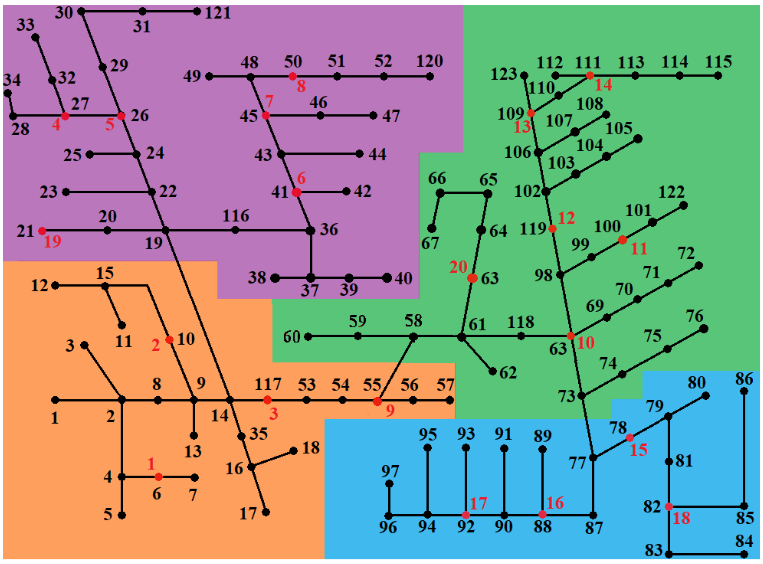


Figure 5.2.2: The partitioned IEEE 123 network. Red nodes denote PV installations.

capability is considered for the reactive power optimization. Lastly, the bus numbering is rearranged for clarity. The modified IEEE 123-bus network is shown in fig 5.2.2; here, colored areas denote the network partitions as obtained by the application of the community detection algorithm described in section 2.2. Next, 20 PV installations are placed throughout the grid, spanning capacities from 140-280 kW. Each inverter's nominal power is +10% of its respective installed PV capacity, as is usual practice. The inverters can curtail the generated PV power and control the power factor of the injected power in the grid.

To simulate different DN states, three different scenarios are created. The first two are snapshot scenarios (i.e., static), and are used in order to infer statistical conclusions for the performance of the proposed method. To be more specific, scenario 1 is used to assess performance for the VDM objective and represents the phenomenon of partial cloudiness, resulting in severe undervoltage. Scenario 2 is employed to evaluate the performance of the RPLM objective and assumes full solar irradiance, resulting in overvoltage in certain buses of the grid. Scenario 3 is used to demonstrate the applicability of the proposed method, and refers to the hourly setpoint optimization of PV inverters for a full day; here, both VDM and RPLM objectives are addressed. PV system specifications and information for scenarios 1 and 2 are shown in Tables 5.2.1, 5.2.2. The load scaling factor, as well as the solar irradiation percent for each network zone corresponding to scenario 3 are shown in fig. 5.2.3. In order to illustrate the effectiveness of the proposed method, two competing schemes are introduced: the first scheme, based on [121] formulates a decentralized optimization problem for each network partition, and solves each problem independently using a PSO algorithm with adaptive weights (the scheme is hereby referred to as "dPSO"). The second scheme applies a standard centralized PSO algorithm [137] to the original problem. The proposed method, as well as the two competing schemes, are contrasted with the network's default state, where all PV inverters operate at unity power factor and zero APC. The tuning parameters for each one of the competing schemes are shown in Table 5.2.3; they were selected based on indicative values found in the literature [1], [123], [138], in conjunction with a trial-and-error procedure. To be more specific, exponentially decreasing inertia was selected for the proposed CPSO method, while an adaptive inertia

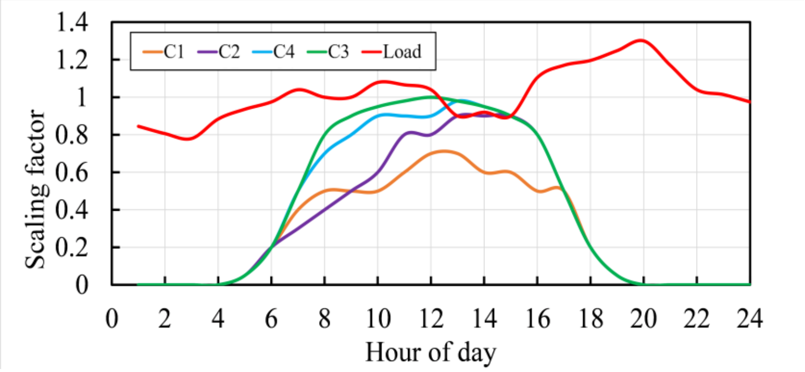


Figure 5.2.3: Scenario 3 load and solar irradiance profile for each one of the C_i network partitions. This scenario represents a day with partial cloudiness.

technique [137] was found to produce the best results for standard PSO and dPSO. Velocity clamping constants were selected to a value equal to 20% of the range of each design variable. Finally, each method uses the same value for the acceleration coefficients c_1 and

Table 5.2.1 Most common optimization objectives

#	Bus	PV Capacity	Power rating	Zone	#	Bus	PV Capacity	Power rating	Zone
1	6	140	155	C_1	**11**	100	280	310	C_3
2	10	140	155	C_1	**12**	119	280	310	C_3
3	117	140	155	C_1	**13**	109	280	310	C_3
4	27	180	200	C_2	**14**	111	280	310	C_3
5	26	180	200	C_2	**15**	78	280	310	C_4
6	41	180	200	C_2	**16**	88	280	310	C_4
7	45	180	200	C_2	**17**	92	280	310	C_4
8	50	180	200	C_2	**18**	82	280	310	C_4
9	55	140	155	C_1	**19**	21	180	200	C_2
10	68	280	310	C_3	**20**	63	280	310	C_3

Table 5.2.2 Snapshot Scenario information

Scenario	**Slack bus nominal voltage (p.u.)**	**Average irradiance percent per zone**				**Load multiplier per zone**			
		C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
1	1.00	100	80	50	50	1	1	1	1
2	1.02	100	100	100	100	1.6	1	0.4	0.4

Table 5.2.3 Tuning parameters for all methods

Scheme	**Swarm size**	**Stall iterations**	**Coefficients c₁, c₂**	**Function Tolerance**	**Inertia type**	**Inertia range**
CPSO	30	40	1.2	10^{-6}	Exponential	[1, 0.75]
PSO	30	40	1.49	10^{-6}	Adaptive	[1.1, 0.1]
dPSO	30	40	1.49	10^{-6}	Adaptive	[1.1, 0.1]

c_2.

5.2.4.2 Results & Discussion

Since the three competing schemes are based on stochastic search, multiple runs are needed to properly assess their performance. To be more specific, a total number of 20 runs for each scenario is performed, starting from different randomly chosen initial particle positions in each run. In order to reach valid conclusions regarding the statistical superiority of the proposed scheme, a *t*-test between CPSO and each one of its rivals has been applied for scenarios 1-2. The null hypothesis is that the results produced by the two competing methodologies are generated by populations with the same mean. Tables 5.2.4, 5.2.5 depict the average and standard deviation values, as well as the best value for the objective function from the 20 runs, together with the *p*-value corresponding to the *t*-test and the average number of function evaluations of each method for scenarios 1 and 2, respectively; the voltage profiles for an indicative run of each scenario are shown in Figs. 5.2.4a-4b. Regarding scenario 1, CPSO achieves 50% and 68% lower average objective value (sum of voltage deviations) in comparison to PSO and dPSO, respectively. Similar performance is recorded for scenario 2 where the objective of RPLM (total power losses in MW) is addressed: a 28% and 76% improvement is achieved over PSO and dPSO, respectively. The superiority of CPSO is also confirmed when comparing the best runs of each method: in scenario 1, CPSO achieves a 25% and 64% improvement over PSO and dPSO, respectively, while in scenario 2, CPSO scores 14% and 77% improvements over PSO and dPSO, respectively. The statistical significance of these results is established by the *t*-test

Table 5.2.4 Scenario 1: Statistical results for VDM objective

	Objective value average	**Objective value standard deviation**	**Best Objective value**	***p*-value**	**Average function evaluations**[1]
CPSO	0.7006	0.0603	0.6241	-	5260
PSO	1.4192	0.4183	0.8364	1.73E-08	1760
dPSO	2.2109	0.2295	1.7594	1.71E-25	6520

Table 5.2.5 Scenario 2: Statistical results for RPLM objective

	Objective value average	**Objective value standard deviation**	**Best Objective value**	***p*-value**	**Average function evaluations**[2]
CPSO	0.01035	0.00072	0.00918	-	8520
PSO	0.01395	0.00202	0.01068	2.513E-08	4560
dPSO	0.04307	0.00230	0.04027	5.543E-37	9240

[1] convergence to the 1st decimal

[2] convergence to the 3rd decimal

with a confidence interval of over 99.99%, as indicated by the produced *p*-values. It should be noted that the proposed method achieves this performance while staying within voltage magnitude bounds as specified in subsection 5.2.2, in contrast to dPSO which violates the upper voltage limit on buses 85-98 for some of the runs corresponding to scenario 2, as also shown in the indicative run of Fig. 5.2.4b. In addition, the superior performance of the proposed method retains consistency, i.e., the method converges around the same solution for each different run. This is indicated by the low value of standard deviation for both scenarios, and is testimony to the increased search space exploitation capabilities inherent to the CPSO algorithm. Lastly, it should be emphasized that the proposed method achieves the aforementioned performance improvements with a reasonable computational burden, as indicated by the number of average objective function evaluations: when compared to dPSO, CPSO exhibits consistently lower computational requirements in both scenarios. However, CPSO is surpassed by standard PSO in this aspect; this is to be expected, as CPSO

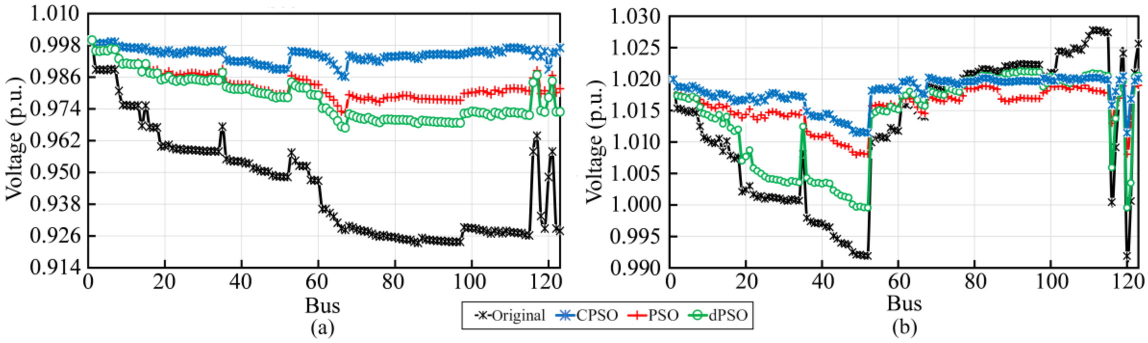


Figure 5.2.4: (a) Bus voltages for the VDM objective on scenario 1 (b) Bus voltages for the RPLM objective on scenario 2

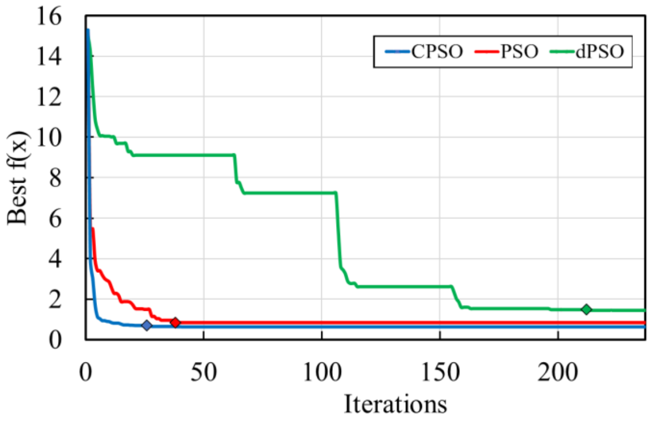


Figure 5.2.5: Convergence graph for the best runs of scenario 1. Coloured rectangles denote convergence to the 1st decimal. Note that the convergence curve of dPSO corresponds to the successive minimization of the four zone-based optimization problems

applies more objective function evaluations per algorithm iteration than PSO, but on the other hand, it manages to greatly outperform the latter in terms of optimization performance.

There exist multiple reasons for the superior statistical performance of CPSO in scenarios 1 & 2. As mentioned earlier, standard PSO suffers from the “two steps forward - one step back” problem and cannot effectively explore the available search space. This is evident from the convergence curves of the best runs of scenario 1, that are shown in Fig. 5.2.5; considering the first 30 iterations, CPSO achieves a rapid improvement in objective value, in contrast to PSO, which appears to stall multiple times over the same period. Moreover, CPSO achieves superior exploitation characteristics, since it converges to the 1st decimal much earlier than PSO. dPSO also sufficiently exploits the search space of each of the four zone-based optimization problems, which appear as distinct “steps” on the convergence curve; the quick convergence of each problem to the 1st decimal confirms this observation. It should be noted that, on one hand, dPSO retains the important practical advantage of complete decentralization [121], which CPSO and PSO lack. On the other hand, dPSO exhibits worse exploration capabilities out of the other two schemes. This can be attributed to the fact that as the algorithm progresses from one zone-based problem to the next, it cannot account for inter-zone effects, which are strong for the test case selected in this work. The result is an overall deterioration of optimization performance.

To assess the performance of the proposed scheme in a more demanding application, scenario 3 was employed, which concerns intra-day hourly voltage control using PV-inverters. The optimization results for both the VDM and the RPLM objectives are shown

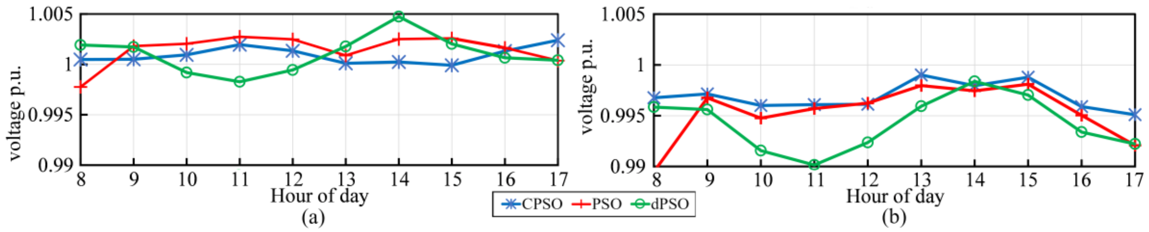


Figure 5.2.6: (a) Voltage profiles for the VDM control objective on scenario 3 for bus 28 (b) Voltage profiles for the VDM control objective on scenario 3 for bus 121

in Table 5.2.6 for each daylight hour, while voltage profiles for the VDM objective for 2 indicative buses are shown in Figs. 5.2.6a and 5.2.6b, respectively. In scenario 3, the superiority of the proposed scheme is also confirmed: for the VDM objective, CPSO scores a 35% and 47% improvement of performance over the PSO and dPSO schemes respectively, when assessing the intraday aggregated sums of voltage deviations.

The application of the proposed method for the RPLM objective yields similar performance, since the proposed method achieves 9.5% and 18% lower real power losses aggregated over the 8-17hr timeframe, when compared to PSO and dPSO, respectively. Even higher differences are observed in the 12-15hr timeframe, where the irradiance profiles for each zone reach their highest values: 21% and 40% real power losses reduction is achieved compared to PSO and dPSO, respectively; this is due to the higher power output the of PV panels that enable increased RPC capabilities for the PV inverters. Meanwhile, the lowest differences between the three methods are observed at 17:00, where the available power output drops sharply in conjunction with a network load increase; this leads to all required power being drawn from the slack bus (hence the proportionally higher losses in comparison to past hours). All in all, the properties of scenario 3 clearly illustrate the applicability of the proposed scheme: the differences in the available power between PV installations, which are in turn owed to the zone-based fluctuations of solar irradiance, greatly hinder the effective exploitation and exploration of the search space; yet, CPSO manages to exploit candidate solutions in a zone-based manner, while also retaining its original exploration capabilities, thus arriving at better solutions overall compared to its rivals. The standard PSO scheme generally outperforms the dPSO one for the VDM and RPLM objectives for the same reason; the aforementioned fluctuations of solar irradiance induce strong inter-zone effects, that render dPSO ineffective.

Table 5.2.6 Scenario 3: Results for the intraday application of VDM and RPLM objectives

	VDM objective (p.u. voltage)			RPLM objective (kW)		
Time Instance	**CPSO**	**PSO**	**dPSO**	**CPSO**	**PSO**	**dPSO**
8 hr	0.0954	0.3650	0.3944	36.7	38.8	41.9
9 hr	0.1072	0.1450	0.1555	30.0	32.5	35.6
10 hr	0.1278	0.1663	0.1498	32.8	33.8	40.6
11 hr	0.1260	0.1651	0.3624	22.9	26.1	26.5
12 hr	0.1203	0.1855	0.2627	20.4	23.8	24.5
13 hr	0.0717	0.1314	0.1374	11.4	14.6	23.1
14 hr	0.0797	0.1252	0.1564	12.2	19.4	25.3
15 hr	0.0729	0.1393	0.1154	11.6	13.2	23.5
16 hr	0.1429	0.1874	0.2073	28.6	31.7	34.3
17 hr	0.2087	0.2310	0.2295	126.6	134.5	131.8
Sum:	1.1530	1.8416	2.1714	333.2	368.4	407.1

5.2.5 Conclusions & Future Prospects

In this section, a cooperative PSO algorithm employing multiple swarms that are dedicated to the exploration of different search space partitions is introduced. Moreover, a practical methodology for the identification of the aforementioned search space partitions for the ORPF problem is presented, which is built upon the underlying topological characteristics of the network at hand. This methodology groups highly-coupled design variables together, with respect to voltage fluctuations incurred by active and reactive power perturbations. The effectiveness of the CPSO algorithm was demonstrated in simulation studies carried out in the IEEE 123-bus distribution system, and its performance improvements over a standard PSO and a decentralized PSO formulation were statistically evaluated.

It should be noted that the proposed community-detection-based CPSO optimization algorithm can be generalized to other engineering systems that contain a distinct structural topology, e.g., large industrial processes or other utility networks. To the author’s best knowledge, no other optimization approach, whether deterministic or metaheuristic, can encode topological information of the system to be optimized. Lastly, from an academic perspective, the community detection approach also lays the groundwork for future research and development of other cooperative metaheuristics [133].

Chapter 6:

Data-driven Tracking Nonlinear Model Predictive Control

MPC has emerged as a highly successful control scheme with applications in various fields during the last decades [74]. The MPC algorithm makes use of a dynamic model of the plant in order to calculate an approximation of the plant's response to the control inputs. Based on this model, a constrained optimization problem is formulated online to obtain the optimum sequence of control moves for a given time horizon. It is apparent that the MPC scheme effectiveness relies on the prediction accuracy of the plant model, as model inconsistencies can lead to poor choices regarding the control moves. Thus, in cases of highly nonlinear plant dynamics, it is a viable strategy to consider nonlinear plant models [139], integrated using typical Runge-Kutta techniques; on the other hand, such an approach adds a significant computational load in solving the optimization problem. Due to this reason, nonlinear MPC approaches are usually coupled with techniques for alleviating the increased computational burden, like online linearization [140], [141]; still, such techniques are not always attainable, especially for cases of large models.

In contrast, methods based on computational intelligence, e.g. neural networks (NNs), seem to be an attractive alternative, as they are very effective in modeling nonlinear plants [26], and are usually cheaper to evaluate online than Runge-Kutta integrations of first-principle models. In addition, for systems where proper ODEs are either unavailable or are simply unable to encapsulate the complexity of the real process, data-driven computational intelligence methods can be employed due to their black-box nature. Radial basis function networks in particular are widely considered for modeling nonlinear dynamics, mainly because of their simple structure and increased accuracy [34]. As these advantages are of paramount importance in the context of predictive control, RBFNs constitute a popular choice in conjunction with MPC [36]. This data-driven computational intelligence approach has enhanced the capabilities of the MPC algorithm with various applications [142]–[144].

The purpose of this chapter is to present data-driven tracking MPC controllers that employ RBF models in order to address nonlinear or otherwise hard-to-model processes within MPC's prediction module. For this reason, two representative cases studies have been chosen: The first refers to the control of a vehicle's active suspension system, which exhibits relatively high-dimensionality and significant nonlinearity, thus rendering linear models or ODE RK4-based approaches prohibitive. The second addresses the trajectory-following control of a vessel with the objective of avoiding moving obstacles; the future trajectory of these obstacles is unknown, and a data-driven trajectory prediction model built on past AIS data is required to be employed.

6.1 Data-driven tracking MPC for active suspension control

Active suspension systems in road vehicles are applied in order to mitigate the road-induced chassis vertical accelerations more effectively than standard passive suspensions, thus increasing comfort and handling. Such systems are greatly assisted by road preview schemes, consisting of special sensors usually based on laser scanners (e.g. LiDAR sensors), which detect road irregularities ahead of the vehicle and feed this information to a control system, designed to manipulate the active suspension accordingly. In this section, a model predictive controller with road preview, incorporating radial basis function models, is presented as a control scheme for a full car active suspension system. Substituting the standard linear predictive models with RBF ones, helps to approximate efficiently the significant nonlinearities present in the suspension system, so as to improve MPC performance. Special care is taken to alleviate the increased computational complexity entailed in the RBF models, in order to ensure that online implementation of the controller is feasible. The proposed scheme is evaluated on a detailed simulated full car model under various road excitation types, while making use of a realistic approach for incorporating LiDAR road scanner noise. Comparisons to a passive suspension system, as well as a standard MPC controller with a fully linear plant model, demonstrate the performance potential of using RBF prediction models in a road preview MPC context.

6.1.1 Introduction

The suspension system of a conventional road vehicle serves to keep the wheels in a relative position with the chassis while traveling. The two main design objectives are ride comfort for passengers, which is directly linked to the vertical acceleration of the vehicle's chassis, and road holding capabilities, often expressed as a load variation on the vehicle's tire [145]. Active suspension systems are usually implemented in vehicles through a hydraulic system that powers a piston placed in parallel to a conventional spring and damper, allowing for the direct pursuit of both objectives, i.e. ride comfort and vehicle handling [75]. The piston is controlled so as to exert desired forces on the wheel in the vertical direction. This system can achieve the aforementioned objectives in varying road conditions, as well as compensate for changes in the suspension dynamic behavior, which are inevitable through the life cycle of the vehicle [143], The challenge of controlling the vertical dynamics of a vehicle is that its behavior is far from ideal. Notably, hydraulic and geometrical non-linearities, as well as cubic terms of displacement and velocity exhibited by the springs and dampers, are the main sources of this behavior; ignoring these characteristics can lead to sub-par results. In this respect, nonlinear control strategies need to be considered, so that the above requirements are addressed.

The development of such strategies has been a topic of research since 1970, and several methods have been proposed to date. In [146], an adaptive backstepping control method with a grey signal predictor used for the estimation of system states is applied for the integrated control of heave and pitch dynamics of a vehicle. A control method that also utilizes a state predictor is presented in [147], where a Kalman filter is used for the estimation of the road type, in order to toggle between energy-saving and high-performance control modes. A similar scheme is presented in [148] that takes the nonlinear dynamics of the actuators into account, albeit for a half-car suspension model. Various techniques that can circumvent the suspension nonlinearities have been built upon the classic methodology of skyhook damping. For example, in [149] a 6 DoF half-car active suspension is controlled by a combined skyhook damping and fuzzy logic controller, where the original, non-linear actuator dynamics are approximated by a linear equation. In [150], an adaptive neuro fuzzy inference controller is presented for the control of a full car active suspension. Initially, a fraction-order-PID controller is implemented, so that the necessary data for training the inference model can be generated. Then, the proposed controller is implemented on an FPGA module in order to accelerate computations. Other neural network-based approaches have been applied in feedback linearization control schemes [151], [152] for the control of the full car active suspension vertical dynamics. Lastly, a fuzzy-PID control strategy is presented in [153] for the vibration control of a linear quarter-car active suspension, using an evolutionary computation algorithm for the optimization of the control parameters.

A control scheme that can directly accommodate for plant nonlinearities is model predictive control [74]. Though, as stated earlier, the vertical vehicle dynamics exhibit significant nonlinearities, up to now it was not the ability of MPC to incorporate nonlinear models, but rather its other merits, that have made it popular for active suspension design. An additional motivation is that MPC is suitable for integrating a road preview scheme, which can feed the active suspension control scheme with valuable information (Fig. 6.1.1). To be more specific, the vehicle can scan the road ahead using appropriate sensors, and supply this information to the controller, thus greatly enhancing performance [76]. The performance potential of MPC preview schemes has been confirmed for active suspensions in terms of

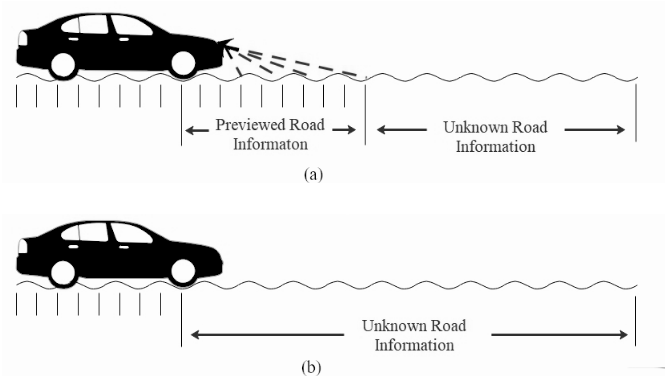


Figure 6.1.1: Active suspension (a) with and (b) without road preview information

energy expenditure [154]. In real-world applications, this preview technology is realized with LiDAR sensors, which, nevertheless exhibit a varying degree of measurement noise [155]. In [75] an MPC controller with preview (MPC-P) was applied to the active suspension full car problem to minimize roll and heave accelerations under several constraints, using a reduced model of the plant. Even though the wheel dynamics are not included in the MPC model on the basis that they exhibit higher frequency dynamics than the actuator, the proposed scheme achieves satisfactory performance. It should be noted that the hydraulic actuator dynamics are not included in the modeling stage but are substituted by the hydraulic actuators' displacement - this significantly reduces the plant nonlinearity, since hydraulic fluid flow through the actuator is a nonlinear phenomenon. In [76] an MPC-P controller with a linear model is created based on a simplified quarter car plant and compared to a skyhook-damping controller in a real world application.

Even though the results reported in these works are very encouraging, the potential of using MPC for active suspension design is not yet fully reaped, as the employed linear models cannot account for the nonlinearities present. Methods based on computational intelligence, e.g. neural networks (NNs), seem to be an attractive alternative, as they are very effective in modeling nonlinear plants [26] and can be configured to accommodate for changes in plant parameters. Radial basis function networks (RBFN) in particular are widely considered for modeling nonlinear dynamics, mainly because of their simple structure and increased accuracy [34]. It should be noted that RBF networks have been used extensively in automotive system modeling and control: in [144] an RBFN is used for the modeling of the wheel-slip dynamics of an anti-lock braking system. In [143] adaptive RBFNs are applied for the modeling of lateral and longitudinal dynamics of a 3 DoF vehicle model, for usage in a highway lane tracking PD-controller. In [36] an RBF network is used to explicitly model an MPC controller for the control of a semi-active suspension. Surprisingly, it seems that no RBFN approaches for modeling the vertical dynamics of a full car model have been reported in literature.

The main contributions of the proposed approach are the following:

- A method for developing a full car vertical dynamics model based on RBFNs is introduced. The new approach (a) is purely data-driven and does not make use of cumbersome first-principle equations, (b) can take into account plant nonlinearities, and (c) is computationally efficient as it makes use of the FM algorithm for training the networks.
- A nonlinear MPC framework is introduced for active suspension design, making use of road preview information. Nonlinearities that arise when modeling the vertical chassis dynamics of the full car are approximated with the aforementioned RBFN model. To the author's best knowledge, no MPC-P full car active suspension control scheme that directly accommodates for model nonlinearities exists, much less one that applies RBFN models.
- A new method for initializing the MPC optimization problem using an inverse model of the plant [156], which is also based on RBFNs, is used in order to minimize the computational burden needed for calculating the control actions, without compromising the model's predictive abilities. This is an important practical consideration, so as to ensure real-time implementation of the proposed approach.
- A realistic representation of noise induced by the road preview functionality is devised, based on studies regarding LiDAR sensors [157], [155]. The aim is to simulate the effect of road preview inaccuracies on the MPC controller performance, in order to confirm the robustness of the proposed scheme.

The rest of this section is structured as follows: In the next section, the plant equations of the full car and electrohydraulic actuating system are presented, together with the LiDAR road preview scheme. Subsection 6.1.3 starts with an introduction to the RBFN architecture and the FM algorithm and continues with a detailed description of the modeling approach followed and the produced results. Subsection 6.1.4 introduces the proposed controller, including a description of the MPC formulation and a discussion about expediting the solution to the optimization problem through an appropriate solution initialization technique. Then, in subsection 5 the case study is presented in detail, containing a description of the setup and the controller tuning procedure, followed by the produced results and relative discussion. Finally, the chapter ends with concluding remarks and directions for future work.

6.1.2 Active Suspension Control Problem Statement

6.1.2.1 Active suspension plant

The electrohydraulic piston-valve system is the powerhouse of an active suspension, since it is where the control force originates from. Hydraulic fluid is pumped from the oil sump by the axial pump and raised to the supply pressure. When the power servovalve is open, high pressure fluid flows to either one of the actuator chambers, while low pressure fluid flows from the other one back to the sump. Between the two actuator chambers a pressure difference is created, which results in the actuator force. The dynamics of the actuator pressure are highly nonlinear [158], since they are nonlinearly related to the hydraulic fluid flow through the servovalve.

Next, the vertical dynamics of the vehicle are presented. These can be approximated using a 7 DoF full car model. The model is based on an abstraction of the actual car, which is modeled as a rectangle representing the sprung inertia, with four masses at each corner, amounting to the unsprung masses, as shown in Fig. 6.1.2. This model can describe the heave z, roll θ and pitch φ modes of the sprung mass m_{spr}, as well as each unsprung mass heave displacement z_w. Each unsprung mass m_{uns} can be displaced along the vertical axis, and is connected to the sprung mass by a spring k, a damper d and a hydraulic actuator that exerts a force of F_a in parallel. The tire stiffness is modeled as a spring k_t between each unsprung mass and the road profile w_i. Each actuator is described by a valve displacement $x_{i,6}$ and a valve input u_i. The inputs to the full car plant are the four valve inputs and the four road profiles for each wheel. Table 6.1.1 depicts the derivative state equations for the model; for simplicity, numbered notation is adopted for each state, or input variable. The index $i \in \{1, 2, 3, 4, 5\}$ corresponds to {*Front Left, Front Right, Rear Left, Rear Right, Chassis*}. The index $j \in \{1, 2, 3, 4, 5, 6\}$ denotes a specific state, as shown in Table 6.1.1. For example, $x_{1,5}$ indicates the actuator pressure state of the front left wheel. The values employed for model parameters are shown in Table 6.1.2; these values were chosen to provide a realistic full car representation based on [158] and [159]. Note the nonlinear spring and damping terms.

Table 6.1.1: Full car state equations and functions

A. State Differential Equations	
$\dot{x}_{\mathrm{i},1} = x_{5,1}(b_{track}/2)\left(-1\right)^{i} + x_{5,2}(L_{base}/2)\,s(i) + x_{5,3}$	(6.1.1)
$\dot{x}_{\mathrm{i},2} = x_{\mathrm{i},4}$	(6.1.2)
$\dot{x}_{\mathrm{i},3} = \dot{x}_{5,1}(b_{track}/2)\left(-1\right)^{i} + \dot{x}_{5,2}(L_{base}/2)\,s(i) + \dot{x}_{5,3}$	(6.1.3)
$\dot{x}_{\mathrm{i},4} = (1/m_{unsp})\left(-F_{ks}\left(x_{\mathrm{i},1}, x_{\mathrm{i},2}\right) - F_{bs}\left(x_{\mathrm{i},3}, x_{\mathrm{i},4}\right) - F_{kt}\left(\mathrm{w}_{i}, x_{\mathrm{i},2}\right) + A_{p}x_{\mathrm{i},5}\right)$	(6.1.4)
$\dot{x}_{\mathrm{i},5} = a\left[Q\left(x_{\mathrm{i},6}, x_{\mathrm{i},5}\right) - C_{tm}x_{\mathrm{i},5} - A_{p}\left(x_{\mathrm{i},3} - x_{\mathrm{i},4}\right)\right]$	(6.1.5)
$\dot{x}_{\mathrm{i},6} = (1/\tau)\left(-x_{\mathrm{i},6} + u_{i}\right)$	(6.1.6)
$\dot{x}_{5,1} = (b_{track}/2Ixx)\left(F_{strut}\left(2\right) + F_{strut}\left(4\right) - F_{strut}\left(1\right) - F_{strut}\left(3\right)\right)$	(6.1.7)
$\dot{x}_{5,2} = (L_{base}/2Iyy)\left(F_{strut}\left(1\right) + F_{strut}\left(2\right) - F_{strut}\left(3\right) - F_{strut}\left(4\right)\right)$	(6.1.8)
$\dot{x}_{5,3} = (1/\mathrm{m}_{spr})\left(F_{strut}\left(1\right) + F_{strut}\left(2\right) + F_{strut}\left(3\right) + F_{strut}\left(4\right)\right)$	(6.1.9)
$\dot{x}_{5,4} = x_{5,1}$	(6.1.10)
$\dot{x}_{5,5} = x_{5,2}$	(6.1.11)
$\dot{x}_{5,6} = x_{5,3}$	(6.1.12)
B. Functions	
$s(i) = \left\{-1 \;\; i \le 2, \;\; 1 \;\; i > 2\right\}$	(6.1.13)
$Q\left(x_{\mathrm{i},6}, x_{\mathrm{i},5}\right) = C_{dP}S_{xv}x_{\mathrm{i},6}\sqrt{(P_{s} - sign\left(x_{\mathrm{i},6}\right)x_{\mathrm{i},5})/\rho}$	(6.1.14)
$F_{bs}\left(x_{i,3}, x_{i,4}\right) = b_{s}^{lin}\left(x_{i,4} - x_{i,3}\right) - b_{s}^{sym}\lvert x_{i,4} - x_{i,3}\rvert + \mathrm{b}_{s}^{nonlin}\sqrt{\lvert x_{i,4} - x_{i,3}\rvert}\,\mathrm{sign}\left(x_{i,4} - x_{i,3}\right)$	(6.1.15)
$F_{ks}\left(x_{i,1}, x_{i,2}\right) = k_{s}^{lin}\left(x_{i,2} - x_{i,1}\right) + k_{s}^{nonlin}\left(x_{i,2} - x_{i,1}\right)^{3}$	(6.1.16)
$F_{kt}\left(x_{i,2}, w_{i}\right) = k_{t}\left(x_{i,2} - w_{i}\right)$	(6.1.17)
$F_{strut}(i) = F_{ks}\left(x_{i,1}, x_{i,2}\right) + F_{bs}\left(x_{i,3}, x_{i,4}\right) - A_{P}\,x_{i,5}$	(6.1.18)

In short, the full car active suspension plant can be described at each continuous time instant t as a system comprised of 30 states, stored in state matrix X:

$$X(t) = \begin{bmatrix} x_{1,1}(t) & \cdots & x_{1,6}(t) \\ \vdots & \ddots & \vdots \\ x_{5,1}(t) & \cdots & x_{5,6}(t) \end{bmatrix} \tag{6.1.19}$$

that accepts as input a vector of each wheel's control signal u:

$$u\left(t\right) = \left[u_{1}(t), u_{2}(t), u_{3}(t), u_{4}(t)\right] \tag{6.1.20}$$

and a road profile vector **w**, consisting of the load profile for each separate wheel:

$$w\left(t\right) = \left[w_{1}(t), w_{2}(t), w_{3}(t), w_{4}(t)\right] \tag{6.1.21}$$

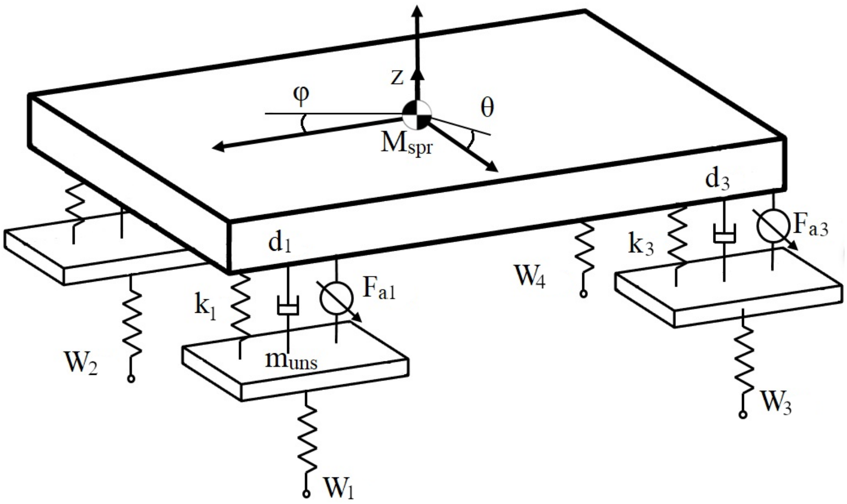


Figure 6.1.2: 7 DoF full car model

Table 6.1.2: Full car plant and electrohydraulic system parameters

Symbol	Description	Value	Units
P_s	Supply Pressure to Valve	$5\ 10^{6}$	Pa
A_p	Actuator Piston Area	$3.35\ 10^{-4}$	m^2
C_{tm}	Leakage constant	$22\ 10^{-11}$	-
α	Hydraulic Coefficient	$4.51\ 10^{12}$	N/m^5
ρ	Density of Hydraulic Fluid	850	kg/m^3
τ	Time constant of valve	$5\ 10^{-3}$	s
C_d	Discharge Coefficient	0.7	-
S_{xv}	Valve width	$4.1\ 10^{-3}$	m
b_s^{lin}	Linear damping coefficient	2000	N s/m
b_s^{sym}	Symmetric damping coefficient	400	N s/m
b_s^{nonlin}	Nonlinear damping coefficient	100	$N\ (m/s)^{-1/2}$
k_s^{lin}	Linear spring coefficient	25000	N/m
k_s^{nonlin}	Nonlinear spring coefficient	$25\ 10^{4}$	N/m^3
k_t	Tire spring coefficient	$19\ 10^{4}$	N/m
b_{track}	Wheel track width	2.5	m
L_{base}	Wheelbase	4	m
Ixx	X moment of inertia	550	$kg\ m^2$
Iyy	Y moment of inertia	300	$kg\ m^2$
m_{spr}	Sprung mass	1800	kg
m_{uns}	Unsprung mass	40	kg

6.1.2.2 Road Preview System

In order to measure the real road height profile vector $\mathbf{w}(t)$ onboard LiDAR sensors that scan the road ahead can be used, such as in [75], [76]. A very detailed description about the practical aspects of road previewing in active suspensions takes place in [160], while a real-world implementation is given in [157]. LiDAR sensors are subject to a variety of measurement errors, mainly arising from photon interference and scattering [155]. In order to model such errors, an additive Gaussian noise to all measurements is assumed in [160]. However, it has been observed that LiDAR measurement errors can vary in magnitude, in relation to the distance of the measured profile, e.g. an almost linearly-increasing error in relation to distance for a kinematic scanning LiDAR is reported in [161]. A simple experiment is set up in [157], and an increasing relative error in relation to distance is also observed. In order to provide a more realistic representation of measurement error induced by the LiDAR sensor, we assume an additive Gaussian noise for the previewed road, with increasing standard deviation as the distance from the sensor increases:

$$w_m(d) = w_r(d) + L\left(0, \sigma^2(d)\right), \quad d \in [0, l_p] \tag{6.1.22}$$

Here, w_m denotes the road profile height as measured by the LiDAR sensor, w_r the real road profile height at distance *d* from the sensor, l_p the preview length, and L is a random value sampled from a Gaussian distribution, where the standard deviation σ is a linear function of *d*. An example of the measured road w_m is shown in Fig. 6.1.3.

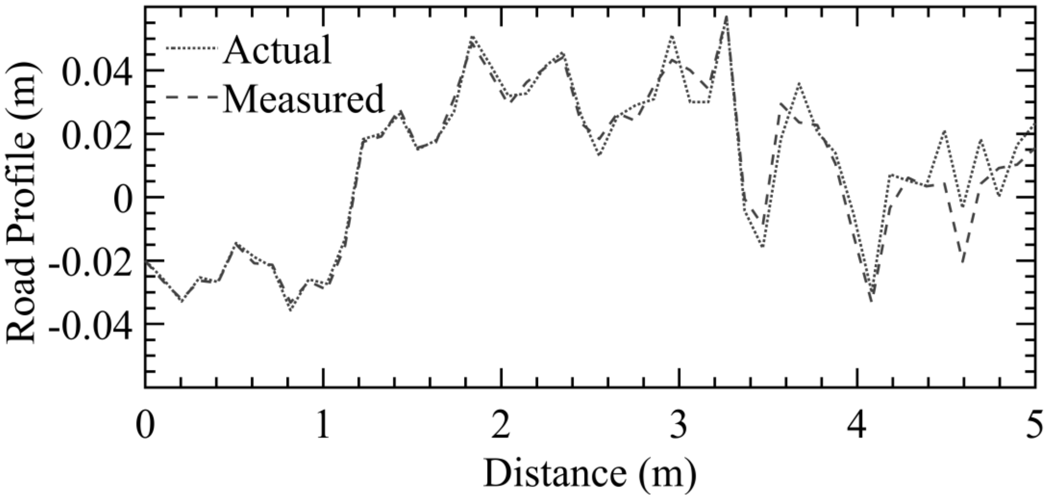


Figure 6.1.3: 1.5s of scanned-ahead road, in a road preview MPC context. The relative error increases with the distance from the LiDAR sensor

6.1.3 Methodology

6.1.3.1 Creation of the controller model

The general mathematical notation for a discrete model *f* of the full car plant is:

$$X(k+1)=f\left(X(k),\mathrm{u}(k),\mathrm{w}(k)\right) \tag{6.1.23a}$$

$$w(k)=\left[w_1(k),w_2(k),w_3(k),w_4(k)\right] \tag{6.1.23b}$$

$$u(k)=\left[u_1(k),u_2(k),u_3(k),u_4(k)\right] \tag{6.1.24}$$

The model f accepts as inputs the current state matrix $X(k)$ (19), the current control vector $u(k)$ and the current road profile vector $w(k)$ at discrete time k, and generates the prediction for the state matrix $X(k+1)$ at the next discrete instance. In order to approximate the function matrix f and build a discrete data-driven model of the full car plant, a suitable input-output dataset must be created. The input data are comprised of a random road signal to the four wheels and a random valve displacement signal to the four actuators. The road signal is generated as per the ISO 8606 standard that specifies the power spectral density of random road profiles of different quality [162] – for the task at hand, a type 'E' road is chosen. The valve displacement signal is generated from a uniform distribution within the valves' operating range, namely ±10mm. The aforementioned input is applied on the full car plant, and its output, which consists of the state variables as described by (6.1.19), is used to create an input-output dataset of 40000 datapoints. The dataset is subsequently split in a 50-25-25% manner, in order to create the training, validation and testing subsets, respectively; the first subset is used for calculating the model parameters, the second for model selection and the third for independently evaluating model performance.

Once those datasets are acquired, least squares regression on the training subset can be used in order to obtain the parameters for a discrete linear state model. The goodness of fit is evaluated on the testing subset using the coefficient of determination (R^2) and the mean absolute error (MAE). The results are shown in Table 6.1.3: as expected, the states (or their derivatives - for the chassis modes) that contain nonlinear terms, namely $x_{i,4}, x_{i,5}, \dot{x}_{5,1}, \dot{x}_{5,2}$ and $\dot{x}_{5,3}$, fail to be modeled sufficiently.

Having in mind that the produced model will be incorporated in a control scheme, it is important to note that the $x_{i,5}$ state corresponds to the actuator pressure of each wheel (and thus to the control force), hence it is directly related to the effectiveness of the applied control actions (6.1.5). If the MPC controller over/underestimates it after a control move, then the control performance will be directly compromised. Similar concerns are raised for the $\dot{x}_{5,1}$ state derivative, as it represents the controlled variable. Lastly, state derivatives $\dot{x}_{5,2}, \dot{x}_{5,3}$, which concern roll and pitch accelerations, indirectly influence all plant states. Correctly estimating the particular quantities can have a significant effect to the overall model performance, but unfortunately the respective equations are nonlinear in nature; this is apparent from (6.1.1-12), but also confirmed in practice, as shown in Table 6.1.3. For this reason, RBFN dynamic models were used for the aforementioned states / state derivatives, while for the rest of the states, linear models were applied, in order to reduce the computational cost of calculating the overall model response; it should be noted that when used for MPC, this model will be evaluated multiple times when solving the optimization problem, in order to calculate each control action.

A main consideration when training the RBFN models using the FM algorithm is the choice of the number of fuzzy sets s. The performance of an RBFN model will increase with increasing fuzzy sets, until it begins to overfit on the training data, thus compromising its generalization ability, i.e. its performance on new data. Overfitting occurs because the increasing network complexity allows for modeling the noise present in the training data [26], a fact which bears a detrimental effect on the network's ability to make accurate predictions for new data not included in the training subset. In order to establish the optimum value for s, a validation data subset must be also employed; to be more specific, a different network is trained for each value of s in the range {4-20}, using only the training subset, and then the value of s that maximizes the performance on the validation subset is chosen [108].

The results for the individual RBFN models used on the independent testing subset are shown in Table 6.1.3. These include the number of fuzzy sets and centers generated from the proposed training procedure, as well as prediction metrics on a random road testing subset. It is clear that the one-step ahead prediction performance of the individual RBFN models is superior to their linear counterparts, when modelling the highly nonlinear states/derivatives $x_{i,4}, x_{i,5}, \dot{x}_{5,1}, \dot{x}_{5,2}, \dot{x}_{5,3}$. Thus, two full car models are created; one with linear models for all the states, and one applying RBFN models for the nonlinear ones.

The two resulting car models (hereby referred to as "Linear" and "RBF-Linear") are essentially one-step-ahead prediction models, but their performance when predicting multiple steps ahead must be also assessed, in order to evaluate their suitability for integration in an MPC context. This can be accomplished by recurrently applying the model, each time using as input states the state predictions produced by the model for the previous time step. Fig. 6.1.5 compares the 35-step ahead prediction performance of the two models for a random road profile and control input signal, selected from the aforementioned testing subset. The RBF-Linear car model clearly outperforms the linear car model for the multiple-step-ahead prediction of the controlled variable (chassis heave

Table 6.1.3: Model comparison statistics

States	Linear model		RBFN Model			
	R^2	MAE	# of Centers	# of Fuzzy Sets	R^2	MAE
$x_{1,1}$	0.9960	0.0021	-	-	-	-
$x_{1,2}$	0.9990	6.1e-4	-	-	-	-
$x_{1,3}$	0.9855	0.0371	-	-	-	-
$x_{1,4}$	0.9893	0.0287	5121	9	0.9910	0.0156
$x_{1,5}$	0.9652	0.5159	8131	12	0.9908	0.2425
$x_{1,6}$	0.9998	3.36e-5	-	-	-	-
$\dot{x}_{5,1}$	0.9865	0.3213	5121	9	0.9920	0.1656
$\dot{x}_{5,2}$	0.9909	0.1021	5121	9	0.9946	0.0628
$\dot{x}_{5,3}$	0.9875	0.1204	5121	9	0.9918	0.0852
$x_{5,4}$	0.9924	0.0014	-	-	-	-
$x_{5,5}$	0.9962	5.2e-4	-	-	-	-
$x_{5,6}$	0.9993	5.0e-4	-	-	-	-

The metrics were generated from a random road testing run. Only the results of the front left wheel are presented; the rest of the wheels are omitted because they exhibit almost identical metrics.

acceleration), as the former achieves a MAE of 0.1082 and an R^2 of 0.9812, while the latter

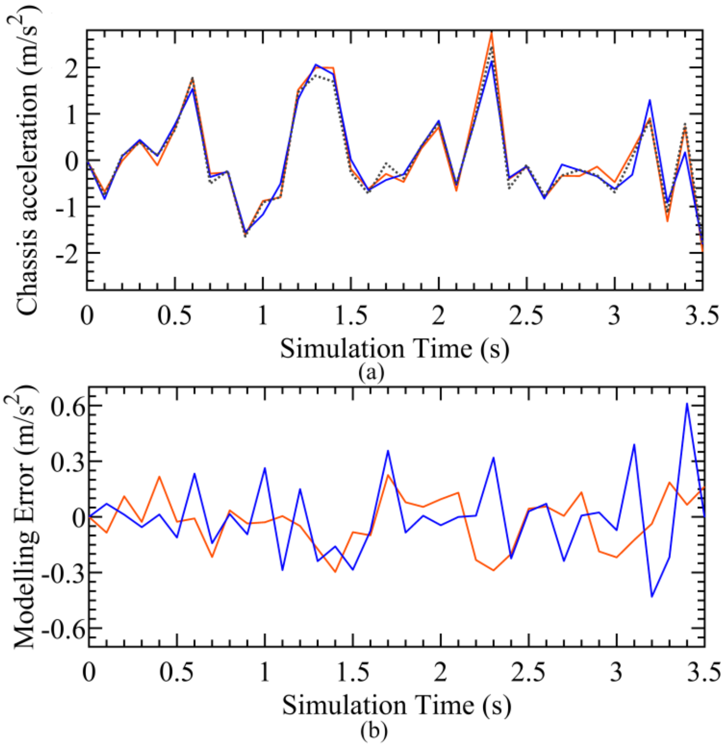


Figure 6.1.4: Linear and RBF-linear car model comparison for multiple-step-ahead evaluation. (a) heave acceleration modeling comparison, (b) modeling error comparison

a MAE of 0.1483 and an R^2 of 0.9528.

6.1.3.2 Model Predictive Controller

The proposed active suspension control scheme involves using an MPC controller, in conjunction with road preview information. There exist numerous formulations for the active suspension problem, as far as the design variables in the MPC objective function are concerned. In [76] wheel load and suspension displacement are explicitly minimized. This work aims to minimize the chassis heave acceleration, as the proposed scheme is concerned with vehicle passengers comfort [163].

At each discrete time point k, the following OCP is formulated:

$$\min_{x,u} J_N(x,u) \tag{6.1.25a}$$

$$\text{s.t.}\,(x,u) \in \mathbb{Z},\;\; x_N \in \mathbb{X}_f \tag{6.1.25b}$$

$$x_0 = x_0' \tag{6.1.25c}$$

$$x^+ = f_{RBF}(x,u) \tag{6.1.25d}$$

Where the objective function $J_N(\cdot)$, quadratic stage cost $\ell(\cdot)$ and terminal penalty $V_f(\cdot)$ are denoted as follows:

$$J_N(x,u) = \sum_{k=0}^{N-1} \ell(x_k,u_k) + V_f(x_N) \tag{6.1.26a}$$

$$\ell(x_k,u_k) = {x_k}^T Q x_k + {u_k}^T R u_k \tag{6.1.26b}$$

$$V_f(x_N) = {x_N}^T Q_f x_N \tag{6.1.26c}$$

Here, $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ denote the admissible state and input space, $\mathbb{X}_f = \{0\}$ is the terminal set, $Q \in \mathbb{R}^{n_x \times n_x}$, $R \in \mathbb{R}^{n_u \times n_u}$ and $Q_f \in \mathbb{R}^{n_x \times n_x}$ are positive-definite weighting matrices, N is a prediction horizon, and f_{RBF} is the "RBF-Linear" controller model that was constructed in subsection 6.1.3.1.

Corollary 6.1.1 (Asymptotic stability of MPC (6.1.25-26)): *Suppose assumptions of Theorem 4.4.2 are satisfied as well as*

A. *State transition function $x^+ = f_{RBF}(x,u)$ is bounded & continuous $\forall (x,u) \in \mathbb{Z}$, $f_{RBF}(x,u) \in \mathbb{X}_f$ and $f_{RBF}(0,0) = 0$*

Then, the origin of $x^+ = f_{RBF}\big(x, \kappa_N(x)\big)$ is asymptotically stable in $\mathcal{X}_N$, where $\kappa_N(x)$ is the MPC control law described in (6.1.25).

Proof: Given that the stage cost $\ell(\cdot)$ and the terminal cost $V_f(\cdot)$ are positive definite functions, together with the assumption that the $f_{RBF}(\cdot)$ responses are bounded and continuous for all admissible (x,u), then Theorem 4.4.2 can be leveraged to show asymptotic stability of the scheme. ■

To ensure real-time implementation of the proposed control scheme 6.1.25-26, the optimization problem should be solved within one sampling time period, thus enabling the controller to calculate new actions for each discrete time step. In addition, in order to accurately track the fast dynamics imposed to the car by the high frequency road excitations, the controller sampling period should be kept as small as possible [164]. In the case of MPC controllers, the sampling time is essentially dictated by the time needed to solve the optimization problem, this being the bottleneck in the control action calculation procedure. Unfortunately, notwithstanding the improved approximation capabilities throughout the whole operating region offered by an RBF model, its use in MPC transforms the optimization problem to a nonlinear one, thus bearing a negative effect on the time required for obtaining a satisfactory solution.

A standard approach for speeding up the optimization procedure in MPC, is to initialize the optimization problem formulated during each time step with the optimal control moves $\mathbf{u}^*$ resulting from the previous time step, after removing the first move which has already been implemented. However, this technique may not be adequate for the particular application; the fast-changing road conditions may in turn cause the car's state to change rapidly, thus rendering the previous time step solution obsolete. In this case, the optimizer starts from a poor initial value, which can lead to longer computational times for solving the problem.

In this work, a more elaborate initialization technique is used, which has been shown to significantly expedite the solution procedure [156]. To be more specific, an inverse RBF model of the plant is trained offline. This model can be then applied at each time step to provide a sequence of actions $\mathbf{u}_{inv}$ for the entire control horizon; calculating u_{inv} is very fast, because no optimization procedure is involved. On the other hand, it should be noted that u_{inv} offers only a feasible trajectory to the setpoint, without taking into account the aspect of optimality; though directly applying it to the real plant could yield subpar results, it can be used as a hot-start for the MPC optimizer. Thus, for every new formulation of the MPC optimization problem occurring at each discrete time step, a suboptimal solution is first obtained using the inverse model, and then passed on to the optimization solver as an initial guess. More details about this type of initialization can be found in [156].

6.1.4 Results

6.1.4.1 Case Study & Setup

In this case study, the proposed RBFN-MPC approach is evaluated on a simulated full car plant, under various road excitation types. A linear MPC-P controller and a passive full car suspension are also presented for comparison purposes. The full car plant described in subsection 6.1.2 is simulated by numerically solving the state equations (6.1.1-12) using a Runge-Kutta 4-5 formula [165]. Two MPC controllers were created, hereby referred to as MPC-L and MPC-RBF-L, employing the linear and RBF-linear full car models, respectively. In both controllers, the MPC optimization problem was solved using an active-set method, as described in [156]. All simulations were realized in MATLAB environment and a desktop computer with an Intel i9-9960X CPU and 64 GB of RAM. The sampling time for both controllers was set to 100ms of simulated time.

The control objective is to maximize passenger comfort. This could be taken as equivalent to minimizing the heave acceleration, as is done in [75]. However, different, more sophisticated performance metrics for comfort quantification also exist. ISO 2631 [166] sets standards for exposure of humans to vibration, and applies not only to vehicles, but to all vibrating environments as well. According to this standard, humans are more sensitive to heave acceleration in the range of 4-8 Hz. A more road-vehicle-oriented discussion about passenger comfort criteria takes place in [163]. It should also be noted that vehicle comfort should be evaluated both on bump and random road tests, for completeness sake.

In light of the above, it was decided to compare the competing schemes in three different tests (Fig. 6.1.5): a one-sided pulse bump with a height of 5 cm, a one-sided symmetric ramp bump with a maximum height of 12 cm and a duration of 0.3 s, and a 30m stretch of random road of class “E” [162], traversed at 10 m/s vehicle speed. For the bump tests, importance is placed on the max absolute acceleration (MAA), the sum of absolute errors (SAE) and the settling time; these are deemed important, since any changes to them are directly perceived by the passengers. For the random road test, besides the sum of absolute errors, the fast Fourier transform (FFT) of the heave acceleration is computed, since it gives a clearer picture of controller performance, together with the squared heave displacement, as dictated by the Steffens comfort metric [163]:

$$X\left[cm^2\right] = 7.62 \cdot 10^{-3}\left(1 + \frac{125}{f_c^2}\right) \tag{29}$$

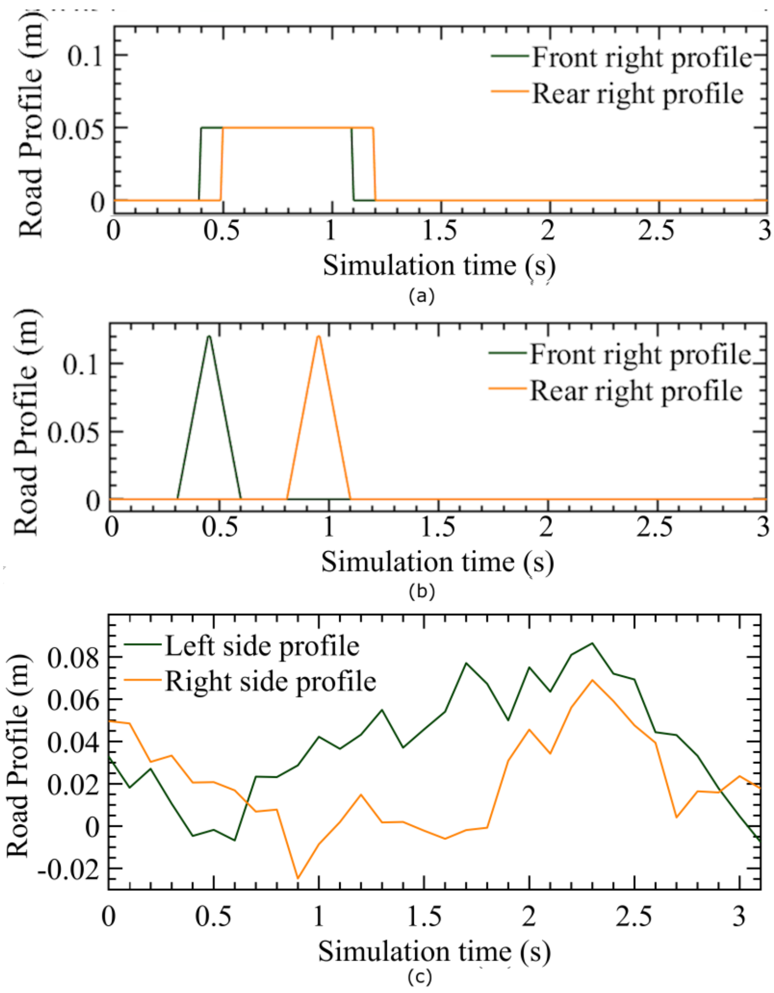


Figure 6.1.5: Road scenarios for simulation: (a) Right-sided pulse bump, (b) Right-sided ramp bump, (c) Random road profile

X is the square of the chassis displacement in cm, and f_c is its frequency in Hz. The Steffens criterion sets a bound on the FFT of the chassis heave displacement, which signifies discomfort. Thus, the furthest away the frequency response of the chassis displacement is from that bound, the better the vehicle's suspension is performing. In addition, an important consideration must be made regarding the road holding capability of the proposed scheme. This capability can be quantified by the vertical acceleration of each wheel, which should not be too high, otherwise the tire contact patch may be perturbed or completely detached from the road, resulting in loss of road holding [145], [167]. While increasing ride comfort remains as the sole control objective of the proposed scheme, it must be verified that the road holding capability does not deteriorate in comparison to the passive suspension case. Therefore, the average root mean square (RMS) values of the vertical acceleration of each wheel are calculated for the random road case, as it is the case most likely to cause such a phenomenon.

Finally, the tuning parameters of the two MPC controllers are presented. Tuning of the two MPC controllers amounts to determining the optimum values for standard MPC parameters, namely the prediction and control horizons h_p and h_c, the **Θ, Ω** vectors and the objective function weights ω and θ, but also for the road preview length l_p. The reason that l_p constitutes a tuning parameter is that the previewed road information quality deteriorates over distance due to noise, as discussed in section 2.3; so, there exists a maximum effective length of previewed road that is useful for the controller.

A low value for the h_c parameter is chosen a priori, since it will result to a lower number of control variables to be optimized online, and thus, to a faster solution of the MPC problem. The rest of the tuning parameters are optimized offline using the particle swarm optimization (PSO) method [66]. PSO is a favorable choice, because of its effectiveness in solving nonlinear optimization problems, in conjunction with its simplicity. The tuning procedure was run on two cases: the first involves a one-sided pulse bump of 8 cm height, and the second a random road of class “E”. The tuning optimization objective for both cases is the minimization of the sum of absolute heave acceleration over the timespan of the simulation. The results of the tuning procedure for the bump and random road cases for both controllers are shown in Tables 6.1.4 and 6.1.5, respectively, which contain the selected values for the tuning parameters described in the first paragraph of this subsection. Using two different tuning scenarios aims to cater to the differences of two distinct types of roads that can occur – a well-paved road with the occasional anomaly (bump case) or a badly paved road or dirt track (random road case). Note that this practice does not violate the practicality of the proposed scheme in a real-world implementation; the controller could

Table 6.1.4: Bump road tuning parameters

	MPC Linear	**MPC Linear - RBF**
h_p	15	15
l_p (m)	5	5
Q	$9.42\ 10^4\ I_{n_x \times n_x}$	$8.17\ 10^4\ I_{n_x \times n_x}$
R	$1.45\ 10^4\ I_{n_u \times n_u}$	$1.99\ 10^4\ I_{n_u \times n_u}$
Q_f	$9.30\ 10^4\ I_{n_x \times n_x}$	$8.23\ 10^4\ I_{n_x \times n_x}$

Table 6.1.5: Random road tuning parameters

	MPC Linear	**MPC Linear - RBF**
N	7	7
l_p (m)	3.1	3.2
Q	$9.42\ 10^4\ I_{n_x \times n_x}$	$9.73\ 10^4\ I_{n_x \times n_x}$
R	$1.15\ 10^4\ I_{n_u \times n_u}$	$5.93\ 10^4\ I_{n_u \times n_u}$
Q_f	$9.40\ 10^4\ I_{n_x \times n_x}$	$9.60\ 10^4\ I_{n_x \times n_x}$

toggle between on-road and off-road modes either manually or automatically [147], as is the case in most modern vehicles with adjustable suspension today.

6.1.4.2 Results & Discussion

The proposed MPC-RBF-L controller is applied to the road scenarios shown in Fig. 6.1.6, and compared to MPC-L and a passive suspension. The responses of the two controllers, along with the passive suspension, are shown in Figs. 6.1.7-9, while numerical results are given in Tables 6.1.6-8. Each table corresponds to a specific road scenario, and contains the simulation metrics for the two controllers and the passive suspension, as discussed in subsection 6.1.4.1.

As far as the pulse bump scenario (Fig. 6.1.7) is concerned, the MPC-RBF-L controller is able to reduce the SAE by 10% compared to MPC-L, and 28% compared to the passive

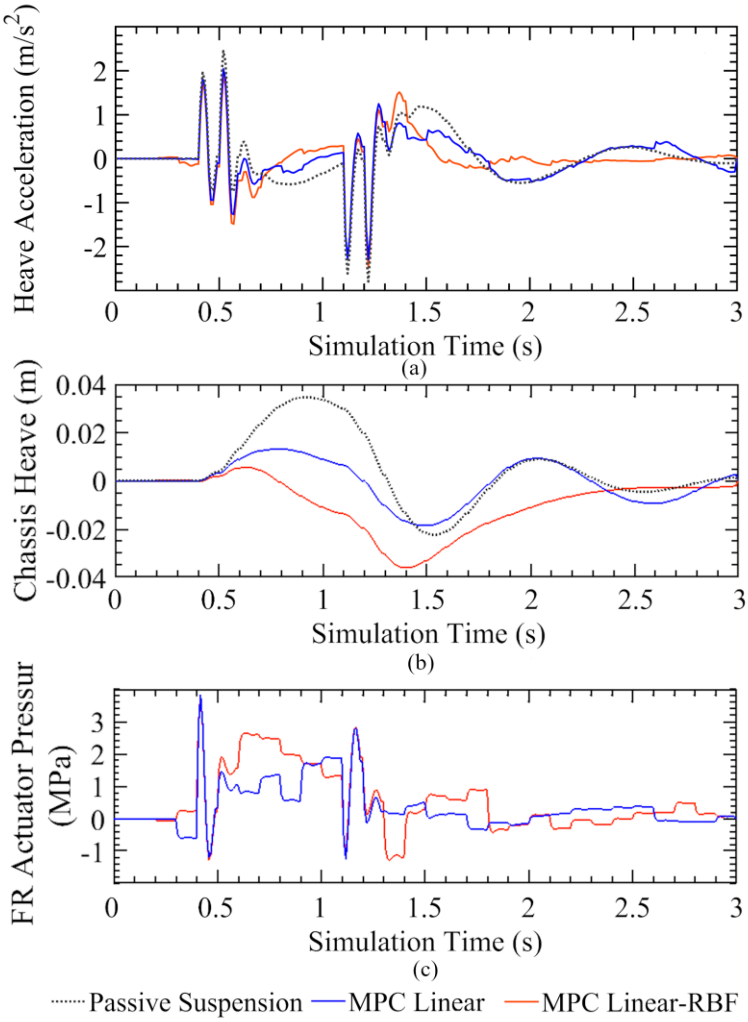


Figure 6.1.6: Results of a pulse bump test: (a) chassis heave acceleration response, (b) chassis heave displacement, (c) front right pressure of the hydraulic actuator

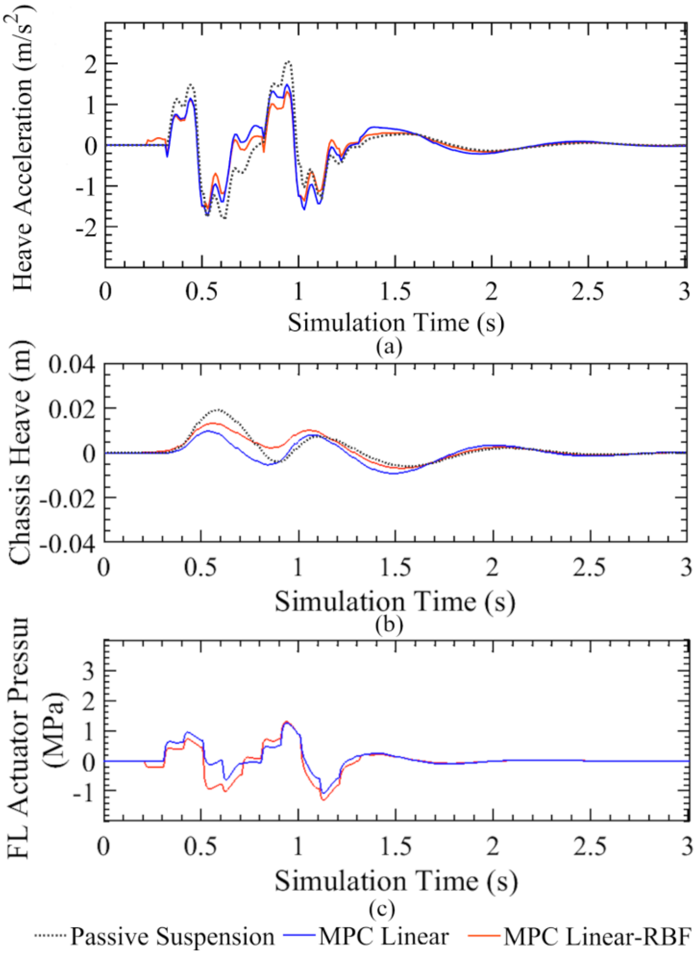


Figure 6.1.7: Results of a ramp bump test: (a) chassis heave acceleration response, (b) chassis heave displacement, (c) front right pressure of the hydraulic actuator

suspension. The settling time also shows significant improvement. In the ramp bump test (Fig. 6.1.7), the MPC-RBF-L controller reduces the SAE by 11.5% compared to MPC-L, and 37% compared to the passive suspension. No reduction in settling time is achieved by the two MPC controllers when compared to the passive suspension in this case – this can be attributed to the fact that when the road disturbance ended, the absolute value of the heave acceleration was so small that the controllers decided not to act. The SAE is overall more effectively reduced on the ramp than the bump test for both MPC controllers. The reason is that the road disturbance is less abrupt and therefore both controllers can track it better.

On the random road (Fig. 6.1.8) the proposed controller reduces the SAE by 10% and 20%, compared to MPC-L and the passive suspension, respectively. Here, the frequency-domain response is of larger importance: the MPC-RBF-L generally performs well in the 4-8 Hz range, with a 44% reduction of the chassis acceleration response at eigenfrequency compared to the passive car and a 22% reduction compared to MPC-L. This frequency range is of interest, since any vibrations herein incur the most discomfort for humans, according to ISO 2631 [166]. The chassis heave frequency response peaks at 4 Hz and then trails off towards zero for higher frequencies. It appears that both MPC controllers fail to control any heave acceleration responses over 10Hz. This can be attributed to the high MPC sample time, which renders both controllers unable to monitor higher frequency responses. The Steffens control metric though, gives a clear indication of the superiority of the proposed controller over the rest of the competing schemes. This metric describes a threshold for vibratory environments that, once exceeded, human discomfort is induced; as shown in Fig. 6.1.8c, the proposed scheme manages to remain largely below this threshold. In particular, for the eigenfrequency of the chassis, the MPC-RBF-L reduces the FFT of the heave displacement by 45.7% compared to the passive suspension and 45.4% compared to the MPC-L controller. Lastly, by assessing the average RMS value for the vertical acceleration of each wheel, one sees that both MPC controllers perform slightly worse than

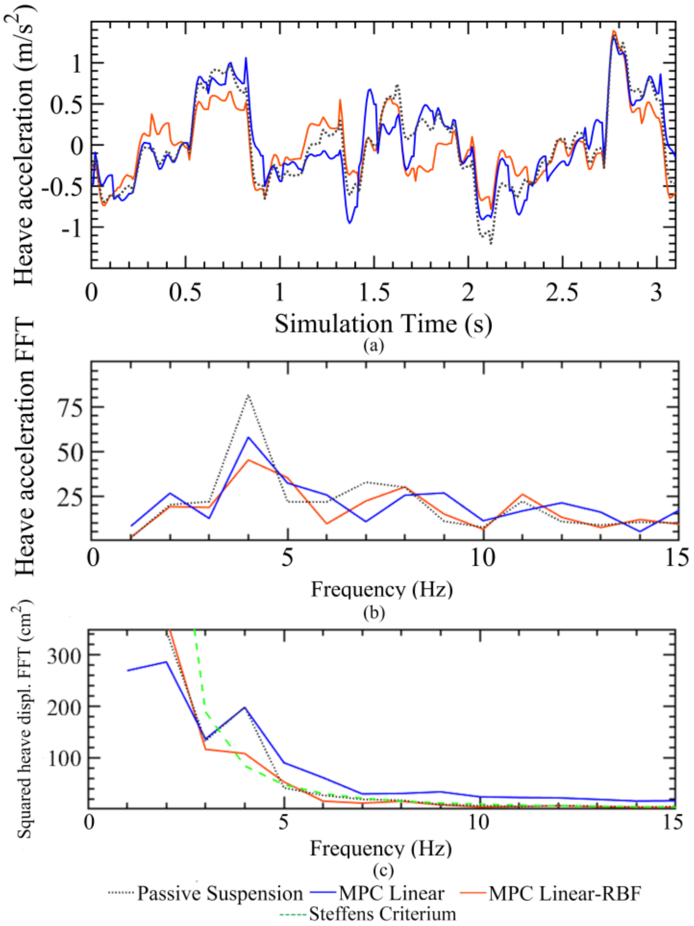


Figure 6.1.8: Results of a random road test: (a) chassis heave acceleration, (b) chassis heave acceleration FFT response, (c) chassis heave displacement FFT response

the passive suspension on this respect; the MPC-L vertical wheel acceleration value is 6.31% larger than the passive suspension one, while the respective MPC-RBF-L value is 4.5% larger. This result is to be expected, since the sampling frequency of 10 Hz of both MPC controllers is lower than the wheel-hop mode frequency, which usually resides in the 10-12 Hz range [167]. Still, this small difference in the aforementioned RMS values does not constitute a significant deterioration of road holding performance for the proposed scheme, thus its potential for practical applications in chassis vibration control is not diminished.

In general, when evaluating SAE as an overall performance metric on the three tests, both MPC controllers significantly increase riding comfort compared to the passive suspension. The MPC-L performance, while being better than the passive suspension, significantly deteriorates when applied to random road conditions; this is due to the limited accuracy of the linear prediction model. The MPC-RBF-L superior performance in all three tests is attributed to the better state approximation capabilities of the RBF-linear model, which enable the computation of more accurate control moves. It should be noted that the actual time needed by the MPC-RBF-L controller for solving the optimization problem was measured to be equal, or smaller to the selected controller sampling time, thus allowing for real-time implementation of the proposed scheme.

6.1.5 Conclusion & Future Prospects

In this section, the application of RBFN models trained with the FM algorithm for the nonlinear states resulted in the successful modeling of the full car plant. Because of the increased computational requirements that the usage of nonlinear prediction models entails, efforts to reduce the overall computational burden of the MPC solution process were made, namely, through the usage of an inverse model of the plant for initializing the optimization

Table 6.1.6: Pulse bump simulation results

Performance metrics	**Passive Car**	**MPC Linear**	**MPC Linear - RBF**
Max absolute acceleration (m/s^2)	2.79	2.3	2.33
Settling time (s)	2.85	2.85	1.81
Sum of absolute error (m/s^2)	113	90.6	81.4

Table 6.1.7: Ramp bump simulation results

Performance metrics	**Passive Car**	**MPC Linear**	**MPC Linear – RBF**
Max absolute acceleration (m/s^2)	2.054	1.489	1.31
Settling time (s)	1.66	2.02	1.65
Sum of absolute error (m/s^2)	101	97.1	81.0

Table 6.1.8: Random road simulation results

Performance metrics	**Passive Car**	**MPC Linear**	**MPC Linear - RBF**
Sum of absolute error (m/s^2)	126.61	112.50	101.05
Heave acceleration FFT at eigenfrequency (m/s^2)	81.6	57.92	45.14
Squared heave displacement FFT at eigenfrequency (cm^2)	199.1	198	108.1
Average RMS value for the vertical acceleration of each wheel (m/s^2)	5.494	5.864	5.754

problem. The proposed MPC scheme was implemented using a realistic LiDAR scanning noise model, tuned using PSO and simulated in three different road scenarios, namely pulse bump, ramp bump and random road.

In general, this case study has demonstrated the effectiveness of a data-driven MPC controller constructed using CI techniques for problems with high-dimensionality and high-nonlinearity. The RBF models used herein are cheaper to evaluate online than the respective ODE integrations of the first-principle equations of the full car plant, and can achieve increased accuracy compared to a linearized model. The proposed controller can be extended to other engineering systems; furthermore, future research plans include the use of RBF networks for modeling the vehicle lateral and yaw dynamics [168], in order to accommodate for the significant nonlinearities arising from lateral tire force behavior. The resulting models could be incorporated in data-driven advanced driver assistance systems (ADAS), thus producing vehicle stability [169], or active torque vectoring [170] controllers of increased performance.

6.2 Data-driven tracking MPC for vessel trajectory control with collision avoidance using real AIS data

The field of automatic collision avoidance for surface vessels has been an active field of research in recent years, aiming for the decision support of officers in conventional vessels, or the creation of autonomous vessel controllers. In this chapter, the multi-ship control problem is addressed using a data-driven model predictive controller that makes use of obstacle ship trajectory prediction models built on the RBF framework and trained on real AIS data sourced from an open-source database. The usage of such sophisticated trajectory prediction models enables the controller to correctly infer the existence of a collision risk and apply evasive control actions in a timely manner, thus accounting for the slow dynamics of a large vessel, such as container ships, and enhancing the cooperation between controlled vessels. The proposed method is evaluated on a real-life case from the Miami port area, and its generated trajectories are assessed in terms of safety, economy and COLREG compliance by comparison with an identical MPC controller utilizing straight-line predictions for the obstacle vessel.

6.2.1 Introduction

In the last two decades, research on automatic collision avoidance (CA) and optimal path planning for surface vessels has intensified, driven by the ever-growing density of maritime traffic in narrow waterways, such as gulfs, ports and canals [171]. Motivated by the design of autonomous surface vehicles (ASV) controllers, but also aiming for the decision support of officers on watch (OOW) of conventional vessels [172], control and optimization tools that ensure the safety and the cost effectiveness of navigational actions are being intensively developed. These tools are perceptive of the surrounding environment through arrays of sensors, radars and other positioning and communication aids. In this context, the automatic identification system encompasses most aforementioned technologies in order to gather positioning and other vessel data. The already vast AIS comprises an ever-expanding worldwide maritime trajectory dataset which is made available by vessels, port authorities and other platforms in charge of efficient and safe maritime path planning. Given the fact that the majority of vessel accidents are related to erroneous handling rather than equipment failure or environmental conditions [173], these tools aim to phase out the human OOW as a vessel controller, or at least augment their navigational decision-making using optimization- and prediction-based methods.

The formulation of the trajectory optimization problem used in CA controllers must take multiple aspects of vessel navigation into account, while being perceptive of their surrounding environment in real time. The generation of control actions that will result in a trajectory remaining sufficiently clear from any stationary or moving objects is not the sole objective: an efficient controller should also ensure the economy of the control actions, as well as the adherence to the collision avoidance rules, commonly known as the COLREGs [174]. Multiple CA controllers have been proposed that fulfil the aforementioned specifications; in [175] a hierarchical multiobjective optimization problem is formulated, that generates an intermediate waypoint for the controlled vessel while accounting for the good seamanship rules. In [176] a fuzzy-Bayesian CA controller is formulated capable of addressing multiple obstacle vessels at once. In [177], optimal trajectories for the CA problem are generated using a B-Spline-based search algorithm. Lastly, in [178] a CA controller utilizes a probabilistic method in order to infer the one-step-ahead position of obstacle vessels, while also accounting for non-COLREG-compliant obstacle vessels.

In general, it has been observed that controllers that are not model-based can have trouble incorporating crucial aspects of the trajectory optimization problem, thus compromising practicality. Without a working model of the controlled vessel, its maneuvering capabilities cannot be easily included in the formulation, neither can the effect of environmental conditions be quantified [12], [179]. For these reasons, model predictive control emerges as an effective control method for the problem at hand, because it utilizes a model of the plant in order to compute an optimal control trajectory based on the predicted trajectory of other ships in the vicinity. As a framework, MPC can account for the uncertainties of both the utilized model of the plant and the trajectory prediction models of other ships, while also incorporating all possible control objectives (such as navigational risk, course smoothness or deviation from the original path) in a single cost function. Some CA controllers based on MPC have been proposed in the literature; a robust MPC controller utilizing straight-line obstacle vessel trajectory predictions is proposed in [12], capable of COLREG compliance and handling of multiple obstacles. In [180], motion planning for an autonomous vessel using a sampling-based MPC method takes place. In [181], an MPC controller for the CA task is built by approximating the behavior of an LQR controller, thus ensuring asymptotic stability of the system. In [13], a neural network is used to approximate the MPC response for the generation of COLREG compliant trajectories for multi ship encounters, is presented. In addition, MPC has been integrated in distributed control frameworks of multi-ship schemes; for example, a distributed MPC scheme has been employed for a multi-vessel formation controller with collision avoidance capabilities [182], or for the robust distributed control of multiple vessels operating for the inter-terminal transport of containers [183].

It becomes apparent that for the scope of the CA task, information about the future trajectories of other ships plays a central role. Prevalent in non-data driven methodologies already used for the vessel trajectory prediction (VTP) problem is the first principles-based modeling technique [184], carrying a number of significant shortcomings, such as their inherent complexity, which has a greater negative impact due to the fact that the model is usually employed multiple times within the duration of each MPC sample. In order to simplify the solution of the employed kinematic differential equations and facilitate the real-time prediction of future states, these types of models are usually created using several assumptions which try to approximate real-world conditions, but also make the final model far less accurate. Therefore, one should employ a more sophisticated, data-driven approach for the creation of effective trajectory prediction models that are included in MPC controllers. Machine learning has answered the call of producing highly accurate models which may be easily integrated in predictive frameworks through the use of black-box modeling, and more specifically artificial neural network approaches [185]. NNs employ different architectures in order to remap the original non-linear problem to a higher-dimensional input space and approximate its dynamics utilizing standard functions. In this context, various NN techniques have been successfully utilized in control frameworks solving the vessel trajectory prediction problem.

Feedforward NN architectures, most commonly represented by the multilayer perceptrons have been employed to solve the VTP problem as in [186], [187], where MLP NNs are trained using the well-established backpropagation algorithm (BP) outperforming rival methodologies i.e., linear models and Kalman filters. In [187] a real AIS dataset gathered from the confined space of a river waterway is used to approximate the vessel dynamics in such environments. BP has been the baseline of more efficient training methods as in [188], where different computational intelligence (CI) approaches like differential evolution, genetic algorithms, and swarm-based techniques are used to modify the original BP algorithm in order to create more accurate feedforward NN models. Other NN architectures, like support vector machines have been employed in conjunction with CI optimization techniques i.e., the particle swarm optimization algorithm, on AIS datasets to solve the VTP problem [189]. In most cases the inherent abilities of NN architectures which can meet the standard of high accuracy are limited to a one-step ahead prediction horizon, in the sense that multi-step ahead predictions would require an approximation of unknown future states to be made and present an error enlarged through propagation to the end of the prediction horizon. Such an error would become critically high after a small number of steps rendering the control framework useless.

To overcome this problem, long-term trajectory prediction approaches have been devised with the inclusion of memory features, such as the recurrent neural networks (RNNs) with their most notable representative i.e., the long short-term memory (LSTM) NNs already used in the context of the VTP problem [190]–[193]. Besides trajectory modeling and prediction in open waters, advances have also been made in crowded port waters as in [194], where another modification of the RNNs, namely the bidirectional gated recurrent unit (Bi-GRU), is used to address the VTP problem outperforming standard NN methods in such scenarios. GRUs are promising candidates for predicting the collective behavior of vessel fleets [195]. Within the context of VTP, RBFs have been integrated in control frameworks by approximating unknown vessel parameters [196]–[198]. Recently, RBFs have been applied on real AIS data in order to produce highly accurate models for one-step and multi-step ahead predictions [6], showing their potential in being integrated to receding horizon control methodologies.

Remarkably, in the research works regarding the design of CA controllers, there are no instances where the multi-step-ahead trajectory prediction of moving obstacles is addressed in such a systematic manner; in most of the reviewed case studies these trajectories are either known a priori, or there are no obstacle ships present whatsoever. An exception occurs in [12], where straight line trajectory predictions are employed, based on estimated current course and speed for the moving obstacle. Indeed, such an approach yields satisfactory approximation results in an open sea setting where ships are expected to travel in a straight line but is of little practical use for the cases of narrow gulfs, ports, or canals where ships need to maneuver in order to navigate through. To the author’s best knowledge, no such implementation occurs for the design of a multi-ship MPC CA controller, much less a nonlinear one.

In this section, a multi-ship MPC controller utilizing RBF prediction models is presented for the CA task. The RBF prediction models are trained using real AIS datasets sourced from an open-source database, and are integrated in an MPC controller for the trajectory prediction of obstacle ships that may pose a CA threat. The usage of such sophisticated prediction models supplies the controller with high-accuracy information, thus allowing for the timely application of appropriate control moves that will result in a safe and efficient trajectory. The proposed method is tested in a CA setting occurring at the Miami port area, and its performance is illustrated by the comparison with an MPC controller utilizing straight-line prediction models.

The section is structured as follows. In subsection 6.2.2 the AIS-data-driven methodology for the creation of the RBF trajectory prediction models is presented. In section 6.2.3 some preliminaries on maritime collision avoidance and optimal trajectory generation are described, and later, the proposed method is presented. In section 6.2.4, the case study based on the port of Miami is outlined, and the simulation results are discussed in depth. Lastly, in section 6.2.5, concluding remarks are made.

6.2.2 Creation of RBF-based Trajectory Prediction Models

Recently it has been shown [6] that RBF NNs trained with the fuzzy means algorithm are ideal candidates for integration to receding horizon control frameworks in the context of the VTP problem. RBFs are already very popular in numerous diverse applications where they have been successfully employed to approximate nonlinear system and process dynamics in order to predict future states and subsequently participate in the formulation of robust control frameworks [199]. However, best modeling practices for RBF networks mandate that a training dataset should be error- and noise-free, a case which is far from truth when using data from AIS transceivers. AIS data are irregularly sampled and contain heavy noise, missing data, and erroneous values. Thus, before employing any modeling technique, proper preprocessing is in order.

The Marine Cadastre service (www.marinecadastre.gov) has been the source of all data used in this work. MarineCadastre.gov is a is a service which gathers and publicly provides AIS data to marine planning initiatives. In this work, data from all days between 1st July 2019 and 30th June 2020 have been included and filtered to keep vessels sailing an area around the port of Miami covered by the geolocation rectangle defined by the latitudes of 25.720° through 25.840° and the longitudes of -80.145° through -80.042°. To conform to the initial assumption of similar size and similar dynamics we allowed only cargo ships sailing on engine power into the dataset, further filtering the dataset to yield a total of 180 vessels.

To address the problems of sample irregularity, noise, and erroneous values, the dataset was resampled to 120 seconds, which was deemed enough to capture the high inertia dynamics of large cargo ships. The interpolation technique applied on the data to perform the resampling was the Akima piecewise cubic interpolation [200] which is quite effective on geolocation data, performing a mild denoising as well. A heuristic which rejects very far-off outlier values due to GPS errors was also applied. The trajectories were split in data samples each one containing ten consecutive vessel positions. Mind that each trajectory's starting point should be the last point of the previous one resulting in an overlap of one point, but this final position will be used as the model's output, so no actual overlapping exists within the input data. The resampling and splitting process yielded a total of about 14k samples from 3.1k resampled trajectories of the initial 180 vessels. Algorithm 6.2.1 depicts the step-by-step procedure of preprocessing.

Algorithm 6.2.1 Preprocessing algorithm for AIS data

```
Initialize for every entry i:
    Keep Vessel ID, timestamp, latitude, and longitude.
    Sort dataset by vessel ID and sort each vessel data by date.
    Apply resampling and outlier filtering on the data of each vessel to achieve a resampling
    of 120 seconds.
For vessel i:
    Split vessel data into trajectories containing ten consecutive vessel positions each.
    Create final preprocessed dataset which should contain the vessel ID and final 10-position
    trajectories. Reject timestamp information.
```

Once the AIS transceiver data has been preprocessed, VTP algorithms can be employed together with CA techniques in order to identify imminent threats and navigate safely and efficiently within heavily crowded port areas or open seas. Let us suppose an available AIS dataset, comprising an arbitrary number of T^v trajectories for a total of V vessels, where $v = 1, 2, \ldots, V$. Let us also suppose that the included trajectories contain an arbitrary number of $K^{v,t}$ AIS messages $AISm_k^{v,t}$ (timestamped geolocation and other data). In this work, for simplicity reasons, we employ the following format in AIS messages

$$AISm_k^{v,t} = \left\{ dt_k^{v,t} \quad y_k^{v,t} \quad x_k^{v,t} \right\} \tag{6.2.1}$$

where $k = 1, 2, \ldots, K^{v,t}$, $dt_k^{v,t}$ denotes the message timestamp, while $y_k^{v,t}$ and $x_k^{v,t}$ are the respective latitude and longitude contained in the *k*-th AIS message for the *t*-th trajectory of the *v*-th vessel. The fact that there are unknown parameters e.g., the state and controls of the vessels, prohibits the use of kinematics in calculating future vessel states. Nevertheless, the vessel dynamics exist in the information hidden within the dataset and can be extracted and, in most cases, approximated by using a black-box modeling technique such as RBF NNs. We can assume that a common underlying pattern exists in the dynamics of same-size vessels executing similar maneuvers, for example when approaching or leaving a port, when berthing, when crossing waterway paths, etc. Thus, if a suitable dataset of sufficient size is made available, an RBF NN can be trained to perform one-step-ahead predictions about a vessel's future geolocation by using past AIS messages as seen in the following equations

$$\begin{Bmatrix} d\hat{y}_{k+1}^{v,t} \\ d\hat{x}_{k+1}^{v,t} \end{Bmatrix} = RBFNN\left(AISm_{k}^{v,t} \quad \ldots \quad AISm_{k-N}^{v,t} \right) \tag{6.2.2}$$

$$\begin{aligned} \hat{y}_{k+1}^{v,t} &= y_{k}^{v,t} + d\hat{y}_{k+1}^{v,t} \\ \hat{x}_{k+1}^{v,t} &= x_{k}^{v,t} + d\hat{x}_{k+1}^{v,t} \end{aligned} \tag{6.2.3}$$

where *N* is the number of past AIS messages given as inputs to the RBF NN.

Delta values of the last position of each sample were used as the model's output, while the first nine positions were the model's input

$$\begin{Bmatrix} d\hat{y}_{k+1}^{v,t} \\ d\hat{x}_{k+1}^{v,t} \end{Bmatrix} = RBFNN\left(y_{k}^{v,t} \quad x_{k}^{v,t} \quad \ldots \quad y_{k-8}^{v,t} \quad x_{k-8}^{v,t} \right) \tag{6.2.4}$$

The $d\hat{y}_{k+1}^{v,t}$ and $d\hat{x}_{k+1}^{v,t}$ values may be added to the last input position to calculate the final predicted vessel position. Based on the above procedure, the results of the modeling process produced an RBF model of very high accuracy [6]. The step-by-step procedure of the modeling stage can be seen in Algorithm 6.2.2.

Note that, the number of past inputs was determined after a trial-and-error procedure, where several RBF models were trained using a different number of inputs. After testing inputs in the range of 3 to 15 past vessel positions, data obtained on model performance showed that using less than nine inputs produced models with reduced prediction accuracy, while using

Algorithm 6.2.2 Creation of RBF models for the VTP task

```
Initialize:
    Load final preprocessed dataset
For vessel i:
        Replace the final value of all included 10-position trajectories with the
        respective delta value according to $\begin{Bmatrix} dy_{10}^{v,t} \\ dx_{10}^{v,t} \end{Bmatrix} = \begin{Bmatrix} y_{10}^{v,t} - y_9^{v,t} \\ x_{10}^{v,t} - x_9^{v,t} \end{Bmatrix}$, so that each trajectory
        sample is in the form $\left[ y_1^{v,t} \quad x_1^{v,t} \quad y_2^{v,t} \quad x_2^{v,t} \quad \dots \quad y_9^{v,t} \quad x_9^{v,t} \quad dy^{v,t} \quad dx^{v,t} \right]$.

        Randomly permute the trajectory samples of i vessel.

        Split the trajectory samples of i vessel into training, validation, and testing
        subsets (in this work a 50%-25%-25% percentage split is used). Do this so that
        all vessels contribute to all three subsets according to the chosen splitting.

Merge all subset samples e.g., all training samples of all vessels together in one
single dataset that will be used for training. Do the same for the validation and testing subsets.

Normalize the inputs and outputs of the training subset. Apply the normalization coefficients to
the validation and testing subsets.

Apply the fuzzy means algorithm on the training and validation dataset using the nine first sets
of $y^{v,t} - x^{v,t}$ values as inputs and the last set of $dy^{v,t} - dx^{v,t}$ values as output.

Yield RBF model trained on VTP data
```

more than nine inputs increased the model's complexity without any accuracy gain compared to the model using nine inputs.

Moreover, a series of tests has been performed by the recurrent application of this model based on a horizon of 5 timesteps for all trajectories of the testing subset, where at each successive timestep the model had to use an increasing number of its own previous predictions. As the model uses more of its past predictions, accuracy decreases due to the enlargement of the propagated prediction error. Such a test can provide intuition on the models' ability to be incorporated in receding horizon predictive frameworks. The quality metrics used for these tests were the root mean squared error (RMSE) and the root mean squared haversine formula distance (RMSHFD). The haversine formula is commonly used to measure great circle distances on spherical surfaces. Table 6.2.1 presents the performance metrics obtained after the recurrent application of the chosen model in order to make predictions for the full length of the trajectories included in the testing subset of the training procedure. Mean RMSE values for the two outputs of the model, namely the latitude and longitude, are provided in degrees, where in can be seen that the error lies in the order of 1.5 thousandth of a degree. The mean RMSHFD metric shows the respective error margin in meters when combining the two model outputs to get the actual predicted future vessel

Table 6.2.1. Performance metrics of the produced RBF NN model

	RBF NN	
	Latitude (y)	**Longitude** (x)
Mean RMSE (deg)	$1439 \cdot 10^{-6}$	$1567 \cdot 10^{-6}$
	Best combined RBF models	
Mean RMSHFD (m)	1200	

position for all tested trajectories. More details on the modeling procedure for the one-step ahead models, including detailed results and comparison with other machine learning approaches can be found in [6].

6.2.3 Vessel Collision Avoidance Problem Statement

The objective of maritime collision avoidance is the generation of a risk-free trajectory which the controlled vessel should follow. A well-defined and effective method of assessing collision risk in the near future is the closest point of approach (CPA). Stemming from the concept of the CPA, two metrics are defined: time to CPA (TCPA) and distance to CPA (DCPA) (see Fig. 6.2.2). A discussion regarding the quick calculation of TCPA and DCPA using the line-of-sight (LOS) distance between the controlled vessel and the obstacle ship is presented in [175]. These metrics depict the urgency of the collision danger of vessel i with another vessel j as well as its magnitude, and by specifying lowest acceptable thresholds $d_{min}$and t_{min} concerning the minimum DCPA and minimum TCPA, respectively, one can construct a risk cost function, as presented in [175]:

$$f_{r,ij} = \begin{cases} \exp\left(a_0\left(d_{min} - DCPA(T_i,T_j) + t_{min} - TCPA(T_i,T_j)\right)\right) - 1, \\ \quad if\ DCPA(T_i,T_j) \leq d_{min} \ \ and \ \ TCPA(T_i,T_j) \leq t_{min} \\ \\ \quad 0,\ if\ otherwise \end{cases} \tag{6.2.5}$$

Here, a_0 is a scaling parameter, and T_i denotes the trajectory matrix containing the x-y position of the *i* vessel for every timestep:

$$T_i = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \tag{6.2.6}$$

By combining TCPA and DCPA, the spatial-temporal nature of a maritime collision risk with vessel *i* is successfully reflected. The physical interpretation of equation (6.2.12) is that a candidate trajectory with larger minimum distance from an obstacle ship occurring at an earlier time will always be safer than a path with a smaller minimum distance and/or earlier time of occurrence. Common values for $t_{min}$and d_{min} are 10min and 0.6nm; because

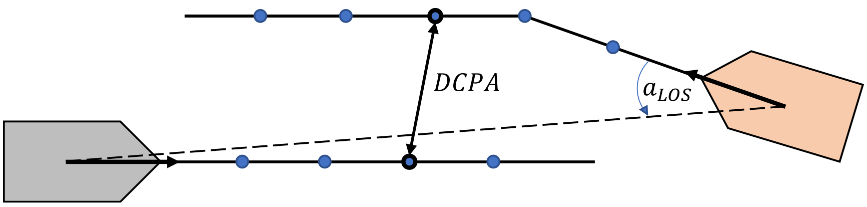


Figure 6.2.1. Illustration of the CPA metrics, as well as the LOS angle concept.

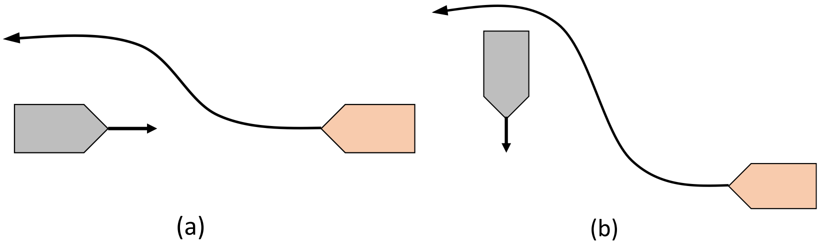


Figure 6.2.2. (a) a head-on situation between two ships (b) a crossing situation between two ships (give-way); the orange ship must give way to the crossing ship on its starboard side.

the present work is concerned with CA in busy waterways such as ports, a lower d_{min}value of 0.4nm is used. In any case, equation (6.2.12) can be readily incorporated in the cost function of an MPC optimization problem formulation.

A second item in the domain of trajectory generation is efficiency. Vessels should strive to not deviate too much from their original course, when addressing a collision risk with another vessel. The efficiency of the generated trajectory T_i for vessel i can be reflected by calculating the sum of absolute deviations from the original trajectory $T_{OG,i}$

$$f_{d,i} = \left\|T_{OG,i} - T_i\right\|. \tag{6.2.7}$$

Next, an important requirement to be fulfilled when addressing the problem of CA are the COLREGs [174]. The implementation of the COLREGs restrict the domain of possible candidate paths according to the type of encounter, for example “head-on”, “crossing” and “overtaking”. Head-on vessels should pass each other on the port side, while a vessel crossing from the starboard side should be given way. A visual depiction of the encounter rules takes place in Fig. 6.2.2. Multiple approaches for the modeling of the COLREG rules have been made in the literature [172], [175], [178], although these are usually concerned with a one-step-ahead calculation**.** However, for the case of an MPC controller, in order to ensure COLREG compliance for a candidate trajectory, all of its waypoints must be taken into account. By assuming that the LOS angle is increasing in the anti-clockwise direction, one needs to evaluate whether the LOS angles of each sequential trajectory timestep position are increasing monotonically, in order to confirm the compliance of the trajectory for the ‘head-on’ and ‘give-way’ situations.

The idea is depicted in Fig. 6.2.4, where a head-on encounter between vessels *i* and *j* occurs; here, the LOS angles for trajectory T_i monotonically increase, therefore it is deemed as compliant. In contrast, the monotonically decreasing LOS angles of the T_i' trajectory confirm its non-compliance as per the COLREGs intentions. A penalty for non-compliance of a vessel *i* encountering a vessel *j* in a 'head-on' or 'give-way' situation can therefore be formulated,

$$P_{ij} = \begin{cases} 1, & if\ a_{LOS_{ij}} \searrow \\ 0, & if\ otherwise \end{cases} \tag{6.2.8}$$

where $a_{LOS_{ij}}$ is the LOS angle vector, calculated for each trajectory point of encountering vessel *i* and the current position of encountered vessel *j*.

6.2.4 Methodology

6.2.4.1 Vessel Kinematic Model Creation

The generated vessel trajectory, apart from being safe and COLREG compliant, should also take into account the maneuvering capabilities of the controlled vessel, i.e. it should be guaranteed that the trajectory is kinematically possible to be tracked by the vessel. The feasible search domain of the trajectory optimization problem can be constructed by a purely geometric approach in the case of a one-step-ahead calculation, such as in [175], where the design variables are the vessel's next position and course. However, the extension of this geometric approach to multiple-steps-ahead requires the application of nonlinear constraints that would bound every sequential vessel position with its previous one, in order to enforce technical feasibility. For this reason, a model-based approach is preferred. The Nomoto models constitute a class of vessel course models that are tailored for this task, and have been widely adopted, not only for the design of CA schemes [201], but also for path tracking controllers [202]. The 1st order linear Nomoto model is shown as follows:

$$T_s \frac{d\omega}{dt} + \omega = K_s a \tag{6.2.9}$$

Here, ω is the angular velocity of the vessel, while a is the control input to the vessel's rudder. The maneuvering capabilities of the vessel are reflected by the T_s and K_s constants, called time constants and rudder gain constants, respectively, while typical values are in the [0.5, 2] range for both. Solving the differential equation (16) by assuming constant rudder angle input for a t time interval, the 1st order linear Nomoto model can be discretized as follows [178]:

$$\varDelta\theta(t) = K_s\, a \left(t - T_s + T_s \exp\left(\frac{t}{T_s}\right)\right) \tag{6.2.10}$$

Here, $\varDelta\theta$ is the course change that would occur if a control input of a was applied and held for a time period of t. By setting this time period t as the discretization interval $\varDelta t$, a course model can be used to create a discrete vessel position model as follows:

$$\theta_{k+1} = \theta_{k-1} + \varDelta\theta_k(a_k) \tag{6.2.11a}$$

$$x_{k+1} = x_k + \cos(\theta_{k+1})\, V_k\, \varDelta t \tag{6.2.11b}$$

$$y_{k+1} = y_k + \sin(\theta_{k+1})\, V_k\, \varDelta t \tag{6.2.11c}$$

Here, θ_k, x_k, y_k is the current course, horizontal displacement and vertical displacement according to a global reference frame, respectively, while V_k is the vessel velocity. The discretization interval $\varDelta t$ can be set according to the simulation resolution required. Eqs. (6.2.11) constitute a discrete position model L_i for the i-th vessel,

$$x_i(k+1) = L_i(u_i, x_i(k)) \tag{6.2.12}$$

with input vector $u_i = [a \quad V]$ and state vector $x_i = [\theta \quad x \quad y]$. By evaluating the discrete vessel position model L_i for $\{1,2,\dots,n\}$ consecutive timesteps, where n the total timesteps, a trajectory T_i can be created for the i-th vessel, as shown in eq. (13).

6.2.4.2 MPC scheme with Collision Avoidance

The MPC framework has demonstrated its aptitude in handling the uncertainties and nonlinearities of the CA problem multiple times in the literature [12], [24] , however, no other works have incorporated a data-based obstacle trajectory prediction model in their formulation, much less a nonlinear one. In MPC, the optimal control moves of the controlled vessels are calculated for multiple steps ahead by solving a constrained optimal control problem, with constraints in real time, for each controller sample time t_{cst}. The cost function of the OCP is constituted by a prediction horizon N. Given a set of controlled vessels $\mathcal{V}_c = \{1, 2, \dots, N_c\}$ and a set of non-controlled or obstacle vessels $\mathcal{V}_o = \{1, 2, \dots, N_o\}$ where N_c and N_o are the total number of controlled and non-controlled vessels, respectively, the OCP's cost function can be formulated as the summation of all the cost functions of the respective controlled vessels for the kth timestep:

$$\min_{U(k)} J_N(X(k), U(k), X_o(k)) \tag{6.2.13a}$$

$$s.t. \;\; U(k) \in \mathbb{U} \tag{6.2.13b}$$

$$x_0 = x_0' \tag{6.2.13c}$$

$$P(\mathrm{X}) = 0 \qquad (6.2.13\mathrm{d})$$

$$N_d\left(T_i, T_j\right) \geq \mathrm{d}_e \qquad (6.2.13\mathrm{e})$$

$$X^+ = f_{Nom}(\mathrm{X}, U) \qquad (6.2.13\mathrm{f})$$

$$X_o = f_{RBF}(\tilde{X}_o) \qquad (6.2.13\mathrm{g})$$

$X(k)$ and $X_o(k)$ contain the states of the controlled and obstacle vessels, respectively, in a single state matrix. $U(k)$ is the input matrix and is created by the horizontal concatenation of the input vectors of all controlled vessels $\mathcal{V}_c$, up to horizon N at timestep k:

$$U(k) = \begin{bmatrix} u_1(k) & u_1(k+1) & \cdots & u_1(k+N-1) \\ \vdots & & \ddots & \vdots \\ u_{N_c}(k) & u_{N_c}(k+1) & \cdots & u_{N_c}(k+N-1) \end{bmatrix} \qquad (6.2.14)$$

Next $X(k)$ and $X_o(k)$ are created by the horizontal concatenation of the state vectors of all controlled and non-controlled vessels $\mathcal{V}_c$ and $\mathcal{V}_o$, respectively, up to the prediction horizon N.

$$X(k) = \begin{bmatrix} x_1(k) & x_1(k+1) & \cdots & x_1(k+N-1) \\ \vdots & & \ddots & \vdots \\ x_{N_c}(k) & x_{N_c}(k+1) & \cdots & x_{N_c}(k+N-1) \end{bmatrix} \qquad (6.2.15\mathrm{a})$$

$$X_o(k) = \begin{bmatrix} x_{o,1}(k) & x_{o,1}(k+1) & \cdots & x_{o,1}(k+N-1) \\ \vdots & & \ddots & \vdots \\ x_{o,N_o}(k) & x_{o,N_o}(k+1) & \cdots & x_{o,N_o}(k+N-1) \end{bmatrix} \qquad (6.2.15\mathrm{b})$$

For simplicity, because consecutive state vectors $x_i(k)$ up to $x_i(k+N-1)$ constitute a single trajectory $T_i(k)$, one can write $X(k)$ and $X_o(k)$ as the concatenation of the trajectories of the respective vessel sets $\mathcal{V}_c$, $\mathcal{V}_o$ as per equation (6.2.15):

$$X(k) = \begin{bmatrix} T_1(k) \\ \vdots \\ T_2(k) \end{bmatrix} \quad X_o(k) = \begin{bmatrix} T_{o,1}(k) \\ \vdots \\ T_{o,N_o}(k) \end{bmatrix}. \qquad (6.2.16)$$

State transition function $X^+ = f_{Nom}(X, U)$ represents the kinematic model that is used for the controlled vessels. $\mathbb{U}$ represents the admissible input space, $\mathbb{X}$ the admissible state space denoted by constraints (6.2.13c-g), and $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$ the admissible state-input space. One can also rewrite $X(k)$ as the vertical concatenation of the two state matrices $X(k)$, $X_o(k)$, containing the trajectories of all vessels $\mathcal{V} = \mathcal{V}_c \cup \mathcal{V}_o$:

$$X(k) = \begin{bmatrix} T_1(k) & \dots & T_{N_c}(k) & T_{o,1}(k) & \dots & T_{o,N_o}(k) \end{bmatrix}^T. \qquad (6.2.17)$$

Returning to OCP (6.2.13), $J_N(\cdot)$ represents the cost function:

$$J_N(X,U,X_o)=\sum_{j=0}^{N-1}\left[\sum_{i\in V}\ell_i\big(X(k+j),U(k+j),X_o(k+j)\big)\right.\\ \left.+\sum_{i\in V}V_{f_i}\big(X(k+N-1)\big)\right]+\sum_{i\in V}G(U(k)) \tag{6.2.18}$$

The $G(\cdot)$ function, weighted by the a_G parameter, encourages the smoothness of the control actions and consequently, the generated trajectories of the controlled vessels:

$$G(U_i(k))=a_G\sum_{j=1}^{N-1}\left\|U_{i,j+1}(k)-U_{i,j}(k)\right\|_2. \tag{6.2.19}$$

Next, $\ell_i(\cdot)$ is the stage cost function of the i-th controlled vessel, and consolidates the collision avoidance and course keeping objectives:

$$\ell_i(X,U,X_o)=a_r\left(\sum_{j\in\mathcal{V}_c\backslash i}\left({f_{r,ij}}^2(X_i,X_j)\right)+\sum_{j\in\mathcal{V}_o}\left({f_{r,ij}}^2\left(X_i,X_{o_j}\right)\right)\right)\frac{1}{|\mathcal{V}\backslash i|}\\ +a_d\,{f_{d,i}}^2(X_i). \tag{6.2.20}$$

In equation (6.2.20), $f_{r,ij}(\cdot)$ is the collision risk between the i-th and the j-th vessel, as calculated using their respective trajectories $X_i(k)$, $X_j(k)$ by applying equation (6.2.5), and $f_{d,i}(\cdot)$ is the deviation from the original trajectory $T_{OG,i}$, as expressed in equation (6.2.7). Both terms are weighted by the a_r and a_d weighting parameters, respectively. Since we are concerned with the safety of the generated trajectory throughout the whole prediction horizon, the mean collision risk from all vessels in set $\mathcal{V}\backslash i$ is evaluated, in contrast to other approaches [175], where only the maximum collision risk at time *k* is minimized. This way, all possible collision risks are addressed and reduced simultaneously, thus avoiding the adverse possibility of evading one collision risk and increasing another. Moreover, the reason that risk avoidance is entered as a control objective in equation (6.2.20) and not as a hard optimization constraint is to ensure the feasibility of the OCP (6.2.13) in the case of the existence of an inescapable collision risk; as shown in equation (6.2.5), risk is a function of distance to CPA, meaning that the controller will continue to attempt to maximize that distance, thus continuing to fulfil the control intention. Lastly, the terminal cost for each vessel V_{f_i} is calculated as the cost of deviation from the original trajectory for the last state:

$$V_{f_i}(X_i(k))={f_{d,i}}^2\left(X_{i,N}(k)\right). \tag{6.2.21}$$

At this point, it must be noted that since the state matrix $X(k)$ consolidates all controlled vessel trajectories, a degree of cooperation is induced. Lastly, returning to the OCP denoted in equation (6.2.13), the $U(k)$ input matrix is bounded by the upper and lower matrices U_u, U_l, respectively. The function $P(\cdot)$ returns the COLREG non-compliance penalties for the controlled vessels V_c as calculated in equation (6.2.8), and are required to be zero via an equality constraint. The function $N_d(\cdot)$ returns the DCPAs of all controlled vessels $\mathcal{V}_c$, requiring them to be above an emergency distance d_e *(*$d_e \leq d_{min}$*)*. The inclusion of this constraint in the OCP constitutes a hard guarantee of collision avoidance.

The next item to be addressed regarding the MPC formulation is the used model that maps the input variables U to the state variables of the controlled vessels X_c. Here, the 1st order linear Nomoto model is used, as described in equation (6.2.11):

$$x_i(k+1) = f_{Nom}\big(u_i,\ x_i(k)\big), \qquad i \in \mathcal{V}_c, \tag{6.2.22a}$$

$$x_i(k+1) = f_{Nom,real}\big(u_i + e(u_i),\ x_i(k)\big), \qquad e(u_i) = u_i\ N_G(0, \sigma^2) \tag{6.2.22b}$$

Note that $f_{Nom}(\cdot)$ represents the controller model and $f_{Nom,real}$ the plant model used in simulations, which contains input noise that accounts for modeling error e and environmental parameters; N_G is a random variable sampled from a Gaussian distribution with a standard deviation of σ.

Finally, the state matrix of the non-controlled vessels $X_o(k)$ representing the trajectories of the obstacles is unknown, and thus an estimation is required, based on past positions. For this task, the prediction model $f_{RBF}(\cdot)$ presented in Chapter 3 is employed for each non-controlled vessel j, and its trajectory is estimated using its past nine positions $\tilde{x}_{o,j}$ that were measured:

$$\hat{T}_{o,j}(k') = f_{RBF}\left(\tilde{x}_{o,j}(k'), \tilde{x}_{o,j}(k'-1), \dots, \tilde{x}_{o,j}(k'-9)\right), \qquad j \in \mathcal{V}_o. \tag{6.2.23}$$

Note that, in order to filter out possible noise during obstacle position measurement $\tilde{x}_{o,j}$, a larger timestep is used, which is a multiple of the controller timestep k, i.e. $k' = \alpha k$, where α is an integer. In other words, the obstacle trajectory predictions are refreshed every k', meaning that the controller uses the same predictions during the range $[k', k'+1]$.

Still, the deployment of $f_{RBF}(\cdot)$ in MPC poses a significant challenge, since its generated predictions $\hat{T}_{o,j}$ that form $X_o(k)$ enter the objective function $J_N(\cdot)$ as shown in eq. (6.2.18), and can thus affect its Lyapunov characteristics. It is noted that since $X_o(k)$ is refreshed on a larger sampling time than MPC, it can be considered as a piece-wise constant signal within the OCP (6.2.13). By furthermore treating it as a control reference signal, one can possibly utilize stability results on tracking NMPC for piece-wise constant reference signals, such as the one recently proposed by Limon et al. [203]: The idea is to add an artificial reference signal as an extra decision variable in the OCP, in order to circumvent any loss of feasibility originating from external changes to the setpoint. The convergence to the (original) setpoint is subsequently pursued by extending the OCP's objective function $J_N(\cdot)$ with a term that penalizes the difference of the artificial reference from the setpoint. The new OCP is thus formulated:

$$\min_{U(k),X_o^{(a)}(k)} \; J_N^{(a)}\left(X(k),U(k),X_o(k),X_o^{(a)}(k)\right) \tag{6.2.24a}$$

$$s.t. \;\; (6.2.13\text{b} - \text{h}) \tag{6.2.24b}$$

Where the new objective function $J_N^{(a)}$ is given by:

$$\begin{aligned} &J_N^{(a)}\left(X,U,X_o,X_o^{(a)}\right) = \\ &\sum_{j=0}^{N-1}\left[\sum_{i\in V} \ell_i\left(X(k+j),U(k+j),\, X_o(k+j)-X_o^{(a)}(k+j)\right)\right. \\ &\left. + \sum_{i\in V} V_{f_i}\big(X(k+N-1)\big)\right] + \sum_{i\in V} G(U(k)) + V_O\left(X_o,X_o^{(a)}\right) \end{aligned} \tag{6.2.24c}$$

Here, $V_O: \mathbb{R}^{N_o\times N} \to \mathbb{R}$ is called the offset cost function, and penalizes the difference between X_o and $X_o^{(a)}$. One can consider $J_N^{(a)}(\cdot)$ as a relaxation of $J_N(\cdot)$, which is performed in order to ensure feasibility in the presence of an externally-set piece-wise constant reference signal.

Corollary 6.2.1 (Asymptotic stability of MPC based on eqs. *6.2.24*): *Suppose assumptions of Theorem 4.3.3 are satisfied as well as*

A. *Vessel trajectory prediction function* $x^+ = f_{RBF}(x,u)$ *is bounded & continuous* $\forall(x,u) \in \mathbb{Z}$, $f_{RBF}(x,u) \in \mathbb{X}_f$ *and* $f_{RBF}(0,0) = 0$

B. *Vessel kinematic model function* $x^+ = f_{Nom}(x,u)$ *is bounded & continuous* $\forall(x,u) \in \mathbb{Z}$

C. *Initial state* x_0' *lies in the region of attraction & prediction horizon N is sufficiently long*

D. *Assumption 2 of [203] is satisfied.*

Then, the origin is asymptotically stable in $\mathcal{X}_N$, *where* $\kappa_N(x)$ *is the MPC control law derived from 6.2.24.*

Proof: Vessel stage cost ℓ_i is a positive definite function, representing the collision risk comprised by $f_{r,ij}(\cdot)$ and $f_{d,i}(\cdot)$, which are also positive definite; $f_{r,ij}(\cdot)$ is built using $exp(\cdot)$ (positive definite as well), and $f_{d,i}(\cdot)$ which represents the norm of state deviations from the original trajectory $T_{OG,i}$. Since the MPC stage cost is the summation of ℓ_i for all $i \in \mathcal{V}_c$, then the MPC stage cost is also positive definite. Next, the MPC terminal penalty $\sum_{i \in \mathcal{V}_c} \left[V_{f_i}(X) \right]$ is also positive definite. Moreover, Assumption D can be satisfied by opting the offset function V_O to be any $\mathcal{K}_\infty$ function.

Therefore, given Assumptions A-D, Theorem 1 of [203] can be leveraged to show asymptotic nominal stability of the scheme. ■

Next, in order to alleviate a possible computational burden for the MPC optimization problem, an important assumption should be made. The formulation of the control scheme as-presented would give rise to a high-dimensional search space for the MPC optimization problem, thus greatly hindering its effective solution. It is assumed then, that all vessels retain their initial speed, with the only controllable variable being the vessel's rudder angle; this way, the total number of control variables is reduced. This approach to the CA problem has occurred in the literature [175] and is not simplistic, for two reasons: first, good seafaring practice dictates that course change maneuvers are preferred over speed ones, not only because they conserve energy, but also because they better emphasize the intentions of the vessel to outside observers, such as other vessels in the vicinity. Secondly, since large container ships will be examined in the scope this case study, their large longitudinal inertia [202] confirms the assumption that the speed remains almost constant during the timeframe of a typical CA maneuvering scenario. Therefore, for the scope of this work, the input matrix U at timestep *k* is formulated as follows:

$$U(k) = \begin{bmatrix} a_1(k) & a_1(k+1) & \cdots & a_1(k+N-1) \\ \vdots & & \ddots & \vdots \\ a_{N_c}(k) & a_{N_c}(k+1) & \cdots & a_{N_c}(k+N-1) \end{bmatrix} \qquad (6.2.24)$$

Where $a_{\mathrm{i}}(k)$ is the rudder angle of vessel *i* at timestep *k*.

Having defined all aspects of the MPC optimization problem, a reiteration of the challenges of the CA control problem and how they are addressed by the controller is in order: Firstly, the goal of the control design is to generate trajectories for the controlled vessels that are risk-free (equation (6.2.5)), smooth (equation (6.2.19)), COLREG-compliant (equation (6.2.8)) and not deviative from the original course (equation (6.2.7)). Possible collision risks are assessed by utilizing trajectory predictions for non-controlled (obstacle) vessels in the vicinity. The controllable variables are the rudder angles of the vessels (vessel speed is considered constant), while a discrete 1st order Nomoto model (equation (6.2.21)) is used for the modelling of the vessel dynamics, which was also infused with a noise signal for the purpose of accounting for uncertainties and environmental factors. The aforementioned vessel dynamics model has been compared to its higher-order nonlinear counterparts in [204], and it was shown that vessel course inaccuracies occur only for high yaw rates. Given the fact that the proposed CA method is concerned with large vessels with slow dynamics, the used vessel dynamics model is adequate for the case. In addition, MPC has shown to be robust against model uncertainties or input noise [199]. Finally, the constraints that must be adhered to when searching for the optimal solution (Problem (6.2.13)) are the technical bounds on the controlled variables (i.e. maximum and minimum rudder angles) and the COLREG compliance of the result trajectory.

6.2.4.3 Data-driven collision avoidance control framework

Having presented the proposed MPC controller, this section describes its integration within a general control framework. As shown in fig. 6.2.3, the framework is comprised by an offline and an online process. The offline process corresponds to the RBF trajectory prediction model training, using data from a specific area of interest (for example, a port) - naturally then, it could be undertaken by the port authority. The online process corresponds to the real-time control of autonomous vessels in the presence of obstacle vessels in the area of interest. The MPC CA controller, as described in subsection 6.2.4.2, is integrated here and is supplied with real time trajectory predictions of all obstacle vessels in order to calculate the optimal control actions for the controlled vessels. Since the RBF trajectory prediction model has been trained offline in the port authority premises, it is sensible to place the MPC controller there too, and communicate the computed control actions per control timestep via a communications link with the controlled vessels. Figure 6.2.4 demonstrates this concept.

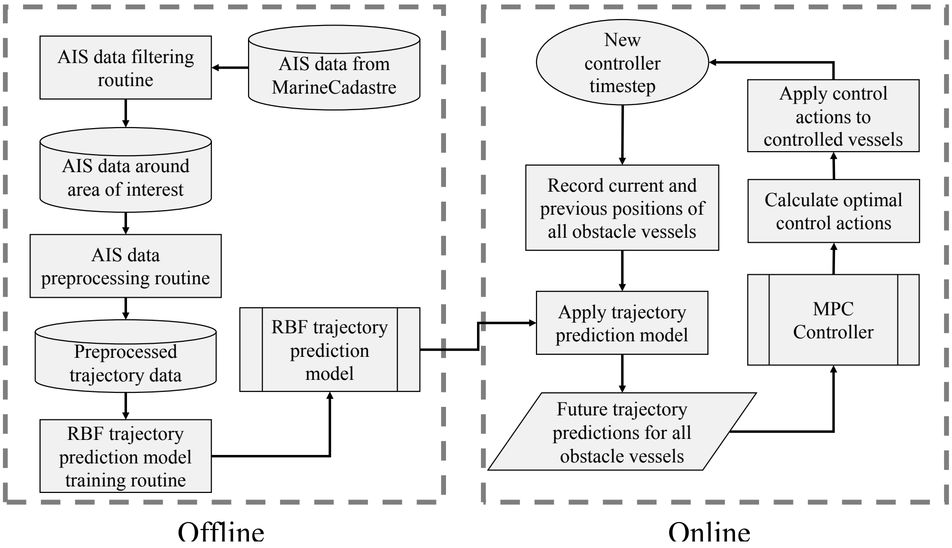

AIS data filtering routine
AIS data from MarineCadastre
AIS data around area of interest
AIS data preprocessing routine
Preprocessed trajectory data
RBF trajectory prediction model
RBF trajectory prediction model training routine
New controller timestep
Apply control actions to controlled vessels
Record current and previous positions of all obstacle vessels
Calculate optimal control actions
Apply trajectory prediction model
MPC Controller
Future trajectory predictions for all obstacle vessels
Offline
Online

Figure 6.2.3. The proposed control framework

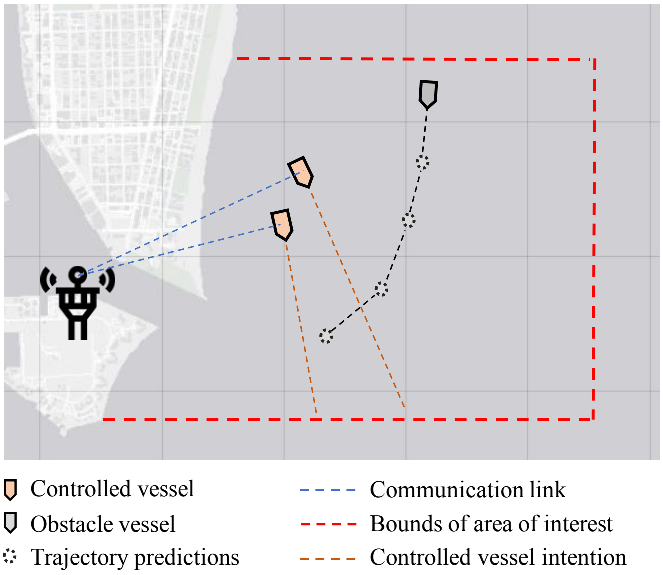


Figure 6.2.4. An example of two cooperating vessels with a central controller

The OCP (6.2.13) is solved using an active-set sequential quadratic programming (SQP) algorithm described in (2.1.4), which involves iterative calls to the objective function [205]. As shown in fig. 6.2.5, the integration of the MPC controller in the control framework requires the calculation of the obstacle vessel trajectory predictions for every controller timestep. Therefore, two main sources of computational complexity arise: The first is the evaluation of the RBF trajectory prediction model, which is shown to be in the order of magnitude of milliseconds [6], meaning that multiple obstacle vessels can be accounted for by the control scheme. The second is the solution of the optimization problem (equation (6.2.13)) by the SQP algorithm, which is known to converge quickly and with few objective function calls [206]. It is concluded that a typical controller timestep duration, comprised by the two aforementioned sources, will not exceed the order of magnitude of seconds, which is considered reasonable given the slow dynamics of large vessels.

6.2.5 Results

6.2.5.1 Case Study

In this subsection, the performance of the proposed multi-ship MPC controller is assessed using real-life obstacle ship trajectories, which were sourced and preprocessed as described in section 6.2.4. In order to underline the importance of using sophisticated trajectory prediction models in the context of CA controller design, the proposed method is compared to an MPC controller that uses straight-line predictions for the trajectories of obstacle ships, based on their current course and speed [12]. To this end, two crossing scenarios are examined, while performance indicators of the generated trajectories are extracted and discussed in detail. The simulations were coded and executed on MATLAB 2020b, on a computer with an Intel i7 processor and 16 GB RAM. The simulation sample time is 30”. Lastly, the tuning and parameters of the methods are shown in Table 6.2.2, while the vessel parameters are shown in Table 6.2.3.

For this case study, two controllable vessels are chosen, moving in parallel to each other and encountering an obstacle vessel moving into the port of Miami. For the performance evaluation of the two controllers, two scenarios are created; the first contains a head-on encounter type, while the second an overtaking maneuver that changes into a crossing encounter as time progresses. In the first scenario, the two controlled vessels are leaving the port of Miami at a course of 110°, when they encounter a single obstacle on their

Table 6.2.2 MPC tuning parameters

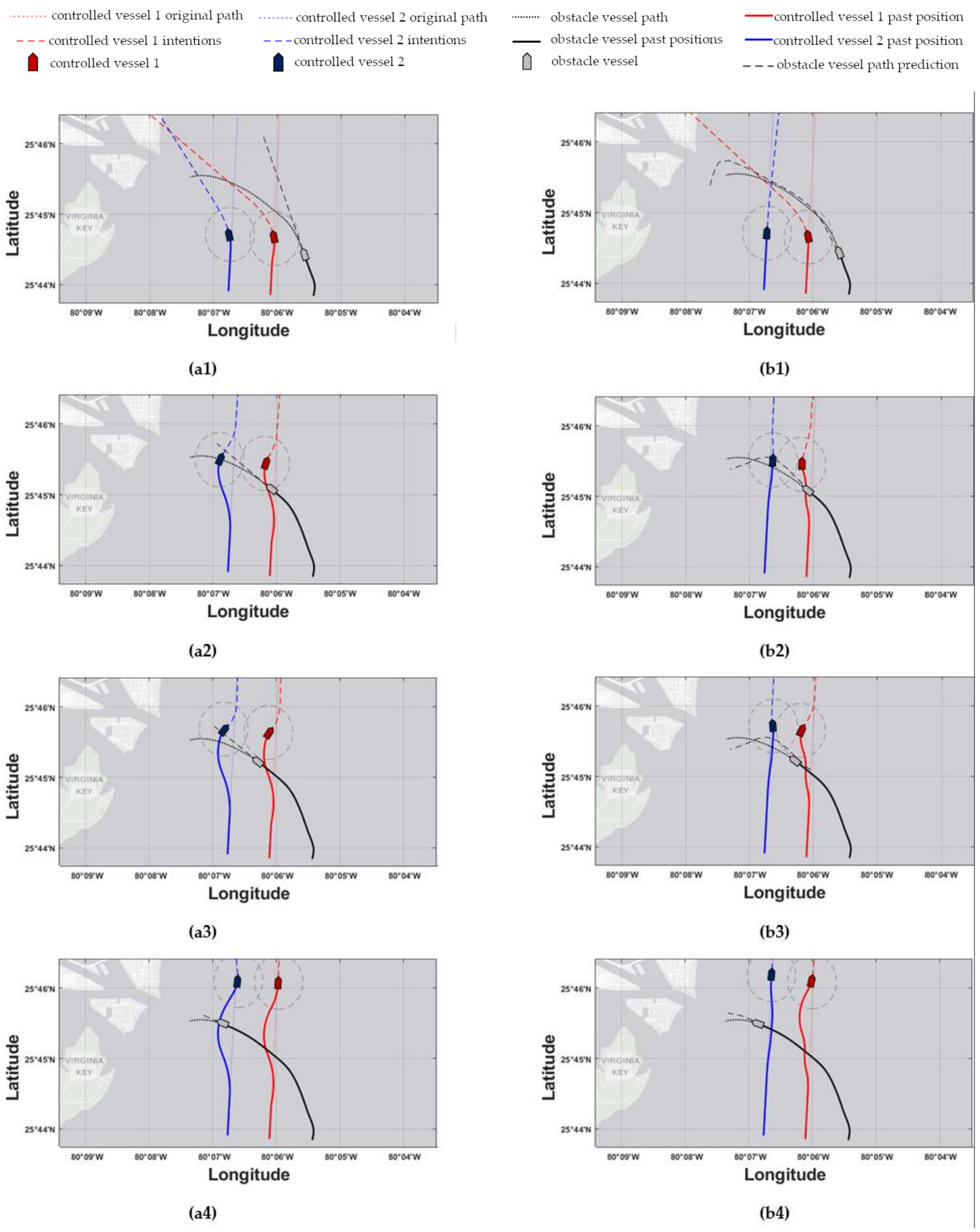
Parameter	Description	Value
t_{cst}	Controller sample time	1′
h_p	Control horizon	5
h_c	Prediction horizon	15
a_0	Risk function scaling parameter	3
a_r	Risk term weighting parameter	1
a_d	Course deviation term weighting parameter	0.05
a_G	Control action smoothness term weighting parameter	5

Table 6.2.3. Vessel parameters

Parameter	Description	Value
d_{min}	Minimum allowable DCPA for risk calculation	750m
d_e	Emergency distance	200m
t_{min}	Minimum allowable TCPA for risk calculation	10′
K_s	Rudder gain constant	0.5
T_s	Rudder time constant	2

starboard side which, in turn, is looking to enter the port. In the second scenario, the two controlled vessels are overtaking an obstacle vessel on her port side when suddenly, she turns port-side in order to enter the port of Miami, crossing into their intended path. The challenge posed by the two scenarios is that the two controllable vessels should maintain a safe distance between each other and the obstacle vessel, while also navigating smoothly and without unnecessary deviation from their original course. It should also be noted that the obstacle vessel is non-controllable and therefore follows a predetermined path, without considering other vessels.

6.2.5.2 Results & Discussion


controlled vessel 1 original path
controlled vessel 1 intentions
controlled vessel 1
controlled vessel 2 original path
controlled vessel 2 intentions
controlled vessel 2
obstacle vessel path
obstacle vessel past positions
obstacle vessel
controlled vessel 1 past position
controlled vessel 2 past position
obstacle vessel path prediction
Latitude
25°46'N
25°45'N
25°44'N
VIRGINIA KEY
Longitude
80°09'W
80°08'W
80°07'W
80°06'W
80°05'W
80°04'W

(a1)

(b1)

(a2)

(b2)

(a3)

(b3)

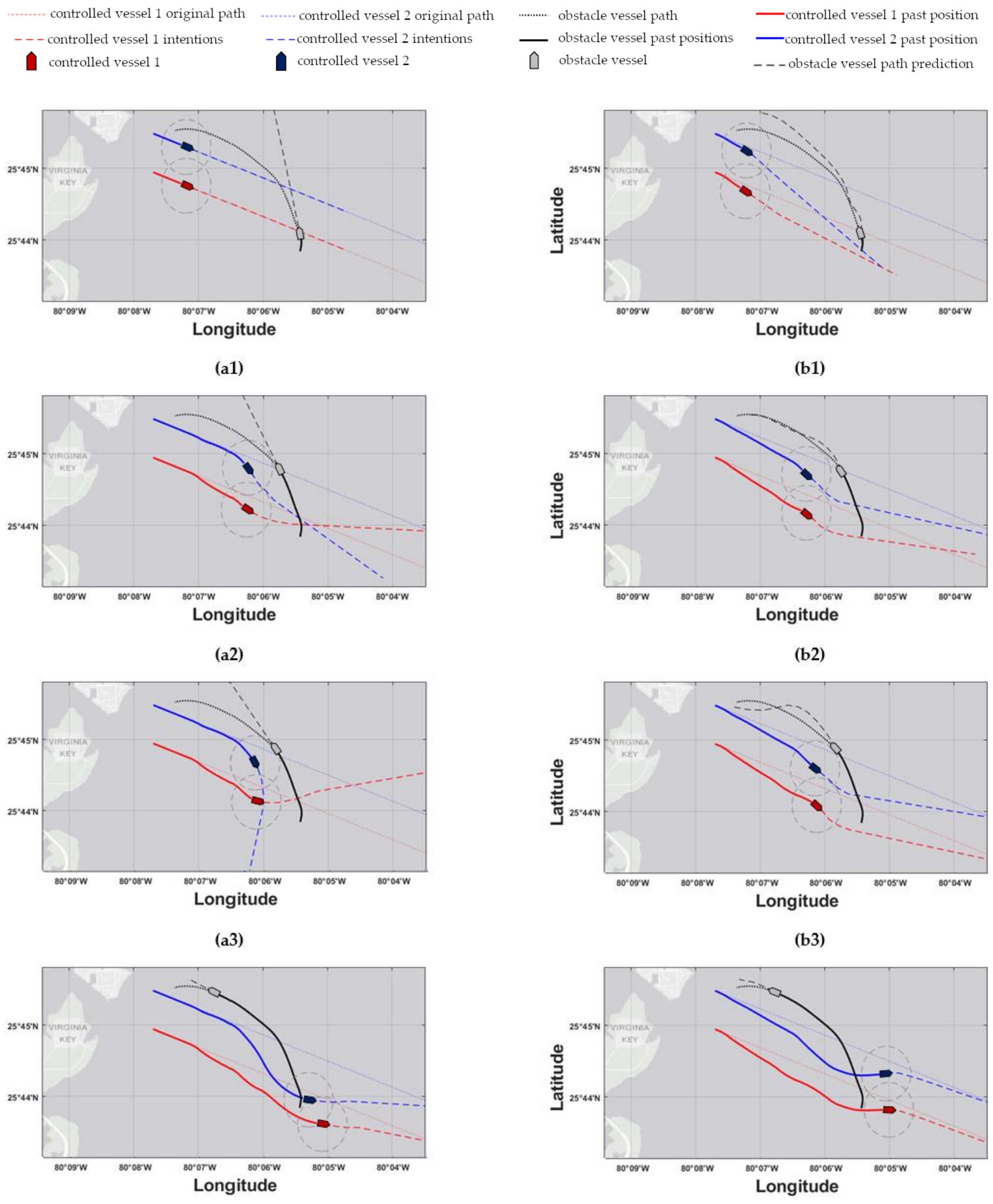
(a4)

(b4)

Figure 6.2.5. Scenario 1. The left subfigure column refers to the MPC-SLP scheme, while the right to the MPC-RBFP scheme. Subfigures **(a1, b1)** refer to time instance 3′, **(a2, b2)** to 9′, **(a3, b3)** to 10′ and finally, **(a4, b4)** to 16.5′

The response of the MPC controller utilizing straight-line prediction models (hereby referred to as 'MPC-SLP' for the first scenario is shown in the left column of subfigures within Fig. 6.2.5, for the 3-, 9-, 10- and 16.5-minute timesteps. The response of the proposed MPC controller utilizing RBF prediction models (hereby referred to as 'MPC-RBFP') for the same scenario and same time instances are shown in the right column of subfigures within Fig. 6.2.6. Next, the responses of MPC-SLP and MPC-RBFP for the second scenario are shown in the left and right subfigure columns of Fig. 6.2.8 respectively, for the 6-, 12-, 13.5-, and 17-minute timesteps. In the aforementioned response figures, the red and blue dotted lines denote the original, undisturbed trajectory for controlled vessels 1 & 2, respectively, while the black dotted line shows the predetermined path that the obstacle ship will follow as the simulation progresses. Next, the red and blue dashed lines denote the trajectory that controlled vessels intend to follow, as calculated by the current MPC iteration, while the black dashed line shows the current trajectory prediction of the obstacle ship, as utilized by the MPC controller. The grey dashed circles have a radius of d_{min} and denote the safe ship domain for the two controlled vessels; should any vessel enter another's domain at any time, a collision risk arises. Lastly, the red-colored and blue-colored rectangles mark the controlled vessels 1 & 2 positions, respectively, while the grey rectangle marks the obstacle ship's position; it should be noted that the markers are not to-scale with the real dimensions of the vessels, since they have been enlarged for graphical convenience.

Firstly, in order to assist the discussion in this subsection, distance plots are generated for the controlled vessels that are in closest proximity with the obstacle ship, for each scenario (see Fig. 6.2.7). In addition, the performance metrics for each controller in each scenario are shown in Table 6.2.4. For the head-on encounter of scenario 1, the correct trajectory prediction of the obstacle ship proves vital for the success of the proposed scheme. Considering timestep 3 (see Figure 6.2.5.a1, 6.2.5.b1) the MPC-RBFP scheme is already applying evasive control actions, since the correct inference of the general direction of the obstacle ship has given rise to a possible collision risk in the near future. In contrast, the MPC-SLP controller does not apply any control actions yet, because, based on the straight-


controlled vessel 1 original path
controlled vessel 2 original path
obstacle vessel path
controlled vessel 1 past position
controlled vessel 1 intentions
controlled vessel 2 intentions
obstacle vessel past positions
controlled vessel 2 past position
controlled vessel 1
controlled vessel 2
obstacle vessel
obstacle vessel path prediction
VIRGINIA KEY
25°45'N
25°44'N
80°09'W
80°08'W
80°07'W
80°06'W
80°05'W
80°04'W
Longitude
Latitude

(a1)

(b1)

(a2)

(b2)

(a3)

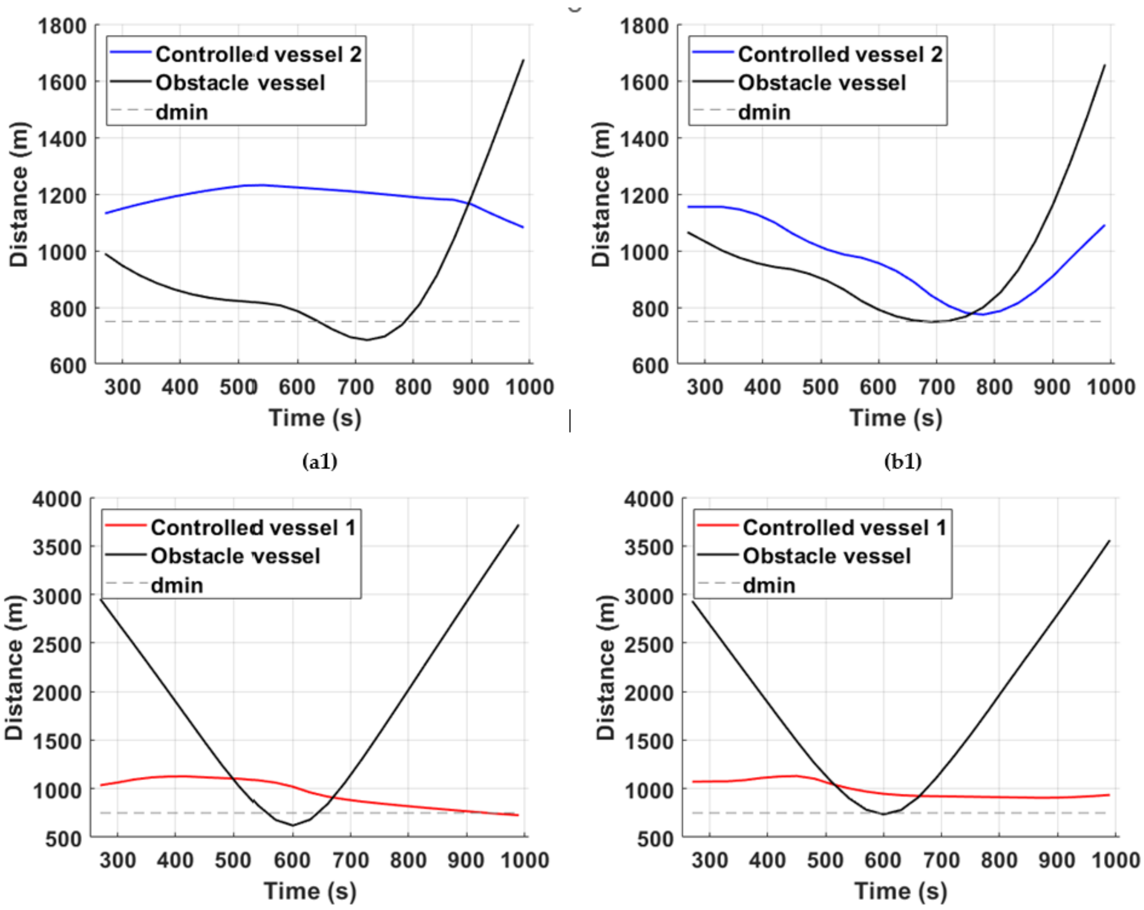
(b3)

(a4)

(b4)

Figure 6.2.6 Scenario 2. The left subfigure column refers to the MPC-SLP scheme, while the right to the MPC-RBFP scheme. Subfigures **(a1, b1)** refer to time instance 6′, **(a2, b2)** to 12′, **(a3, b3)** to 13.5′ and finally, **(a4, b4)** to 17′

line prediction model that it utilizes, the obstacle vessel will continue north and thus remain well clear of the controlled vessels. For the same reason, it takes MPC-SLP another 5’ minutes in order to correctly assess the collision risk and apply decisive control actions, but by then it is too late; by timestep 9’ (see Figures 6.2.7.a3, 6.2.7.b3), controlled vessel 2 reaches its CPA with the obstacle ship, with a DCPA of 680m for controlled vessel 2, well



(a1)

(b1)

(a2)

(b2)

Figure 6.2.7. Distance plots for scenarios 1 & 2. The left subfigure column refers to the MPC-SLP scheme, while the second to the MPC-RBFP scheme. Note that for scenario 1 **(a1, b1)** and for scenario 2 **(a2, b2)**, the MPC-SLP violates the lower limit on distance from CPA, therefore its trajectories are deemed unsafe.

below the acceptable minimum distance d_{min}, as shown in Figure 6.2.7.a1. In contrast, the MPC-RBFP controller generates a smooth, safe, and consistent trajectory, owed to the correct trajectory prediction of the obstacle vessel. Not only does it reach an acceptable DCPA of 751 meters for controlled vessel 2, but it also manages to apply consistent control actions and not significantly deviate from the original course, as shown in Table 6.2.4**.**

Next, the performance of the two controllers is assessed in an overtaking/crossing encounter in scenario 2. Here, the effect of the used trajectory prediction models is once again eminent: At timestep 6 (see Figures 6.2.6.a1, 6.2.6.b1), MPC-RBFP calculates a sharp control move to port-side for controlled vessel 1 in anticipation of the obstacle ship's crossing towards the port of Miami; in contrast, MPC-SLP applies a lower rate of steering for controlled vessel 1, because the straight-line trajectory prediction places its CPA with the obstacle ship at a later time instance. This failure to correctly place the CPAs has adverse effects on vessel 2 trajectory, too, since it is displaced unnecessarily to the left in false anticipation of a collision risk. In addition, the obstacle ship crosses into the domain of

Table 6.2.4. Performance metrics for the generated trajectories of the MPC-RBFP and MPC-SLP schemes for the two simulation scenarios

		Scenario 1		**Scenario 2**	
	Controlled Vessel	**MPC-RBFP**	**MPC-SLP**	**MPC-RBFP**	**MPC-SLP**
Course deviations (1)	1	1.31 10^4	2.21 10^4	0.658 10^4	0.521 10^4
	2	1.49 10^4	2.85 10^4	0.404 10^4	0.529 10^4
Control action smoothness (2)	1	307.35	476.59	242.12	167.85
	2	290.94	424.43	92.47	128.41
Risk of trajectory (3)	1	0	4.032 10^6	0	0
	2	0	0	0	3.949 10^6
Cost of trajectory (4)	1	9.05 10^6	1.62 10^{13}	2.63 10^6	1.49 10^6
	2	2.63 10^6	4.15 10^7	8.58 10^5	1.55 10^{13}

(1) As calculated by equation (14)
(2) As calculated by equation (26)
(3) As calculated by equation (12)
(4) As calculated by equation (24)

controlled vessel 1 (see Figure 6.2.6.a2) once it changes course towards the Miami port at timestep 8'. On the other hand, the MPC-RBFP scheme places controlled vessel 1 in a better position to narrowly evade the breach of its safe domain (see Figure 6.2.8.b2), throughout the simulation. This performance is owed to the trajectory that the RBF model generated for the obstacle vessel, placing its predicted CPA much closer to the real CPA for both controlled vessels. Also, it should be noted that for scenario 2, unnecessary deviations from the original course are avoided for controlled vessel 2, as indicated by the total deviation values in Table 6.2.4. In general, the proposed method achieves a lower overall cost for the generated trajectories, as shown in Table 6.2.4, while obtaining a certain degree of cooperation between the two vessels, where one makes way for the other, in anticipation of their upcoming evasive maneuvers. Moreover, the results show that the proposed method exhibits robust characteristics against environmental effects, which are modelled as input noise in the vessel dynamic model for the scope of the simulations, while accounting for COLREGs. Lastly, the average CPU time evaluation of the MPC calculation was recorded as 7 s for both scenarios, which is well within the allocated simulation controller timestep t_{cst} of 60 s, proving that the proposed method is scalable to more controlled vessels and obstacle vessels.

6.2.6 Conclusions and Future Prospects

In this section, a data-driven tracking MPC controller utilizing RBF obstacle ship trajectory prediction models trained on real AIS data was proposed for the collision avoidance task in busy ports or waterways. The simulations have shown that the incorporation of a trajectory prediction model with a moderate degree of accuracy greatly benefits the performance of a CA controller.

The future prospects of the proposed data-driven MPC CA controller extend further than the specific case study presented; other types of systems such as underwater or aerial vehicles could benefit from the black-box models of obstacles that the RBF networks permit. Since sophisticated RBF obstacle models can detect a collision risk correctly and in time, the planning of more economic trajectories for the controlled vehicles can be executed. This constitutes a significant benefit towards the economic navigation of vehicles, which can be further complemented by the design and development of data-driven economic propulsion controllers, as will be shown in the later chapter.

Chapter 7:

Data-Driven Economic Nonlinear Model Predictive Control

Economic MPC address the need of optimal economic performance of a process, rather than the tracking of a specific setpoint. Contrary to tracking NMPC, where the stage cost is lower bounded by a $\mathcal{K}_\infty$ function and is usually of the form $\ell_t(x,u) = Q^T x Q + R^T u R$, economic NMPC can handle generic stage costs, provided that the formulation of its respective optimal control problem adheres to the EMPC theoretical framework. As discussed in Chapter 4, earlier research works on EMPC obtained nominal asymptotic stability, either by using stage cost rotations by some storage function [79] or by terminal-constraint-based approaches [78]. EMPC has enjoyed multiple applications, with significant economic benefit [207]–[209].

However, the performance of EMPC controllers depends on the existence of an accurate controller model. This motivates the development of data-driven EMPC techniques that could handle model discrepancies as well as unknown disturbances so that superior economic performance is achieved in practical settings. Several theoretical developments of data-driven EMPC based on reinforcement learning (RL) have been proposed in the past three years, pertaining to safety and stability of such schemes [210]–[212]. As one of the most promising intersections of machine learning and control, RL is a sequential decision-making computational intelligence algorithm that adapts a parametric internal representation of the control process through trial-and-error [38]. Due to its black-box nature, RL is well-oriented for usage in conjunction with an MPC formulation.

The purpose of this chapter is to leverage the latest theoretical developments in EMPC and RL-MPC in order to create data-driven EMPC approaches for the economic control of a vessel propulsion system. There is significant motivation for this particular choice of case study: First, it is naturally related to the work presented in section 6.2, regarding the data-driven MPC for vessel trajectory control with collision avoidance, since the combination of both can result in autonomous data-driven controllers for the economic propulsion and navigation of vessels. Second, the creation of economic marine propulsion controllers is of significant consequence for the national Greek economy, which heavily relies on international freight shipping. In short, the impact of this case study has both academic and real-world merits.

The chapter is structured as follows: First, an initial EMPC control law utilizing the latest theoretical developments is constructed for the vessel propulsion case, in order to theoretically establish the concept of using EMPC instead of tracking MPC for this specific task. Next, the EMPC control law previously created is leveraged to create a data-driven propulsion controller based on RL, capable of handling modelling discrepancies and disturbances. In addition, a more detailed vessel propulsion plant model is introduced, together with an economic stage cost that accurately reflects the high-level chartering economics of freight shipping, thus further establishing the practicality and applicability of the data-driven EMPC approach. Lastly, the possibility of extending the created RL-MPC controller in order to include navigation objectives of the controlled vessel is explored.

7.1 A Vessel Propulsion Controller based on Economic Model Predictive Control

The importance of reducing energy expenditure in vessel propulsion is underlined by recent environmental mandates in the maritime sector. Vessel propulsion is a multi-objective problem, since the overall energy expenditure of the powertrain must be minimized, while the vessel speed must be maximized. This section proposes an economic model predictive control approach, which can accommodate powertrain efficiency maps and thus evaluate candidate input trajectories in terms of energy efficiency. The proposed EMPC controller utilizes recent theoretical developments in order to guarantee stability. Simulation results are presented in comparison to a standard MPC scheme, for two different vessel sizes under environmental disturbances, and are evaluated in terms of the overall energy expenditure and the settling time to the desired vessel speed. Lastly, it is demonstrated that the proposed approach achieves a reduction in energy consumption of up to 1.9% in a rough sea scenario.

7.1.1 Introduction

Even though shipping remains the most efficient type of bulk transportation of goods, recent environmental mandates [213], as well as fuel cost increases, have ushered the maritime sector to seek further fuel saving measures, particularly in vessel propulsion. On one hand, fuel constitutes 75% of the total expenses of a vessel in a long-distance voyage [214]; this incentivizes reduced engine loads and thus, cruising speeds [215]. On the other hand, vessels are chartered under tightly-constrained port arrival times, meaning that there exists an economic incentive for maximizing the vessel’s speed [216]. It appears then, that these two conflicting objectives can be advanced concurrently only if the vessel propulsion controller (VPC) operates the powertrain in an energy-efficient manner.

Previous research works have identified the engine speed of an internal combustion powertrain as the parameter of interest regarding fuel consumption [214]. In [217], a dynamic optimization problem is formulated over the whole span of the ship’s voyage in order to generate economically optimal steady-state engine speeds. In [218], a model predictive controller with a standard tracking stage cost that penalizes the distance from a reference steady-state engine operation is presented; the aforementioned stage cost implies the fuel consumption rate, while improvements are shown to be achieved in simulations.

Still, even though such control approaches commonly state to minimize energy expenditure, in practice the actual economic criteria are not included in the objective function, and the desired results are achieved only indirectly. Moreover, it should be noted that energy efficiency of powertrains, whether of electric or internal combustion type, should not be assessed only by the engine speed at steady-state. In marine vessels there are multiple cases where the engine is displaced from its optimal engine speed; propeller ventilation in rough seas can momentarily reduce engine load, thus increasing engine speed for a given power input [219], or adverse wind conditions may require the pursuit of a different engine setpoint in order to retain the same vessel speed. In other words, optimal economic vessel performance must accommodate the energy-efficiency of transient powertrain states, which can only be attained by encoding the powertrains' efficiency map in the objective.

Application of optimal control with such explicit economic criteria commonly requires the problem to be formulated in an economic MPC framework. The consideration of purely economic objectives in a receding horizon fashion has shown significant benefits in various engineering domains [220]. Traditionally, stability of EMPC schemes was enforced by dissipativity-based cost rotations, with or without terminal penalties and Lyapunov-based stability [220]. Recently, a gradient-correcting terminal penalty was shown to be necessary for stability whenever the economic cost has a non-zero gradient at the steady-state [221]. Lastly, this theoretical result has successfully been applied to the energy-optimal coordination of autonomous ground vehicles at intersections [207]. To the author's best knowledge, no application of EMPC for the vessel propulsion problem exists.

The main contribution of this section is the development of a nonlinear EMPC controller for the vessel propulsion task under environmental disturbances. The proposed controller directly incorporates the economic criterion in the objective function, while also taking into account the transient powertrain states. The asymptotic stability of the proposed controller is guaranteed by adding a gradient-correcting end penalty in the cost function, to account for the non-zero gradient at the economically optimal steady state. Two scenarios are simulated; the first is a velocity setpoint change, and the second simulates a high-sea condition that induces propeller thrust loss [222]. Comparisons to a standard MPC controller for two different vessel types, in order to demonstrate the capabilities of the proposed EMPC controller.

The rest of this section is structured as follows: Subsection 7.1.2 presents the nonlinear plant dynamics, as well as the propulsion objectives. In subsection 7.1.3, the proposed EMPC formulation is detailed. Subsection 7.1.4 presents the simulation results for the two scenarios, and finally, conclusions and plans for future research are drawn in subsection 7.1.5.

7.1.2 Vessel Propulsion Control Problem Statement

7.1.2.1 Vessel dynamics

Since this section is concerned with the performance of a vessel propulsion controller, only the surge dynamics of the vessel are considered. These consist of the vessel's hull dynamics, as well as the powertrain's rotational dynamics. The hull dynamics are based on the standard non-dimensional form of SNAME [202], while the powertrain dynamics take into account friction terms as well as the energy efficiency map. The longitudinal hull dynamics are as follows:

$$\dot{v} = \frac{1}{L\,m}\left(X_{uu}v^2 + L^2(1-\tau_n)T_{nn}d_p n^2 + W_{uu}d_w\right) \tag{7.1.1}$$

Here, v is the longitudinal speed of the vessel, n is the angular velocity of the propeller, and d_w weighed by W_{uu} is the longitudinal acceleration induced by the environmental conditions such as waves or wind gusts. The parameter m is the inertia (including the added mass along the longitudinal axis), X_{uu} is the dissipation parameter, L is the vessel's length, τ_n is a scaling parameter, and lastly T_{nn} translates the propeller's angular velocity to longitudinal thrust, and is weighed by the propeller submergence coefficient d_p [219]. The propeller itself is powered by the powertrain, and assuming that there is no gearbox, n also corresponds to the crankshaft's angular velocity. Therefore, an internal combustion engine can be modeled as follows [219]:

$$\dot{n} = \frac{30}{\pi\,I_{ep}}\left(Q_a - 6.28a_f n - 39.43a_p d_p n^2\right) \tag{7.1.2}$$

$$\dot{Q}_a = \frac{1}{t_e}(-Q_a + Q_c) \tag{7.1.3}$$

Here, I_{ep} is the total rotational inertia of the crankshaft and propeller, Q_a is the generated torque by the powertrain, a_f is a rotational friction coefficient, and a_p relates propeller angular velocity to propeller torque. The generated torque Q_a lags behind the commanded torque Q_c by the delay parameter t_e, due to the fuel regulator dynamics. The powertrain is subject to the following operational constraints

$$Q_c^{\min} \le Q_c \le Q_c^{\max} \tag{7.1.4a}$$
$$Q_a^{\min} \le Q_a \le Q_a^{\max} \tag{7.1.4b}$$
$$n^{\min} \le n \le n^{\max} \tag{7.1.4c}$$
$$P^{\min} \le P \le P^{\max} \tag{7.1.4d}$$

where P is the output power of the engine. Eqs. (4a-d) form the admissible set of states and inputs $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$, where $\mathbb{X}$ and $\mathbb{U}$ are the admissible state and input spaces, respectively

Note that this formalism is not restrictive; other engine types can be accommodated, i.e. AC motors [207]. To conclude, the plant model (7.1.1-3) has the following states:

$$x = [v \;\; n \;\; Q_a], x \in \mathbb{X} \tag{7.1.5}$$

and accepts the following inputs and disturbance variables:

$$\begin{aligned} u &= Q_c, \;\; u \in \mathbb{U} \\ d &= \left[d_p \, d_w\right]^T, d \in \mathbb{D} \end{aligned} \tag{7.1.6}$$

7.1.2.2 Propulsion control objectives

The two control objectives of a vessel propulsion controller are the minimization of the total powertrain energy usage and the minimization of total shipping voyage time. The first objective corresponds to the efficient utilization of the powertrain, while the second is equivalent to the maximization of the vessel's surge velocity v. Considering a discrete finite horizon *N* ahead, the objectives to be minimized can be written as follows at discrete timestep *k*:

$$E\big(x(t), u(t)\big) = \sum_{k=0}^{N-1} \left(\int_{kT_s}^{(k+1)T_s} P\big(x(t), u(t)\big) \, dt \right) \tag{7.1.7}$$

$$V\big(x(t), u(t)\big) = -\sum_{k=0}^{N-1} \left(\int_{kT_s}^{(k+1)T_s} v(t) \, dt \right) \tag{7.1.8}$$

Here, $E\big(x(t), u(t)\big)$ and $V\big(x(t), u(t)\big)$ are the average consumed energy and average velocity over the horizon *N* for the given state and input vectors $x(t), u(t)$. Next, $P\big(x(t), u(t)\big)$ corresponds to the power generated by the powertrain

$$P\big(x(t), u(t)\big) = Q_a(t) \; n(t) \; \eta_e\big(x(t), u(t)\big) \tag{7.1.9}$$

where $\eta_e\big(x(t), u(t)\big)$ is the inverse efficiency map of the powertrain (lower values correspond to an energy-efficient engine operation); $\eta_e(\cdot) \in [0,1]$ is a continuously differentiable function, the arguments of which are commonly engine speed n and torque output Q_a. For a combustion engine, this is equivalent to a fuel consumption map. Lastly, note that the form of η_e is not binding and can accommodate any type of engine mapping.

7.1.3 Methodology

It is intuitive to consider the propulsion control of a vessel as a continuous process, the output of which must be pursued economically. For this reason, an economic stage cost must be created and minimized under a suitable control scheme.

7.1.3.1 Creation of the economic stage cost

One can create a multi-objective economic stage cost that reflects the objectives as presented in subsection 7.1.2. Considering the state and input vectors x_k, u_k at the discrete timestep k of the horizon N, this stage cost is comprised by the expended energy cost

$$\ell_f(x_k, u_k) = SFC\, p_f \int_{kT_s}^{(k+1)T_s} \Big(P\big(x(t), u(t)\big)\Big)\, dt \tag{7.1.10}$$

and the velocity:

$$\ell_v(x_k, u_k) = \int_{kT_s}^{(k+1)T_s} v(t)\, dt \tag{7.1.11}$$

Here, *SFC* and p_f are the specific fuel consumption of the engine and the fuel price, respectively. In order to simplify the analysis and to accommodate for other type of powertrains such as electric motors, *SFC* and p_f take on the nominal value of 1. Thus, the objectives (7.1.7), (7.1.8) are written as $E\big(x(t), u(t)\big) = \sum_{k=0}^{N-1} \ell_f(x_k, u_k)$ and $V\big(x(t), u(t)\big) = -\sum_{k=0}^{N-1} \ell_v(x_k, u_k)$, respectively.

Note that the two stage costs (7.1.10), (7.1.11) are conflicting, i.e., the pursuit of a maximized velocity $V(\cdot)$ would lead to increased energy consumption and thus energy cost $E(\cdot)$. Consequently, it is sensible to weigh the two costs using the tradeoff parameter $\beta > 0$ in order to create a suitable stage cost:

$$\ell_\beta^{f,v}(x_k, u_k) = \ell_f(x_k, u_k) - \beta\, \ell_v(x_k, u_k) \tag{7.1.12}$$

Here, $\ell_\beta^{f,v}$ is a weighted sum of the multi-objective economic cost for the vessel propulsion task. It is apparent that β acts as a regulation parameter that weighs the preference between energy expenditure and vessel velocity. By considering the vessel propulsion task as a steady-state process, one can formulate an optimization problem in order to yield the optimal steady-states for a given β. Therefore, $\ell_\beta^{f,v}$ can be considered in the following steady-state optimization problem:

$$\min_{v,Q} \ell_\beta^{f,v}(x_k, u_k) \tag{7.1.13}$$

$$\text{s.t. } v = F_v(x, u),\ (x, u) \in \mathbb{Z}$$

Here, $F_v(x, u)$ is an integrator function that returns the terminal state value of v resulting from the integration of (7.1.1) over one discretization interval, assuming an initial condition x and constant input u. Solving for a variety of β values, the Pareto front of the multi-objective problem can be constructed, as shown in Fig. 7.1.1. The desired tradeoff parameter β_d can be computed a priori given a desired steady-state velocity, by solving the following problem:

$$\min_{x,u} \ell_f(x,u) \tag{7.1.14a}$$
$$\text{s.t. } v = F_v(x,u) \tag{7.1.14b}$$
$$(x,u) \in \mathbb{Z} \tag{7.1.14c}$$
$$\ell_v(x_k, u_k) = v_r \tag{7.1.14d}$$

As outlined in [207], the optimal Lagrange multiplier μ_r of the constraint (7.1.14d) is equal to the desired tradeoff parameter β_d. Note that (7.1.14b) represents the surge velocity dynamics equation. In other words, the desired reference velocity v_r can be assigned by setting $\beta_d = \mu_r$. It is noted that this desired reference velocity can be computed offline during standard voyage estimation/chartering; the economic advantage of short voyage time is assessed, as influenced by current freight rates and penalties for delayed arrival at the destination [216].

7.1.3.2 Creation of EMPC controller with gradient-correcting end penalty

Standard MPC stage costs penalize the distance to a steady-state reference setpoint $(x_s, u_s) \in int(\mathbb{X} \times \mathbb{U})$, where $\mathbb{X}$ and $\mathbb{U}$ are the admissible state and input spaces, respectively. This formulation ensures that the stage cost is bounded from below by a $\mathcal{K}_\infty$ function, which is a necessary prerequisite for asymptotic stability. One observes that the stage cost (12) presented in subsection 7.1.3.1 is generic, meaning that it is not designed to track a target setpoint, but rather economically optimize a process. When such user-provided stage costs are applied, the resulting nonlinear MPC is considered as an economic MPC.

The stability analysis of EMPC schemes differs from the standard MPC ones, and multiple research work has been devoted to this end [220]. The turnpike property of optimal control problems, implied by the reachability and the dissipativity properties of the problem at hand, has recently been exploited to enable stability guarantees [221], [223]. However, EMPC without penalties and constraints cannot stabilize to the optimal steady-state (x_s, u_s) in the cases where the stage cost has a non-zero gradient there [221]. Indeed, the vessel propulsion control problem is such a case; as computed by the steady-state Problem (7.1.13), the gradients $\nabla_Q \ell_\beta^{f,v}$, $\nabla_v \ell_\beta^{f,v}$ at (x_s, u_s) are nonzero for any nonzero β values. In [221], it is shown that in these cases, a linear terminal penalty term that corrects the gradient at (x_s, u_s) can be introduced in order to guarantee stability. This linear terminal penalty is a cost rotation by the optimal steady-state Lagrange multiplier vector $\bar{\lambda}_s$, as calculated by the steady-state Problem (7.1.13)

$$P_\lambda(x_N) = x_N^T \bar{\lambda}_s \tag{7.1.15}$$

where x_N is the state vector at the end of the prediction horizon. In order to apply the theoretical results of [9], the initial assumptions for the problem must be met. These are the linear independence of the state-input constraints (Assumption 1) and the regularity of the steady-state optimal problem (7.1.13) (Assumption 2) which is trivially shown for the problem at hand. Next, one must verify that the Jacobian linearization of the plant model at the optimal steady-state (x_s, u_s) is controllable, so that the standard P_{LQR} terminal penalty can be applied, which is again true for this system. Then, the EMPC formulation with the terminal penalty

$$P(x_N) = P_\lambda(x_N) + P_{LQR}(x_N) \tag{7.1.16}$$

given a finite horizon *N* and sample time T_s will exponentially stabilize the system (1)-(6) at the optimal steady state [9]. Eq. (7.1.16) comprises two terms; the gradient-correcting terminal cost P_λ constitutes a type of storage function that is necessary for the local uniform exponential stabilization of EMPC, but also for guaranteeing the economic performance of the proposed scheme. Its omission would result in the following: Firstly, as outlined in Theorem 3 of [221], the EMPC scheme would not stabilize at the economically optimal steady state. Secondly, limit-cycle phenomena around (x_s, u_s) could arise, resulting in unnecessary torque command modulation and thus, reduced economic performance (see Remark 7 of [221]). It should be noted that P_λ is not substitutive to the cost-to-go term P_{LQR}, which is typically added in MPC schemes to account for the infinite-horizon cost. Omitting this cost-to-go may lead to the optimal trajectory's divergence from (x_s, u_s) towards the end of the MPC prediction horizon. Thus, the terminal penalty (7.1.16) ensures the stabilization of the proposed EMPC scheme, while also accounting for the infinite-horizon cost.

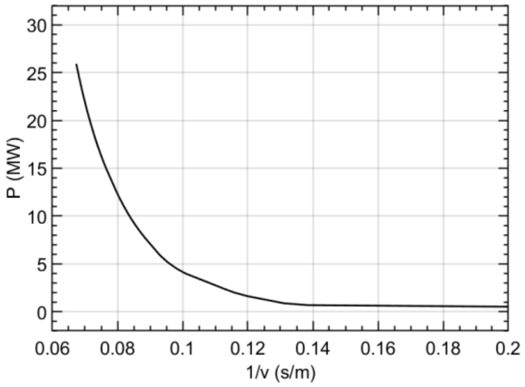


Figure 7.1.1 Pareto front for different values of the tradeoff parameter β

Note that here, it is assumed that $(x_s, u_s) = (0,0)$. However, this assumption is not binding, since for the problem at hand the optimal steady-state is calculated a priori by problem (7.1.13); therefore, the state and input space $\mathbb{X}$ and $\mathbb{U}$ can be shifted by x_s and u_s, respectively.

7.1.3.3 OCP formulation of the EMPC controller

The optimal control problem corresponding to the proposed approach is formulated as follows:

$$\min_{x,u} J_{EMPC}(x,u) \tag{7.1.17a}$$

$$\text{s.t.} \quad x(0) = \hat{x}_0 \tag{7.1.17b}$$

$$(x,u) \in \mathbb{Z} \tag{7.1.17c}$$

$$x_{k+1} = f^{(T_s)}(x_k, u_k) \tag{7.1.17d}$$

$$\beta = \beta_d \tag{7.1.17e}$$

where

$$J_{EMPC}(x,u) = \sum_{k=0}^{N-1} \left(\ell_{\beta}^{f,v}(x_k, u_k)\right) + P(x_N) \tag{7.1.18}$$

is the objective function cost and $f^{(T_s)}$ is the discretized model with zero-order hold over T_s.

7.1.4 Results

In this subsection, the performance of the proposed EMPC scheme for the task of vessel propulsion control is evaluated and compared to a standard MPC (SMPC) controller for two different vessel types. The two controllers are assessed on their overall economic performance in a step and an environmental disturbance scenario.

7.1.4.1 Setup

First, the quadratic stage cost of the SMPC controller is introduced:

$$\ell_q = \frac{1}{2}\left((v - v_s)^2 Q_v + \left(Q_a - Q_{a,s}\right)^2 R\right) \tag{7.1.19}$$

This stage cost is of tracking type; it penalizes the distances from the optimal steady-state state and input x_s and u_s, which are weighted by the Q and R parameters, respectively. In order to perform a fair comparison, these parameters are tuned as follows:

$$Q_v = \left(\frac{1}{v^{max}}\right)^2, R = \left(\frac{1}{Q_a^{\max}}\right)^2 \tag{7.1.20}$$

The full OCP of the SMPC is formulated as follows:

$$\min_{x,u} J_{SMPC}(x, u) \tag{7.1.21a}$$

$$\text{s.t.} \quad x(0) = \hat{x}_0 \tag{7.1.21b}$$

$$(x, u) \in \mathbb{Z} \tag{7.1.21c}$$

$$x_{k+1} = f^{(T_s)}(x_k, u_k) \tag{7.1.21d}$$

where

$$J_{SMPC}(x, u) = \sum_{k=0}^{N-1}\left(\ell_q(x_k, u_k)\right) + P_{LQR}(x_N) \tag{7.1.22}$$

Next, the two simulation scenarios are detailed. For completeness' sake, a very large tanker vessel, commonly known as KVLCC2 in the marine engineering literature, as well as a medium sized vessel, are examined in the first scenario [224]. The objective of scenario 1 is to drive the vessel to an optimal steady-state setpoint, where it is assumed that no disturbances exist, i.e. $d = \left[d_p, d_w\right]^T = [1, 0]^T$. Scenario 2 represents a sea condition with large head waves, inflicting a decelerating force expressed by $W_{uu} d_w$, as well as a thrust loss due to propeller ventilation, expressed by $T_{nn d_p}$. Disturbance variable $d_w \in [0,1]$ represents the modulation of the head wave decelerating force and has a value of 0 when no head waves are present. Disturbance variable $d_p \in [0,1]$ multiplies both the propeller thrust coefficient T_{nn} and the propeller torque constant a_p. When ventilation is present due to large waves, the propeller torque load and thrust periodically drop. When no large waves are present, d_p is 0. The parameters of the two vessels, the vessel bounds and constraints, as well as scenario information are shown in Tables 7.1.1-4, while the disturbance variable

Table 7.1.1 Vessel parameters

#	I_{ep} (kg m^2)	a_f $\left(\frac{MNm}{RPM}\right)$	a_p $\left(\frac{MNm}{RPM^2}\right)$	m (-)	X_{uu} (-)	τ_n (-)	T_{nn} (-)	t_e (-)	L (m)	W_{uu} (m/s^2)
1	1 10^5	560	390	1.050	-0.043	0.22	3.5 10^{-5}	3	300	-3
2	0.1 10^5	168	39	0.500	-0.018	0.22	1.54 10^{-4}	3	100	-1

Table 7.1.2 Vessel bounds & constraints

#	$Q_c^{\min}$ (MNm)	$Q_c^{\max}$ (MNm)	$n^{\min}$ (RPM)	$n^{\max}$ (RPM)	$P^{\min}$ (MW)	$P^{\max}$ (MW)
1	0.5	3	20	100	1	30
2	0.1	0.75	20	120	0.2	9

Table 7.1.3 Scenario 1: Initial conditions and optimal steady-states per vessel

Vessel	Initial conditions			Optimal steady-state		
	v_0 (m/s)	$Q_{a,0}$ (MNm)	n_0 (RPM)	v_s (m/s)	Q_s (MNm)	n_s (RPM)
1	6.0	1.0	30	7.0	1.62	55.2
2	7.0	0.2	58	11.65	0.47	85.97

Table 7.1.4 Scenario 2: Initial conditions and velocity setpoint per vessel

Vessel	Initial conditions			Velocity setpoint
	v_0 (m/s)	$Q_{a,0}$ (MNm)	n_0 (RPM)	v_s (m/s)
1	6.6	1.75	85	6.6
2	7.5	0.45	75	7.5

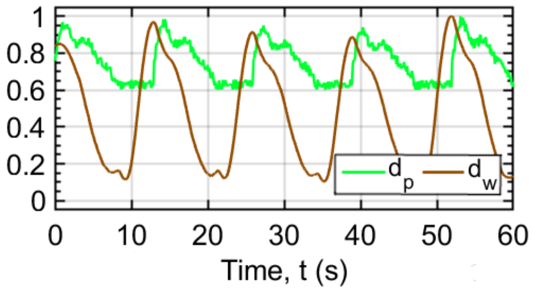


Figure 7.1.2: Scenario 2: Normalized head wave and propeller submergence profiles representing a high sea condition

profiles d for scenario 2 are shown in Fig. 7.1.2; note that these are typical profiles of propeller ventilation and wave force upon a vessel for a moderately high sea, as sourced from the literature [219], [222], [224]. For both MPC controllers the sample time T_s and horizon length N are set as $1s$ and 60, respectively. The plant dynamics $f(x(t), u(t))$ are integrated using the explicit Runge-Kutta (4,5) formula for a simulation timestep of $0.5s$, and the MPC optimization problem is solved using an adaptive interior-point solver.

Finally, the computational framework was built in MATLAB and the simulations were ran on a i9-9960x processor with 64 GB RAM.

7.1.5 Results & discussion

Regarding scenario 1, in Figs. 7.1.3,4, the velocity profiles, as well as the required power outputs of the SMPC and EMPC controllers are shown for both vessels, respectively. As expected, both controllers converge to the optimal steady-state velocity. However, EMPC requires less average power over the simulation period, which corresponds to lower total fuel expenditure. Namely, for vessel 1, the SMPC controller over the initial 750 seconds, requires a higher engine power input than EMPC, while in vessel 2 this is observed over the complete time range. Also, it should be noted that the velocity profiles achieved by the two controllers are almost identical, with regards to settling time and steady-state value. For easier comparison, analytical results for the two vessels are shown in Table 7.1.5.

It appears that for both vessels in scenario 1, the proposed EMPC controller successfully stabilizes to the desired steady-state, while minimizing energy expenditure in transient. In order to reveal the causes of the increased economic performance, one must examine the

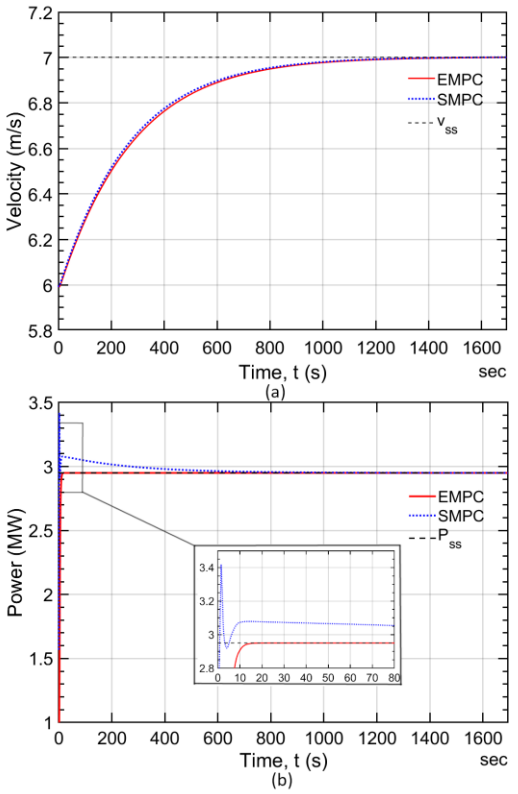


Figure 7.1.3: Scenario 1, Vessel 1: (a) Velocity profile (b) Engine power output

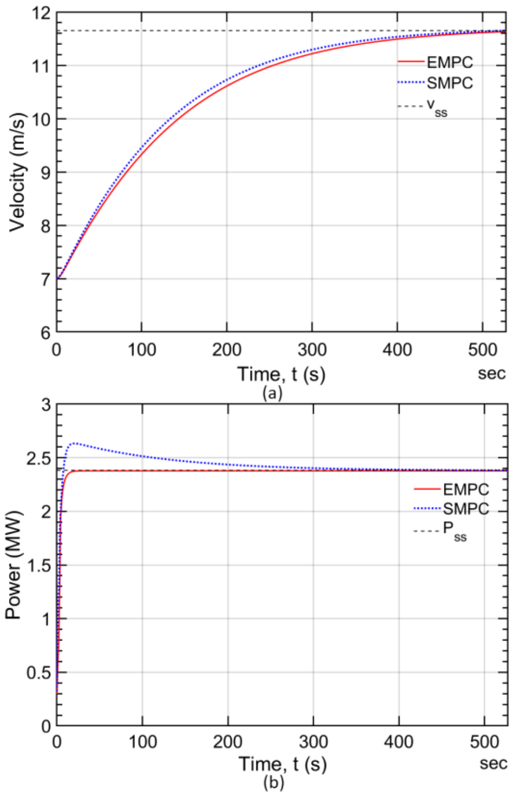


Figure 7.1.4: Scenario 1, Vessel 2: (a) Velocity profile (b) Engine power output

state trajectory in the engine's inverse efficiency map $\eta_e(\cdot)$ for each scenario. In Figs. 7.1.5-6, the respective maps of the vessels are shown for scenario 1 (low $\eta_e(\cdot)$ values correspond to high efficiency). In vessel 1, it is apparent that the EMPC opts to remain in the high-efficiency area denoted by blue, whereas SMPC traces an inefficient trajectory by applying an aggressive initial control input Q. Next, in vessel 2, suitable engine torque values are commanded by the EMPC, so that the engine state (Q, n) remains in the high-efficiency "ridge" of the map for as long as possible. Here too, the SMPC controller departs from the efficient area and traces an economically suboptimal engine map trajectory. The superior transient economic performance of EMPC is owed to the formulation of the economic stage cost (7.1.12), that incorporates the powertrain information $\eta_e(\cdot)$. In contrast, SMPC can only act based upon its perceived distance from the steady-state setpoint, as is evident by its stage cost (7.1.19). This means that it cannot assess the energy efficiency of a candidate input trajectory, and therefore its overall economic performance is reduced.

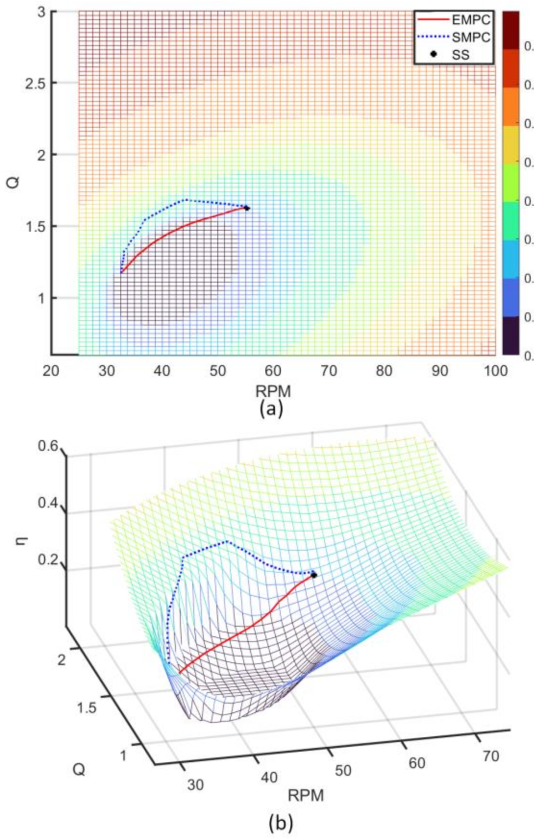


Figure 7.1.5: Engine map trajectories for scenario 1, vessel 1 in (a) 2D and (b) 3D. Redder colours denote higher energy consumption.

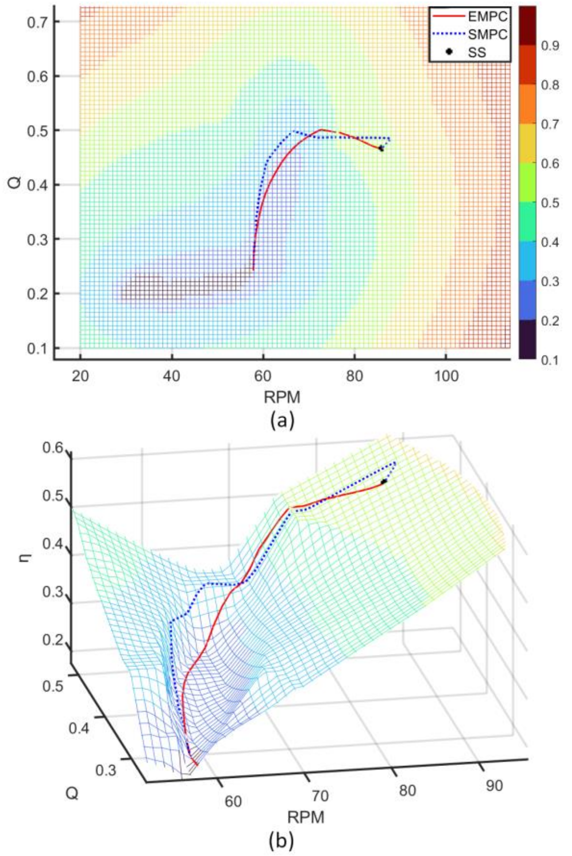


Figure 7.1.6: (a) Engine map trajectories for scenario 2, vessel 1 in (a) 2D and (b) 3D. Redder colours denote higher energy consumption.

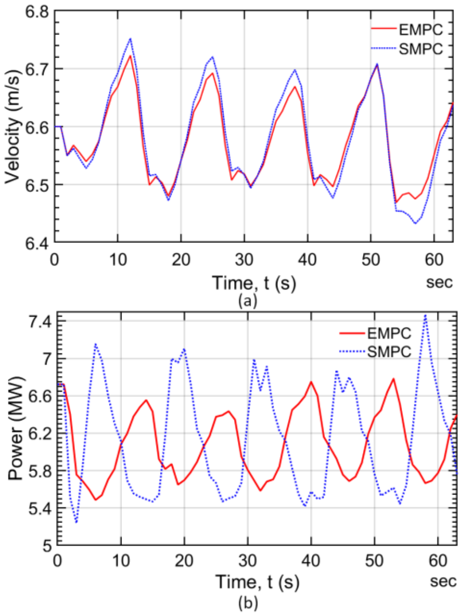


Figure 7.1.7: Scenario 2, Vessel 1: (a) Velocity profile (b) Engine power output

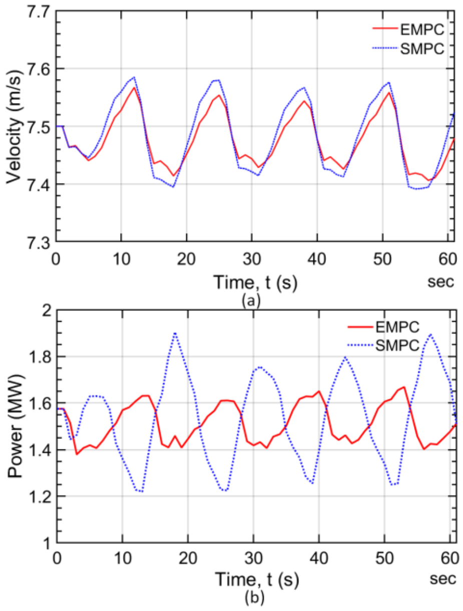


Figure 7.1.8: (a) Scenario 2, Vessel 2: (a) Velocity profile (b) Engine power output

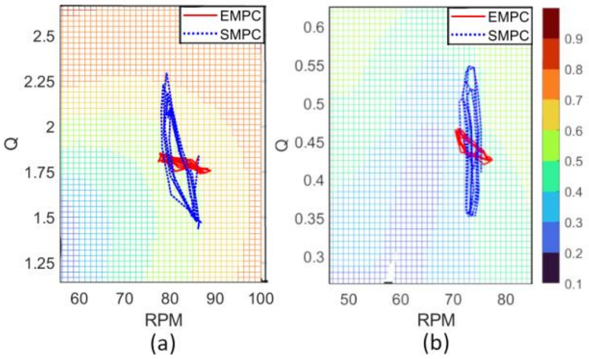


Figure 7.1.9: Scenario 2, engine maps of (a) Vessel 1 and (b) Vessel 2

The merits of the EMPC approach are highlighted in transient engine operation during scenario 2, which is bound to occur due to propeller ventilation [219] or wave forces (Fig. 7.1.2). In Figs. 7.1.7-8, the velocity profiles and the engine power outputs of the SMPC and EMPC controllers are shown for vessels 1, 2 respectively. Indeed, both controllers manage to track a reference velocity as desired, however with different power profiles; from Figs 7.1.7b, 7.1.8b it is apparent that the EMPC controller draws less power in average from the powertrain, for both vessels, by using the engine map efficiently, as demonstrated in Fig. 7.1.9. In Table 7.1.6, the results of scenario 2 are summarily presented for both vessels, and they confirm the merit of the EMPC approach: For a negligible reduction in average sailing speed, energy saving returns of up to 1.95% can be reaped. For vessel 1, this corresponds

Table 7.1.5 Scenario 1 Simulation Results

	Vessel 1			Vessel 2		
	t_s (s)	v_s (m/s)	E (MWh)	t_s (s)	v_s (m/s)	E (MWh)
EMPC	1262	7.00	1.3868	495	11.65	0.3463
SMPC	1250	7.01	1.3991	490	11.65	0.3563
Difference %	0.9	-0.1	0.9	1	0	2.8

t_s denotes the settling time

Table 7.1.6 Scenario 2 Simulation Results

	Vessel 1			Vessel 2		
	Average speed (m/s)	E (MWh)	avg. MPC evaluation time (s)	Average speed (m/s)	E (MWh)	avg. MPC evaluation time (s)
EMPC	6.575	0.1045	0.553	7.478	0.0261	0.551
SMPC	6.576	0.1060	0.432	7.482	0.0266	0.431
Difference %	-0.02	-1.52	+21.88	-0.05	-1.95	+21.66

to almost 1.5 kWh of energy per minute of operation, while for vessel 2 the respective number is 0.5 kWh – therefore, considering the typical cargo ships' voyage duration of multiple months, the economic effects of the EMPC approach are expected to compound significantly. It should be noted, that although the EMPC performance comes at a slightly increased computational cost, owed to the non-quadratic economic cost function, the average computational time needed per iteration is still perfectly acceptable for this application.

7.1.6 Conclusion

In this section, the concept of using an EMPC approach for the vessel propulsion problem has shown its merits in simulation. The proposed EMPC controller enabled the computation of efficient engine trajectories, resulting in better transient economic performance. It is noted that this EMPC approach can be further extended to other types of propulsion systems that prioritize fuel efficiency, such as turbofan engines on airliners.

Still, the performance of the proposed controller rests on the assumption that there are no modelling discrepancies between controller model and real plant; if they exist, then this approach will not yield the optimal economic control law. Building on the theoretical foundation presented in this section, a data-driven scheme that will achieve the optimal economic performance for the problem at hand will be constructed next.

7.2 A Data-Driven Vessel Propulsion Controller based on Reinforcement Learning and Economic Model Predictive Control

In the previous section the proof of concept of an EMPC scheme for the vessel propulsion control was successfully established, for the case of a perfect controller model. However, an effective vessel propulsion controller should not only account for powertrain efficiency information & modelling errors, but also encompass the ship's economic performance objectives. In this section we introduce an economic stage cost that reflects the actual vessel chartering economics instead of an arbitrary tracking setpoint, in order to create a stable economic MPC control law, which is then approximated up to 1st order by a standard MPC. This approximation is then used to build a data-driven economic model predictive control approach based on reinforcement learning (RL-MPC) for the economic control of vessel propulsion. The RL component uses a temporal difference learning scheme to generate stable controller parametrizations, which are iteratively applied in order to achieve improved closed-loop economic performance. Contrary to other data-driven or adaptive MPC approaches, RL-MPC is capable of handling significant structural & parametric plant-model mismatches and disturbances in real time. Simulation comparisons are performed, for off-design and on-design disturbance scenarios, where the merits of the proposed method are showcased. The RL-MPC economic controller presented herein is shown to track the optimal economic policy, thus bridging efficient real-time propulsion control with high-level ship chartering economics.

7.2.1 Introduction

During voyage chartering operations, two objectives must be taken into account in order to generate the service speed setpoint: the first is the minimization of diesel fuel cost, while the second refers to the financial incentive of voyage time minimization [216] due to freight transport commercial costs (cost of opportunity, chartering time clauses, etc) [225], and is achievable by higher cruising speeds. Typically, a shipowner or charterer would formulate an optimization problem involving these two objectives using a model of the real vessel in order to determine the economically optimal service speed for their case [226]. Unavoidably, this model could account neither for environmental disturbances encountered along the voyage, nor for nominal modelling errors; in fact, standard practice in the maritime industry has been to simplistically assume that fuel consumption has a cubic relationship to vessel speed [227].

Even though the vessel propulsion control schemes in the literature claim to practically achieve efficient propulsion operation [228], the associated economic objectives do not appear explicitly in the formulated stage cost, meaning that the desired economic performance is achieved only indirectly. The EMPC controller presented in section 7.1.confirmed the economic performance improvement margin over tracking MPC, under the assumption that the prediction model was perfect, i.e. there was no plant-model mismatch. Unfortunately, this assumption is impossible to guarantee, not only because of nominal modelling inaccuracies during control design, but also due to the constantly changing characteristics of the vessel propulsion system during operation [229], owed to hull & propeller fouling, component deterioration and fuel contamination [202]. Typical adaptive control techniques for MPC such as the prediction error minimization (PEM) could alleviate modeling inaccuracies by fitting the model on data that are sampled over predetermined intervals [25]. This model fitting may succeed whenever the plant-model mismatch is merely parametric, however, if there also exist structural differences between the two, e.g. the plant is of higher order and/or contains additional terms, then such approaches may not deliver the optimal MPC control law [230]. Plant-model structural mismatches is very often the case with vessel propulsion systems, since the real system dynamics can become highly intricate and detailed, as the large body of related vessel modelling literature reveals [202], [222], [224], [231].

Recently, a data-driven MPC was presented, capable of handling structural and parametric plant-model mismatches by employing reinforcement learning (RL) [212], [232]. As one of the most promising intersections of machine learning and control, RL is a sequential decision-making algorithm that adapts a parametric internal representation of the control process through trial-and-error [38]. By assigning rewards to control actions that result in desirable state transitions, a policy (equivalent to a control law) is learned as a state-action mapping. By leveraging this central idea, [212] showed that if an RL algorithm uses a parametrized MPC controller as the internal representation of the control problem coupled with an economic reward function, then the optimal economic policy will be yielded, even if the MPC prediction model is structurally mismatched. The definitive advantage of this RL-MPC fusion over using generalized parametrizations for RL, such as deep neural networks [38], is the existence of safety & stability guarantees for the learned policy, which were established by Gros and Zanon [211] using the MPC theoretical framework.

In this section, an RL-MPC controller is presented for the economic control of the vessel propulsion system. First, a tracking MPC is initialized using a tuned stage cost that is a first-order approximation of the respective EMPC control law [233]. Then, by letting RL adapt the stage and terminal costs, the constraints & the model parameters, the optimal economic vessel propulsion control law in the case of parametric and structural plant-model mismatches is yielded. Comparisons to the initial tracking MPC are performed and the economic capabilities of the proposed RL-MPC are demonstrated in two simulation scenarios. The main contributions of this work are as follows:

- An MPC economic stage cost is presented that encompasses the actual economics of vessel propulsion control as denominated during chartering operations [225], namely the bunker fuel cost & the freight commercial-related profits for the ship owner or charterer. Instead of devising arbitrary control objectives for the vessel propulsion system, this approach bridges the high-level charter planning stage with the real-time powertrain control of the vessel, with significant economic benefit. This economic stage cost can also be readily built by a practitioner with access to the economic characteristics of the voyage, enhancing the overall applicability of the proposed approach.
- A data-driven MPC controller is introduced for the economic vessel propulsion task, capable of handling parametric & structural plant-model mismatches. These mismatches, manifesting as engine efficiency map discrepancies, model-order differences & nominal model parametric errors, can render typical adaptive MPC techniques invalid. Thus, the proposed scheme paves the way for achieving true economic performance in real vessel propulsion systems. In addition, the research results yielded in this work can be easily extended to other propulsion systems.

Next, this section is structured as follows: a more detailed nonlinear vessel plant than the one presented in section 7.1., as well as a practical economic stage cost formulation, are laid out in subsection 7.2.2. Subsection 7.2.3 describes the theoretical foundation of the proposed controller. In subsection 7.2.4, simulation results for two environmental scenarios are presented & discussed, and lastly, conclusions are drawn in subsection 7.2.5.

7.2.2 Vessel Propulsion Control Problem Statement

7.2.2.1 Vessel dynamics

Since the objective is to develop economic vessel propulsion controllers, only the vessel surge dynamics are included, namely, its hull dynamics and its 4-stroke diesel mechanical powertrain propulsion system (fig. 7.2.1). Both the plant and the controller model were based on the vessel model of [234], with some modifications necessary for usage as a controller simulation model. The motivation of not using the model already presented in section 7.1 pertains to its overall simplistic structure, which would not permit the construction of plant and controller model with sufficient structural discrepancy, thus dismissing the merits of the controller presented herein.

The powertrain dynamics are modeled as differential-algebraic equations (DAEs):

$$\dot{Q_e} = \frac{1}{t_Q}(-Q_e + M_e) \tag{7.2.1}$$

$$\dot{r_p} = \frac{1}{t_e}\left(-r_p + r_p^{nom}\frac{u_{ice}}{100}\right) \tag{7.2.2}$$

Here, Q_e is the current engine output brake torque, r_p is an internal engine state variable describing turbocharger spooling and fuel rack time-delay dynamics, u_{ice} is the input variable representing the torque command as a percentage of the maximum torque that is currently available, and lastly t_Q & t_e are time constants. Next, M_e is the target engine output torque, written as:

$$M_e = Q_e^{nom}\left(p_1\left(\frac{n_e}{n_e^{nom}}\right)^2\frac{r_p}{r_p^{nom}} + p_2\left(\frac{n_e}{n_e^{nom}}\right)\frac{r_p}{r_p^{nom}} + p_3\right) \tag{7.2.3}$$

This algebraic equation is comprised by a 2nd degree polynomial function with constants (p_1, p_2, p_3) describing the engine speed – torque envelope of the engine [235], multiplied by the maximum indicated brake torque output Q_e^{nom}. Here, n_e is the current engine speed, and n_e^{nom} is the nominal engine speed. Next, the rotational dynamics of the engine are modelled as follows:

$$\dot{n_e} = \frac{Q_{gb} - Q_p}{2\pi J_t}i_{gb} \tag{7.2.4}$$

Here, Q_{gb} is the output torque of the gearbox, Q_p is the propeller torque, i_{gb} is the gearbox speed reduction ratio, and J_t is the total inertia of the powertrain, gearbox, axle & propeller. Taking into account gearbox losses, Q_{gb} is denoted as

$$Q_{gb} = 10^3\left(Q_e\, i_{gb} - Q_l^{nom}\left(a_{gb} + b_{gb}\left(\frac{n_e}{n_e^{nom}}\right) + c_{gb}\frac{Q_e}{Q_e^{nom}}\right)\right) \tag{7.2.5}$$

where Q_l^{nom} is the nominal gearbox torque loss due to friction, and a_{gb}, b_{gb}, c_{gb} are the torque loss parameters.

Next, assuming a propeller with a constant pitch, Q_p is described as:

$$Q_p = \frac{1}{8\eta_p}C_Q\rho v_h^2\pi D^3 \tag{7.2.6}$$

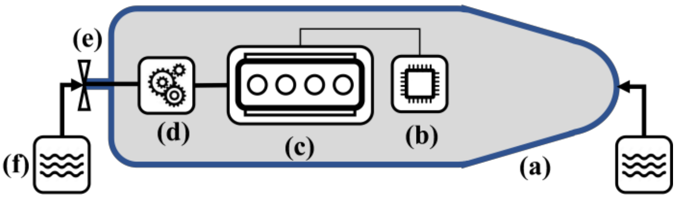


Figure 7.2.1 Vessel dynamics schematic representation. a) Vessel hull, b) Vessel Propulsion Controller, c) Powertrain, d) Gearbox, e) Propeller, f) Sea effects

Here, η_p is the relative rotative efficiency of the propeller, C_Q is the torque coefficient, ρ is the seawater density, D is the propeller diameter, and v_h is the propeller's hydrodynamic velocity, modelled as follows:

$$v_h = \sqrt{v_a^2 + \left(c_h \, \pi \frac{n_e}{i_{gb}} D\right)^2} \tag{7.2.7}$$

In the previous equation, c_h is a dimensionless constant and v_a is the advance speed of water w.r.t the propeller, calculated as

$$v_a = v_s(1 - f_w) + v_w \tag{7.2.8}$$

where f_w is the wake fraction constant and v_w the wave orbital speed. As the final component of plant modelling, the vessel surge speed v_s is described as follows:

$$\dot{v_s} = \frac{1}{m}\left(K_p - R_v \frac{1 + w_f}{1 - f_t}\right) \tag{7.2.9}$$

Here, m is the total mass of the vessel, f_t is the thrust deduction factor, w_f is a dimensionless variable accounting for hull resistance due to waves, while K_p is the propeller thrust and R_v is the hull resistance:

$$K_p = k_p C_T \rho v_h^2 \pi D^2 \tag{7.2.10}$$

$$R_v = c_0 y v_s^2 \tag{7.2.11}$$

In (7.2.10), k_p is a propeller constant and C_T is the propeller thrust coefficient, while in (7.2.11), c_0 is the nominal hull resistance and y a multiplication factor accounting for fouling [236].

Lastly, the wave model describing wave speed v_w w.r.t time is

$$v_w(t) = \zeta\omega \exp\left(\omega^2 \frac{z}{g}\right) sin\big(t(-w_k v_s - \omega)\big) \mathcal{X}(t) \tag{7.2.12}$$

where ζ, ω, z and w_k are the significant wave amplitude, the wave radial frequency, the water depth of the propeller and the propeller wave fraction, respectively [237]. In order to account for random effects, a random variable $\mathcal{X}(1, 0.05)$ is added as a multiplying factor.

To summarize, the plant model comprised by the DAEs (7.2.1-3) has the state variables

$$x = \left[v_s \ n_e \, r_p \, Q_e\right]^T, x \in \mathbb{X} \tag{7.2.13}$$

and accepts the input and disturbance variables

$$u = u_{ice}, \quad u \in \mathbb{U} \tag{7.2.14a}$$

$$d = \left[v_w \, w_f\right]^T, d \in \mathbb{D} \tag{7.2.14b}$$

where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{D}$ are the admissible state, input, and disturbance spaces, respectively.

7.2.2.2 Propulsion economic objectives

The minimization of the total powertrain fuel consumption and the total shipping voyage time are the two economic objectives of a vessel propulsion controller that correspond directly to shipping operational costs [8]. These two objectives are translatable as the efficient powertrain utilization and the surge velocity maximization, respectively, and can be written as follows for a discrete finite horizon *N*:

$$F(x,u) = \sum_{k=0}^{N-1} \left(\int_{kT_s}^{(k+1)T_s} P(x,u) \; SFC(x,u) \; dt \right) \tag{7.2.15a}$$

$$V(x,u) = -\sum_{k=0}^{N-1} \left(\int_{kT_s}^{(k+1)T_s} v_s \; dt \right) \tag{7.2.15b}$$

Here, for state-input vectors $x(t), u(t)$, the sums $F\big(x(t), u(t)\big)$ and $V\big(x(t), u(t)\big)$ represent the expended fuel and the surge velocity, averaged over a horizon of length *N*. The power output of the engine at the current timestep is written as

$$P(x,u) = Q_e \, n_e \tag{7.2.16}$$

which, multiplied with the current specific fuel consumption $SFC(x,u)$, results in the current fuel consumption. Typically, SFC is represented as a 2D look-up table with engine power and speed as indices, however this formulation can be extended to any type of mapping describing different powertrains. In addition, similarly to [235], an emission minimization objective can also be included using a powertrain’s emission map

7.2.3 Methodology

The RL-EMPC vessel propulsion controller that will be presented in this section is initialized on a regular tracking MPC formulation (TMPC), which, in turn, is required to be locally-equivalent to a corresponding EMPC control law. This section presents the proposed approach and its associated theoretical foundation.

7.2.3.1 Creation of the improved economic stage cost

Using the objectives presented in section 7.1, one can create an improved economic stage cost for the vessel propulsion task, that reflects the actual operational economic performance. Considering the discretized system with state-input vectors x_k, u_k, the expended fuel and surge velocity values at timestep k are written as

$$g_f = \int_{kT_s}^{(k+1)T_s} P(x,u)\, SFC(x,u) dt \tag{7.2.17}$$

$$g_v = \int_{kT_s}^{(k+1)T_s} v_s\, dt \tag{7.2.18}$$

respectively, and the corresponding costs as

$$\ell_f(x_k, u_k) = p_f\, g_f \tag{7.2.19a}$$

$$\ell_v(x_k, u_k) = p_v\, g_v \tag{7.2.19b}$$

where is p_f the fuel unit price and p_v the surge velocity “price”, with the latter quantifying the economic incentive of voyage time minimization [238]. Price p_v is nominally composed as $p_v = s\, C/D$, where s is the spot rate that the vessel has been chartered at (in \$/ton), C is the vessel’s cargo (in ton), and D is the voyage distance (in m). Note that p_v is normally negative, i.e., it represents profit. An economic stage cost can then be constructed for the vessel propulsion task,

$$\ell_{p_f,p_v}^{f,v}(x_k, u_k) = p_f\, g_f(x_k, u_k) + p_v\, g_v(x_k, u_k) \tag{7.2.20a}$$

which can be equivalently also written as

$$\ell_{p_f,p_v}^{f,v}(x_k, u_k) = p_f \left(g_f(x_k, u_k) + \frac{p_v}{p_f} g_v(x_k, u_k) \right) \tag{7.2.20b}$$

by factoring p_f out. Then, one can formulate a steady-state optimization problem that calculates the optimal economic steady-states:

$$\begin{gathered} \min_{x,u} \ell_\beta^{f,v}(x_k, u_k) \\ \text{s.t. } v_s = F_{v_s}(x,u),\ (x,u) \in \mathbb{Z}, v_s \geq v_{cl} \end{gathered} \tag{7.2.21}$$

where:

$$\ell_{\beta}^{f,v}(x_k, u_k) = g_f(x_k, u_k) + \beta g_v(x_k, u_k) \qquad (7.2.22)$$

Here, $\beta = p_v/p_f$ stands as a trade-off parameter, F_{v_s} is the integrator of the surge velocity DAE (7.2.9), which returns the terminal value of v_s at the end of the discretization interval, $\mathbb{Z}$ is the admissible state-input space, and finally v_{cl} is the minimum service speed that the charter clause permits. Note that cost functions (7.2.22) and (7.2.20b) are interchangeable in an optimization context. In addition, the corresponding β of an a-priori desired steady-state velocity v_r can also be computed for practical purposes, by solving the following optimization problem:

$$\begin{aligned} &\min_{x,u} g_f(x,u) && (7.2.23a)\\ &\text{s.t. } v_s = F_{v_s}(x,u) && (7.2.23b)\\ &(x,u) \in \mathbb{Z} && (7.2.23c)\\ &g_v(x_k, u_k) = v_r && (7.2.23d) \end{aligned}$$

The solution of Problem (7.2.23) will yield the optimal Lagrange multiplier of constraint (7.2.23d). By setting β equal to this Lagrange multiplier in the original Problem (7.2.21), the desired steady-state velocity $v_s^{(ss)} = v_r$ will be returned. Note that, in the real-world case, β would be properly set as $\frac{p_v}{p_f}$.

7.2.3.2 Economic MPC and its locally-equivalent tracking MPC

Regular TMPC schemes act on a perceived deviation from a reference setpoint, using stage costs that are positive definite (usually quadratic). In contrast, EMPC is designed to handle the generic stage cost $\ell_{\beta}^{f,v}$, which, even though indefinite, it reflects the actual economic objectives of the vessel propulsion system. The OCP corresponding to an EMPC-based vessel propulsion controller can then be described as:

$$\begin{aligned} &\min_{x,u} J_{EMPC}(x,u) && (7.2.24a)\\ \text{s.t.}\quad &x(0) = \hat{x}_0 && (7.2.24b)\\ &(x,u) \in \mathbb{Z} && (7.2.24c)\\ &x_{k+1} = f^{(T_s)}(x_k, u_k) && (7.2.24d)\\ &\beta = \beta_d && (7.2.24e)\\ &v_s \geq v_{cl} && (7.2.24f) \end{aligned}$$

where β_d is the desired tradeoff parameter resulting from Problem (7.2.14), $f^{(T_s)}$ is the prediction model of the propulsion system discretized using a sample time equal to T_s, and J_{EMPC} is the economic objective function

$$J_{EMPC}(x,u) = \sum_{k=0}^{N-1} \left(\ell_{\beta}^{f,v}(x_k,u_k)\right) + P(x_N) \tag{7.2.25}$$

formulated using stage cost (22) and a stabilizing terminal penalty $P(x_N)$. As shown in subsection 7.1.3, a special terminal penalty is required for the stability of this EMPC controller, since the gradients $\nabla_x \ell_{\beta}^{f,v}$, $\nabla_u \ell_{\beta}^{f,v}$ at the optimal economic steady state (x_s, u_s) are non-zero [221]:

$$P(x_N) = P_{LQR}(x_N) + P_{\lambda}(x_N) \tag{7.2.26}$$

Penalty function P is comprised by the infinite horizon penalty P_{LQR} typically included in MPC schemes, plus the linear terminal penalty

$$P_{\lambda}(x_N) = x_N^T\, \lambda_s \tag{7.2.27}$$

where λ_s is the optimal Lagrange multiplier vector at (x_s, u_s). P_{λ} represents a storage function that allows the exponential stabilization of the vessel propulsion EMPC scheme at the optimal steady state.

Notwithstanding its stability and economic optimality for the case where there are no plant-model mismatches, the EMPC scheme (7.2.24) presents two significant practical shortcomings. The first pertains to the tractability of the OCP; algorithms capable of handling the non-convexity of Problem (7.2.1) are often hard to deploy in practice. The second is related to the practical computation of the optimal Lagrange multiplier $\bar{\lambda}_s$; inaccuracies between the actual and computed $\bar{\lambda}_s$ may arise in the case of plant-model mismatches, which would compromise the stabilization of the system at (x_s, u_s) [21]. In order to circumvent these obstacles, [233], [239] proposed an algorithm for tuning a regular TMPC with a quadratic stage cost so as to become a first-order approximation of the original EMPC control law, provided that the latter is stabilizing. Given the steady-state optimization problem (7.2.21), its Lagrange function $\mathcal{L}(w)$ can be defined, where $w = (x, u, \lambda)$ is the primal-dual variable vector. By performing a convexification procedure represented by the following semi-definite program (SDP), one can yield a positive-definite Hessian matrix $\widetilde{H}$ of a new quadratic stage cost $\ell_{Q,\beta}$:

$$\min_{\Pi,\Gamma,\delta,\zeta} \gamma\delta - \zeta + \rho\|\Gamma\| \tag{7.2.28a}$$

$$\text{s.t.} \quad \zeta I \preccurlyeq H + \mathcal{H}(\Pi) + G_{s,A}^T \Gamma G_{s,A} \preccurlyeq \delta I \tag{7.2.28b}$$

$$\mathcal{H}(\Pi) = \begin{bmatrix} A_s^T \Pi A_s - \Pi & A_s^T \Pi B_s \\ B_s^T \Pi A_s & B_s^T \Pi B_s \end{bmatrix} \tag{7.2.28c}$$

$$\widetilde{H} \equiv H + \mathcal{H}(\Pi) + G_{s,A}^T \Gamma G_{s,A} \tag{7.2.29}$$

Here, γ and ρ are problem tuning variables, A_s, B_s represent the x, u gradient vectors of system (7.2.13,14) respectively, $G_{s,A}$ is the x, u gradient of the active inequality constraints, and lastly, H is the Hessian of the Lagrangian, all computed at the primal dual solution w_s of Problem (7.2.21).

7.2.3.3 Reinforcement-Learning-EMPC

The idea behind RL-MPC is to encapsulate a parametrized MPC controller within RL so as to approximate the optimal action-value & value functions, as well as the optimal policy, for the economic process at hand. To this end, the vessel propulsion process is denoted as a discrete Markov decision process (MDP) with the following stochastic dynamics,

$$\mathcal{P}[s_+|s,a] \tag{7.2.30}$$

where s and a are the state and action vectors, and s_+ is the state transition of the discretized system represented by (7.2.13,14) (in control notation, x_k, u_k and x_{k+1}, respectively). For process $\mathcal{P}$, an economic reward function is created:

$$L(s,a) = \ell_\beta^{f,v}(s,a) + I_\infty(h(s,a)) + I_\infty(g(a)) \tag{7.2.31}$$

The reward function is comprised by the economic stage cost (7.2.22), augmented with penalty functions I_∞ that are activated when the process constraints are violated, as denoted by $\mathbb{X}_{\mathcal{P}} = int(\mathbb{X} \times \{v_s \geq v_{cl}\})$. The associated value and action-value functions $V_\star$ and $Q_\star$ are then written as:

$$Q_\star(s,a) = L(s,a) + \gamma\mathbb{E}[V_\star|s,a] \tag{7.2.32}$$

$$V_\star(s) = min_a\, Q_\star(s,a) = Q_\star\big(s, \pi_\star(s)\big) \tag{7.2.33}$$

These are the Bellman equations for the policy at hand. Here, γ is a discount factor, $\mathbb{E}[\cdot]$ is the future expectation and $\pi_\star$ is a state-action mapping function called the policy. Eq. (7.2.32) states that the application of action a while at state s will yield the immediate reward $L(s,a)$ plus the discounted expectation of future rewards $\gamma\mathbb{E}[V_\star|s,a]$ that are retrieved by following $\pi_\star$ from the next state and onwards.

Next, the MDP of a mismatched model of the real plant dynamics is

$$\mathcal{P}[\hat{s}_+|s,a] \tag{7.2.34}$$

and has an associated stage cost defined as

$$\hat{L}(s,a) = Q_\star(s,a) - \gamma\mathbb{E}[V_\star(\hat{s}_+)|s,a] \tag{7.2.35}$$

with $|\mathbb{E}[V_\star(\hat{s}_+)|s,a]| < \infty$. Then, Theorem 1 of [212] has shown that an optimal value function of the form

$$\hat{V}_N(s) = \min_{\pi} \mathbb{E}\left[\gamma^N V_\star(\hat{s}_N^\pi) + \sum_{\kappa=0}^{N-1} \gamma^k \hat{L}\big(\hat{s}_k^\pi, \pi(\hat{s}_k^\pi)\big)\right] \tag{7.2.36}$$

can, in theory, yield the optimal policy $\pi_\star$ of the real plant dynamics s_+, even by using the wrong model $\hat{s}_+$. Here, N is an optimization horizon and $\hat{s}_k^\pi$ are discrete states generated over N by applying π. This central idea enables the use of fully parametrized EMPC schemes as function approximators within RL. Consider an EMPC controller with its individual constituents parametrized by a θ vector - namely a prediction model $f_\theta^{(T_s)}$, a stage cost l_θ, a terminal penalty P_θ & a state-input constraint map h_θ - characterized by the following OCP:

$$\min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma) \tag{7.2.37a}$$
$$\text{s.t.} \quad x_0 = x_0{}' \tag{7.2.37b}$$
$$x_{k+1} = f_\theta^{(T_s)}(x_k, u_k) \tag{7.2.37c}$$
$$g(u_k) \le 0 \tag{7.2.37d}$$
$$h_\theta(x_k, u_k) \le \sigma_k \tag{7.2.37e}$$

Here, σ is the slack variable vector of the ℓ_1 relaxation of the mixed state-input constraints h_θ denoting $\mathbb{X}_{\mathcal{P}}$, g are the constraints denoting $\mathbb{U}$, and the objective function $J_{MPC,\theta}$ is

$$J_{MPC,\theta}(x,u,\sigma) = \lambda_\theta(x_0) + \gamma^N P_\theta(x_N) + \sum_{\kappa=0}^{N-1} \gamma^k \big(l_\theta(x_k,u_k) + w^T\sigma_k\big) \tag{7.2.38}$$

where w is a slack weight vector and λ_θ an initial cost function, the role of which will be discussed later. Using Problem (7.2.37), the following value and action-value functions can be created,

$$V_\theta(s) = \min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma), \qquad s.t\ (7.2.37\text{b}) - (7.2.37\text{e}) \tag{7.2.39a}$$

$$\begin{gathered} Q_\theta(s,a) = \min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma) \\ s.t\ (7.2.37\text{b}) - (7.2.37\text{e}) \\ u_0 = a \end{gathered} \tag{7.2.39b}$$

which correspond to the policy $\pi_\theta = u_0^\star$ (as retrieved from the optimal input sequence yielded by Problem (7.2.37)). It is trivially shown that this formulation satisfies the Bellman equations:

$$\begin{gathered} \pi_\theta(s) = arg \min_{\alpha} Q_\theta(s,a) \\ V_\theta(s) = \min_{\alpha} Q_\theta(s,a) \end{gathered} \tag{7.2.40}$$

As discussed in Corollary 2 of [212], this scheme can, in theory, deliver the optimal policy $\pi_\star$; however, in most practical cases, this is not likely to happen. This is partly due to the intricacy of creating an MPC parametrization elaborate enough to capture the structure of any $Q_\star$ and $V_\star$, and partly because, even if such a parametrization could be created, the computation of the related $\theta^\star$ would be extremely demanding. In short, only an approximation of the optimal vessel propulsion policy $\pi_\star$ can be expected in practice.

Still, by letting θ be adapted by classic RL tools, this approximation will be attained, even if the vessel propulsion model $f_\theta^{(T_s)}$ used in Problem (7.2.37) is structurally or parametrically mismatched to the real plant $f^{(T_s)}$, denoted by (7.2.13, 14). However, since the corresponding stage cost $\ell_\beta^{f,v}(s,a)$ of $\pi_\star$ is generic, the adaptation of θ may yield a non-positive-definite value function, therefore compromising the standard, Lyapunov-based stability guarantees of the MPC scheme. The inclusion of the initial cost function λ_θ can alleviate this problem [212]: Function λ_θ acts as a cost rotation per classic economic MPC stability theory [79], which enables $J_{MPC,\theta}(x,u,\sigma)$ to remain positive-definite without disturbing the optimal solutions of Q and V (7.2.39). Provided that the stage cost l_θ also remains positive-definite throughout θ-adaptation, the scheme (7.2.37-40) as discussed here can successfully approximate the optimal vessel propulsion economic policy $\pi_\star$.

7.2.3.4 Learning for RL-MPC

Given the value and action-value functions introduced in subsection 7.2.3.1, classic RL methods that tune the θ-vector can be employed in order to yield increased closed-loop propulsion performance, such as Q-learning. Using the temporal-difference (TD) learning method [240], θ-updates can be computed as:

$$\theta \leftarrow \theta + a\tau_k \nabla_\theta Q_\theta(s_k, a_k) \tag{7.2.41}$$

Here, θ-updates are computed instantaneously at timestep k as the gradient-descent step of the action-value function Q_θ, using a step size $a \in (0,1]$. The TD error τ_k is computed as:

$$\tau_k = L_f(s_k, a_k) + \gamma V_\theta(s_{k+1}) - Q_\theta(s_k, a_k) \tag{7.2.42}$$

$$L_f(s_k, a_k) = \ell_\beta^{f,v}(x_k, a_k) + w^T \max\big(0, h(x_k, a_k)\big) \tag{7.2.43}$$

where action $a_k = \pi_\theta(s_k)$, and L_f represents the economic reward function corresponding to the real process. This initial version of TD-learning, notwithstanding its simplicity, has some drawbacks. Firstly, instantaneous parameter updates may become counter-productive in the presence of noise and disturbances, which would be prevalent in the system of interest. Secondly, as a gradient-based method, it cannot ensure convergence or globalization if the θ-parametrization is nonlinear. Lastly, it does not guarantee that the Hessians of the quadratic cost functions l_θ and P_θ remain positive-definite as the algorithm progresses, therefore undermining the well-posedness and stability of the MPC scheme. The formulation of an SDP optimization problem that minimizes the TD error over an interval spanned by the last N_{upd} number of steps, subject to positive-definitive constraints, alleviates these drawbacks [211]:

$$\min_{\theta} \sum_{j=0}^{N_{upd}} \tau_{k-j}^2 \tag{7.2.44a}$$

$$\text{s.t.} \quad H_{l_\theta} \succ 0, H_{P_\theta} \succ 0, \tag{7.2.44b}$$

Problem (7.2.44) is solved every N_{upd} timesteps and converges to a $\theta^\star$ that is used to construct a step $a\theta^\star$, which is then applied at a convenient time as $\theta \leftarrow \theta + a\theta^\star$. Note that for $aH_{l_\theta}^\star$ & $aH_{P_\theta}^\star$, which are the positive-definite step matrices generated by (7.2.44)**,** it is trivially shown that their respective additions to H_{l_θ} & H_{P_θ} result in positive-definite matrices. Therefore, learning as presented here is guaranteed to generate stable MPC control laws. In general, it should also be noted that RL parameter update schemes such as (7.2.44) are not expected to yield the global θ optimum, especially in the case of nonlinear value function parametrizations; This drawback is universal in cases of such elaborate parametrizations, even when RL is commonly paired with neural-network-based function approximators.

7.2.3.5 RL-MPC agent for vessel propulsion control

Using the results of the previous subsections, the RL-MPC agent for vessel propulsion control can now be constructed. In order to adhere to the central motivation behind RL-MPC and to ensure the tractability of the OCP in the case of real-life deployment, a lower-order vessel model f_θ is used as the controller prediction model. This model is parametrically and structurally mismatched with regards to the vessel plant (7.2.13,14) presented in subsection 7.2.2**;** the structural differences pertain to the absence of the internal engine state variable r_p, the simplified modelling of engine target torque M_e, and the omission of some terms from gearbox friction torque Q_{gb}:

$$M_e = M_e^{nom}\left(p_1\left(\frac{n_e}{n_e^{nom}}\right)^2\frac{u_{ice}}{100} + p_2\left(\frac{n_e}{n_e^{nom}}\right)\frac{u_{ice}}{100} + p_3\right) \tag{7.2.45}$$

$$Q_{gb} = 10^3\left(Q_e\, i_{gb} - Q_l^{nom} b_{gb}\left(\frac{n_e}{n_e^{nom}}\right)\theta_{Q_l}\right) \tag{7.2.46}$$

Model f_θ is thus represented by the following DAE's,

$$\dot{Q}_e = \frac{1}{t_Q}(-Q_e + M_e) + \theta_{Q_e,c} \tag{7.2.47}$$

$$\dot{n}_e = \frac{Q_{gb} - Q_p\theta_{C_Q}}{2\pi J_t} i_{gb} + \theta_{n_e,c} \tag{7.2.48}$$

$$\dot{v}_s = \frac{1}{m}\left(K_p\theta_{C_T} - \frac{R_v\theta_{R_v}}{1-f_t}\right) + \theta_{v_s,c} \tag{7.2.49}$$

The model has the following state and input variables,

$$x = [v_s\ n_e\ Q_e]^T\ \ x \in \mathbb{X},\ \ u = u_{ice}\ \ u \in \mathbb{U} \tag{7.2.50}$$

parametrized by $\theta^{f_\theta} = \left[\theta_{Q_l}\theta_{Q_l}\ \theta_{R_v}\ \theta_{C_Q}\ \theta_{C_T}\ \theta_{Q_e,c}\ \theta_{n_e,c}\ \theta_{v_s,c}\right]^T$, with default values $\theta_{def}^{f_\theta} = [1\ 1\ 1\ 1\ 0\ 0\ 0]^T$.

Using the discretized model $f_\theta^{(Ts)}$, Problems (7.2.21), (7.2.22) are formulated that yield an economic steady state (x_s, u_s) for a given trade-off parameter β_d. Note that (x_s, u_s) does not necessarily correspond to the optimal economic steady state $(x_s^\star, u_s^\star)$ that would result from the real plant $f^{(T_s)}$. Next, using $f_\theta^{(Ts)}$, a stabilizing EMPC control law is constructed for the vessel propulsion problem**,** which is then employed in the convexification procedure described in subsection 7.2.3.1**,** in order to yield the first-order quadratic approximation of the control law. One can model the newly-created tracking stage cost, the classic LQR terminal penalty & the initial cost as fully parametrized quadratic functions of the type:

$$F(\Delta w) = \Delta w^T H \Delta w + h\Delta w + c,\ \ \Delta w = \begin{bmatrix} x \\ u \end{bmatrix} - \begin{bmatrix} x_s \\ u_s \end{bmatrix} \tag{7.2.51}$$

Now, the RL-MPC value functions can be initialized with the default parameter vector:

$$\theta_{def} = \left(\theta_{def}^{f_\theta}\ H_l\ h_l\ c_l\ H_P\ h_P\ c_P\ H_\lambda\ h_\lambda\ c_\lambda\right) \tag{7.2.52}$$

where H_*, h_*, c_* are the Hessian matrix, gradient and constant of the respective quadratic functions.

The RL-MPC agent for vessel propulsion control can now be deployed to the real system. Actions are applied based on the policy π_θ resulting from the parametrized action-value and value functions $Q_\theta(s,a)$ and $V_\theta(s)$, and TD-learning is employed in a batch update fashion using Problem (7.2.44) in order to approximate the optimal economic policy $\pi_\star$, corresponding to the vessel plant f (7.2.13,14). The algorithmic framework is summarized in fig. 7.2.2**.**

7.2.4 Results

In this section, the proposed RL-EMPC vessel propulsion controller is evaluated through comparisons to a tracking MPC controller, for two types of control scenarios. Simulation results are then presented and discussed.

7.2.4.1 Case study & Setup

First, the full OCP of the tuned-tracking MPC (TTMPC) used for comparison is presented,

$$\min_{x,u,\sigma} J_{TTMPC,\theta_{def}}(x,u,\sigma) \tag{7.2.53a}$$

$$\text{s.t.} \quad x_0 = x_0{}' \tag{7.2.53b}$$

$$x_{k+1} = f_{\theta_{def}}^{(T_s)}(x_k,u_k) \tag{7.2.53c}$$

$$g(u_k) \leq 0 \tag{7.2.53d}$$

$$h_{\theta_{def}}(x_k,u_k) \leq \sigma_k \tag{7.2.53e}$$

with objective function:

$$J_{TTMPC,\theta_{def}} = \sum_{\kappa=0}^{N-1} \left(l_{\theta_{def}}(x_k,u_k) + w^T \sigma_k \right) + P_\theta(x_N) \tag{7.2.53f}$$

The choice of TTMPC as a comparison controller is highly intuitive for this context: First of all, since it is a 1st-order tracking MPC equivalent of the EMPC vessel propulsion controller originally presented in section 7.1 it can be considered as state-of-the-art for this application. Secondly, the original EMPC control law may lose the stability guarantees granted by its gradient-correcting terminal penalty (this is due to plant-model mismatch, as discussed in subsection 7.2.3.2). Third, OCPs (53) & (37) are almost equivalent for $\theta = \theta_{def}$, meaning that RLMPC and TTMPC are also similar at initialization, therefore ensuring the fair evaluation of the proposed TD-learning scheme.

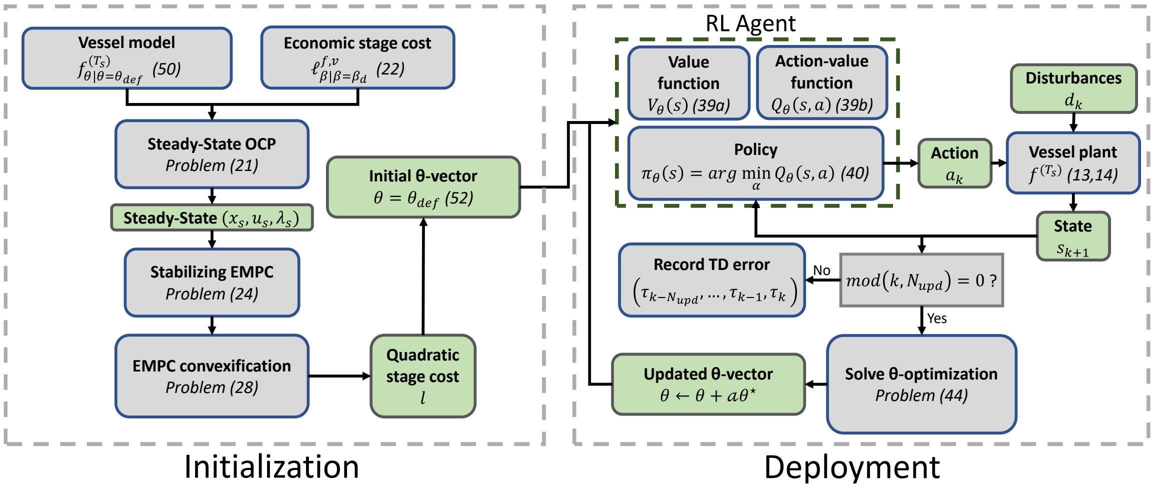


Figure 7.2.2 Schematic representation of the proposed vessel propulsion control approach.

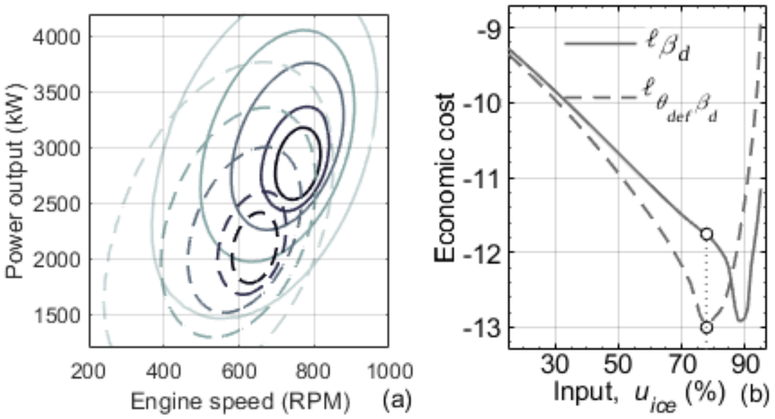


Figure 7.2.3 a) Contour plot of the engine's normalized SFC map. Solid lines denote the actual plant map, while dashed lines the model's map. b) The corresponding steady-state economic profile for a given $\beta_d = 24.73$ and various u. Note that $(x_s, u_s) \neq (x_s^\star, u_s^\star)$, i.e. the model's optimum differs from the plant's one.

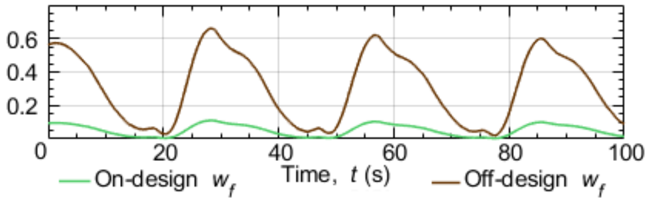


Figure 7.2.4 Wave hull disturbance w_f modulation for the on and the off-design case.

Next, the real plant f and the controller model f_θ parameters that is used by both controllers are shown in Table 7.2.1. Here, the evident parametric discrepancy, coupled with the plant-model structural mismatch detailed in subsection 7.2.3.5, are expected to challenge the RLMPC controller in two respects: First, the computed economic steady state may not be optimal for the real plant, i.e., $(x_s, u_s) \neq (x_s^\star, u_s^\star)$ and second, the applied policy may differ significantly from the optimal one. This challenge is especially salient whenever the model's economic cost formulation does not correspond to the plant's – for the case study at hand, this manifests through an inaccurate specific fuel consumption map $SFC(x, u)$ [234]. This inaccuracy is evident in fig. 7.2.3, where the plant & model engine SFC contours, as well as the corresponding steady-state cost profiles are shown. Lastly, the constraints that denote the admissible state-input space $\mathbb{Z}$ for both controllers are shown in Table 7.2.2.

In order to showcase the merit of the RLMPC method, two environmental scenarios are used, pertaining to normal (on-design) and extreme weather conditions (off-design) situations. Information regarding initialization and wave characteristics are shown in Table 7.2.3, while the modulation of the w_f parameter that describes retrograde force due to waves acting on the hull is shown in fig. 7.2.4, for both scenarios. In addition, the tuning constants of the two controllers are shown in Table 7.2.4. Here, horizon N and sample time T_s are chosen with regards to the dynamics of the vessel model in order to ensure feasibility and tractability of the OCP problems, while slack weight w is determined in relation to the magnitude of the stage cost. The RL parameters N_{upd}, a and γ are tuned by trial-and-error in order to achieve a reasonable learning speed while mitigating the effect of disturbance noise in the TD-learning Problem (7.2.44).

Table 7.2.1 *Plant & model parameters*

Symbol	Description	Plant	Model	Unit
t_Q	Torque time constant	5	10	s
J_t	Total rotational inertia	5000	5000	kg m^2
i_{gb}	Gearbox speed reduction ratio	4.355	4.355	-
t_e	Turbo spooling time constant	20	-	s
r_p^{nom}	r_p nominal value	100	-	-
n_e^{nom}	n_e nominal value	950	950	RPM
m	Total mass	38 10^5	37.5 10^5	10^3 kg
f_t	Thrust deduction factor	0.155	0.155	-
Q_e^{nom}	Q_e nominal value	53	52	kNm
p_1	M_e 2nd deg polynom. param. 1	0.9297	0.9297	-
p_2	M_e 2nd deg polynom. param. 2	0.0333	0.0333	-
p_3	M_e 2nd deg polynom. param. 3	0.15	0.17	-
Q_l^{nom}	Q_l nominal value	1.04	1.04	kNm
a_{gb}	Q_l param. 1	0.1	-	-
b_{gb}	Q_l param. 2	0.65	0.75	-
c_{gb}	Q_l param. 3	0.25	-	-
f_w	Wake fraction constant	0.08	0.08	-
D	Propeller diameter	3	3	m
C_Q	Propeller torque coefficient	0.036	0.035	kNm
ρ	Seawater density	1024	1024	kg/m^3
k_p	Propeller constant	2	2	-
C_T	Propeller thrust constant	0.16	0.155	kN
c_0	Nominal hull resistance	6350	6450	kN
y	Hull fouling factor	1	1	-
z	Propeller water depth	6.5	6.5	m
g	Gravitational acceleration	9.81	9.81	m/s^2
c_c	RPM to rad/s conversion	0.1042	0.1042	-

Table 7.2.2 *Bounds & Constraints*

	$\boldsymbol{u_{ice}}$ (%)	$\boldsymbol{n_e}$ (rad/s)	$\boldsymbol{Q_e}$ (kNm)	$\boldsymbol{r_p}$ (%)
Lower	20	15	5	10
Upper	95	95	$Q_e^{max}(n_e)$	100
$Q_e^{max}(n_e) = Q_e^{nom}\left(p_1\left(\frac{n_e}{n_e^{nom}}\right)^2 + p_2\left(\frac{n_e}{n_e^{nom}}\right) + p_3\right)$				$[p_1, p_2, p_3]$= [0.90, 0.03, 0.22]

Table 7.2.3 *Scenario Setup: Initial Conditions & Disturbance parameters*

	$\boldsymbol{x_0}$	$\boldsymbol{u_0}$	$\boldsymbol{p_f}$	$\boldsymbol{p_v}$	ζ	ω	w_k
On-design	[8, 80, 65, 35]	85	10^{-5}	-2.312	10	0.3	0.020
Off-design	[7, 70, 40, 30]	80	10^{-5}	-2.45	18	0.3	0.025

Lastly, the simulation framework was built in Python, the OCPs (7.2.37) & (7.2.53) were

Table 7.2.4 *MPC & RL Parameters*

Symbol	Description	Value
N	MPC prediction horizon	100
T_s	MPC sampling time	1 s
w	Constraint slack weight	1000
N_{upd}	RL sample batch length	40
a	RL step constant	0.35
γ	RL discount factor	0.99

constructed with CasADi tools [241] and solved with qpOASES [242], while the simulations took place on a i9-9960x processor with 64 GB RAM.

7.2.4.2 Results & Discussion

Regarding the on-design scenario, the state and input profiles are shown for both controllers in fig. 7.2.5. The TTMPC controller exhibits a tracking offset from the computed economic setpoint computed by Problem (7.2.21) due to environmental disturbances v_w, w_f, which decrease propulsion efficiency and increase ship resistance, respectively. The RL-MPC however, slowly adjusts its parameters and moves toward a different, more economical steady-state setpoint, as evident from fig. 7.2.5f. This result can be interpreted by examining fig. 7.2.3a**,** which shows the contour of the engine maps used by the model and the plant, respectively. The nominal operation points of the two maps, which correspond to the lowest SFC value, are different. The effect of this modelling discrepancy is reflected at the steady-state economic cost map, plotted throughout the input range for indicative β_d values and shown in fig. 7.2.3b. It appears that Problem (21), formulated with $l_{\theta=\theta_{def}}$ and the engine map shown in fig. 7.2.3a with dashed lines, miscalculates the desired economic setpoint; consequently, RL-MPC adapts its θ parameters towards the optimal setpoint. The engine map trajectories for this scenario are shown in fig.7.2.6, where it can be seen that RLMPC slowly moves towards the true nominal engine operation point, while modulating the power in circles due to the disturbances affecting on the propeller. Also, the evolution of some indicative θ-parameters is shown in fig. 7.2.7, where it is shown that the θ-learning process tends to converge to a parameter set that corresponds to the new setpoint, which is testimony to the effectiveness of the TD-learning optimization Problem (7.2.44).

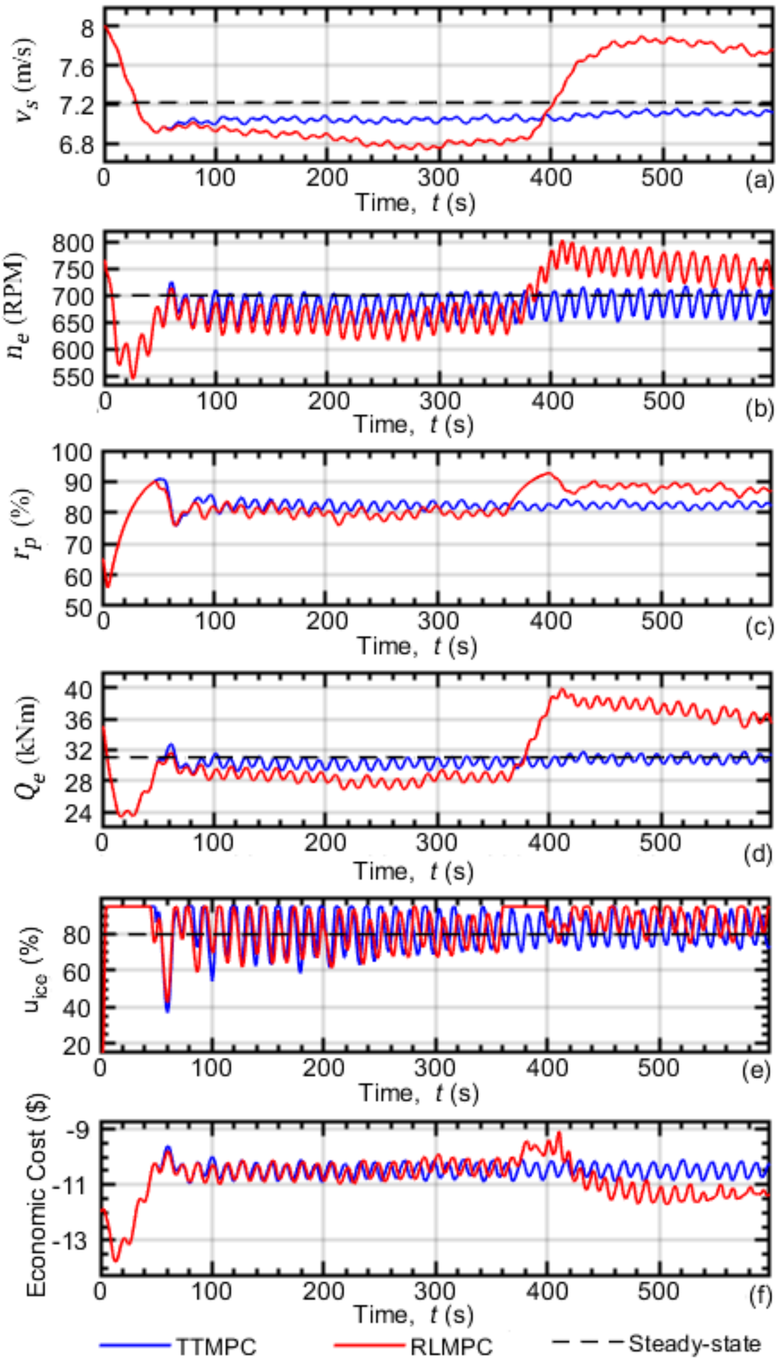


Figure 7.2.5 On-design scenario, state & input results: a) Vessel speed, b) Engine speed, c) Spool percent, d) Engine torque output, e) Engine controller command, f) Vessel propulsion economic cost.

The economic efficiency of a vessel propulsion controller is also assessed in an off-design scenario setting that represents severe weather conditions. Figs. 7.2.8a-e depict the state and feedback profiles, while in fig. 7.2.8f, the plant's economic stage cost L_f is shown. In this scenario, it appears that both controllers fail to track the pre-computed setpoint vessel speed due to the severe longitudinal wave forces acting on the hull, even though the propulsion system is operated at near-maximum capacity. Here, the results of RL-MPC are

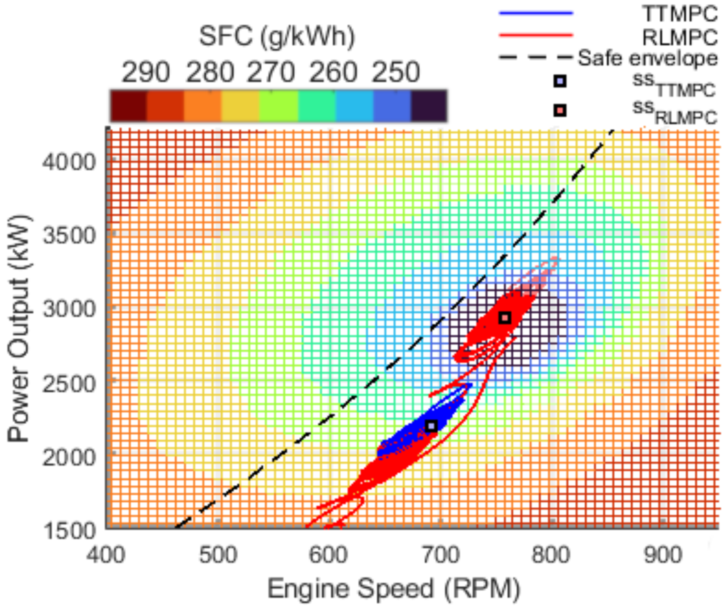


Figure 7.2.6 On-design scenario, engine trajectories for the two controllers

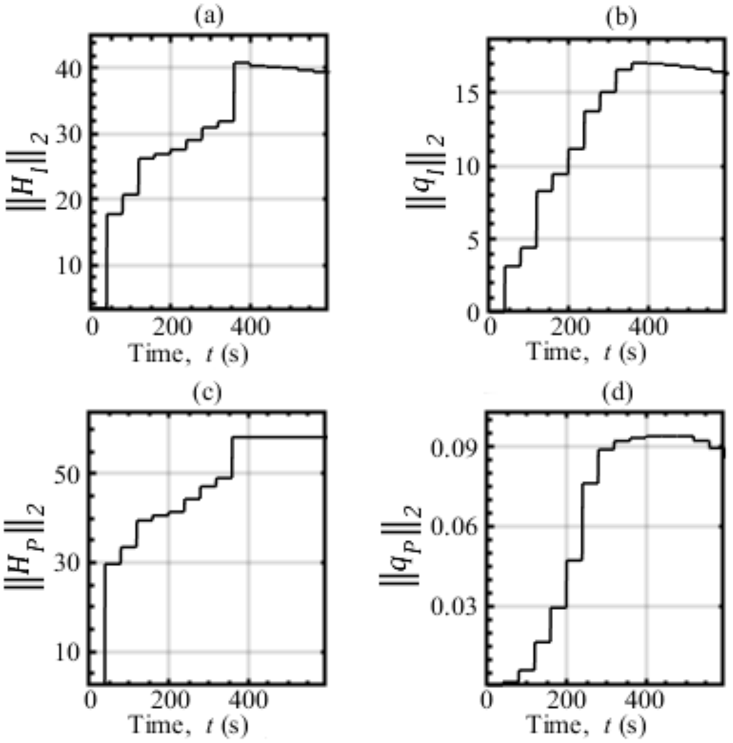


Figure 7.2.7 On-design scenario, 2-norm of cost θ-parameters of RLMPC: a) Stage cost Hessian, b) Stage cost gradient, c) Terminal cost Hessian, d) Terminal cost gradient.

highly intuitive: During severe weather, higher engine setpoint operation does not correspond to higher velocity due to the wave forces acting on the hull and propeller. This means that the economically sensible propulsion strategy is to track a (negligibly) reduced speed while applying conservative control actions. The latter is apparent if one examines the input profile shown in fig.7.2.8e: TTMPC consistently saturates the input in a naïve attempt to track the original velocity setpoint, while RL-MPC modulates the engine torque

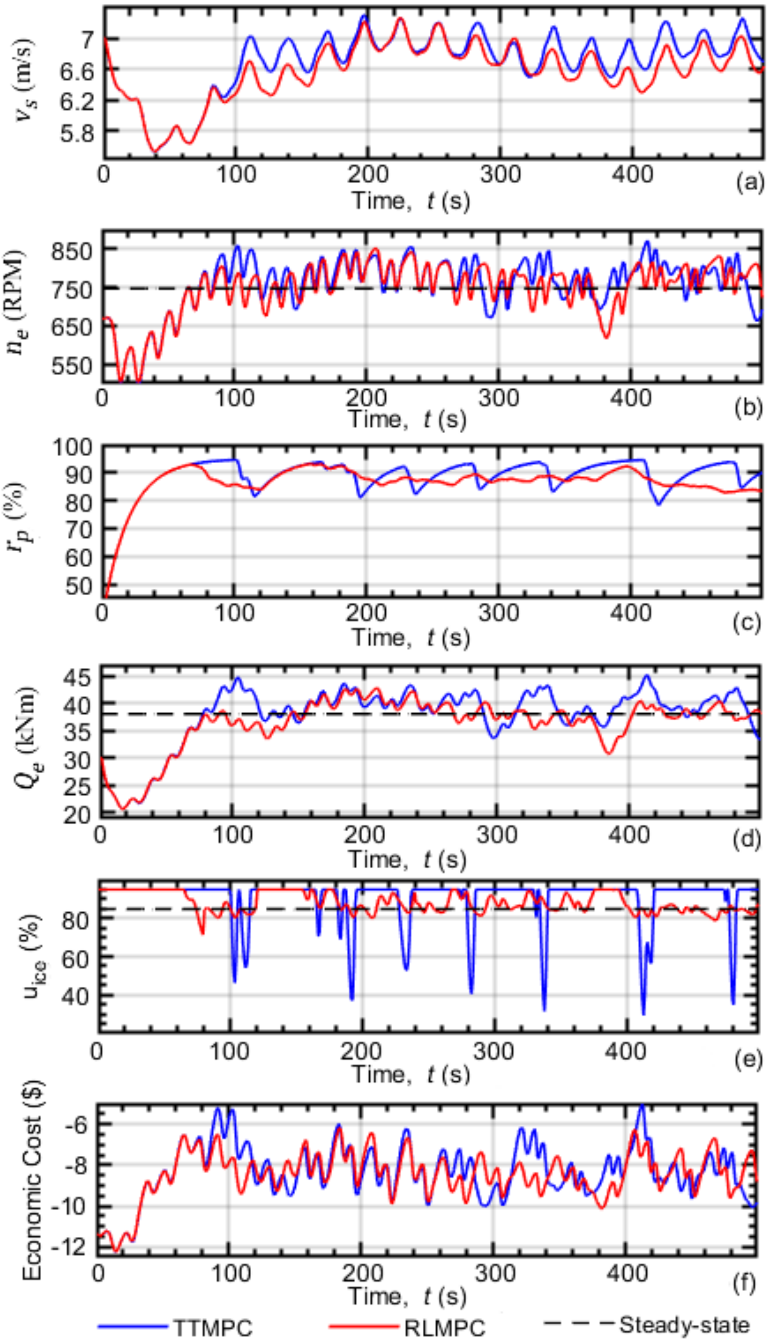


Figure 7.2.8 Off-design scenario, state & input results: a) Vessel speed, b) Engine speed, c) Spool percent, d) Engine torque output, e) Engine controller command, f) Vessel propulsion economic cost.

command in order to achieve economic performance, as encapsulated by L_f. This conservative propulsion control strategy also avoids engine overloading, as evident by the engine map trajectories for this scenario shown in fig. 7.2.9. Both controllers considerably modulate the power output of the engine due to the heavy disturbances acting on hull and propeller; however, TTMPC narrowly violates the safe engine operation envelope due to

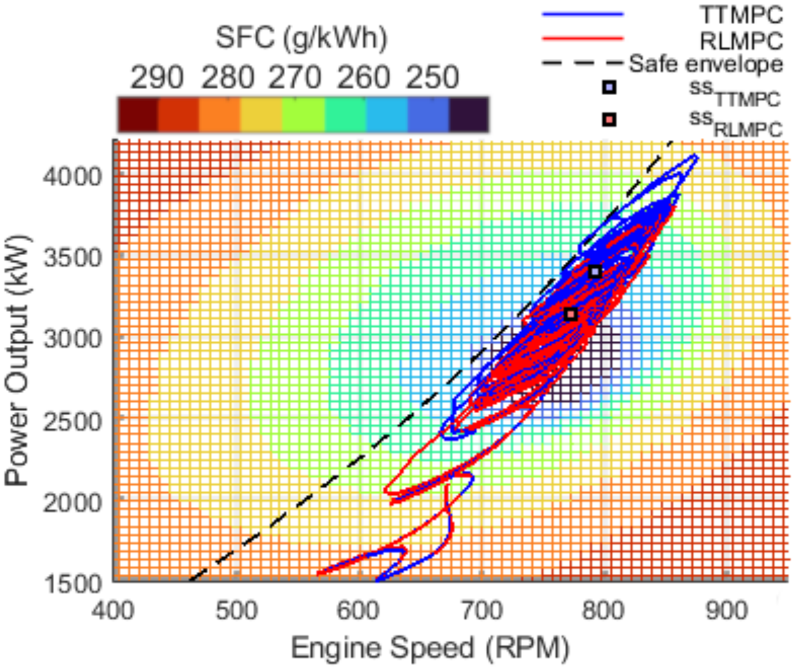


Figure 7.2.9 Off-design scenario, engine trajectories for the two controllers

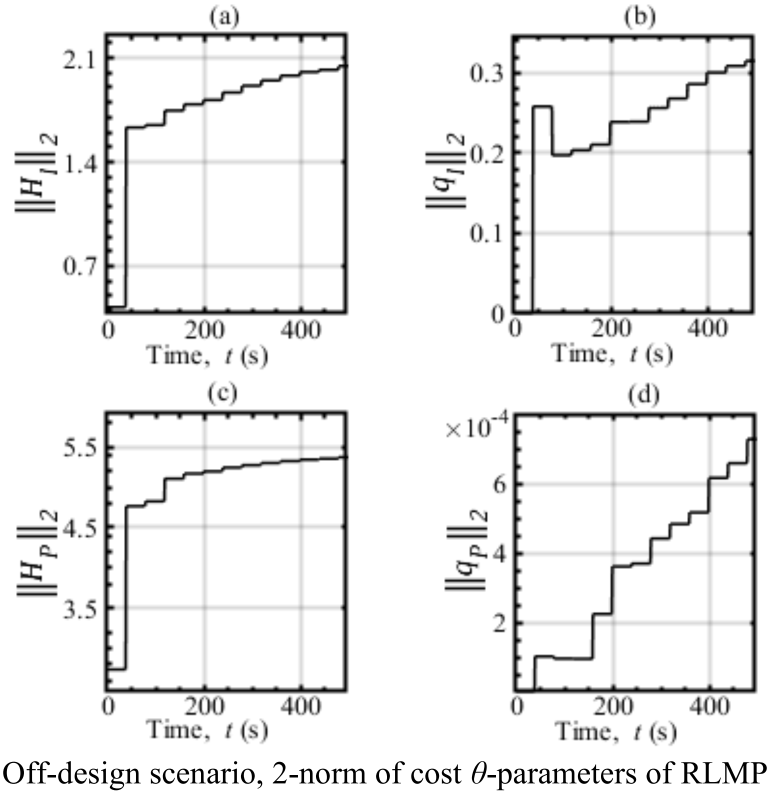


Figure 7.2.10 Off-design scenario, 2-norm of cost θ-parameters of RLMPC: a) Stage cost Hessian, b) Stage cost gradient, c) Terminal cost Hessian, d) Terminal cost gradient.

the saturation of the input command. In fig. 7.2.10, the evolution of the θ-parameters for the off-design scenario are presented. It should be noted that the convergence of the θ-parameters is less clear in this case; however, RL parameter convergence is not generally expected [212].

As discussed in Section 7.2.3.1, the economic performance of the vessel propulsion process is not arbitrarily quantified. Instead, it results as a trade-off between the fuel and

Table 7.2.5 On-Design Scenario Economic Results

	Total simulation time ($\boldsymbol{T_{sim}}$ = 600s)			t > 400s (new setpoint)		
	$\boldsymbol{L_f}$ ($)	ℓ_f ($)	ℓ_v ($)	$\boldsymbol{L_f}$ ($)	ℓ_f ($)	ℓ_v ($)
RLMPC	-6468.4	3523.7	-9992.1	-2223.4	1374.9	-3598.2
TTMPC	-6369.5	3462.2	-9831.6	-2104.4	1195.5	-3300.1
Diff. %	1.53	1.75	1.61	5.35	13.04	8.29

Table 7.2.6 Off-Design Scenario Economic Results

	Total simulation time ($\boldsymbol{T_{sim}}$ = 500s)		
	Total $\boldsymbol{L_f}$ **($)**	ℓ_f ($)	ℓ_v ($)
RLMPC	**-4237.2**	3820.0	-8057.1
TTMPC	**-4169.0**	4047.2	-8216.2
Diff. %	**1.61**	5.61	1.96

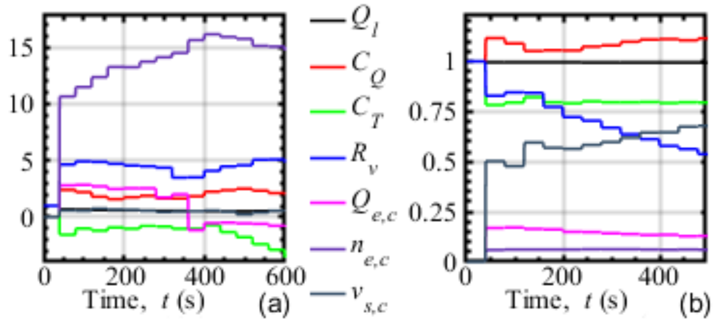


Figure 7.2.11: Model parametrization evolution θ^{f_θ}: a) On-design scenario, b) Off-design scenario

velocity prices p_f and p_v respectively, with the latter representing the operational incentive of freight commercial profit maximization. Therefore, one should not expect the tracking of a pre-specified velocity setpoint from RL-MPC, but rather the minimization of the economic cost L_f. The optimal policy $\pi_\star$ that minimizes this L_f cannot be known a-priori, for two reasons: Firstly, a wrong model f_θ may lead to inefficient control laws, as the on-design scenario results show; the engine map mismatch led to the generation of an unsuitable economic setpoint, which RL-MPC had to alter in order to coincide with the powertrain's nominal operation point. Secondly, external disturbances may invalidate a pre-computed economic policy; as shown in the off-design scenario, RL-MPC successfully takes heed of the reduced propulsion efficiency due to the severe sea conditions and adjusts its policy accordingly.

In addition, one should examine the economic performance of the two controllers, calculated using the economic stage cost $\ell^{f,v}_{p_f,p_v}$ that was presented in Section 7.2.3.1, using prices p_f, p_v sourced from Table 7.2.3. It is first reminded that this stage cost, shown in figs. 7.2.5f & 8f for the on- and off-design scenarios, respectively, reflects the actual operational returns of the vessel, since it prices the fuel cost plus the freight commercial profit. For ease of comparison, the economic results of the two simulation scenarios are summarized in Tables 7.2.5, 7.2.6. For the on-design scenario shown in figs. 7.2.5-7, RLMPC achieves 1.53% higher operational profit throughout the whole simulation time. However, once RLMPC settles in the new setpoint at t=400 s, a significant 5.35% increase in operational profit is recorded from that time, up to the end of the simulation. By observing the TD-error containing the true rewards $\ell^{f,v}_{p_f,p_v}$, RLMPC slowly adjusts the θ-parameters, in order to converge towards the true optimal economic setpoint, thus achieving this increased performance. In the off-design scenario, a 1.61% profit increase is seen throughout the simulation time; here, RL-MPC correctly identifies that the economic trade-off represented by $\ell^{f,v}_{p_f,p_v}$ requires a reduced service speed in the face of adverse weather conditions. To put the results of both scenarios in perspective: For the on-design scenario, RLMPC would increase profit by 2142 $/hr when operating at the new setpoint, while 491 $/hr would be saved for the off-design scenario case. It should be noted that RL-MPC achieves this increased economic performance without necessarily generating a sensible prediction model f_θ, as evident by the model parametrization θ^{f_θ} evolution for the two scenarios, shown in fig. 7.2.11. As discussed in Section 7.2.3.3, the objective of RL-MPC is to approximate $\pi_\star$ by toggling the entirety of the MPC constituents (cost, constraints, & model), rather than merely focusing on reducing model prediction error, as in typical system identification schemes.

7.2.5 Conclusion & Future Prospects

In this section, an RL-MPC controller is presented for the maximization of the economic performance of a vessel propulsion system. First, an economic MPC stage cost that incorporates the actual high-level chartering economic objectives is built. Then, using this stage cost, a data-driven RL-MPC controller for economic vessel propulsion is developed. Simulations show that the proposed vessel propulsion controller is able to maximize the economic performance by accommodating for model discrepancies and external disturbances.

It should be noted that the results of this section can be further extended to other types of propulsion systems, while also incorporating recent research results from the literature that combine RL-MPC with system identification [243], or account for safety-critical applications [211]. Still, the case of efficient operation of an autonomous vessel's requires not only the economic utilization of the propulsion system, but also its efficient navigation; for this reason, the unification of the results presented herein with the data-driven tracking controller capable of & navigation and collision avoidance presented in Section 6.2 is well-motivated.

7.3 A Control Law for the Data-Driven Navigation & Economic Propulsion Control of Vessels

As a continuation of the work presented in Sections 7.2 & 6.2, this chapter constructs a stable control law for the data-driven navigation & economic propulsion control of vessels (it should be noted that this chapter does not constitute a complete study to this end). There exists significant motivation for this endeavour, both from a practical & academic perspective: From a practical perspective, the proposal of a data-driven control law that incorporates actual vessel economic performance as well as path following can lead to the creation of fully autonomous & economic vessel agents, with tangible economic benefit. Next, from an academic perspective, a data-driven approach for the combined navigation & economic propulsion control of vehicles has not been created so far; two research works currently exist that are most relevant: The first constructed a data-driven trajectory tracking navigation control law for autonomous vessels using RL-based MPC, however without including a detailed modelling of the propulsion system and its respective economic objectives [244]. The second research work presented a stabilizing, dual-objective continuous-time MPC which did handle both a trajectory tracking & a propulsion energy objective for underactuated vehicles [208]; still, since it was not data-driven, it cannot handle modelling discrepancies, while also its economic objective formulation remained rather simple. To reiterate, the development of a novel, data-driven navigation & economic propulsion control law is well-motivated, and this section lays the necessary theoretical groundwork to this end.

7.3.1 Vessel Navigation & Propulsion Control Problem Statement

7.3.1.1 Kinematics & dynamics of vessel

In this subsection, the vessel model that incorporates 3DoF in motion as well as the propulsion system states is presented. The pose of a vessel is described by $\eta = [x, y, \psi]^{\mathsf{T}} \epsilon \mathbb{R}^2 \times \mathbb{S}$ where $\mathbb{S} = [0, 2\pi]$ is the set of yaw angles ψ, while its cartesian position in the fixed reference frame described by $p = [x, y]^{\mathsf{T}}$. Next, the velocity vector $\nu = [u_v, v, r]^{\mathsf{T}} \epsilon \mathbb{R}^3$ includes the body-fixed velocities u, v and the yaw velocity r. The dynamic model of the vessel is then given as [202]:

$$\dot{\eta} = J(\eta)\nu \tag{7.3.1a}$$

$$M\dot{\nu} + D(\nu)\nu + C(\nu)\nu = \tau_p + \tau_w \tag{7.3.1b}$$

Here, $M\epsilon\mathbb{R}^{3\times3}$ is the inertia matrix, $D(\nu)\epsilon\mathbb{R}^{3\times3}$ is the water damping matrix, $C(\nu)\epsilon\mathbb{R}^{3\times3}$ is the Coriolis matrix, τ_p, τ_w are the vessel thrust and wave forces, respectively, and lastly, $J(\cdot)$ is the North East Down (NED) reference transformation matrix, denoted as:

$$J(\eta) = \begin{bmatrix} \cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.3.2}$$

Typically, evaluating the thrust vector τ_p would involve the mapping of the actuators installed on the vessel to generate the respective forces and moments for each body-fixed direction. Here, without loss of generality, it is assumed that there exists a single thruster generating thrust magnitude f_0 at an azimuth angle of a_0. Thus, the vessel's thrust force vector τ_p can be written as follows:

$$\tau_p = \begin{bmatrix} f_0\cos(a_0) \\ f_0\sin(a_0) \\ f_0\left(d_x\sin(a_0) - d_y\cos(a_0)\right) \end{bmatrix} \tag{7.3.3}$$

Here, (d_x, d_y) is the distance of the thruster from the body-fixed axis of the vessel.

Next, the thruster dynamics can be presented; typically, the azimuth angle dynamics can be described by a time-delay ODE

$$\dot{a}_0 = \frac{1}{t_a}\left(-a_0 + a_{0,c}\right) \tag{7.3.4}$$

where t_a is a time constant and $a_{0,c}$ is the azimuth angle command. The thrust magnitude f_0 however, encapsulates the propulsion system dynamics and thus requires detailed modeling. For this reason, we leverage the internal combustion engine plant model presented in Section 7.2.2.1 with DAEs 7.2.1-13, which contains the states $\left[v_s \; n_e \; r_p \; Q_e\right]^T$, where v_s is the longitudinal velocity, n_e is the engine angular velocity, r_p is an internal state variable describing turbocharger spooling and fuel rack time-delay dynamics, and Q_e is the current engine output brake torque. This model accepts the input u_{ice}, which represents the torque command to the engine, as a percentage of the maximum torque that is currently available. The first step to creating the propulsion system model $f_{prop}(\cdot)$ is to omit the velocity v_s (7.2.9), since a representation already exists (7.3.1). Next, the propeller thrust magnitude f_0 is written as :

$$f_0 = C_T\rho v_h^2\pi D^2 \tag{7.3.5}$$

Here, C_T is a propeller constant, ρ the water density, v_h is the propeller's hydrodynamic velocity (which is a function of engine speed n_e), and D the propeller's diameter. Using $f_{prop}(\cdot)$ and (7.3.1) the implicit dynamics of the full vessel model $f_{full}(\cdot)$ encompassing the 3DoF hull dynamics as well as the propulsion system can be created:

$$\begin{bmatrix} \dot{\eta} - J(\eta)\nu \\ M\dot{\nu} + D(\nu)\nu + C(\nu)\nu - a_0 f_0 - \tau_w \\ \dot{a_0} - \frac{1}{t_a}\left(-a_0 + a_{0,c}\right) \\ \dot{f}_{prop} - f_{prop}\left(\left[u_v\ n_e\ r_p\ Q_e\right], \left[u_{ice}\right]\right) \end{bmatrix} = 0 \tag{7.3.6}$$

The model $f_{full}(\cdot)$ contains the states

$$x = \left[\eta\ \nu\ a_0\ n_e\ r_p\ Q_e\right]^T, x \in \mathbb{X} \tag{7.3.7}$$

and accepts the input and environmental disturbance variables

$$u = \left[u_{ice}, a_{0,c}\right], \quad u \in \mathbb{U} \tag{7.3.8a}$$

$$d = \tau_w, d \in \mathbb{D} \tag{7.3.8b}$$

where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{D}$ are the admissible state, input, and disturbance spaces, respectively.

7.3.1.2 Control objectives

The first control objective pertains to the trajectory tracking of a desired pose η_d. Splitting the task into position and heading, two respective functions are created [244]:

$$C_{pos}(\eta) = \delta^2\left(\sqrt{1 + \frac{(x - x_d)^2 + (y - y_d)^2}{\delta^2}} - 1\right) \tag{7.3.9a}$$

$$C_{head}(\eta) = \frac{1 - \cos\left(\psi - \psi_d\right)}{2} \tag{7.3.9b}$$

The $C_{pos}(\cdot)$ function represents a pseudo-Huber distance cost, with attractive numerical characteristics [245] for usage in MPC, while $C_{head}(\cdot)$ ensures that no heading angle wrap-around occurs during calculation. The two functions are combined to form the following positive-definite tracking cost function:

$$\ell_t(x, u) = q_{pos}\, C_{pos}(\eta) + q_{head}\, C_{head}(\eta) + q_a {a_0}^2 \tag{7.3.7}$$

Here, the last term penalizes excessive thruster angle changes and the q terms are weights. Next, the second control objective pertains to the economic propulsion trade-off $\ell_\beta^{f,v}$ and is already described in subsection 7.2.2.2:

$$\ell_{p_f,p_v}^{f,v}(x, u) = p_f\, g_f(x, u) + p_v\, u_v \tag{7.3.8}$$

Here g_f is the expended fuel and p_f, p_v are the fuel unit and cost-of-opportunity prices, respectively. Note that $\ell_{p_f,p_v}^{f,v}$ is a generic function. The two cost functions (7.3.7,8) can be combined in order to create the final stage cost, as follows:

$$\ell_0(x,u) = \ell_t(x,u) + \ell_{p_f,p_v}^{f,v}(x,u) \tag{7.3.9}$$

Note that this is also a generic stage cost, even though $\ell_t(\cdot)$ is quadratic.

7.3.2 Methodology

The creation of the control law for the data-driven vessel navigation & economic propulsion control follows the same methodological outline as subsection 7.2.3 : The idea is to use stage cost $\ell_0(\cdot)$ to generate a stabilizing EMPC control law, which can then be approximated up to 1st order by a properly tuned MPC through the procedure described in section 7.2 & [239]. Next, this tuned MPC can be paired with RL tools as in 7.2.3.5 in order to yield the desired control law, capable of handling modeling mismatches and disturbances.

7.3.2.1 Stabilizing EMPC

Before formulating the stabilizing EMPC control law, the optimal steady-state for the propulsion system must be calculated. First, the propulsion system model $f_{prop}(\cdot)$ is augmented by the surge velocity part u_v of the hull dynamics kinematic model:

$$\dot{f}_{prop}^{(aug.)}(x,u) = \left\{ f_{prop}(\cdot), \left(-M^{-1}(D(v)v - C(v)v + a_0 f_0 + \tau_w)\right)_{u_v} \right\} \tag{7.3.10}$$

Model $f_{prop}^{(aug)}(x,u)$ consolidates the states $x_{prop} = \left[u_v \; n_e \; r_p \; Q_e\right]$ and input $u_{prop} = u_{ice}$, thus describing the propulsion system & surge dynamics of the vessel. Similar to 7.2.3.1, a steady-state optimization problem can be formulated using economic stage cost $\ell_{p_f,p_v}^{f,v}(\cdot)$:

$$\begin{gathered} \min_{x,u} \ell_{p_f,p_v}^{f,v}(x_k,u_k) \\ \text{s.t. } x_{k+1} = f_{prop}^{(aug),(Ts)}(x_k,u_k),\; (x,u) \in \mathbb{Z} \end{gathered} \tag{7.3.11}$$

This will yield the propulsion system's optimal economic steady-states $x_{prop,(s)}$ and $u_{prop,(s)}$, for given pricing parameters p_f, p_v.

Next, using the economic stage cost $\ell_0(\cdot)$ (7.3.9) of the full system, the OCP of an EMPC controller can be formulated as:

$$\min_{x,u} J_{EMPC}(x,u) \tag{7.3.12a}$$

$$\text{s.t.} \qquad x(0) = \hat{x}_0 \tag{7.3.12b}$$

$$(x,u) \in \mathbb{Z} \tag{7.3.12c}$$

$$x_{k+1} = f_{full}{}^{(T_s)}(x_k, u_k) \tag{7.3.12d}$$

$$p_f = p_{f,d}, p_v = p_{v,d} \tag{7.3.12e}$$

$$\eta = \eta_d \tag{7.3.12f}$$

$$x(N) = x_N \tag{7.3.12h}$$

Where $p_{f,d}$, $p_{v,d}$ are the desired pricing parameters for $\ell_{p_f,p_v}^{f,v}(\cdot)$, $f_{full}{}^{(T_s)}(\cdot)$ is the system model discretized by the sample time T_s, $\mathbb{Z}$ is the admissible state-input space, and η_d is a vector of reference poses up to the prediction horizon *N*. Eq. (7.3.10f) represents the stability-enforcing terminal constraint of the scheme: Here, x_N depends on whether $\eta_d(N)$, i.e. the desired pose at the end of the prediction horizon, refers to an intermediate or final waypoint; for the first case, $x_N = [\eta_d(N)\ v_N\ a_{0,N}\ n_{e,(s)}\ r_{p,(s)}\ Q_{e,(s)}]$, i.e. (the desired terminal position and velocity, together with the propulsion system's optimal economic steady-states, while for the second case, $x_N = [\eta_d(N)\ 0\ a_{0,N}\ 0\ 0\ 0]$, (i.e. the desired static terminal position, with inactive propulsion). Next, J_{EMPC} is the OCP's objective function:

$$J_{EMPC}(x,u) = \sum_{k=0}^{N-1} \big(\ell_0(x_k, u_k)\big) + P(x_N) \tag{7.3.13}$$

Where $P(\cdot)$ is a suitably-chosen terminal penalty. Here, it can be argued that the conditions exist for the usage of the gradient-correcting terminal penalty used in subsection 7.1.3: First, the system (7.3.7,8) is dissipative, and second, the stage cost $\ell_0(\cdot)$ has a non-zero gradient at the optimal steady-state [221] - more specifically, it holds that $\nabla_x \ell_{p_f,p_v}^{f,v}(x_s, u_s) \neq 0$, $\nabla_u \ell_{p_f,p_v}^{f,v}(x_s, u_s) \neq 0$, even though $\nabla \ell_t(x_s, u_s) = 0$. Therefore, the EMPC control law resulting from (7.3.10) is stabilizing.

7.3.2.2 Outline of RL-based MPC for navigation & economic propulsion

The stabilizing EMPC control law developed in subsection 7.3.2.1 must first be convexified up to 1st order so as to produce a specially-tuned tracking MPC control law, using Problem (7.2.28) presented in 7.2.3.2. The solution of this Problem admits a quadratic stage cost ℓ_Q that can subsequently be used to develop a θ-parametrized MPC for usage within RL tools:

$$\min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma) \tag{7.3.14a}$$

$$\text{s.t.} \quad x_0 = x_0{}' \tag{7.3.14b}$$

$$x_{k+1} = f_{full,\theta}^{(T_s)}(x_k, u_k) \tag{7.3.14c}$$

$$g(u_k) \leq 0 \tag{7.3.14d}$$

$$h_\theta(x_k, u_k) \leq \sigma_k \tag{7.3.14e}$$

This OCP is comprised of a discretized model $f_{full,\theta}^{(T_s)}$, a terminal penalty P_θ, a state-input constraint map h_θ and a bound constraint map g. Note that the convexified stage cost $l_{Q,\theta}$ can be extended to include collision avoidance objectives built upon data-driven obstacle vessel models, as presented in subsection 6.2.4.2. The objective function $J_{MPC,\theta}(\cdot)$ is:

$$J_{MPC,\theta}(x,u,\sigma) = \lambda_\theta(x_0) + \gamma^N P_\theta(x_N) + \sum_{\kappa=0}^{N-1} \gamma^k (l_\theta(x_k, u_k) + w^T \sigma_k) \qquad (7.3.15)$$

Here, l_θ is the parametrized stage cost, λ_θ a parametrized initial cost function required for stability as discussed in subsection 7.2.3.4, and P_θ a parametrized terminal penalty. This MPC is then used to construct the following value and action-value functions for the navigation & economic propulsion problem:

$$V_\theta(s) = \min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma), \qquad s.t\ (7.2.37\text{b}) - (7.2.37\text{e}) \qquad (7.3.16\text{a})$$

$$\begin{gathered} Q_\theta(s,a) = \min_{x,u,\sigma} J_{MPC,\theta}(x,u,\sigma) \\ s.t\ (7.2.37\text{b}) - (7.2.37\text{e}) \\ u_0 = a \end{gathered} \qquad (7.3.16\text{b})$$

These value and action-value functions correspond to the policy $\pi_\theta = u_0^\star$, as retrieved from Problem (7.2.26). Then, RL tools that toggle the θ-parametrization of π_θ can be applied, in order to approximate the optimal economic policy $\pi_\star$.

Corollary 1 (Asymptotic stability of data-driven MPC control law π_θ): *Suppose assumptions of Theorem 4.3.3 are satisfied as well as*

A. Initial state $x_0{}'$ lies in the region of attraction & prediction horizon N is sufficiently long.

B. Initial cost $\lambda_\theta(\cdot)$, terminal penalty $P_\theta(\cdot)$, and stage cost $l_\theta(\cdot)$ are all positive definite.

Then, the origin is asymptotically stable in $\mathcal{X}_N$, and the optimal economic policy $\pi_\star$ can be learned.

Proof: Initial positive-definite values for $l_\theta(\cdot)$ and $P_\theta(\cdot)$ are obtained by using Problem (28) presented in 7.2.3.2, and by solving the DARE as discussed in intro, respectively. Function $\lambda_\theta(\cdot)$ can be initialized using any positive-definite function [212]. This means that Theorem 4.3.3 can be employed to show that $J_{MPC,\theta}(\cdot)$ is a Lyapunov function ■

Using a batch TD-learning scheme such as the one presented in subsection 7.2.3.4 which ensures the positive-definitiveness of $l_{\theta}(\cdot)$, $P_{\theta}(\cdot)$, and $\lambda_{\theta}(\cdot)$ guarantees that the resulting RL-MPC data-driven control law is stable.

Remark: The full creation of the aforementioned control law requires the proper parametrization of the vessel model (7.3.7,8) and the tuning of the RL parameters, which, together with the actual simulation studies, remain as items of future research. However, their similarity to the work already carried out in subsection 7.2.3 is noted. In general, it is the author's opinion that the concept presented here can be further developed & eventually applied in practice with little extra effort using section 7.2 as a roadmap.

Conclusion & Outlook

This thesis focused on the construction of data-driven MPC schemes and optimization methods by employing computational intelligence & machine learning tools. The topic was approached from both a theoretical and practical perspective, and the economic benefits of the proposed algorithms were showcased over the optimization & MPC control of a wide variety of engineering applications.

First, CI-based methods were developed that addressed the load forecasting task and the optimal dispatch of distributed renewable sources. The load prediction task was addressed using a pool of selected ML algorithms with an online dynamic selection system; therefore, the advantages of each algorithm are combined in order to create a prediction ensemble of high accuracy. It is noted that this approach can be generalized to other types of timeseries forecasting problems, characterized by stochasticity. Next, a metaheuristic optimization algorithm capable of exploiting structural & topological information of networked problems was created, & applied to the optimal power flow problem of smart distribution grids. The proposed method employed a community-detection algorithm in order to assign effective cooperation sets for a cooperative PSO algorithm, with significant economic results. The author notes that this optimization approach could be extended to the optimization of other large-scale engineering systems that similarly exhibit topological structure. Lastly, it is noted that the two load forecasting & OPF methods presented in this thesis could be leveraged to create a smart-grid decision support system.

Next, two data-driven tracking MPC methods based on RBFN were designed, in order to capture nonlinear or otherwise hard-to-model processes. Two representative case studies were chosen: The first case study concerned the control of a vehicle’s active suspension, the dimensionality of which prohibits the usage of standard linear or ODE-integrated prediction models; the application of RBF models for this task leads to significant control performance improvement, over various performance indexes. The second data-driven MPC controller addressed the vessel collision avoidance task using obstacle trajectory models trained on real vessel data; here, the usage of RBF models is mandated due to their black-box nature. It is noted that the future prospects of the proposed MPC controllers extend further than the two specific case studies presented here; other types of systems that exhibit high nonlinearity & complexity, could benefit from the data-driven approach.

Finally, a data-driven nonlinear EMPC scheme was presented for the economic control of a vessel propulsion system: First, a stabilizing EMPC propulsion control law was constructed, with significant fuel economy benefits. Serving as a proof of concept, this control law was extended to a data-driven EMPC scheme based on reinforcement learning, using an economic stage cost that accurately described the real, high-level chartering economics of freight shipping. This learning controller successfully handled structural modelling discrepancies between plant & model, leading to higher closed loop economic performances. Finally, the theoretical groundwork of a control law that leverages previous results was laid, pertaining to the data-driven autonomous navigation & economic propulsion control of vessels.

Future research will first focus on further developing the aforementioned control law and simulating it with various scenarios. Next, more applications will be explored that would benefit from learning-based MPC controllers with theoretical guarantees, such as wastewater plants. In general, it is the author's opinion that the potentialities of CI/ML tools for optimization & data-driven control have not yet been fully exploited, both from a practical and a theoretical perspective. Therefore, he hopes that this thesis will serve as a starting point for future endeavours towards this end.

References

[1] M. Papadimitrakis, N. Giamarelos, M. Stogiannos, E. N. Zois, N. A. I. Livanos, and A. Alexandridis, "Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications," *Renew. Sustain. Energy Rev.*, vol. 145, no. April, p. 111072, 2021.

[2] M. Papadimitrakis, M. Stogiannos, H. Sarimveis, and A. Alexandridis, "Multi-Ship Control and Collision Avoidance Using MPC and RBF Based Trajectory Predictions," *Sensors*, 2021.

[3] N. Giamarelos, E. N. Zois, M. Papadimitrakis, M. Stogiannos, N. A. I. Livanos, and A. Alexandridis, "Short-Term Electric Load Forecasting with Sparse Coding Methods," *IEEE Access*, vol. 9, pp. 102847–102861, 2021.

[4] M. Papadimitrakis and A. Alexandridis, "Active vehicle suspension control using road preview model predictive control and radial basis function networks," *Appl. Soft Comput.*, vol. 120, p. 108646, 2022.

[5] M. Papadimitrakis, A. Kapnopoulos, S. Tsavartzidis, and A. Alexandridis, "A cooperative PSO algorithm for Volt-VAR optimization in smart distribution grids," *Electr. Power Syst. Res.*, vol. 212, p. 108618, Nov. 2022.

[6] M. Stogiannos, M. Papadimitrakis, H. Sarimveis, and A. Alexandridis, "Vessel Trajectory Prediction Using Radial Basis Function Neural Networks," in *2021 IEEE 19th International Conference on Smart Technologies (EUROCON)*, 2021.

[7] N. Giamarelos, M. Papadimitrakis, M. Stogiannos, E. N. Zois, N.-A. I. Livanos, and A. Alexandridis, "A Machine Learning Model Ensemble for Mixed Power Load Forecasting across Multiple Time Horizons," *Sensors*, vol. 23, no. 12, 2023.

[8] M. Papadimitrakis and A. Alexandridis, "A Vessel Propulsion Controller based on Economic Model Predictive Control," in *American Control Conference (in press)*, 2023.

[9] J. A. Momoh, M. E. El-Hawary, and R. Adapa, “A review of selected optimal power flow literature to 1993 Part I: Nonlinear and Quadratic Programming Approaches,” *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–103, 1999.

[10] J. A. Momoh, M. E. El-Hawary, and R. Adapa, “A Review of Selected Optimal Power Flow Literature to 1993 Part II: Newton, Linear Programming and Interior Point Methods.,” *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–104, 1999.

[11] S. Frank, I. Steponavice, and S. Rebennack, “Optimal power flow: A bibliographic survey I Formulations and deterministic methods,” *Energy Syst.*, vol. 3, no. 3, pp. 221–258, 2012.

[12] T. A. Johansen, T. Perez, and A. Cristofaro, “Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3407–3422, 2016.

[13] S. Xie, V. Garofano, X. Chu, and R. R. Negenborn, “Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks,” *Ocean Eng.*, vol. 193, no. October, p. 106609, 2019.

[14] J. Mattingley and S. Boyd, “CVXGEN: A code generator for embedded convex optimization,” *Optim. Eng.*, vol. 13, no. 1, pp. 1–27, 2012.

[15] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, “Efficient interior point methods for multistage problems arising in receding horizon control,” *Proc. IEEE Conf. Decis. Control*, pp. 668–674, 2012.

[16] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd Editio. Springer New York, 2007.

[17] D. Q. Mayne and H. Michalska, “Receding horizon control of nonlinear systems,” *IEEE Trans. Automat. Contr.*, vol. 35, no. 7, pp. 814–824, 1990.

[18] S. S. Keerthi and E. G. Gilbert, “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations,” *J. Optim. Theory Appl.*, vol. 57, no. 2, pp. 265–293, 1988.

[19] R. R. Negenborn and J. M. Maestre, “Distributed Model Predictive Control: An overview of features and research opportunities,” *Proc. 11th IEEE Int. Conf. Networking, Sens. Control. ICNSC 2014*, no. April, pp. 530–535, 2014.

[20] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design (2nd Edition)*, vol. 18, no. 8. 2008.

[21] M. Zanon, T. Faulwasser, and S. Gros, “Practical Economic MPC,” *Pamm*, vol. 20, no. 1,

pp. 2–5, 2021.

[22] A. Alexandridis, H. Sarimveis, and G. Bafas, "Modeling and control of continuous digesters using the PLS methodology," *Chem. Eng. Commun.*, vol. 191, no. 10, pp. 1271–1284, 2004.

[23] T. Sardarmehni, J. Keighobadi, M. B. Menhaj, and H. Rahmani, "Robust predictive control of lambda in internal combustion engines using neural networks," *Arch. Civ. Mech. Eng.*, vol. 13, no. 4, pp. 432–443, 2013.

[24] H. Zheng, R. R. Negenborn, and G. Lodewijks, *Trajectory tracking of autonomous vessels using model predictive control*, vol. 19, no. 3. IFAC, 2014.

[25] L. F. Recalde, B. S. Guevara, C. P. Carvajal, V. H. Andaluz, J. Varela-Aldás, and D. C. Gandolfo, "System Identification and Nonlinear Model Predictive Control with Collision Avoidance Applied in Hexacopters UAVs," *Sensors*, vol. 22, no. 13, pp. 1–29, 2022.

[26] S. Haykin, *Nueral Network*, 2nd editio. Pearson Prentice Hall, 2005.

[27] D. Karamichailidou, V. Kaloutsa, and A. Alexandridis, "Wind turbine power curve modeling using radial basis function neural networks and tabu search," *Renew. Energy*, 2021.

[28] Y. I. Alamin, J. D. Álvarez, M. del Mar Castilla, and A. Ruano, "An Artificial Neural Network (ANN) model to predict the electric load profile for an HVAC system," *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 26–31, Jan. 2018.

[29] A. P. Alexandridis, C. I. Siettos, H. K. Sarimveis, A. G. Boudouvis, and G. V Bafas, "Modelling of nonlinear process dynamics using Kohonen's neural networks, fuzzy systems and Chebyshev series," *Comput. Chem. Eng.*, vol. 26, no. 4, pp. 479–486, 2002.

[30] M. Papadimitrakis, A. Kapnopoulos, S. Tsavartzidis, and A. Alexandridis, "Cooperative PSO algorithm for Volt-Var optimization in smart distribution grids," *(submitted)*, 2021.

[31] R. A. El Sehiemy, F. Selim, B. Bentouati, and M. A. Abido, "A novel multi-objective hybrid particle swarm and salp optimization algorithm for technical-economical-environmental operation in power systems," *Energy*, vol. 193, p. 116817, 2020.

[32] A. Kapnopoulos and A. Alexandridis, "A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme," *Aerosp. Sci. Technol.*, vol. 127, p. 107725, 2022.

[33] D. Kandris, A. Alexandridis, T. Dagiuklas, E. Panaousis, and D. D. Vergados, "Multiobjective Optimization Algorithms for Wireless Sensor Networks," *Wirel. Commun. Mob. Comput.*, vol. 2020, p. 4652801, 2020.

[34] A. Alexandridis, E. Chondrodima, N. Giannopoulos, and H. Sarimveis, “A fast and efficient method for training categorical radial basis function networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 11, pp. 2831–2836, Nov. 2017.

[35] D. Karamichailidou, S. Koletsios, and A. Alexandridis, “An RBF online learning scheme for non-stationary environments based on fuzzy means and Givens rotations,” *Neurocomputing*, vol. 501, pp. 370–386, 2022.

[36] L. H. Cseko, M. Kvasnica, and B. Lantos, “Explicit MPC-Based RBF Neural Network Controller Design with Discrete-Time Actual Kalman Filter for Semiactive Suspension,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1736–1753, 2015.

[37] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-Based Model Predictive Control: Toward Safe Learning in Control,” *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 3, pp. 269–296, 2020.

[38] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, “Reinforcement learning for control: Performance, stability, and deep approximators,” *Annu. Rev. Control*, vol. 46, pp. 8–28, 2018.

[39] X. Li and X. Yao, “Cooperatively coevolving particle swarms for large scale optimization,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, 2012.

[40] Y. Ling, H. Li, and B. Cao, “Cooperative co-evolution with graph-based differential grouping for large scale global optimization,” *2016 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov. ICNC-FSKD 2016*, vol. X, no. X, pp. 95–102, 2016.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2017.

[42] Y. Y. DG Luenberger, *Linear and Nonlinear Programming*, 3rd ed. Springer, 1984.

[43] A. P. Ruszczyński, *Nonlinear optimization*. Princeton University Press, 2006.

[44] G. Sagnol and M. Stahlberg, “PICOS: A Python interface to conic optimization solvers,” *J. Open Source Softw.*, vol. 7, no. 70, p. 3915, 2022.

[45] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming: Theory and Applications*. SIAM, 1994.

[46] A. Askarzadeh, “A Memory-Based Genetic Algorithm for Optimization of Power Generation in a Microgrid,” *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1081–1089, Jul. 2018.

[47] H. Pulluri, R. Naresh, and V. Sharma, “An enhanced self-adaptive differential evolution

based solution methodology for multiobjective optimal power flow," *Appl. Soft Comput. J.*, vol. 54, pp. 229–245, 2017.

[48] M. O. W. Grond, H. N. Luong, J. Morren, P. A. N. Bosman, H. J. G. Slootweg, and H. La Poutré, "Practice-oriented optimization of distribution network planning using metaheuristic algorithms," in *Proceedings - 2014 Power Systems Computation Conference, PSCC 2014*, 2014.

[49] P. Patrinos, A. Alexandridis, K. Ninos, and H. Sarimveis, "Variable selection in nonlinear modeling based on RBF networks and evolutionary computation," *Int. J. Neural Syst.*, vol. 20, no. 5, pp. 365–379, 2010.

[50] Sarjiya, A. B. Mulyawan, A. Setiawan, and A. Sudiarso, "Thermal unit commitment solution using genetic algorithm combined with the principle of tabu search and priority list method," *Proc. - 2013 Int. Conf. Inf. Technol. Electr. Eng. "Intelligent Green Technol. Sustain. Dev. ICITEE 2013*, pp. 414–419, 2013.

[51] V. Roberge, M. Tarbouchi, and F. A. Okou, "Distribution System Optimization on Graphics Processing Unit," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1689–1699, 2017.

[52] A. Asrari, S. Lotfifard, and M. Ansari, "Reconfiguration of Smart Distribution Systems with Time Varying Loads Using Parallel Computing," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2713–2723, 2016.

[53] M. Dorigo and T. Stützle, "Ant Colony Optimization: Overview and Recent Advances," 2010, pp. 227–263.

[54] A. A. A. Mousa, "Hybrid ant optimization system for multiobjective economic emission load dispatch problem under fuzziness," *Swarm Evol. Comput.*, vol. 18, pp. 11–21, 2014.

[55] B. Allaoua and A. Laoufi, "Optimal power flow solution using ant manners for electrical network," *Adv. Electr. Comput. Eng.*, vol. 9, no. 1, pp. 34–40, 2009.

[56] A. Silva, J. Marinheiro, H. L. Cardoso, and E. Oliveira, "Demand-side management in power grids: An ant colony optimization approach," in *Proceedings - IEEE 18th International Conference on Computational Science and Engineering, CSE 2015*, 2015, pp. 300–306.

[57] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[58] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.

[59] A. Alexandridis and E. Chondrodima, “A medical diagnostic tool based on radial basis function classifiers and evolutionary simulated annealing,” *J. Biomed. Inform.*, vol. 49, pp. 61–72, 2014.

[60] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, “Unit commitment by an enhanced simulated annealing algorithm,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 68–76, 2006.

[61] M. A. Abido, “Optimal power flow using tabu search algorithm,” *Electr. Power Components Syst.*, vol. 30, no. 6, pp. 469–483, 2002.

[62] M. Stogiannos, A. Alexandridis, and H. Sarimveis, “An enhanced decentralized artificial immune-based strategy formulation algorithm for swarms of autonomous vehicles,” *Appl. Soft Comput. J.*, vol. 89, p. 106135, Apr. 2020.

[63] S. S. F. Souza, R. Romero, J. Pereira, and J. T. Saraiva, “Artificial immune algorithm applied to distribution system reconfiguration with variable demand,” *Int. J. Electr. Power Energy Syst.*, vol. 82, pp. 561–568, 2016.

[64] S. A. Taher and S. A. Afsari, “Optimal location and sizing of DSTATCOM in distribution systems by immune algorithm,” *Int. J. Electr. Power Energy Syst.*, vol. 60, pp. 34–44, Sep. 2014.

[65] L. De Mello Honório, A. M. L. Da Silva, and D. A. Barbosa, “A gradient-based artificial immune system applied to optimal power flow problems,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4628 LNCS, pp. 1–12, 2007.

[66] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4.

[67] N. Borges, J. Soares, and Z. Vale, “A Robust Optimization for Day-ahead Microgrid Dispatch Considering Uncertainties,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3350–3355, Jul. 2017.

[68] M. A. Abido, “Optimal Power Flow using Particle Swarm Optimization,” *Int. J. Electr. Power Energy Syst.*, vol. 24, no. 1, pp. 563–571, 2002.

[69] Y. Shi and R. Eberhart, “Modified particle swarm optimizer,” in *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1998.

[70] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Trans. Evol. Comput.*, 2002.

[71] C. Yanwei and Y. Guofang, “Multi-Sub-Swarm PSO Classifier Design and Rule

Extraction," 2013.

[72] A. Alexandridis and H. Sarimveis, "Nonlinear adaptive model predictive control based on self-correcting neural network models," *AIChE J.*, vol. 51, no. 9, pp. 2495–2506, 2005.

[73] C. Darken and J. Moody, "Fast, Adaptive K-Means Clustering: Some Empirical Results," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1990)*, 1990.

[74] J. H. Lee, "Model predictive control: Review of the three decades of development," *Int. J. Control. Autom. Syst.*, vol. 9, no. 3, pp. 415–424, 2011.

[75] C. Gohrle, A. Schindler, A. Wagner, and O. Sawodny, "Design and vehicle implementation of preview active suspension controllers," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1135–1142, 2014.

[76] J. K. Hedrick and M. D. Donahue, "Implementation of an Active Suspension," *Nonlinear Hybrid Syst. Automot. Control*, vol. XVIII, p. 446, 2003.

[77] D. Limon, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained MPC without terminal constraint," *IEEE Trans. Automat. Contr.*, vol. 51, no. 5, pp. 832–836, 2006.

[78] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *IEEE Trans. Automat. Contr.*, vol. 57, no. 7, pp. 1615–1626, 2012.

[79] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov function for economic optimizing model predictive control," *IEEE Trans. Automat. Contr.*, vol. 56, no. 3, pp. 703–707, 2011.

[80] N. Hatziargyriou, *Microgrids: Architectures and Control*. 2013.

[81] T. Strasser *et al.*, "A Review of Architectures and Concepts for Intelligence in Future Electric Energy Systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4. pp. 2424–2438, 2015.

[82] S. R. Salkuti, "Day-ahead thermal and renewable power generation scheduling considering uncertainty," *Renew. Energy*, vol. 131, pp. 956–965, Feb. 2019.

[83] N. Zhang *et al.*, "A Convex Model of Risk-Based Unit Commitment for Day-Ahead Market Clearing Considering Wind Power Uncertainty," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1582–1592, 2015.

[84] U. B. Tayab, A. Zia, F. Yang, J. Lu, and M. Kashif, "Short-term load forecasting for

microgrid energy management system using hybrid HHO-FNN model with best-basis stationary wavelet packet transform," *Energy*, vol. 203, p. 117857, Jul. 2020.

[85] R. Basir *et al.*, "Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges," *Sensors 2019, Vol. 19, Page 4807*, vol. 19, no. 21, p. 4807, Nov. 2019.

[86] S. E. Haupt *et al.*, "Blending distributed photovoltaic and demand load forecasts," *Sol. Energy*, vol. 157, pp. 542–551, Nov. 2017.

[87] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. 2015.

[88] C. Wang, G. Grozev, and S. Seo, "Decomposition and statistical analysis for regional electricity demand forecasting," *Energy*, vol. 41, no. 1, pp. 313–325, May 2012.

[89] C. N. Yu, P. Mirowski, and T. K. Ho, "A Sparse Coding Approach to Household Electricity Demand Forecasting in Smart Grids," *IEEE Trans. Smart Grid*, 2017.

[90] M. K. Azad, S. Uddin, and M. Takruri, "Support vector regression based electricity peak load forecasting," in *11th International Symposium on Mechatronics and its Applications, ISMA 2018*, 2018, vol. 2018-Janua, pp. 1–5.

[91] N. Korovesis, D. Kandris, G. Koulouras, and A. Alexandridis, "Robot motion control via an eeg-based brain–computer interface by using neural networks and alpha brainwaves," *Electron.*, 2019.

[92] P. Lusis, K. R. Khalilpour, L. Andrew, and A. Liebman, "Short-term residential load forecasting: Impact of calendar effects and forecast granularity," *Appl. Energy*, vol. 205, pp. 654–669, Nov. 2017.

[93] N. Mughees, S. A. Mohsin, A. Mughees, and A. Mughees, "Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting," *Expert Syst. Appl.*, 2021.

[94] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No Free Lunch Theorem: A Review," in *Approximation and Optimization : Algorithms, Complexity and Applications*, I. C. Demetriou and P. M. Pardalos, Eds. Cham: Springer International Publishing, 2019, pp. 57–82.

[95] O. Ahmia and N. Farah, "Multi-model approach for electrical load forecasting," *IntelliSys 2015 - Proc. 2015 SAI Intell. Syst. Conf.*, pp. 87–92, Dec. 2015.

[96] P. Zeng, M. Jin, and M. D. Fazla Elahe, "Short-term power load forecasting based on cross multi-model and second decision mechanism," *IEEE Access*, vol. 8, pp. 184061–184072, 2020.

[97] C. Peng, Y. Tao, Z. Chen, Y. Zhang, and X. Sun, "Multi-source transfer learning guided ensemble LSTM for building multi-load forecasting," *Expert Syst. Appl.*, vol. 202, p. 117194, Sep. 2022.

[98] S. Li, Y. Zhong, and J. Lin, "AWS-DAIE: Incremental Ensemble Short-Term Electricity Load Forecasting Based on Sample Domain Adaptation," *Sustain. 2022, Vol. 14, Page 14205*, vol. 14, no. 21, p. 14205, Oct. 2022.

[99] P. S. G. De Mattos Neto *et al.*, "Energy Consumption Forecasting for Smart Meters Using Extreme Learning Machine Ensemble," *Sensors 2021, Vol. 21, Page 8096*, vol. 21, no. 23, p. 8096, Dec. 2021.

[100] Y. Hu *et al.*, "Short-term load forecasting using multimodal evolutionary algorithm and random vector functional link network based ensemble learning," *Appl. Energy*, vol. 285, p. 116415, Mar. 2021.

[101] Q. Duan, J. Liu, and D. Zhao, "Short term electric load forecasting using an automated system of model choice," *Int. J. Electr. Power Energy Syst.*, vol. 91, pp. 92–100, Oct. 2017.

[102] Q. Chen, W. Zhang, K. Zhu, D. Zhou, H. Dai, and Q. Wu, "A novel trilinear deep residual network with self-adaptive Dropout method for short-term load forecasting," *Expert Syst. Appl.*, 2021.

[103] R. Gao, L. Du, P. N. Suganthan, Q. Zhou, and K. F. Yuen, "Random vector functional link neural network based ensemble deep learning for short-term load forecasting," *Expert Syst. Appl.*, vol. 206, p. 117784, Nov. 2022.

[104] M. Saviozzi, S. Massucco, and F. Silvestro, "Implementation of advanced functionalities for Distribution Management Systems: Load forecasting and modeling through Artificial Neural Networks ensembles," *Electr. Power Syst. Res.*, vol. 167, pp. 230–239, Feb. 2019.

[105] Z. Tan, J. Zhang, Y. He, Y. Zhang, G. Xiong, and Y. Liu, "Short-Term Load Forecasting Based on Integration of SVR and Stacking," *IEEE Access*, vol. 8, pp. 227719–227728, 2020.

[106] M. Zulfiqar, K. A. A. Gamage, M. Kamran, and M. B. Rasheed, "Hyperparameter Optimization of Bayesian Neural Network Using Bayesian Optimization and Intelligent Feature Engineering for Load Forecasting," *Sensors 2022, Vol. 22, Page 4446*, vol. 22, no. 12, p. 4446, Jun. 2022.

[107] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, 1963.

[108] A. Alexandridis, E. Chondrodima, and H. Sarimveis, "Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 2, pp. 219–230, 2013.

[109] L. Sun, K. Zhou, X. Zhang, and S. Yang, "Outlier Data Treatment Methods Toward Smart Grid Applications," *IEEE Access*, vol. 6, no. c, pp. 39849–39859, 2018.

[110] P. Martín, G. Moreno, F. J. Rodríguez, J. A. Jiménez, and I. Fernández, "A Hybrid Approach to Short-Term Load Forecasting Aimed at Bad Data Detection in Secondary Substation Monitoring Equipment," *Sensors 2018, Vol. 18, Page 3947*, vol. 18, no. 11, p. 3947, Nov. 2018.

[111] Y. Wang *et al.*, "Short-term load forecasting of industrial customers based on SVMD and XGBoost," *Int. J. Electr. Power Energy Syst.*, vol. 129, p. 106830, Jul. 2021.

[112] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 109, pp. 470–479, Jul. 2019.

[113] U. Javed *et al.*, "A novel short receptive field based dilated causal convolutional network integrated with Bidirectional LSTM for short-term load forecasting," *Expert Syst. Appl.*, vol. 205, p. 117689, Nov. 2022.

[114] M. M. Rashidi, M. Alhuyi Nazari, I. Mahariq, and N. Ali, "Modeling and Sensitivity Analysis of Thermal Conductivity of Ethylene Glycol-Water Based Nanofluids with Alumina Nanoparticles," *Exp. Tech.*, 2023.

[115] B. Zhou *et al.*, "Multi-energy net load forecasting for integrated local energy systems with heterogeneous prosumers," *Int. J. Electr. Power Energy Syst.*, vol. 126, p. 106542, Mar. 2021.

[116] M. Hosenuzzaman, N. A. Rahim, J. Selvaraj, M. Hasanuzzaman, A. B. M. A. Malek, and A. Nahar, "Global prospects, progress, policies, and environmental impact of solar photovoltaic power generation," *Renew. Sustain. Energy Rev.*, vol. 41, pp. 284–297, 2015.

[117] Verband der Elektrotechnik Elektronik Informationstechnike.V, "Erzeugungsanlagen am Niederspannungsnetz Technische Mindestanforderungen für Anschluss und Parallelbetrieb von Erzeugungsanlagen am Niederspannungsnetz," Berlin, Germany, 2010.

[118] E. Dall'Anese, S. V. Dhople, B. B. Johnson, and G. B. Giannakis, "Decentralized optimal dispatch of photovoltaic inverters in residential distribution systems," *IEEE Trans. Energy Convers.*, vol. 29, no. 4, pp. 957–967, 2014.

[119] Y. P. Agalgaonkar, B. C. Pal, and R. A. Jabr, "Voltage Control in Distribution System Considering the Impact of PV Generation using MPPT Controller on Tap Changers and Autonomous Regulators," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 1–10, 2016.

[120] P. Li, Z. Wu, K. Meng, G. Chen, and Z. Y. Dong, "Decentralized optimal reactive power dispatch of optimally partitioned distribution networks," *IEEE Access*, vol. 6, pp. 74051–74060, 2018.

[121] B. Zhao, Z. Xu, C. Xu, C. Wang, and F. Lin, "Network Partition-Based Zonal Voltage Control for Distribution Networks with Distributed PV Systems," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4087–4098, 2018.

[122] M. Mehdinejad, B. Mohammadi-Ivatloo, R. Dadashzadeh-Bonab, and K. Zare, "Solution of optimal reactive power dispatch of power systems using hybrid particle swarm optimization and imperialist competitive algorithms," *Int. J. Electr. Power Energy Syst.*, vol. 83, pp. 104–116, 2016.

[123] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application," *Swarm Evol. Comput.*, vol. 63, no. February, p. 100868, 2021.

[124] M. Baranwal and S. Salapaka, "Clustering and supervisory voltage control in power systems," *Int. J. Electr. Power Energy Syst.*, vol. 109, no. January, pp. 641–651, 2019.

[125] E. Cotilla-Sanchez, P. D. H. Hines, C. Barrows, S. Blumsack, and M. Patel, "Multi-attribute partitioning of power networks based on electrical distance," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4979–4987, 2013.

[126] M. Emarati, M. Barani, H. Farahmand, J. Aghaei, and P. C. del Granado, "A two-level over-voltage control strategy in distribution networks with high PV penetration," *Int. J. Electr. Power Energy Syst.*, vol. 130, no. October 2020, p. 106763, 2021.

[127] C. Luo, H. Wu, Y. Zhou, Y. Qiao, and M. Cai, "Network partition-based hierarchical decentralised voltage control for distribution networks with distributed PV systems," *Int. J. Electr. Power Energy Syst.*, vol. 130, no. 5, pp. 4087–4098, 2021.

[128] A. Alexandridis, E. Chondrodima, and H. Sarimveis, "Cooperative learning for radial basis function networks using particle swarm optimization," *Appl. Soft Comput. J.*, vol. 49, pp. 485–497, 2016.

[129] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 26113, Feb. 2004.

[130] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, no. September 2017, pp. 87–111, 2018.

[131] Test Feeder Working Group, "IEEE PES AMPS DSAS," *http://sites.ieee.org/pestestfeeders/ resources*. .

[132] K. C. Tan, T. H. Lee, Y. J. Yang, and D. S. Liu, "A cooperative coevolutionary algorithm for multiobjective optimization," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[133] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, 2009.

[134] V. Den Bergh and A. P. Engelbrecht, "Cooperative Learning in Neural Networks using Particle Swarm Optimizers," *South African Comput. J.*, vol. 26, pp. 84–90, 2000.

[135] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, 2004.

[136] K. P. Schneider *et al.*, "Analytic Considerations and Design Basis for the IEEE Distribution Test Feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, 2018.

[137] M. Erik, H. Pedersen, and M. E. H. Pedersen, "Good parameters for particle swarm optimization," *Tech. Rep. HL1001, Hvass Lab.*, vol. HL1001, pp. 1–12, 2010.

[138] A. Rezaee Jordehi and J. Jasni, "Parameter selection in particle swarm optimisation: A survey," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 25, no. 4. Taylor & Francis, pp. 527–542, 2013.

[139] M. Ławryńczuk and P. Tatjewski, "Offset-free state-space nonlinear predictive control for Wiener systems," *Inf. Sci. (Ny).*, vol. 511, pp. 127–151, 2020.

[140] M. Ławryńczuk, "Constrained computationally efficient nonlinear predictive control of Solid Oxide Fuel Cell: Tuning, feasibility and performance," *ISA Trans.*, vol. 99, pp. 270–289, 2020.

[141] M. Ławryńczuk, "Nonlinear predictive control of a boiler-turbine unit: A state-space approach with successive on-line model linearisation and quadratic optimisation," *ISA Trans.*, vol. 67, pp. 476–495, 2017.

[142] A. Alexandridis, H. Sarimveis, and K. Ninos, "A Radial Basis Function network training algorithm using a non-symmetric partition of the input space - Application to a Model Predictive Control configuration," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 830–837, 2011.

[143] S. Kumarawadu and T. T. Lee, "Neuroadaptive combined lateral and longitudinal control of highway vehicles using RBF networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 500–512, 2006.

[144] H. Mirzaeinejad, "Robust predictive control of wheel slip in antilock braking systems based on radial basis function neural network," *Appl. Soft Comput. J.*, vol. 70, pp. 318–329, 2018.

[145] A. Alleyne and J. K. Hedrick, "Nonlinear adaptive control of active suspensions," *IEEE Trans. Control Syst. Technol.*, vol. 3, no. 1, pp. 94–101, 1995.

[146] S. Fergani, O. Sename, and L. Dugard, "An LPV Hinfin; Integrated vehicle dynamic controller," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 1880–1889, 2016.

[147] R. Ding, R. Wang, X. Meng, W. Liu, and L. Chen, "Intelligent switching control of hybrid electromagnetic active suspension based on road identification," *Mech. Syst. Signal Process.*, vol. 152, p. 107355, 2021.

[148] W. Sun, H. Pan, and H. Gao, "Filter-Based Adaptive Vibration Control for Active Vehicle Suspensions with Electrohydraulic Actuators," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4619–4626, 2016.

[149] T. Yoshimura, Y. Isari, Q. Li, and J. Hino, "Active suspension of motor coaches using skyhook damper and fuzzy logic control," *Control Eng. Pract.*, vol. 5, no. 2, pp. 175–184, 1997.

[150] A. Aldair and W. J. Wang, "Adaptive neuro fuzzy inference controller for full vehicle nonlinear active suspension systems," *EPC-IQ01 2010 - 2010 1st Int. Conf. Energy, Power Control*, vol. 1, no. 4, pp. 97–106, 2010.

[151] J. O. Pedro, M. Dangor, O. A. Dahunsi, and M. M. Ali, "Dynamic neural network-based feedback linearization control of full-car suspensions using PSO," *Appl. Soft Comput. J.*, vol. 70, pp. 723–736, 2018.

[152] J. O. Pedro, M. Dangor, O. A. Dahunsi, and M. M. Ali, "Intelligent feedback linearization control of nonlinear electrohydraulic suspension systems using particle swarm optimization," *Appl. Soft Comput. J.*, vol. 24, pp. 50–62, 2014.

[153] W. Wang, Y. Song, Y. Xue, H. Jin, J. Hou, and M. Zhao, "An optimal vibration control strategy for a vehicle's active suspension based on improved cultural algorithm," *Appl. Soft Comput. J.*, vol. 28, pp. 167–174, 2015.

[154] C. Pilbeam and R. S. Sharp, "Performance potential and power consumption of slow-active suspension systems with preview," *Veh. Syst. Dyn.*, vol. 25, no. 3, pp. 169–183, 1996.

[155] Y. Hu, *Theory and technology of laser imaging based target detection*. Springer, 2017.

[156] M. Stogiannos, A. Alexandridis, and H. Sarimveis, "Model predictive control for systems with fast dynamics using inverse neural models," *ISA Trans.*, vol. 72, pp. 161–177, 2018.

[157] A. Schindler, "Neue Konzeption und erstmalige Realisierung eines aktiven Fahrwerks mit Preview-Strategie," Karlsruher Institut für Technologie, 2009.

[158] H. E. Merritt, *Hydraulic control system*. John Wiley & Sons, 1973.

[159] W. R. Garrott, "Measured vehicle inertial parameters-NHTSA's data through September 1992," *SAE Tech. Pap.*, vol. 1, no. November 1998, 1993.

[160] C. Göhrle, A. Schindler, A. Wagner, and O. Sawodny, "Road Profile Estimation and Preview Control for Low-Bandwidth Active Suspension Systems," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2299–2310, 2015.

[161] C. Glennie, "Rigorous 3D error analysis of kinematic scanning LIDAR systems," *J. Appl. Geod.*, vol. 1, no. 3, pp. 147–157, 2008.

[162] F. Tyan, Y.-F. Hong, S.-H. Tu, and W. S. Jeng, "Generation of random road profiles," *J. Adv. Eng.*, vol. 4, no. 2, pp. 151–156, 2009.

[163] A. L. Do, O. Sename, and L. Dugard, "LPV modeling and control of semi-active dampers in automotive systems," in *Control of Linear Parameter Varying Systems with Applications*, vol. 9781461418, J. Mohammadpour and C. Scherer, Eds. Springer, Boston, MA, 2012, pp. 381–411.

[164] K. Worthmann, M. Reble, L. Grüne, and F. Allgöwer, "The role of sampling for stability and performance in unconstrained nonlinear model predictive control," *SIAM J. Control Optim.*, vol. 52, no. 1, pp. 581–605, 2014.

[165] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, Mar. 1980.

[166] International Organization for Standardization, *ISO 2631-5:2018*, 2nd ed. 2018.

[167] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics, Volume 1*. SAE International, 1995.

[168] N. Ahmadian, A. Khosravi, and P. Sarhadi, "Integrated model reference adaptive control to coordinate active front steering and direct yaw moment control," *ISA Trans.*, vol. 106, pp. 85–96, 2020.

[169] M. Choi and S. B. Choi, "Model predictive control for vehicle yaw stability with practical

concerns," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 3539–3548, 2014.

[170] A. Tahouni, M. Mirzaei, and B. Najjari, "Novel Constrained Nonlinear Control of Vehicle Dynamics Using Integrated Active Torque Vectoring and Electronic Stability Control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9564–9572, 2019.

[171] M. B. Zaman, E. Kobayashi, N. Wakabayashi, S. Khanfir, T. Pitana, and A. Maimun, "Fuzzy FMEA model for risk evaluation of ship collisions in the Malacca Strait: Based on AIS data," *J. Simul.*, vol. 8, no. 1, pp. 91–104, 2014.

[172] J. Zhang, D. Zhang, X. Yan, S. Haugen, and C. Guedes Soares, "A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs," *Ocean Eng.*, vol. 105, pp. 336–348, 2015.

[173] R. Puisa, L. Lin, V. Bolbot, and D. Vassalos, "Unravelling causal factors of maritime incidents and accidents," *Saf. Sci.*, vol. 110, no. August, pp. 124–141, 2018.

[174] IMO, "COLREG: Convention on the International Regulations for Preventing Collisions at Sea, 1972," London, UK, 2003.

[175] L. Hu *et al.*, "A multiobjective optimization approach for COLREGs-Compliant path planning of autonomous surface vehicles verified on networked bridge simulators," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1167–1179, 2020.

[176] L. P. Perera, J. P. Carvalho, and C. Guedes Soares, "Intelligent ocean navigation and fuzzy-Bayesian decision/action formulation," *IEEE J. Ocean. Eng.*, vol. 37, no. 2, pp. 204–219, 2012.

[177] X. Zhang, C. Wang, K. T. Chui, and R. W. Liu, "A Real-Time Collision Avoidance Framework of MASS Based on B-Spline and Optimal Decoupling Control," *Sensors MDPI*, 2021.

[178] T. Wang, Q. Wu, J. Zhang, B. Wu, and Y. Wang, "Autonomous decision-making scheme for multi-ship collision avoidance with iterative observation and inference," *Ocean Eng.*, vol. 197, no. November 2019, p. 106873, 2020.

[179] A. P. Aguiar and A. M. Pascoal, "Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents," *Int. J. Control*, vol. 80, no. 7, pp. 1092–1108, 2007.

[180] C. Caldwell, D. Dunlap, and E. Collins, "Motion planning for an autonomous underwater vehicle via sampling based model predictive control," in *Oceans 2010 MTS/IEEE*, vol. 670–671, pp. 1370–1377.

[181] S. D. and S. F. S.Taherian, K.Halder, “Autonomous Collision Avoidance Using MPC with LQR-Based Weight Transformation,” *Sensors MDPI*, 2021.

[182] L. Chen, H. Hopman, and R. R. Negenborn, “Distributed model predictive control for vessel train formations of cooperative multi-vessel systems,” *Transp. Res. Part C Emerg. Technol.*, vol. 92, no. April, pp. 101–118, 2018.

[183] H. Zheng, R. R. Negenborn, and G. Lodewijks, “Robust Distributed Predictive Control of Waterborne AGVs - A Cooperative and Cost-Effective Approach,” *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2449–2461, 2018.

[184] K. D. von Ellenrieder, “Stable Backstepping Control of Marine Vehicles with Actuator Rate Limits and Saturation ⁎,” *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 262–267, 2018.

[185] S. Haykin, *Neural Networks: A Comprehensive Foundation (3rd Edition)*, vol. 13, no. 4. Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©2007, 1999.

[186] H. Zhou, Y. Chen, and S. Zhang, “Ship Trajectory Prediction Based on BP Neural Network,” *J. Artif. Intell.*, vol. 1, no. 1, pp. 29–36, 2019.

[187] T. Xu, X. Liu, and X. Yang, “Ship trajectory online prediction based on BP neural network algorithm,” in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences, ICM 2011*, 2011, vol. 1, pp. 103–106.

[188] S. Ma, S. Liu, and X. Meng, “Optimized BP neural network algorithm for predicting ship trajectory,” in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020*, 2020, pp. 525–532.

[189] X. Liu, W. He, J. Xie, and X. Chu, “Predicting the Trajectories of Vessels Using Machine Learning,” in *2020 5th International Conference on Control, Robotics and Cybernetics, CRC 2020*, 2020, pp. 66–70.

[190] W. Li, C. Zhang, J. Ma, and C. Jia, “Long-term vessel motion predication by modeling trajectory patterns with AIS data,” in *ICTIS 2019 - 5th International Conference on Transportation Information and Safety*, 2019, pp. 1389–1394.

[191] M. Ding, W. Su, Y. Liu, J. Zhang, J. Li, and J. Wu, “A Novel Approach on Vessel Trajectory Prediction Based on Variational LSTM,” in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2020*, 2020, pp. 206–211.

[192] P. Tampakis *et al.*, “Sea Area Monitoring and Analysis of Fishing Vessels Activity: The i4sea Big Data Platform,” in *IEEE International Conference on Mobile Data Management*, 2020, pp. 275–280.

[193] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Prediction of Vessel Trajectories from AIS Data Via Sequence-To-Sequence Recurrent Neural Networks,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8936–8940.

[194] C. Wang, H. Ren, and H. Li, “Vessel trajectory prediction based on AIS data and bidirectional GRU,” in *2020 International Conference on Computer Vision, Image and Deep Learning, CVIDL 2020*, 2020, pp. 260–264.

[195] A. Tritsarolis, E. Chondrodima, P. Tampakis, and A. Pikrakis, “Online Co-movement Pattern Prediction in Mobility Data,” Feb. 2021.

[196] G. Zhu, J. Du, and Y. Kao, “Robust adaptive neural trajectory tracking control of surface vessels under input and output constraints,” *J. Franklin Inst.*, vol. 357, no. 13, pp. 8591–8610, Sep. 2020.

[197] C. Zhang, C. Wang, Y. Wei, and J. Wang, “Robust trajectory tracking control for underactuated autonomous surface vessels with uncertainty dynamics and unavailable velocities,” *Ocean Eng.*, vol. 218, p. 108099, Dec. 2020.

[198] C. Li, Y. Zhao, G. Wang, Y. Fan, and Y. Bai, “Adaptive RBF neural network control for unmanned surface vessel course tracking,” in *6th International Conference on Information Science and Technology, ICIST 2016*, 2016, pp. 285–290.

[199] M. Stogiannos, A. Alexandridis, and H. Sarimveis, “Model predictive control for systems with fast dynamics using inverse neural models,” *ISA Trans.*, vol. 72, pp. 161–177, Jan. 2018.

[200] H. Akima, “A Method of Bivariate Interpolation and Smooth Surface Fitting Based on Local Procedures,” *Commun. ACM*, vol. 17, no. 1, pp. 18–20, Jan. 1974.

[201] J. Zhang, X. Yan, X. Chen, L. Sang, and D. Zhang, “A novel approach for assistance with anti-collision decision making based on the International Regulations for Preventing Collisions at Sea,” *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.*, vol. 226, no. 3, pp. 250–259, 2012.

[202] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Norway: Wiley, 2011.

[203] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo, “Nonlinear MPC for Tracking Piece-Wise Constant Reference Signals,” *IEEE Trans. Automat. Contr.*, vol. 63, no. 11, pp. 3735–3750, 2018.

[204] M. Zhu, A. Hahn, Y. Q. Wen, and A. Bolles, “Identification-based simplified model of large container ships using support vector machines and artificial bee colony algorithm,” *Appl. Ocean Res.*, vol. 68, pp. 249–261, 2017.

[205] A. Alexandridis, H. Sarimveis, and K. Ninos, “A Radial Basis Function network training algorithm using a non-symmetric partition of the input space - Application to a Model Predictive Control configuration,” *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 830–837, Oct. 2011.

[206] R. H. Byrd, J. C. Gilbert, and J. Nocedal, “A trust region method based on interior point techniques for nonlinear programming,” *Math. Program. Ser. B*, vol. 89, no. 1, pp. 149–185, Nov. 2000.

[207] R. Hult, M. Zanon, S. Gros, and P. Falcone, “Energy-Optimal Coordination of Autonomous Vehicles at Intersections,” *2018 Eur. Control Conf. ECC 2018*, pp. 602–607, 2018.

[208] A. Alessandretti, “Continuous-time Model Predictive Control for Economic Optimization: Theory, Design, and Applications to Motion Control of Underactuated Vehicles,” no. March, 2016.

[209] I. Kalogeropoulos, A. Alexandridis, and H. Sarimveis, “Economic Oriented Dynamic Matrix Control of Wastewater Treatment Plants,” *J. Process Control*, vol. 118, pp. 202–217, 2022.

[210] S. Gros and M. Zanon, “Learning for MPC with stability & safety guarantees,” *Automatica*, vol. 146, p. 110598, 2022.

[211] M. Zanon and S. Gros, “Safe Reinforcement Learning Using Robust MPC,” *IEEE Trans. Automat. Contr.*, vol. 66, no. 8, pp. 3638–3652, 2021.

[212] S. Gros and M. Zanon, “Data-driven economic NMPC using reinforcement learning,” *IEEE Trans. Automat. Contr.*, vol. 65, no. 2, pp. 636–648, 2020.

[213] IMO, “2nd IMO Greenhouse Gas Study,” London, UK, 2009.

[214] C. Gkerekos, I. Lazakis, and G. Theotokatos, “Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study,” *Ocean Eng.*, vol. 188, no. June, p. 106282, 2019.

[215] F. Tillig, J. W. Ringsberg, H. N. Psaraftis, and T. Zis, “Reduced environmental impact of marine transport through speed reduction and wind assisted propulsion,” *Transp. Res. Part D Transp. Environ.*, vol. 83, no. May, p. 102380, 2020.

[216] H. Yu, Z. Fang, X. Fu, J. Liu, and J. Chen, “Literature review on emission control-based ship voyage optimization,” *Transp. Res. Part D Transp. Environ.*, vol. 93, no. March, p.

102768, 2021.

[217] R. Zaccone, E. Ottaviani, M. Figari, and M. Altosole, "Ship voyage optimization for safe and energy-efficient navigation: A dynamic programming approach," *Ocean Eng.*, vol. 153, no. March 2017, pp. 215–224, 2018.

[218] J. Wang, H. Han, Y. Yang, and M. Wei, "An improved model predictive control approach for fuel efficiency optimization of vessel propulsion systems," *Control Eng. Pract.*, vol. 109, no. August 2020, p. 104749, 2021.

[219] F. Zhao, W. Yang, W. W. Tan, W. Yu, J. Yang, and S. K. Chou, "Power management of vessel propulsion system for thrust efficiency and emissions mitigation," *Appl. Energy*, vol. 161, no. 2016, pp. 124–132, 2016.

[220] T. Faulwasser, L. Grüne, and M. A. Müller, "Economic nonlinear model predictive control," *Found. Trends Syst. Control*, vol. 5, no. 1, pp. 224–409, 2018.

[221] M. Zanon and T. Faulwasser, "Economic MPC without terminal constraints: Gradient-correcting end penalties enforce asymptotic stability," *J. Process Control*, vol. 63, pp. 1–14, 2018.

[222] E. Ruth, Ø. N. Smogeli, T. Perez, and A. J. Sørensen, "Antispin thrust allocation for marine vessels," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 6, pp. 1257–1269, 2009.

[223] T. Faulwasser, M. Korda, C. N. Jones, and D. Bonvin, "On turnpike and dissipativity properties of continuous-time optimal control problems," *Automatica*, vol. 81, pp. 297–304, 2017.

[224] J. H. Seo, C. M. Lee, J. W. Yu, J. E. Choi, and I. Lee, "Power increase and propulsive characteristics in regular head waves of KVLCC2 using model tests," *Ocean Eng.*, vol. 216, no. December 2019, p. 108058, 2020.

[225] R. T. Poulsen, M. Viktorelius, H. Varvne, H. B. Rasmussen, and H. von Knorring, "Energy efficiency in ship operations - Exploring voyage decisions and decision-makers," *Transp. Res. Part D Transp. Environ.*, vol. 102, no. December 2021, pp. 1–29, 2022.

[226] H. N. Psaraftis and C. A. Kontovas, "Speed models for energy-efficient maritime transportation: A taxonomy and survey," *Transp. Res. Part C Emerg. Technol.*, vol. 26, pp. 331–351, 2013.

[227] H. N. Psaraftis and C. A. Kontovas, "Ship speed optimization: Concepts, models and combined speed-routing scenarios," *Transp. Res. Part C Emerg. Technol.*, vol. 44, pp. 52–69, 2014.

[228] N. Bhati, U. K. Kalla, B. Singh, and A. K. Mishra, “An Intelligent Control Scheme for Optimum Efficiency and Reduced Emission Operation of Marine Transportation System,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17107–17118, 2022.

[229] G. Zhu, Y. Ma, Z. Li, R. Malekian, and M. Sotelo, “Event-Triggered Adaptive Neural Fault-Tolerant Control of Underactuated MSVs With Input Saturation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7045–7057, 2022.

[230] L. Hewing, A. Liniger, and M. N. Zeilinger, “Cautious NMPC with Gaussian Process Dynamics for Autonomous Miniature Race Cars,” in *European Control Conference (ECC)*, 2018.

[231] J. M. Kvale, “Revised simulation model for a Very Large Crude Carrier (VLCC),” *Nor. Univ. Sci. Technol.*, no. June, 2014.

[232] M. Zanon, S. Gros, and A. Bemporad, “Practical reinforcement learning of stabilizing economic MPC,” *2019 18th Eur. Control Conf. ECC 2019*, pp. 2258–2263, 2019.

[233] M. Zanon, S. Gros, and M. Diehl, “A tracking MPC formulation that is locally equivalent to economic MPC,” *J. Process Control*, vol. 45, pp. 30–42, 2016.

[234] R. D. Geertsma, R. R. Negenborn, K. Visser, M. A. Loonstijn, and J. J. Hopman, “Pitch control for ships with diesel mechanical and hybrid propulsion: Modelling, validation and performance quantification,” *Appl. Energy*, vol. 206, no. October, pp. 1609–1631, 2017.

[235] N. Planakis, G. Papalambrou, and N. Kyrtatos, “Predictive power-split system of hybrid ship propulsion for energy management and emissions reduction,” *Control Eng. Pract.*, vol. 111, no. September 2020, p. 104795, 2021.

[236] J. Holtrop, “Statistical re-analysis of resistance and propulsion data,” *Int. Shipbuild. Prog.*, vol. 31, no. 363, pp. 272 – 276, 1984.

[237] R. D. Geertsma, R. R. Negenborn, K. Visser, and J. J. Hopman, “Torque control for diesel mechanical and hybrid propulsion on naval vessels,” 2016.

[238] J. Devanney, “The impact of the energy efficiency design index on very large crude carrier design and CO2 emissions,” *Ships Offshore Struct.*, vol. 6, no. 4, pp. 355–368, 2011.

[239] J. De Schutter, M. Zanon, and M. Diehl, “TuneMPC-A Tool for Economic Tuning of Tracking (N)MPC Problems,” *IEEE Control Syst. Lett.*, vol. 4, no. 4, pp. 910–915, 2020.

[240] Z. Chen, L. Li, X. Hu, B. Yan, and C. Yang, “Temporal-Difference Learning-Based Stochastic Energy Management for Plug-in Hybrid Electric Buses,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2378–2388, 2019.

[241] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[242] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, “qpOASES: a parametric active-set algorithm for quadratic programming,” *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.

[243] A. B. Martinsen, A. M. Lekkas, and S. Gros, “Combining system identification with reinforcement learning-based MPC,” *IFAC-PapersOnLine*, vol. 53, pp. 8130–8135, 2020.

[244] A. B. Martinsen, A. M. Lekkas, and S. Gros, “Reinforcement learning-based NMPC for tracking control of ASVs: Theory and experiments,” *Control Eng. Pract.*, vol. 120, no. August 2021, p. 105024, 2022.

[245] S. Gros and M. Diehl, “NMPC based on Huber penalty functions to handle large deviations of quadrature states,” *Proc. Am. Control Conf.*, pp. 3159–3164, 2013.

Appendix I: Multi-agent Simulation Framework

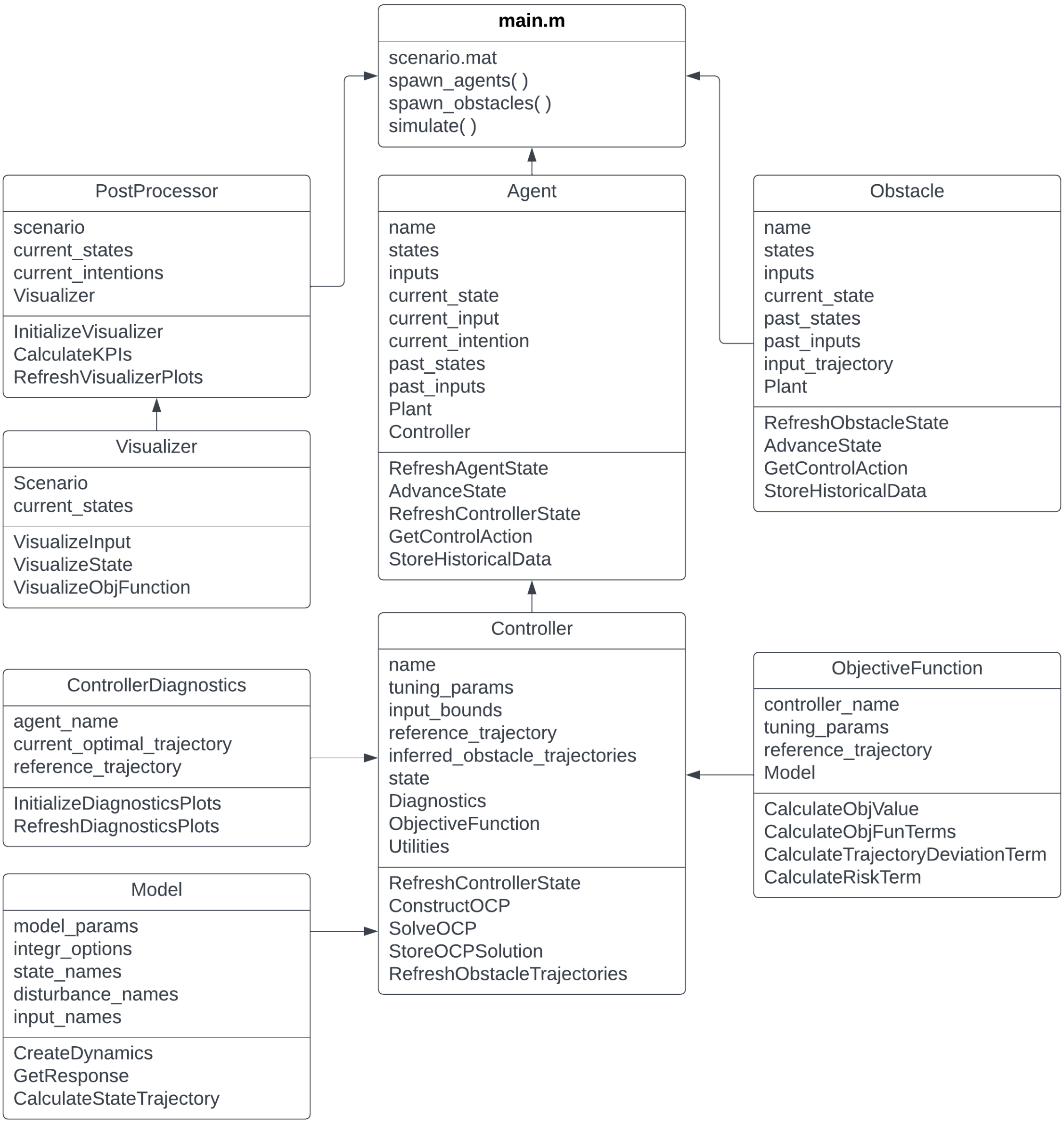


Figure 8.1 Simplified class diagram of the multi-agent simulation framework

This multi-agent simulation framework was built on MATLAB using standard packages. Oriented as a research and development platform for multiagent MPC scenarios, the main objectives were modularity, scalablity, as well as debugging & diagnostics tools for visualizing controller output and agent intention. Fig. 8.1 shows the simplified class diagram of the simulation framework. Access to code repository available upon request to the author.

Appendix II: RL-MPC Simulation Framework

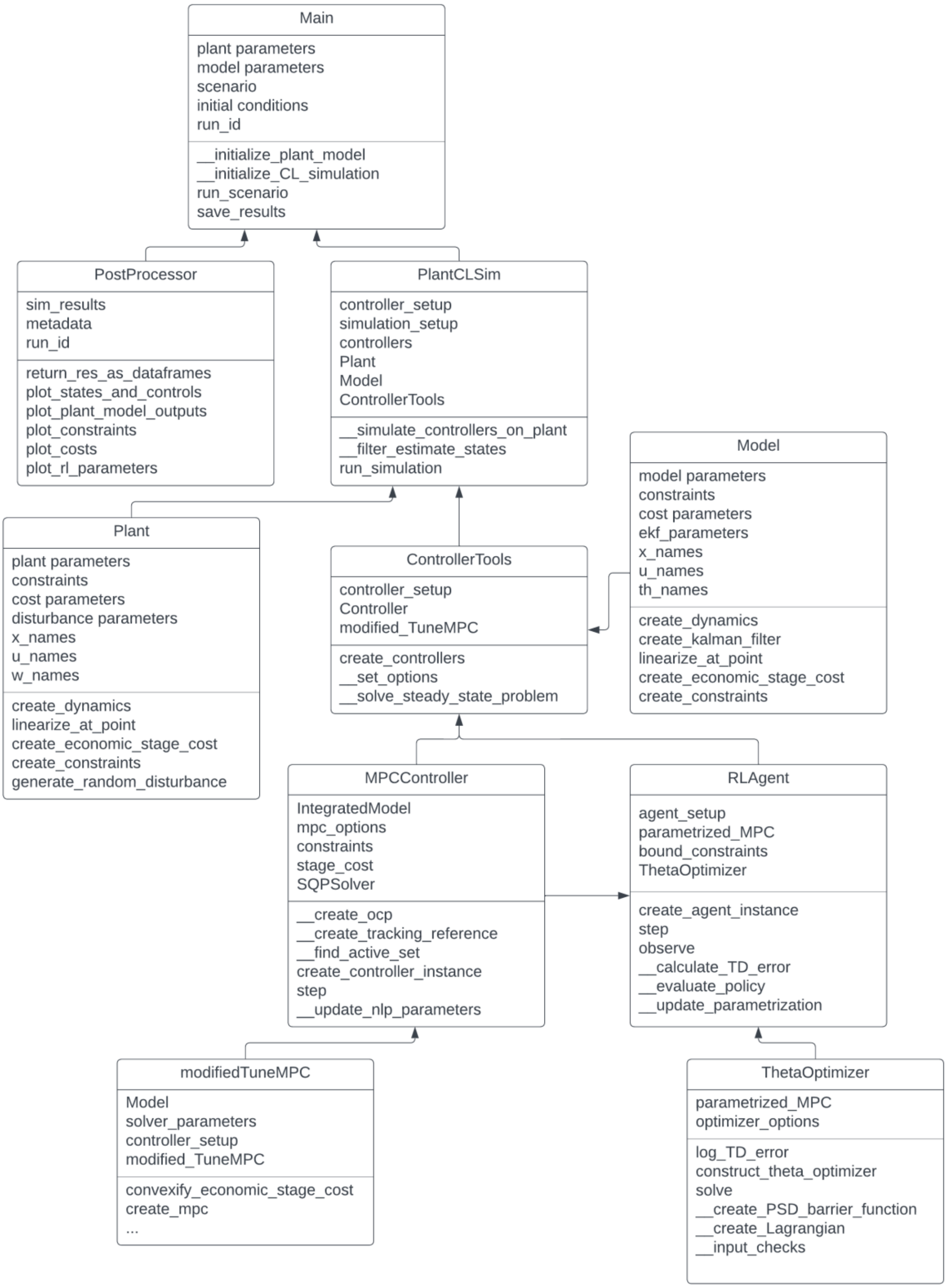


Figure 8.2 Simplified class diagram of the RL-MPC simulation framework

This RL-MPC simulation framework based on CasADi was built on Python by leveraging the capabilities of the tuneMPC package [239]. Main objectives were the standardization of system and OCP description and the employment of cutting-edge solvers & MPC development packages such as PICOS, ACADO, and others. Fig. 8.2 shows a simplified class diagram of the simulation framework. Access to the code repository available upon request to the author.