
Preface

Placeholder text

Erlend Hestvik, 20.12.2021

Abstract

placeholder text

Contents

Preface	i
Abstract	iii
List of Figures	vii
1 Introduction	1
1.1 Motivation	1
1.2 Previous Work	1
1.3 Problem Description	1
1.4 Contributions	1
1.5 Outline	1
1.6 Abbreviations	1
2 Background	3
2.1 Target Ship prediction	3
2.2 ASV modelling	3
2.3 Trajectory Planning	3
2.4 Collision Avoidance	4
2.5 'The complete system'	4
3 Trajectory planner	5
3.1 Dataflow	5
3.2 Setup	6
3.2.1 Simplify Prediction	6
3.2.2 COLREGs assessment	6
3.2.3 Dynamic Horizon	8
3.2.4 CasADi setup	8
3.2.5 Feasibility check	8
3.2.6 Reference from LOS	8
3.3 NLP construction and solver	8
3.3.1 NLP initialization	8
3.3.2 Integration step	8
3.3.3 Dynamic Obstacles constraints	8
3.3.4 Static Obstacles constraints	8
3.3.5 Solver	8

3.4	Alternative ideas and lessons	8
4	Simulation and Results	9
4.1	Situation overview	9
4.2	Simulation Results	10
4.3	Discussion	10
5	Conclusion and Future Work	11
	References	12

List of Figures

1 Introduction

Placeholder text. this is a placeholder citation to remove an error: Eriksen and Breivik 2017.

- hvorfor er det så vanskelig å skrive introduksjon.

1.1 Motivation

- Mye samme som på fordypningsoppgaven.

1.2 Previous Work

- skulle jo helst skrevet masse her.

1.3 Problem Description

- COLREGs-awareness.
- Trajectory planning.
- Target Ship prediction.
- NLP runtime optimization

1.4 Contributions

- Analyse av fordeler med å ha bedre / avansert prediksjon av TS.
-

1.5 Outline

- Samma stil som på fordypningsoppgaven.

1.6 Abbreviations

- abbreviations skal være en EGEN BLOKK i samme stil som list of figures.
- Tenkte det kunne vært lurt å ha en handy liste over alle forkortelser

-
- Selv med denne listen vil jeg fortsatt skrive forkortelser fullstendig ut første gang de brukes.

2 Background

- Husk rød tråd.
- Vær generisk.
- Bare inkluder konsept som blir relevante senere, eller som er brukt i nødvendige antagelser.

2.1 Target Ship prediction

- Gjenfortelling fra fordypningsprosjekt, da kalt traffic pattern
- Fant en annen artikkel fra Kina som skrev om nogenlunde det samme, AIS data -i, prediksjon
- Skiller seg fra fordypningsprosjekt fordi det er egentlig ikke traffic pattern som er den viktige antagelsen, Det er heller viktig at vi antar det finnes en måte å gjette/vite hvor andre båter vil være fremover i tid.
- Andre metoder for target ship prediction kan være f.eks utvidelse av AIS som inkluderer autonav data for de neste 5 minuttene eller noe lignende.

2.2 ASV modelling

Jeg tenker det er best å skrive om modellering i sammenheng med hvordan trajectory planning problemet blir satt opp i MATLAB med CasADi.

- Kinematics & Kinetics -i Begge brukes i CasADi setup
- Her kan det også skrives om de spesifikke tallverdiene som blir brukt i Masse, coriolis og dempnings -matrisene. de er spesifikke til Milliampere, funnet gjennom en rekke forsøk utført av Anders Pedersen.

2.3 Trajectory Planning

- How to get from A to B.
- Multiple methods, all with pros and cons, skriv liten oversikt.

LOS, OCP, Machine Learning, osv.

Kanskje ikke så veldig viktig å snakke om andre metoder enn OCP.

-
- Important factors to consider:

Time horizon / length of planning period.

Trajectory safety with respect to ship capabilities.

COLREGs compliance with respect to expected behaviour.

osv.

- Litt dypere inn i numerisk optimalisering og MPC, og LOS ettersom det kommer til å bli brukt igjen senere.

2.4 Collision Avoidance

- COLREGs

Expected behaviour, situation classification, etc etc.

- dCPA / tCPA

- Other risk assessment? Situation complexity? Det er mer som inngår i "collisions avoidance" som jeg kanskje ikke dekker så veldig bra med min algoritme.

2.5 'The complete system'

- Vet ikke helt om dette kapitlet er nødvendig, men jeg lurte på om det er en god ide å skrive litt om nøyaktig hvor i ett fult funksjonelt system jeg forventer at min algoritme passer inn. Hva de andre delene jeg ikke kommer til å skrive om har ansvar for, og hva som forventes av systemene rundt mitt eget.
- Hvis systemet mitt var en sort boks, hvilke inputs og outputs ville det hatt.

3 Trajectory planner

- Tidligere kjent som 'Method'.
- Har lyst å skrive litt om tankegangen bak utviklingen, ikke bare om hvordan ting endte opp med å bli.
- Ingen 'Preliminaries', alt av forkunnskaper og antagelser burde vært gjort rede for i 'Background'.
- Spesifikt mitt arbeid.
- Tar det fra start til slutt.

Persistent variables & settings.

COLREGs assessment.

Dynamic Horizon.

Casadi setup (generer F)

Feasibility check.

Initial conditions and Reference LOS guidance.

NLP init.

Main loop, med alt som skjer der.

Solve NLP, give output.

- Bit for bit, forklar hva, hvorfor, hvordan, eventuelt andre versjoner eller ideer som ble prøvd.
- forklar informasjonsflyt, kanskje som eget delkapittel.

3.1 Dataflow

- Begin by explaining the idea behind how the algorithm should work.
- This chapter will need diagrams.

input \rightarrow ??? \rightarrow output

show how the internal functions parse data

- Serves as a good overview of the whole algorithm.

3.2 Setup

- All the stuff before main loop.
- subsection for each 'block' as outlined by the dataflow.
- when the trajectory planner is called we need to run through some calculations before constructing the NLP problem
- These calculations are a mix of situation analysis, simulation settings, and CasADi initialization.
- Some of these calculations could be redundant in a complete control and navigation system, where other modules of the system would calculate the same thing.
- It's also important to remember that the value of many parameters are just guesswork, many of the subfunctions would benefit from a more sophisticated design that are tuned based on the situation the vessel finds itself in.

3.2.1 Simplify Prediction

This part of the setup is only required in simulations, the aim is to emulate the 'standard' way target ship (TS) prediction is conducted, which is to say constant course and velocity [Citation needed]. The TS trajectory is changed so that the first waypoint is the current position of the ship, and the next waypoint is one nautical mile in the direction of the ships heading. Ideally course over ground would be used instead of heading, however in the simulator crab angle and sideslip are not accounted for, therefore heading and course are the same angle. Excess waypoints stored in the TS struct are also truncated and the current waypoint index is forcefully set to 1 to prevent index out of range type errors.

3.2.2 COLREGs assessment

The COLREGs assessment function solves two problems; figuring out if when a TS vessel will be in close enough proximity that evasive maneuvers might be considered, and deciding which of the COLREGs rules will apply for the encounter. The design idea is to first find what the distance at closest point (dCPA) of approach with the TS is, and then time until closest point of approach (tCPA) occurs. If both dCPA and tCPA values are under a set threshold we consider the encounter an active event and run the COLREGs

Algorithm 1 getCPAlist

Input: *Agent1.Agent2* ▷ Agent is a struct that includes path waypoints

- 1: $dCPAlist \leftarrow []$
- 2: $tCPAlist \leftarrow []$
- 3: $pos_OS_list \leftarrow []$
- 4: $pos_TS_list \leftarrow []$
- 5: $timer \leftarrow 0$ ▷ Initialize timer used to calculate position of Agent2
- 6: **for** $i \leftarrow Agent1.current_wp : agent_wplist_length - 1$ **do**
- 7: $[pos_{OS}, vel_{OS}] \leftarrow VesselReadout(Agent1, i)$ ▷ VesselReadout explained in algorithm...
- 8: $DisttonextWP \leftarrow$ Distance to Agent1's next waypoint
- 9: $TimetonextWP \leftarrow DisttonextWP \div Agent1's \text{ speed over ground}$
- 10: $[pos_{TS}, vel_{TS}] \leftarrow whereisTS(Agent2, Timer)$ ▷ whereisTS explained in algorithm...
- 11: $[dCPA, tCPA] \leftarrow$ Equation for dCPA & tCPA as shown by...
- 12: $tCPA \leftarrow tCPA + timer$ ▷ Add travel time to reach current wp
- 13: $timer \leftarrow timer + TimetonextWP$
- 14: $pos_OS_list \leftarrow [pos_OS_list, pos_{OS}]$ ▷ Append all values to respective list.
- 15: $pos_TS_list \leftarrow [pos_TS_list, pos_{TS}]$
- 16: $dCPAlist \leftarrow [dCPAlist, dCPA]$
- 17: $tCPAlist \leftarrow [tCPAlist, tCPA]$
- 18: $i \leftarrow i + 1$
- 19: **end for**
- 20: **return** $pos_OS_list, pos_TS_list, dCPAlist, tCPAlist$

situation assessment shown by [cite paper], COLREGs assessment is also explained in [fordypningsoppgaven].

Finding the dCPA and tCPA between two vessels with constant velocity and course is easily done with a formula as shown by [cite paper]. However with our wish for more advanced target ship prediction this formula is not sufficient on it's own. In order to achieve 'full coverage' of our intended path, and the projected path of the target ship, we must check the dCPA and tCPA starting at each and every waypoint in the path of both vessels. A helper function 'getCPAlist' is constructed to get the list of all dCPAs and their respective tCPAs when given two agent structs (AGENT STRUCTS MÅ FORKLARES I BACKGROUND) as inputs. To achieve full coverage the getCPAlist function is ran twice so that the perspective of each agent is considered.

3.2.3 Dynamic Horizon

3.2.4 CasADi setup

3.2.5 Feasibility check

3.2.6 Reference from LOS

3.3 NLP construction and solver

- inputs vessel, ref_trajectory, static_obs, dynamic_obs, F, settings, h, N, previous_w_opt.
- sub funksjoner
 - Dynamic Obs.
 - Static Obs.
 - step.
- output w_opt

3.3.1 NLP initialization

3.3.2 Integration step

3.3.3 Dynamic Obstacles constraints

3.3.4 Static Obstacles constraints

3.3.5 Solver

3.4 Alternative ideas and lessons

Burde kanskje heller gå under discussion, og igjen i future work.

- Change w0 based on previous solution runtime.
- Gamle versjoner av Static_obs.
- eksperimenter med feasibility check.
- Masse styr med COLREGs assessment, tcpa og dcpa.
- ipopt innstillinger.

4 Simulation and Results

- noen større scenarioer, noen enkle situasjoner. For å vise hvordan algoritmen oppfører seg i forskjellige situasjoner med varierende kompleksitet.
- Delkapittel for hver ”stor” scenario, et delkapittel for alle ’enkle’ situasjoner.
- Viktig å analysere både bra, dårlig, og uventet oppførsel.
- annen viktig sak som må diskuteres er hvor ’inconsistent’ oppførselen er, små endringer i scenario innstillinger gir store utslag på oppførselen vår.
- Se på forskjell i oppførsel mellom når vi har ’prediksjon’ av target ships og når vi bare antar fast kurs og hastighet.

4.1 Situation overview

- Havn

crossings, head-on, trangt med statiske hindringer, full blockade av veien vi skal ta.

kan variere stat posisjoner for å se endra flere forskjellige COLREGs situasjoner.

- ’Trondheimsfjord’

Større åpent hav, mange båter på kryss og tvers.

viser at båter som vi vet vi ikke kommer i nærheten av ikke påvirker oppførselen vår.

viser at vi kan tracke en referanse veldig godt.

- ’Skjærgård’

Litt i samme stil som ’Trondheimsfjord’, men flere små statiske hindringer.

viser fint hvordan små statiske hindringer fortsatt blir ’oppdaget’.

stor distanse → lang tidshorisont og hvordan det påvirker oppførselen vår.

- ’usynlig sving’

Traffikert område hvor ’all’ trafikken følger en spesifikk sving.

- enkle situasjoner:

Head-on, Give way, Stand on i ’åpent’ hav med bare et target ship.

med og uten sving inkludert, for prediksjons sammenligning.

4.2 Simulation Results

- 'Dårlig' resultat er fortsatt resultat

4.3 Discussion

- Hvorfor er viktigere en hva
- ikke overanalyser resultat, ikke dra ville konklusjoner.
- Hvis et resultat er mye verre enn forventet kan det godt være det er bugs.
- i tillegg til det resultatene viser kan jeg også skrive om det jeg kan se med debugging.

5 Conclusion and Future Work

- conclusion:

oppsummering, forklaring, avsluttende ord.

- future work:

(variabel) Cost funksjon

'grenseverdier', altså verdier som constraint størrelse, distanse fra statiske hindringer, verdier som egentlig burde tunes basert på situasjonen slik den er i øyeblikket.

plassering av dynamiske constraints.

bedre måte å gjøre COLREGs assessment (ikke bare skjekk waypoints slik jeg gjør).

generelt andre metoder jeg ville foreslått å prøve isteden for spaghettien jeg har kokt sammen.

References

Eriksen, H. Bjørn-Olav and Morten Breivik (2017). ‘MPC-based mid-level collision avoidance for ASVs using nonlinear programming’. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)* (Mauna Lani Bay Hotel). IEEE. Hawaii, USA, pp. 766–772.