Preface

Placeholder text

Erlend Hestvik, 20.12.2021

Abstract

test acronym for error control: Automatic Identification System (AIS)

Contents

List of Figures

Acronyms

AIS Automatic Identification System

COLREGs Convention on the International Regulations for Preventing Collisions at Sea

dCPA distance at Closest Point of Approach

DOF Degrees Of Freedom

NED North East Down

NLP NonLinear Programming

tCPA time to Closest Point of Approach

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.
- Trajecory 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.

2 Background

- Husk rød tråd.
- Vær generisk.
- Bare inkluder konsept som blir relevante senere, eller som er brukt i nødvendige antagelser.
- many of the concepts explained in this chapter were also explained in [TODO: fordypningsprosjekt].
- The goal of this chapter is to explain all the concepts neccessary to understand how the trajeoctyr planning algorithm works. Additionally, it should grant an understanding as to why the algorithm ended up the way it is.

2.1 Target Ship prediction

——— Genereal thoughts ———————

- Gjenfortelling fra fordypningsprosjekt, da kalt traffic pattern
- Fant en annen artikkel fra Kina som skrev om nogenlunde det samme, AIS data - 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.

———- /General Thoughts———————

- Naval navigation is an 'active' task, always looking out for obstacles and making sure the way forward is clear.
- There are no lanes, instead proper conduct is dictated by Convention on the International Regulations for Preventing Collisions at Sea (COLREGs), the rules are laid out so that different situations have different rules
- Knowing which situation you are in is half the battle, therefor the ability to predict and estiamte how encounters will happen is a powerful tool. Human navigators do this by experience.

- Autonomous vessels could also predict by experience, that would be the machine learning approach. AIS can already provide training data.
- But it could be easier than that, what if AIS data packets were expanded to send out autonav data for where a vessel intends to traverse. The assumption here is that vessels using autonav will correct their course when spotting a conflict. Vessels not using autonav would still be able to observe the indended path of other vessels and adjust their plans 'manually'.
- Relying on fully predicting the transit of a target vessel is fragile, relying on a few known waypoints would be much more robust.
- In the end the best solution would be if all vessels could fully communicate their intended path either peer to peer or through a centralized system. Solving conflicting paths would be done on a higher level than individual vessel.

2.2 Vessel modelling

——— Genereal thoughts ———————

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

———- /General Thoughts———————

- 3 Degrees Of Freedom (DOF) non-linear model with thurster dynamics, very common for designing control systems.
- The numerical values for the model will vary based on ship, and the complexity of the model change based on usage and the degrees of freedom included.
- The model used in this paper is based on Fossen's meneuvering theory with 3 degrees of freedom and no disturbances.

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.1}$$

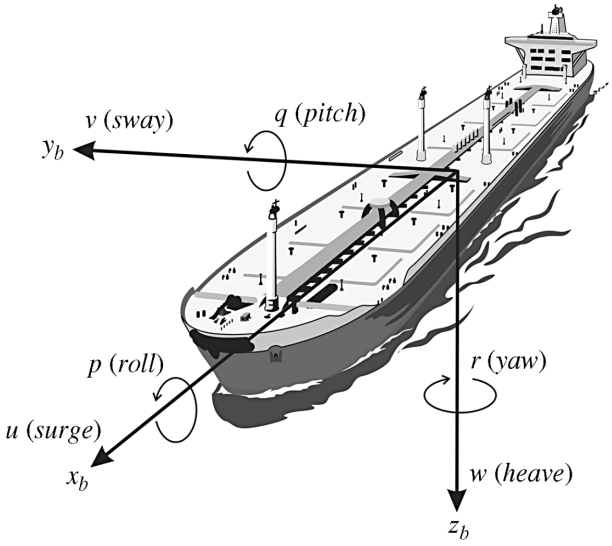

Figure 1: A ships 6 degrees of Freedom, from Fossen 2011

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{2.2}$$

$$\mathbf{R}(\psi) = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \tag{2.4}$$

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^T \tag{2.5}$$

$$\boldsymbol{\tau} = \begin{bmatrix} F_X & F_Y & F_N \end{bmatrix}^T \tag{2.6}$$

- η is defined in the North East Down (NED) reference frame, ν is defined in BODY frame, briefly explain what that means.
- The M and C matrices are a combination of Rigid body and hydrodynamic added mass matrices
- Fully computing the M and C matrices is a lot of work, for milliampere the work has already been conducted by Andreas Pedersen.
- If the model values are to be found through experiementation it's senseless to seperate Rigid body and hydrodynamic added mass. The matrices then become:

$$\mathbf{M} = \begin{bmatrix} m11 & m12 & m13 \\ m12 & m22 & m23 \\ m31 & m32 & m33 \end{bmatrix} \tag{2.7}$$

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & c13(\boldsymbol{\nu}) \\ 0 & 0 & c23(\boldsymbol{\nu}) \\ c31(\boldsymbol{\nu}) & c32(\boldsymbol{\nu}) & 0 \end{bmatrix} \tag{2.8}$$

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} d11(\boldsymbol{\nu}) & 0 & 0 \\ 0 & d22(\boldsymbol{\nu}) & d23(\boldsymbol{\nu}) \\ 0 & d32(\boldsymbol{\nu}) & d33(\boldsymbol{\nu}) \end{bmatrix} \tag{2.9}$$

- The D matrix can sometimes be a bit of a nuisance, a simplified D matrix can be employed to make the model more computationally efficient, at the cost of accuracy.

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} d11 & 0 & 0 \\ 0 & d22 & 0 \\ 0 & 0 & d33 \end{bmatrix} \tag{2.10}$$

2.3 Trajectory Planning

——— Genereal thoughts ———————

- 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.

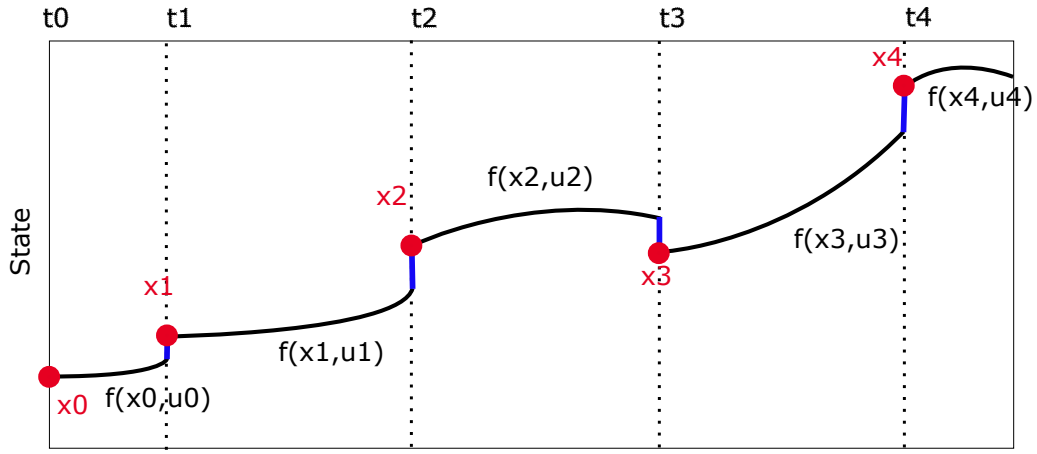

Figure 2: A physically feasible trajectory is formed by "pinching" the shooting gaps so they close. Reproduction of (TODO: CITER GROS), scale arbitrary.

——— /Genereal thoughts ———————————

$$\begin{aligned} \text{Minimize} \quad & \mathbf{F}(\boldsymbol{\eta}(t), \mathbf{u}(t)) && \text{(2.11a)} \\ \text{subject to} \quad & \dot{\boldsymbol{\eta}}(t) = \mathbf{L}(\boldsymbol{\eta}(t), \mathbf{u}(t)) && \text{(2.11b)} \\ & \mathbf{h}(\boldsymbol{\eta}(t), \mathbf{u}(t)) \leq \mathbf{0} && \text{(2.11c)} \\ & \boldsymbol{\eta}(t_0) = \boldsymbol{\eta}_0 && \text{(2.11d)} \end{aligned}$$

(TODO: REWRITE THIS, SOME PARTS MAKE LITTLE SENSE)

where $\mathbf{F}$ is the objective function, $\boldsymbol{\eta}(t)$ is the position and attitude trajectory of the vehicle, $\mathbf{u}(t)$ is the control input trajectory and $\mathbf{L}$ is the kinematic model of the vehicle. Though the OCP can be solved the way it is set up in (2.11) it is more pratical to discretize it into a (NonLinear Programming (NLP)). With a method called direct multiple shooting, both state and control input are defined as decision variables, the NLP with N control intervals is:

$$\begin{aligned} \min_{\boldsymbol{\omega}} \quad & \mathbf{F}(\boldsymbol{\omega}) && \text{(2.12a)} \\ \text{subject to} \quad & \boldsymbol{\omega}_{lb} \leq \boldsymbol{\omega} \leq \boldsymbol{\omega}_{ub} && \text{(2.12b)} \\ & \mathbf{g}_{lb} \leq \mathbf{g} \leq \mathbf{g}_{ub} && \text{(2.12c)} \end{aligned}$$

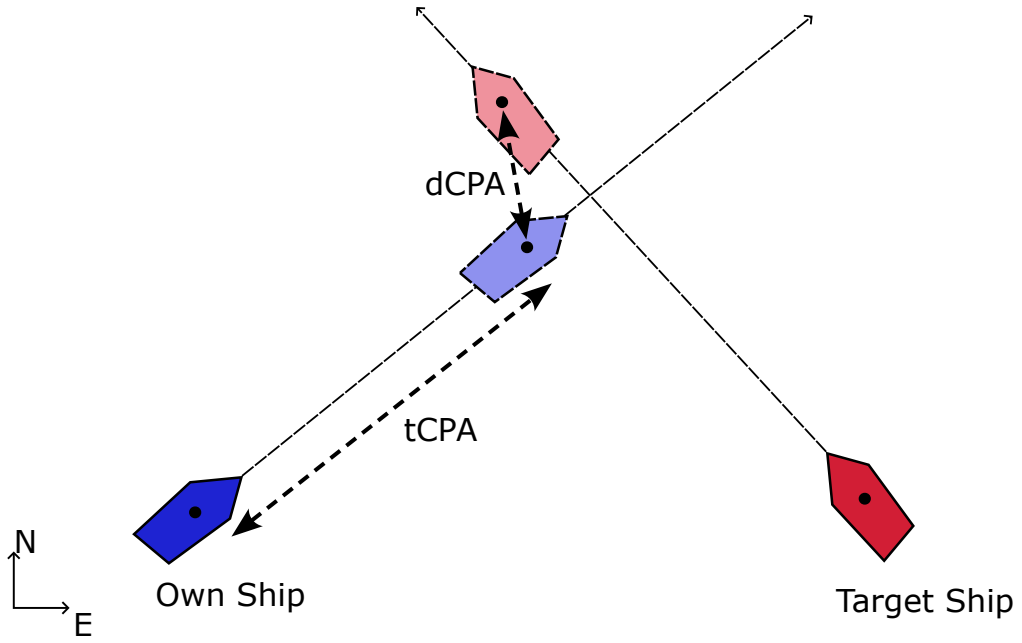

Figure 3: Visualizing dCPA and tCPA.

2.4 Collision Avoidance

- COLREGs

  Expected behaviour, situation classification, etc etc.

- distance at Closest Point of Approach (dCPA) / time to Closest Point of Approach (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 kapittelet er nødvendig, men jeg lurer 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.

2.6 Simulator setup

- liten forklaring av hvordan simulatoren funker selv om jeg ikke har laget den.

- f.eks agent structs kan bli viktig.

- Forklar at det er en liten forsjel på hvordan own ship agenten og andre agenter oppdateres, fører til en maksimal 'feil' på 1 meter.

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, eventuellt 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.

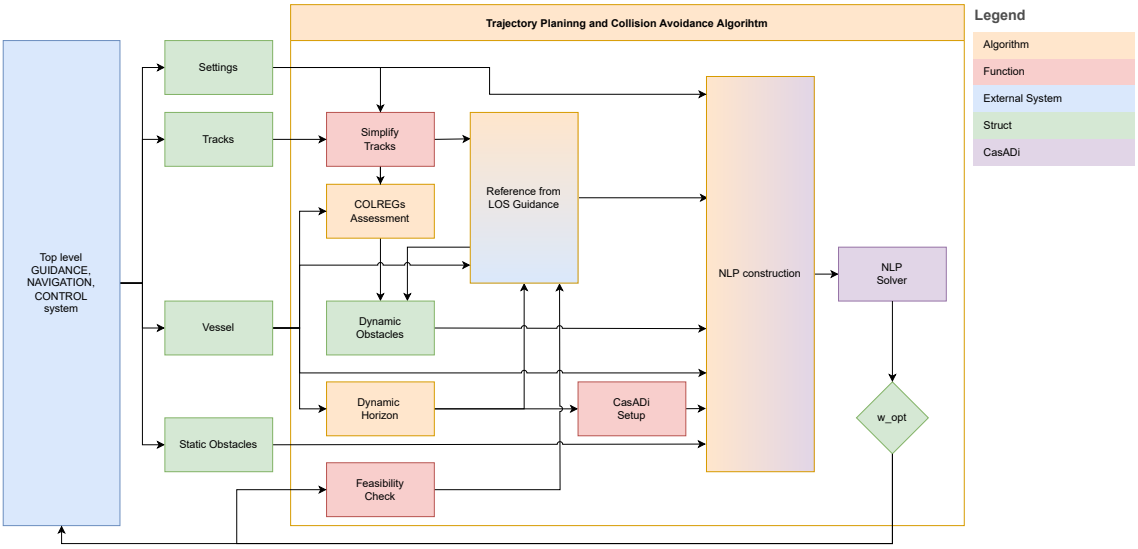

Figure 4: The simple version of the dataflow...

3.2 Setup

- All the stuff before main loop.
- subsubsection 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 calcualtions 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 sophesticated 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 [TODO: 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, therefor 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 ifwhen 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 cloest 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 situation assessment shown by [TODO: cite paper], COLREGs assessment is also explained in [TODO: fordypningsoppgaven].

Finding the dCPA and tCPA between two vessels with constant velocity and course is easily done with a formula as shown by(eller in?) Kufoalor et al. 2018.

$$t_{AB}^{CPA} = \begin{cases} \frac{\mathbf{P}_{BA} \cdot \mathbf{V}_{A|B}}{||\mathbf{V}_{A|B}||^2} & \text{if } ||\mathbf{V}_{A|B}|| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.1a}$$

$$d_{AB}^{CPA} = ||(\mathbf{P}_A + t_{AB}^{CPA}\mathbf{V}_A) - (\mathbf{P}_B + t_{AB}^{CPA}\mathbf{V}_B)|| \tag{3.1b}$$

Where $\mathbf{V}_{A|B} = \mathbf{V}_A - \mathbf{V}_B$ with $\mathbf{V}_A$, $\mathbf{V}_B$, $\mathbf{P}_A$ and $\mathbf{P}_B$ being the respective velocities and positions of vessel A and vessel B parameterized in NED. However if we are to utilize the advanced prediction we have on other agents a bit more logic must be applied to achieve full coverage of our intended path. Presume that our path contains a set of waypoints we intend to pass by, and similarly we know of a set of waypoints another agent intends to pass by. To find the true dCPA and tCPA between our own ship and the target vessel we use equations 3.1 as the situation is when each agent passes by one of their waypoints(TODO: TRENGER FIGUR FOR Å VISE BEDRE HVA JEG MENER). Speed is tougher to account for, unless we know better speed should be assumed to remain constant (TODO: dette hører hjemme i background).

TODO: Ikke ferdig

Algorithm 1 getCPAlist. Denne ble jævelig stygg, beholder den for synlighet

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

$dCPAlist \leftarrow []$
$tCPAlist \leftarrow []$
$pos\_OS\_list \leftarrow []$
$pos\_TS\_list \leftarrow []$
$timer \leftarrow 0$ ▷ Initialize timer used to calculate position of Agent2
for $i \leftarrow Agent1.current\_wp : agent\_wplist\_length - 1$ **do**
    $[pos_{OS}, vel_{OS}] \leftarrow VesselReadout(Agent1, i)$ ▷ VesselReadout explained in algorithm...
    $DisttonextWP \leftarrow$Distance to Agent1's next waypoint
    $TimetonextWP \leftarrow DisttonextWP \div$ Agent1's speed over ground
    $[pos_{TS}, vel_{TS}] \leftarrow whereisTS(Agent2, Timer)$ ▷ whereisTS explained in algorithm...
    $[dCPA, tCPA] \leftarrow$ Equation for dCPA & tCPA as shown by...
    $tCPA \leftarrow tCPA + timer$ ▷ Add travel time to reach current wp
    $timer \leftarrow timer + TimetonextWP$
    $pos\_OS\_list \leftarrow [pos\_OS\_list, pos_{OS}]$ ▷ Append all values to respective list.
    $pos\_TS\_list \leftarrow [pos\_TS\_list, pos_{TS}]$
    $dCPAlist \leftarrow [dCPAlist, dCPA]$
    $tCPAlist \leftarrow [tCPAlist, tCPA]$
    $i \leftarrow i + 1$
end for
return $pos\_OS\_list, pos\_TS\_list, dCPAlist, tCPAlist$

to get a list of all dCPA and tCPAs between two agents, as well as the corresponding positions of both agents as they are when the euqations 3.1 are used. getCPAlist

3.2.3 Dynamic Horizon

- Dynamic horizon is a balancing act between distance to goal, encompassing all active dynamic obstacles, and not looking to far ahead into the future.
- changing the dynamic horizon is really just changing how many control intervals we want the NLP to have.
- As the distance to goal approaches zero we want the number of control intervals to shrink accordingly, otherwise we end up with too many control intervals stationary at the goal, which can cause problems like the cost function becoming unbalanced.
- If there are active dynamic obstacles we need the dynamic horizon to encompass them.
- we don't want the dynamic horizon to be too short during transit (why not?).

3.2.4 CasADi setup

- sym x = [N, E, psi, u, v, r]
- sym tau as a free variable

- sym xref as reference
- model parameters
- M, C, D matrix
- xdot [nu_dot, eta_dot]
- Error in the correct reference frame.
- Why is the cost function the way it is.
- runge-kutta method.
- the final function F that CasADi needs.
- Noe av disse greiene blir dekt av Background, forhåpentligvis.

3.2.5 Feasibility check

- The feasibility check came from the wish to read out the status report CasADi prints to the command window.
- It's very important to know if the previous iteration of the trajectory planner function yielded a feasible result or not.
- if the result is not feasible the path forward might be completely blocked, in which case reducing our vessel speed is the best option.
- Very simple check, just checks if every point in the previously calculated optimal trajectory is within 5 meters of each other. This is very lenient and should of course change depending on vessel speed.

3.2.6 Reference from LOS

- I didn't write this)
- The important part is that the time discretization is consistent with the trajectory planner's time step
- You don't need to use LOS for reference.
- Position reference and speed reference need to be consitent with each other.

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.

  integration step.
- output w_opt

3.3.1 NLP initialization

- Initial conditions and end of interval coditions, we need the end of one control interval and the beginning of the next to match.
- Tror dette delkapittelet er litt unødvendig

3.3.2 Integration step

- Getting the correct reference
- make sure all the indexes are correct!
- put the references in w0, speeds up runtime significantly.

3.3.3 Dynamic Obstacles constraints

- When to place constraints
- Where to place them
- How to place them

3.3.4 Static Obstacles constraints

- Explain static_obstacles_check and the theory for convex-free set
- explain why circles, such as the ones used for dynamic obstacles are insufficient.
- this is sort of similar to finding a cross track error, if that helps to explain what is going on.

3.3.5 Solver

- Options, there are many options.
- things to try / were tried for optimizing runtime.
- CasADi really does all the hard work.

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 versioner 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 uvented 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 $\rightarrow$ 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 værre 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.

4.4 Improvements over previous version

- Definite improvements in terms of computational efficiency. This greatly increases the likelyhood of finding an optimal solution
- Because of the better efficiency the algorithm is also able to handle more control intervals, This means it is better at handling both greater time horizon and shorter control interval steps.
- The new method for handling static obstacles is much less prone to misplaced or inefficnet constraints. (her ta gjerne med figuren som viser problemer med sirkel constraints for statiske hindringer).
- The new way of handling dynamic constraints should in theory make the algorithm better suited for more complex situations with more agents, however the placement of dynamic constraints remains largely unchanged. Dynamic Constraint placement is bigger 'bottleneck' than agent culling for how complex situations are handled.
- More robust when an encounter leads to an infeasible solution.
- Improved COLREGs assessment
- But does it behave *noticeably* better?

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.

Fossen, Thor I (2011). *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons.

Kufoalor, D. K.M., E. F. Brekke and T. A. Johansen (2018). 'Proactive Collision Avoidance for ASVs using A Dynamic Reciprocal Velocity Obstacles Method'. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2402–2409. DOI: 10.1109/IROS.2018.8594382.