Make a simulation of trajectory tracking control system using a machine learning method, with the following scenarios:

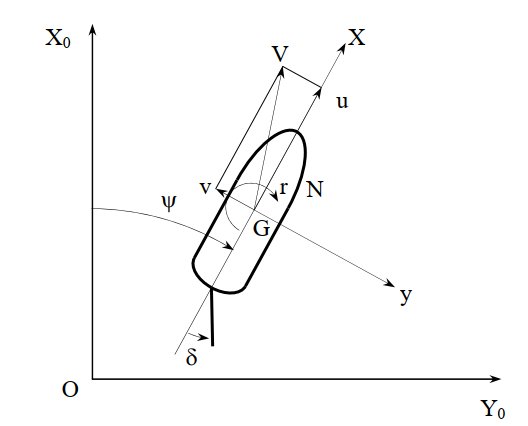
1. Autopilot
2. Speed control
3. Trajectory tracking control with a pattern (zigzag/lawn mowing)

To simulate these scenarios, develop a set of simulation with MATLAB (Problem 1) and Simulink (Problem 2) separately.

3 DOF:

* Motion in x-direction (surge)
* Motion in y-direction (sway)
* Rotational motion about z-axis (yaw)

Diagram of a ship with a diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Surge |  |  |
| Sway |  |  |
| Yaw |  |  |

|  |  |
| --- | --- |
| where | m : mass of ship [kg.s2/m] |
|  | : acceleration in x direction [m/s2] |
|  | : acceleration in y direction [m/s2] |
|  | : angular acceleration [rad/s2] |
|  | IZZ : inertia moment with respect to z-axis [kgs2m] |
|  | XH, YH : Hydrodynamic forces acting on ship’s hull [kg] |
|  | NH : Hydrodynamic moment acting on the ship’s hull [kg.m] |
|  | XP : Propulsive force of propeller [kg] |
|  | XR, YR : Hydrodynamic forces acting on ship’s rudder [kg] |
|  | NR : Hydrodynamic moment acting on ship’s rudder [kg.m] |
|  | YT : Hydrodynamic force induced by thruster [kg] |
|  | NT : Hydrodynamic moment induced by thruster [kg.m] |

The state equations of ship’s heading and position on the earth-fixed coordinate system:

|  |  |  |
| --- | --- | --- |
| Yaw rate |  |  |
| Position in x-axis |  |  |
| Position in y-axis |  |  |

1. Autopilot

* Primary control variable: yaw angle
* Goal: maintain a desired heading while moving at a constant speed
* Approach: the controller adjusts the rudder to correct the heading as disturbances push the vessel off course.

1. Speed control

* Primary control variable: speed
* Goal: maintain a target speed regardless of environmental disturbances
* Approach: the controller adjusts the thruster to keep the vessel’s speed steady

1. Trajectory tracking control

* Primary control variables: heading and speed
* Goal: follow a specific path or pattern to cover an area or reach specific waypoint
* Approach: The controller adjusts both heading and speed simultaneously to navigate the desired trajectory

Literature review

1. Deep reinforcement learning based controller for ship navigation (2023)

* The paper uses DQN to control a 3 DOF vessel (Krisco container ship)
* Tested with turning circle and zigzag
* 4 observation states: course-track error, course angle error, distance to destination, yaw rate.
* 3 discrete actions: rudder angle [-35o, 0, 35o]
* 3 reward functions: cross-track error, course angle error, distance to goal.
* Tested with a model scale vessel

1. Deep Reinforcement Learning Based Optimal Trajectory  
   Tracking Control of Autonomous Underwater Vehicle (2017)

* The authors use DDPG to control an AUV
* 3-DOF => similar to an ASV
* Compared with PID controller
* 2 test scenarios: straight line and curve line
* DRL higher accuracy and more stability than PID

1. Autonomous Surface Vehicle Control Method Using  
   Deep Reinforcement Learning (2020)

* 4 thruster ASV using DDPG
* Simulation in MATLAB with reinforcement learning toolbox
* Higher performance when compared to nonlinear model predictive control (NMPC) method in the fixed-point control experiment
* Have the ability of trajectory tracking

1. Control Design of a Marine Vessel System Using Reinforcement  
   Learning (2018)

* Use Policy Iteration (PI) to control a 3-DOF marine vessel system

X(1): u – surge velocity

X(2): v – sway velocity

X(3): r – yaw velocity

X(4): x – x-position

X(5): y – y-position

X(6): – yaw angle

X(7): p – roll velocity

X(8): – roll angle

X(9): – rudder angle

X(10): n – shaft velocity

Autopilot: input (rudder), output (yaw angle)

Speed control: input n (shaft velocity), output speed =

Trajectory tracking: input (, n), output (x, y, yaw)