TMR4243 - Marine Control Systems II

Case study D: Maneuvering

This exercise investigates maneuvering for CS Enterprise I (CSEI).

The assignment should be completed in groups and a short report is to be delivered on itslearning before a due date, which will be published on its learning. The short report encompasses a theoretical part and a processor-in-the-loop simulation session. Your group must pass the short report in order to be allowed to participate in the scale model testing. The scale testing in the Marine Cybernetics Laboratory will be conducted at the end of the semester. The results of all parts of this assignment must be included in the full report, which is to be delivered at the end of the semester. The nal report will be marked and a due date will be published on it's learning.

System model

Two models are used in this case:

- The kinematics model of the horizontal motion of a vessel,

$$\dot{\eta} = R(\psi)\nu, \tag{1}$$

  where $\eta := (p, \psi) \in \mathbb{R}^3$ and $p = (x, y) \in \mathbb{R}^2$ is the vessel position and heading vector, and the body-fixed velocity vector $\nu \in \mathbb{R}^3$ is viewed a control input. The rotation matrix is

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

  and note that $\dot{R} = R(\psi)S(r)$ where

$$S(r) = \begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -S(r)^{\top}.$$

- The complete C/S Enterprise I mathematical model in the MC-Lab handbook.

Problem statement

The maneuvering problem is comprised of the two tasks, in prioritized order:

Geometric task: For some absolutely continuous function $s\left(t\right)$, force the output η to converge to the desired parametrized path $\eta_d\left(s\right)$, that is,

$$\lim_{t\to\infty} \left|\eta\left(t\right) - \eta_d\left(s\left(t\right)\right)\right| = 0.$$

Dynamic task: Control $\dot{s}$ to converge to a desired speed assignment $U_s\left(s, t\right)$, that is,

$$\lim_{t\to\infty} \left|\dot{s}\left(t\right) - U_s\left(s\left(t\right), t\right)\right| = 0.$$

Part I

Theory

1 Task: Path parametrization

1. Straight-line
   (a) Propose a straight-line parametrization for $p_d(s) = (x_d(s), y_d(s))$ with:
      - initial position $p_d(0) = p_{d,0} = (x_0, y_0)$, and
      - final position $p_d(1) = y_{d,1} = (x_1, y_1)$.

   (b) Derive the tangential velocity vector $p_d^s(s) = (x_d^s(s), y_d^s(s))$.

   (c) Formulate the desired pose $\eta_d(s)$, including $p_d(s)$ together with 1) a constant reference heading $\psi_d(s) = \psi_{ref}$, and 2) desired heading $\psi_d(s)$ tangential to the path.

   (d) Derive expressions for $\eta_d^s(s)$ and $\eta_d^{s^2}(s)$ for the two cases.

   (e) Propose a speed assignment $U_s(s,t)$ for $\dot{s}$ that makes the vessel follow the path at constant reference speed U_{ref} [m/s]. Note that U_{ref} may take both positive and negative values for forward and backwards motion along the path.

2. Ellipsoidal path
   (a) Propose an ellipsoidal parametrization for $p_d(s) = (x_d(s), y_d(s))$ with:
      - center in (c_x, c_y), and
      - radii (r_x, r_y).

   (b) Derive the tangential velocity vector $p_d^s(s) = (x_d^s(s), y_d^s(s))$.

   (c) Formulate the desired pose $\eta_d(s)$, including $p_d(s)$ together with 1) a constant reference heading $\psi_d(s) = \psi_{ref}$, and 2) desired heading $\psi_d(s)$ tangential to the path.

   (d) Derive expressions for $\eta_d^s(s)$ and $\eta_d^{s^2}(s)$ for the two cases.

   (e) Propose a speed assignment $U_s(s,t)$ for $\dot{s}$ that makes the vessel follow the path at constant reference speed U_{ref} [m/s].

2 Task: Kinematic model control design

For the kinematic model, let $z_1 := R(\psi)^\top (\eta - \eta_d(s))$ be the position/heading error vector composed in the body-fixed reference frame.

1. Differentiate z_1 and show that the CLF

$$V_1(\eta, s) = \frac{1}{2} z_1^\top z_1 = \frac{1}{2} (\eta - \eta_d(s))^\top (\eta - \eta_d(s)) \tag{2}$$

   gives

$$\dot{V}_1 = z_1^\top (\nu - \eta_d^s(s)\dot{s}) \tag{3}$$

2. Assuming $\nu = \alpha_1$ is our control law, show that

$$\alpha_1(\eta, s, t) = -K_p z_1 + \eta_d^s(s) U_s(s, t), \qquad K_p = K_p^\top > 0 \tag{4}$$

   results for (2) in

$$\dot{V}_1 \leq -\lambda_{\min}(K_p) |z_1|^2 - V_1^s(\eta, s)(U_s(s, t) - \dot{s}).$$

3. Derive the expression for $V_1^s(\eta, s)$ for the two cases of the desired heading.

3 Task: Update laws

1. Tracking update law:
   (a) Show that
   $$\dot{s} = U_s(s,t)$$
   solves the Maneuvering Problem.
   (b) Explain the reason for the name of this update law.
2. Gradient update law:
   (a) Show that
   $$\dot{s} = U_s(s,t) - \mu V_1^s(\eta, s)$$
   where $\mu \geq 0$ is a gain, solves the Maneuvering Problem.
   (b) Explain the reason for the name of this update law.
3. Modified gradient update law:

   To account for the varying magnitude of $V_1^s(\eta, s)$ along the path, due to a possible nonlinear parametrization, a normalizing modification can be applied.

   (a) Show that
   $$\dot{s} = U_s(s,t) - \frac{\mu}{|\eta_d^s(s)|} V_1^s(\eta, s) \tag{5}$$
   where $\mu \geq 0$ is a gain, solves the Maneuvering Problem.
   (b) Explain mathematically how this normalizes $V_1^s(\eta, s)$ to rather use the unit tangent vector along the path.
4. Filtered gradient update law:

   Let $\omega_s := U_s(s,t) - \dot{s}$ be the speed assignment error and
   $$W_1(\eta, s, \omega_s) := V_1(\eta, s) + \frac{1}{2\lambda\mu}\omega_s^2$$
   be a new CLF (considering ω_s as an additional state in the system).

   (a) Show that
   $$\begin{aligned} \dot{s} &= U_s(s,t) - \omega_s \\ \dot{\omega}_s &= -\lambda\left(\omega_s - \mu V_1^s(\eta, s)\right) \end{aligned}$$
   where $\mu > 0$ is a gain, gives
   $$\dot{W}_1 \leq -\lambda_{\min}(K_p)\,|z_1|^2 - \frac{1}{\mu}\,|\omega_s|^2,$$
   and solves the Maneuvering Problem.
   (b) Explain the reason for the name of this update law.

4 Task: Kinematic model simulation

Implement the kinematic model (1), using control law (4) with gain $K_p = \text{diag}(0.2, 0.2, 0.1)$, and the update law (5) with gain $\mu \in \{0, 0.1, 1.0, 10\}$. Simulate the following cases:

1. DP along a straight line through $p_d(0) = (2, 0)$ and $p_d(1) = (10, 4)$, with initial conditions $p_d(s(0))$ and $p(0)$ set close to each other and $U_{ref} = 0.0$. Use the Tracking update law ($\mu = 0$).
   (a) Stay on DP with zero speed for some time, until steady state is achieved.
   (b) Move the setpoint $p_d(s(t))$ along the path a bit, by setting $U_{ref} \neq 0$, and then come to stop at a new location by again commanding $U_{ref} = 0.0$. Stay here until new steady-state is achieved.
2. Backward and forward motion along a line through $p_d(0) = (2, 0)$ and $p_d(1) = (10, 4)$, with constant speed $U_{ref} = \frac{U_{\max}}{5}$, and initial conditions $s(0) = 0$ and $p(0) = (6, 5)$.
   (a) Tracking update law, i.e. $\mu = 0$.
   (b) Modified gradient update law, with $\mu = 0.1$, then $\mu = 1.0$, and finally $\mu = 10$.
3. Backward and forward motion along a line as in Task 4.2, but with varying speed $U_{ref}(s) = \frac{U_{\max}}{5}\left(1.01 + \sin\left(2\pi s - \frac{\pi}{2}\right)\right)$.
   (a) Tracking update law i.e. $\mu = 0$.
   (b) Modified gradient update law, with $\mu = 0.1$, then $\mu = 1.0$, and finally $\mu = 10$.
4. Infinite ellipsoid motion centered at $(c_x, c_y) = (6, 0)$ with radii $(r_x, r_y) = (5, 3)$, with speed $U_{ref} = \frac{U_{\max}}{10}$, and initial conditions $s(0) = 0$ and $p(0) = (7, 0)$.
   (a) Tracking update law i.e. $\mu = 0$.
   (b) Modified gradient update law, with $\mu = 0.1$, then $\mu = 1.0$, and finally $\mu = 10$.

5 Task: DP maneuvering control design

In the previous sections we have designed and tested a maneuvering control algorithm for the kinematic model of the vessel. In this task, you shall design a backstepping-based control law for τ that uses the maneuvering control law (4) and solves the maneuvering objective for the full vessel model, using the modified gradient update law.

In this task we will NOT change the update law (5) to also include the z_2 dynamics, but rather we will keep the update law (5) as above, depending only on (η, s, t). This is achieved if the control law for τ compensates $\dot{\alpha}_1$ directly.

1. Let $z_2 = \nu - \alpha_1(\eta, s, t)$ be the error between ν and its desired control α_1, and calculate again $\dot{V}_1$ for (2) with z_2 and α_1 included.
2. Calculate $\dot{\alpha}_1$ and $\dot{z}_2$.
3. Let $V_2(\nu, \eta, s, t) = V_1(\eta, s, t) + \frac{1}{2} z_2^\top M z_2$ be the Step 2 CLF. Calculate $\dot{V}_2$.
4. Propose a control law for τ that renders $\dot{V}_2$ negative definite. Make sure that τ cancels $\dot{\alpha}_1$ directly (and not by $\sigma_1(\eta, s, t) + \alpha_1^s(\eta, s, t) U_s(s, t)$), so that the modified gradient update law (5) in previous task can be used as is.

Do not apply integral action directly in the control law. Instead, try to compensate for the unknown bias by using the estimated bias from the DP observer developed in Case C.

Note also that states should be free from noise. Otherwise the term $V_1^s(\eta, s)$ in (5) may make s jump around when μ is large and η is noisy. To avoid this, use the estimated $\hat{\eta}$ from the observer. One can also try to use the filtered gradient update law with cutoff frequency λ set appropriately to attenuate noise.

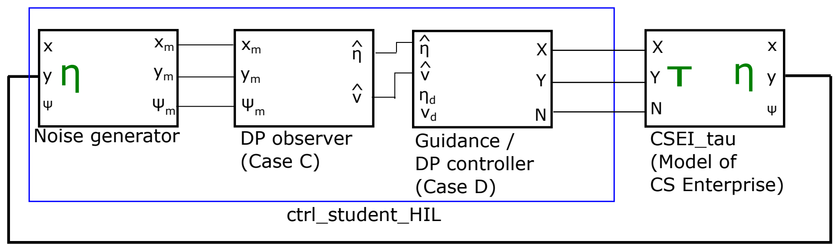

Figure 1: Simulation system structure

Part II

Simulation

Required file, available on GitHub: `CSE1_tau.out`, CSEI model

6 Task: Simulink simulation

1. Implement the control law together with the CSEI model.
2. Simulate DP along straight line path, as in Task 4.1. Tune DP gains with $\mu = 0$.
3. Simulate straight line maneuvering, as in Task 4.3. Tune μ.
4. Simulate ellipsoid maneuvering, as in Task 4.4. Tune μ.

7 Processor-in-the-loop simulation

1. Adjust and compile the control system (without the CSEI control plant model) from Tasks 5 and 6 for use on cRIO. Add input and output ports as necessary.
2. Set up a VeriStand simulation project connecting your control system to the provided CSEI model.
3. Develop a suitable graphical user interface.
4. Simulate scenarios corresponding to Task 6.
5. Tune controller gains online.

Part III

Scale model test

The scale model test will be conducted in the MClab in weeks 13,14 and 17.

1. Implement your controller into the CSEI VeriStand project.
2. Develop a suitable graphical user interface.
3. Test scenarios corresponding to Task 6.
4. Tune controller gains online.

Report requirements

The report should reflect that you have completed the tasks. Include necessary (but short) explanations, steps for derived expressions, calculations, diagrams (including Simulink block diagrams), plots (including relevant system states), user interface screenshots and code (including eventual initialization files and MATLAB functions as appendices) .

The report must be in English. The report should be delivered in portable document format (PDF).

Specifically for this case study, the report shall contain:

- 2D position plots of $p\left(t\right)$ and $p_d\left(s\left(t\right)\right)$ in the same figure.
- Time plots of heading $\psi(t)$ and $\psi_d(s(t))$ in same figure.
- Time-plots of $s\left(t\right)$.
- Time plots of path speed and speed assignment $\dot{s}(t)$ and $U_s(s(t), t)$ in same figure.
- Comparison of the tracking update law versus the modified gradient update law.

All plots, responses, and performance should be discussed.