1 Introduction

The motion of a submarine is influenced by the angles of several control surfaces (inputs) and the goal is to achieve desired motion along several degrees of freedom (outputs). Thus, submarine control requires a multiple-input multiple-output (MIMO or multivariable) control system. The mathematical description of a submarine is a set of nonlinear differential equations, or equivalently, a nonlinear state-space model. It is customary to linearize the model output about an operating point such as a constant-velocity trajectory, and to control deviations from this operating point. It may be necessary to design several linear control systems, each for a different velocity, and put them together with a gain-scheduling algorithm as is done in [3]. In this project, we consider only the design of a single linear multivariable digital tracking system.

2 Description of the Plant

The material from this section is taken from [1,2]. The linearized state-space model for the submarine is

\[ \dot{x} = Ax + Bu \]
\[ y = Cx \]

where the state variables are:

- \( x_1 = \) forward velocity, \( u \), ft/sec
- \( x_2 = \) lateral velocity, \( v \), ft/sec
- \( x_3 = \) vertical velocity, \( w \), ft/sec
- \( x_4 = \) roll rate, \( p \), deg/sec
- \( x_5 = \) pitch rate, \( q \), deg/sec
- \( x_6 = \) yaw rate, \( r \), rad/sec
- \( x_7 = \) roll angle, degrees
- \( x_8 = \) pitch angle, degrees.

The inputs are:

- \( u_1 = \) bow/fairwater planes, degrees
- \( u_2 = \) rudder deflection, degrees
- \( u_3 = \) port stern plane deflection, degrees
- \( u_4 = \) starboard stern plane deflection, degrees

The outputs are:

- \( y_1 = \) roll angle, degrees
- \( y_2 = \) pitch angle, degrees
- \( y_3 = \) yaw rate, deg/sec
- \( y_4 = \) depth rate, ft/sec.
The numerical values for the matrices $A, B, C$ are given in the Matlab function `plant_param.m`, which is available on the course website. The modeled submarine is about 400 ft long. The system variables are indicated in the following figure:

![Figure 1: Sketch showing positive directions of axes, angles, velocities, forces, and moments. From [2].](image)

### 3 Tracking System Design

We begin with the design and analysis of a full-state feedback tracking system and then consider using an observer to estimate the plant state vector from the input and output signals of the plant. **The sampling interval for this project is $T=0.1$.**

#### 3.1 Full-State Feedback

A tracking system is to be designed to follow step commands for each of the four plant outputs. Thus, the discrete-time additional dynamics must have an eigenvalue equal to 1 on each of the four tracking errors. This results in the additional dynamics block consisting of four parallel digital integrators, $\phi_a=\text{eye}(4)$, $\gamma_a=\text{eye}(4)$.

It is not uncommon for high-order control systems to exhibit more than one settling time. That is, different signals settle in different amounts of time. The control systems encountered in class this semester were all designed using a single settling time, $T_s$, which governed all of the state variables. In this project, most of the closed-loop poles may be chosen to have a setting time $T_{s1}$. However, to avoid actuator saturation, at least one of the closed-loop poles must be chosen with a longer settling time, $T_{s2}$. For the particular submarine considered in this project,
realistic values for the two settling times are:

\[ T_{S_1} = 25 \text{ sec} \quad \text{and} \quad T_{S_2} = 85 \text{ sec.} \quad (1) \]

Consider the following choice of closed-loop poles:

\[
\begin{align*}
    cp &= -0.1969 + j \times 0.3130 \\
    spoles &= [-0.5049 \quad -0.4511 \quad cp \quad \text{conj}(cp) \quad s7/Ts1 \quad s1/Ts2].
\end{align*}
\]

1. Explain how the choice of sampling interval given at the beginning of this section agrees with the rule-of-thumb given in class for choosing the sampling interval of a digital tracking system.

2. Explain how the choice of \( spoles \) given above agrees with the information in the Rules for Selecting Pole Locations handout. For each entry in the \( spoles \) vector explain which rule is being used.

3. Call \( tsd \) two times, once with \( \text{place} \) and a second time with \( \text{rfbg} \). Examine the stability robustness of each system and explain which of these systems has adequate robustness to be useful as a real-world submarine control system. Provide a printout of your Matlab code.

3.2 Observer-Based

[Unless otherwise specified, use the feedback gain matrix calculated using \( \text{rfbg} \) for the observer-based tracking systems in this section.] The pole-placement approach to the calculation of observer gains amounts to choosing the observer pole locations. However, when the observer uses more than one measured plant output, there are an infinite number of observer gain matrices that result in the specified observer pole locations. A given pole-placement program selects a particular gain matrix from this infinite set and that selection has an influence on the stability margins of the resulting observer-based tracking system. For this project, the following choice of observer pole locations has been found to work well:

\[
\begin{align*}
    opoles &= [-0.0384 \quad -0.251 \quad s6/(Ts1/5)].
\end{align*}
\]

4. Explain how this choice of observer pole locations agrees with the information in the Rules for Selecting Pole Locations handout. For each entry in the \( opoles \) vector explain which rule is being used.

5. Using the given \( opoles \) vector, calculate observer gain matrices using \( \text{place} \) and using a new function \( \text{obg} \_ts \), which is available on the course web site. Compare the stability robustness bounds for the resulting observer-based tracking systems. Compare these with each other and with the robustness bounds for the state-feedback tracking system from Part 2. Note that the function \( \text{rb} \_tsob \) calculates the stability robustness bounds for any observer-based tracking system. Provide a printout of your Matlab code.

6. Draw a block diagram of the complete observer-based tracking control system used for this project. Show all equations used to implement the digital tracking system, A/D and D/A converters, and a block containing the hardware plant.
4 Simulations

The performance of the observer-based tracking system is demonstrated by simulating a combined maneuver in which step commands for each plant output are applied simultaneously at $t = 5$ sec. The commands are as follows: roll angle ($y_1$) is to be maintained at 0 deg, pitch ($y_2$) is commanded to 1 deg, yaw rate ($y_3$) is commanded to 1 deg/sec, and depth rate ($y_4$) is commanded to -0.5 ft/sec.

The simulations for this section may be obtained using the Simulink model project_sim.slx, which is available on the course website. Note that the plots may be obtained with the plotting script ts_dobp.m. The simulations are to be performed for 200 sec. Put two separate graphs on each Matlab plot.

7. Consider the observer-based tracking system designed using rfbg and obg_ts with spoles and opoles given above. Compare the plant outputs and inputs with those shown in Figs. 2 and 3, which were obtained using an LQG/LTR control system in [1,2].

Suppose you were to do this project only with the standard pole-placement tool available in Matlab, which is the place function:

8. Calculate the feedback and observer gains using the place function with the given vectors for spoles and opoles. Compare robustness bounds of this system with those of the observer-based tracking system designed using the new functions rfbg and obg_ts. Provide a printout of your Matlab code.

9. Simulate the tracking system obtained using only place. Compare the output and input plots with those of the observer-based tracking system designed using rfbg and obg_ts. Do the simulation results for the place tracking system give any cause for concern? Is this tracking system suitable for hardware testing?

References


Figure 2: Plant output simulation results from [2] for the reference signals given above. The first graph is roll angle ($y_1$) and the vertical axis goes from -3 to +3 degrees. The second graph is pitch angle ($y_2$) and the vertical axis goes from -1 to +1 degree. The third graph is yaw rate ($y_3$) and the vertical axis goes from 0 to 1.6 deg/sec. The fourth graph is depth rate ($y_4$) and the vertical axis goes from -0.5 to +0.5 ft/sec.
Figure 3: Plant input simulation results from [2] for the reference signals given above. The first graph is $u_1$ and the vertical axis goes from -5 to +5 deg. The second graph is $u_2$ and the vertical axis goes from -6 to 0 deg. The third graph is $u_3$ and the vertical axis goes from -4 to 0 deg. The fourth graph is $u(4)$ and the vertical axis goes from -4 to +4 deg.