EL2700 - Assignment 2: Finite-time Optimal Control

Group 14: Chieh-Ju Wu and Charles Brinkley

September 14, 2021

1 Q1: System Model

The first task is to implement the open-loop and continuous, linearized plant model. In this scenario, 3-DoF are represented (unlike in the first assignment). These three degrees correspond to movement in the X-Y plane and the angle of rotation around the Z axis. The corresponding state space form describing this dynamic system is shown below and implemented the *cartesian_ground_dynamics* function.

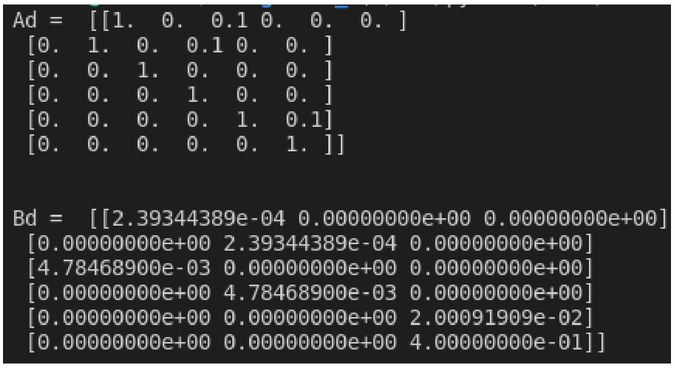
$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} u(t) \tag{1.1}$$

where:

$$\begin{aligned} x(t) &= [p_X \ p_Y \ v_X \ v_Y \ \theta \ \omega]^T \\ u(t) &= [f_X \ f_Y \ \tau_Z]^T \\ m \quad &= 20.9\, kg \\ I_z \quad &= .2517\, kgm^2 \end{aligned}$$

2 Q2: System Model (Discretized)

In the same manner as Assignment 1, the continuous time state space is discretized. This is again performed by sending the continuous matrices to the *casadi_c2d* function which uses the Jacobian derivations to derive the discrete values. The new discretized state space can be seen below in the printed terminal output.



```
Ad =  [[1.  0.  0.1 0.  0.  0. ]
 [0.  1.  0.  0.1 0.  0. ]
 [0.  0.  1.  0.  0.  0. ]
 [0.  0.  0.  1.  0.  0. ]
 [0.  0.  0.  0.  1.  0.1]
 [0.  0.  0.  0.  0.  1. ]]


Bd =  [[2.39344389e-04 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 2.39344389e-04 0.00000000e+00]
 [4.78468900e-03 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 4.78468900e-03 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 2.00091909e-02]
 [0.00000000e+00 0.00000000e+00 4.00000000e-01]]
```

Figure 1: Discretized A and B matrices describing system dynamics

3 Q3: Finite-Time Optimal Control - Initialization Method

In question 3, we began deriving the optimized system input. The objective function is an energy minimizer and provided in the form shown below (when written in discrete terms)...

$$\sum_{t=0}^{T-1} u_t^T R u_t \tag{3.1}$$

Where R is weighted cost matrix of $I_3 * 10$. In conjunction to this, we are given a negative and positive bounding equality constraint for each input variable (f_X f_Y τ_Z) that our objective function is subject to (in addition to the normal discretized ODE of $x_{t+1} = Ax_t + Bu_t$). With the provided code, the optimization is set to run over a 30 second span, which equates to 300 samples for a sampling period of .1 seconds. The desired rendevous time for Bumble with Honey is also set to 25 seconds.

This question lets us know that something is missing in regards to the the rendezvous constraints. It is apparent that the upper/lower bounded inequality tolerance constraints associated to the x, y, and θ position and velocity is missing. This is easily implemented mirroring the example for the 3D scenario below. A screenshot of said code is shown in figure 2 below.

```python
# State constraints for rendezvous
if t > self.Ntr:
    # Make sure that we target the reference then
    if ref_type == "2d":
        x_ref = x_t_ref[(r_i * self.Nr):(r_i * self.Nr + self.Nr)]

        """Question 3"""
        # TODO: use 'x_ref', 'x_t', 'self.pos_tol' and 'self.att_tol'
        #       to define the maximum error tolerance for the position
        #       in X, Y and angle theta, by adjusting 'con_ineq',
        #       'con_ineq_ub' and 'con_ineq_lb' - take inspiration from
        #       the example below for 3D
        con_ineq.append(x_ref[0:2] - x_t[0:2])
        con_ineq_ub.append(self.pos_tol)
        con_ineq_lb.append(-self.pos_tol)

        con_ineq.append(x_ref[2] - x_t[4])
        con_ineq_ub.append(self.att_tol)
        con_ineq_lb.append(-self.att_tol)
```

Figure 2: Code screenshot depicting rendevous inequality constraint additions

The first 3 lines of appending updates the x and y position/velocity variance and tolerance at each iteration after the rendevous time (25 sec). Similarily, the last 3 lines deals with the angular motion around the Z-axis as defined by the θ and ω state variables.

4 Q4: Finite-Time Optimal Control - Solving and Plotting

By filling in the inequality constraints of the quadratic functions, we can ensure that Bumble reaches Honey from its initial condition to x_r ([0.1,0,0,0,0.01,0]). This successful rendezvous can be seen in figure 3 below.

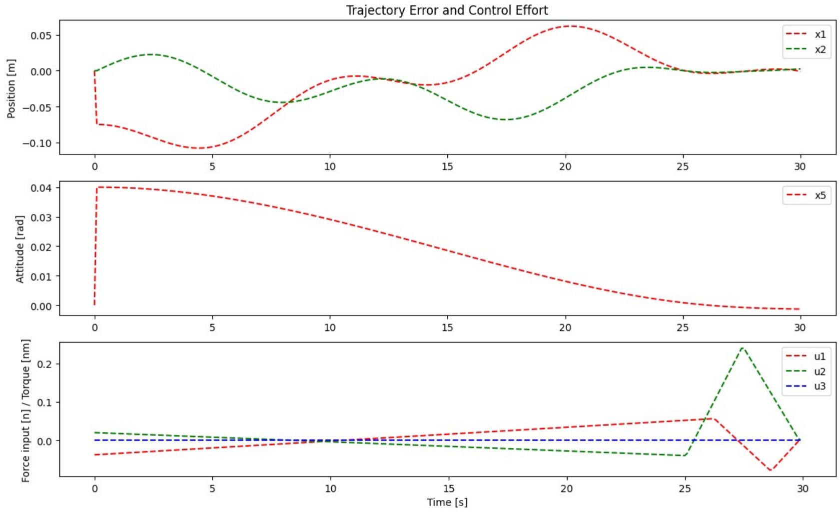


Figure 3: Rendezvous of Honey and Bumble without disturbance in 2d dynamics

Due to the newly added positional tolerance inequality constraints during rendezvous, it can be seen that extra control signals are implemented to realize this accuracy in the last 5 seconds in particular. These control signals are optimized $u*$ parameters determined by the *solve_problem* function.

5 Q5: Broken Thruster - Behavioral Cause

When we added in disturbance to our control signal, we can see that the Honey robot does not always successfully approach the Bumble robot. The disturbance itself is clearly seen by high frequency noise in u_2 of figure **??**; control input corresponding to the the broken thruster in the Y direction.

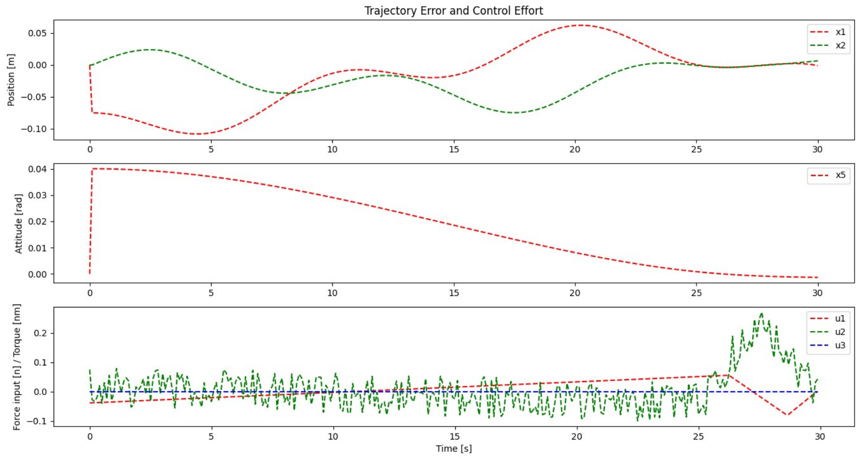


Figure 4: Rendezvous of Honey and Bumble with disturbance in 2d dynamics

We deducted that the reason of failure during the rendezvous was caused by the inherit open-loop system controller, meaning that the error between the reference (position of Bumble robot) and the output (position of Honey robot) is not considered into the updates of input control signal.

6 Q6: Broken Thruster - Suggested Improvement

To solve the disturbance based issues, we suggest the design of a more robust controller that includes a closed-loop system with integral control. This will elminate the steady state error induced by the disturbance. Perhaps specific pole pole placement in an output feedback controller may even be able to filter out some of the high frequency noise as seen by the input. To have an even more robust controller, we recommend implement model predictive control strategy to design the controller.

7 Q7: Finite-Time Optimal Control - Additional Z-dimension

Question 7 begins by extending our 2-D system to a 3rd dimensional model by including translation in the z-direction. The resulting state space model can be seen below.

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} u(t) \tag{7.1}$$

where:

$x = [p_X \ p_Y \ p_Z \ v_X \ v_Y \ v_Z \ \theta \ \omega]^T$

$u = [f_X \ f_Y \ f_Z \ \tau_Z]^T$

When using the new solver (*ctrl_wz*) to solve question 7, we noticed a significant increase of time duration required to solve the problem (.978163 seconds vs. .137324 seconds). We belive this is due to the fact that when we take translation axis Z into consideration, this creates a large increase of states necessary to be calculated for. The resulting plot describing positional translation, Z-axis angle (”attitude”), and control inputs can be seen in figure 5 below.

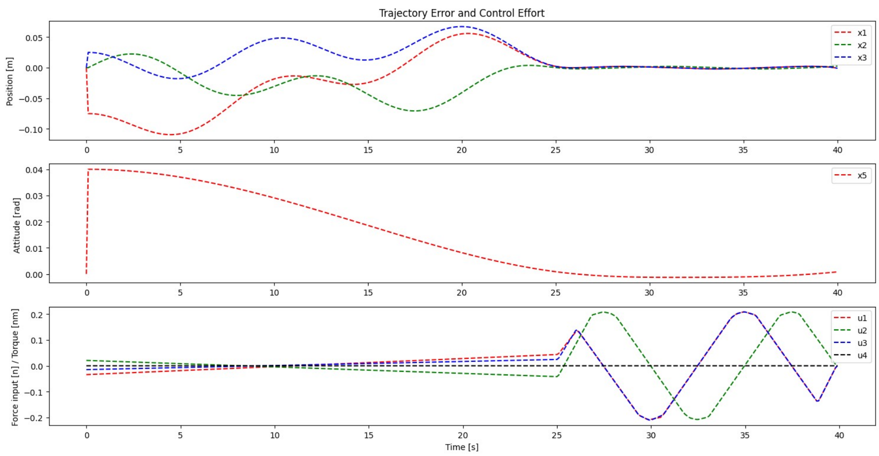


Figure 5: Rendezvous of Honey and Bumble without disturbance in 4 Dof (including Z translation)