AE353: Design Problem 03

Glider Control

March 27, 2019

1 Goal

`DesignProblem03` simulates an unpowered glider controllable only by its elevator. An actuator allows you to specify the angular rate of this elevator. Sensors allow measurements of both the glider pitch angle and elevator relative angle. The goal is to optimize glide distance with initial conditions $h_i = 2\ m$, and $v_i = 6\ \frac{m}{s}$ with a pitch angle of $\theta = -2°$.

1.1 Requirements

The glider must achieve a glide distance of at least 25 meters at an 85% accuracy, reaching and oscillating within the angle of attack of $\theta = -2° \pm 1°$ within 1 second and last for the duration of flight.

1.2 Verification

`DesignProblem02` will run for 1000 trials and the procured data will be analyzed to check if $x_{final} \geq 25$ for at least 85% of the trials. A histogram will then be plotted to show the overall performance and statistics of the trials to verify. The mean and median will be calculated and must be above 20 and 22.5 respectively.

2 State Space Model

The motion of the glider is governed by ordinary differential equations with the form

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = f(\theta, \phi, \dot{x}, \dot{y}, \dot{\theta}, \dot{\phi}), \quad x = \begin{bmatrix} \theta \\ \phi \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \quad u = \begin{bmatrix} \dot{\phi} \end{bmatrix}, \quad y = \begin{bmatrix} \theta - \theta_e \\ \phi - \phi_e \end{bmatrix} \tag{1}$$

With the following definitions:

θ = pitch angle	$\dot{x}$ = horizontal velocity	$\dot{\theta}$ = pitch angular velocity
ϕ = elevator angle	$\dot{y}$ = vertical velocity	$\dot{\phi}$ = elevator angular velocity

By use of the `fsolve(f_numeric, v_guess)` function, the equilibrium points may be found for state space model. Note, v_{guess} values were chosen to match realistic equilibrium values for our simulation:

$\theta_e = 0.0047$ for $v_{guess}(\theta) = -2^\circ$	$\dot{x}_e = 5.9629$ for $v_{guess}(\dot{x}) = 6$	$\dot{\theta}_e = -0.0221$ for $v_{guess}(\dot{\theta}) = 0$
$\phi_e = -0.0674$ for $v_{guess}(\phi) = -1^\circ$	$\dot{y}_e = -0.5609$ for $v_{guess}(\dot{y}) = -1$	$\dot{\phi}_e = -0.0041$ for $v_{guess}(\dot{\phi}) = 0$

The resulting coefficients for the state space model were found by use of the `jacobian()` function and resulted in the following matrices:

eqn:2]

$$\textbf{EDIT!!!}\ \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 \\ -8.76891 & 0.801336 & -0.0162797 & -0.173063 & 0.101591 \\ 115.354 & 23.1587 & 1.45768 & -19.4823 & 1.07654 \\ -53.9872 & -77.1851 & 0.844208 & 8.97444 & -5.26814 \end{bmatrix}, \tag{2}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1.0 \\ 0.0124201 \\ 0.193223 \\ -0.644061 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3}$$

3 Designing the Simulation

The controller will be implemented by including the state space model defined in matrices 2-3. These will form the basis for our controller and observer. In brief terms, both a controller and observer will be used in order to meet the requirement.

3.1 Controller

In order to begin, the input to the simulation must be defined. This will be done by using equation 4:

$$u = -Kx \tag{4}$$

To find a value for K, the following data will be placed into the *linear quadratic regulator* function, $K =$`lqr(A, B, Q_o, R_o)`. Q_o and R_o will be chosen to appropriately weigh the controller to optimize performance by the ratio with one another.

3.2 Observer

It is important to note that the only sensors available in this simulation are θ and ϕ. Therefore, it is paramount to implement an observer. This will be done by finding the matrix for L, achievable by using the function, $L =$`lqr`$(A', C', R_o^{-1}, Q_o^{-1})$ Finally, the observer may be initialized by use of equation 5:

$$\dot{\hat{x}} = A\hat{x} + Bu - L(C\hat{x} - y) \tag{5}$$

The output, y, is defined in equation 6:

$$y = \begin{bmatrix} \theta - \theta_E & \phi - \phi_E & 0 & 0 & 0 \end{bmatrix}^T \tag{6}$$

3.3 Completed Controller

Once the controller and observer are designed, `actuator.phidot =` is assigned to respond to $u = -K\hat{x}$, enabling the simulation to respond appropriately in order to meet the set requirement in section 1.1.

4 Statistical Results

Histogram results: **EDIT!!!**