AE353: Design Problem 03

Glider Control

April 5, 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 $v_i = 6\ \frac{m}{s}$.

1.1 Requirements

The glider must achieve a glide distance of at least 20 *meters at an* 75% *accuracy and mean of at least* 20 *meters, stably reaching it's ideal glideslope angle of* $\theta \approx -13.4957° \pm 2°$ *within* 2 *seconds and remain for the duration of flight.*

1.2 Verification

`DesignProblem02` will run for 1000 *trials and the procured data will be analyzed to check if* $x_{final} \geq 20$ *for at least* 75% *of the trials. A histogram will then be plotted to show the overall performance and statistics of the trials to verify. The mean,* μ*, and median will be calculated and must be above* $\mu \geq 18$ *and* 20 *respectively.*

2 State Space Model

The motion of the glider is governed by ordinary differential equations with the forms defined as **m = state, n = output, u = input**:

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

For:

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

Equilibrium points, θ_E in particular, will be selected to find the optimal glide angle. The simplified method of finding the optimal glide angle is shown in equation 2[source]:

$$\theta_{optimal} = \arctan(\frac{h_i - h_f}{R}) \tag{2}$$

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:

$v_{sol} = 0.011 \rightarrow v_{guess}(\theta) = 1°$	$v_{sol} = 4.6 \rightarrow v_{guess}(\dot{x}) = 10$	$v_{sol} = 0.33 \rightarrow v_{guess}(\dot{\theta}) = 0$
$v_{sol} = 3.0 \rightarrow v_{guess}(\phi) = 2°$	$v_{sol} = -0.7 \rightarrow v_{guess}(\dot{y}) = 1$	$v_{sol} = -0.092 \rightarrow v_{guess}(\dot{\phi}) = 0$

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

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 \\ -8.8866 & 0.63717 & -0.0371266 & -0.194675 & 0.102815 \\ 69.4289 & 14.1415 & 1.88573 & -15.3474 & 0.535567 \\ -21.1239 & -35.3369 & 0.849319 & 4.45345 & -2.35203 \end{bmatrix}, \tag{3}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1.0 \\ -0.0164536 \\ -0.150928 \\ 0.376398 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4}$$

3 Designing the Controller

The controller will be implemented by including the state space model defined in matrices 3-4. 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:

$$u = -Km \tag{5}$$

To find a value for K, the following data will be placed into the *linear quadratic regulator* function, $K =$`lqr`$(\mathbf{A}, \mathbf{B}, \mathbf{Q_o}, \mathbf{R_o})$. The ratio of $\frac{Q_C}{R_C} = 25$ was chosen to appropriately weigh the controller to optimize performance by the ratio with one another. The *open-loop controllability* may be verified by evaluating the condition number of the controller: `cond(ctrb(A,B))=` $1.0264 \cdot 10^4$.

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 include an observer. This will be done by finding the matrix for L, achievable by using the function, $L =$`lqr`$(\mathbf{A'}, \mathbf{C'}, \mathbf{R_o^{-1}}, \mathbf{Q_o^{-1}})$ Once again, the ratio of $\frac{Q_O}{R_O} = 30$ was selected to optimize the performance by weighing the LQR system. To verify the observability, a similar method to section 3.1 is utilized: `cond(obsv(A,C))=` $1.2304 \cdot 10^5$. Finally, the observer design equation may be implemented as shown in equation 6:

$$\dot{\hat{m}} = A\hat{m} + Bu - L(C\hat{m} - n) \tag{6}$$

The output, n, is defined in equation 7:

$$n = \begin{bmatrix} \theta - \theta_E & \phi - \phi_E & 0 & 0 & 0 \end{bmatrix}^T \tag{7}$$
$$\text{for } \theta_E = 0.2355, \;\; \phi_E = 0.01745$$

3.3 Completed Controller

Once the controller and observer are designed, `actuator.phidot` $= u = -K\hat{x}$ is implemented, enabling the simulation to react appropriately to meet the set requirement in section 1.1. To confirm whether the system is asymptotically stable, one final check must be performed: `eig(A − BK)< 0` and `eig(A−LC)< 0`. Both satisfied the test and therefore indicated asymptotic stability. Note, while lower condition numbers are more preferable - such as on the order of 10^2 - resulted in lower stability of the glider regardless of the ratio of $\frac{Q_c}{R_c}$ or $\frac{Q_o}{R_o}$. In the end, the controller's closed loop asymptotic stability was verified through Matlab and succeeded.

4 Simulation Tests

Below, the histogram results of the 1000 trial simulation will be displayed for both `actuator .phidot = 0` and `actuator.phidot=u`. An initial launch angle of 4° will be fixed for each iteration. The selection of this launch angle was made based reaching maximum altitude with the initial velocity in order to increase glide distance.

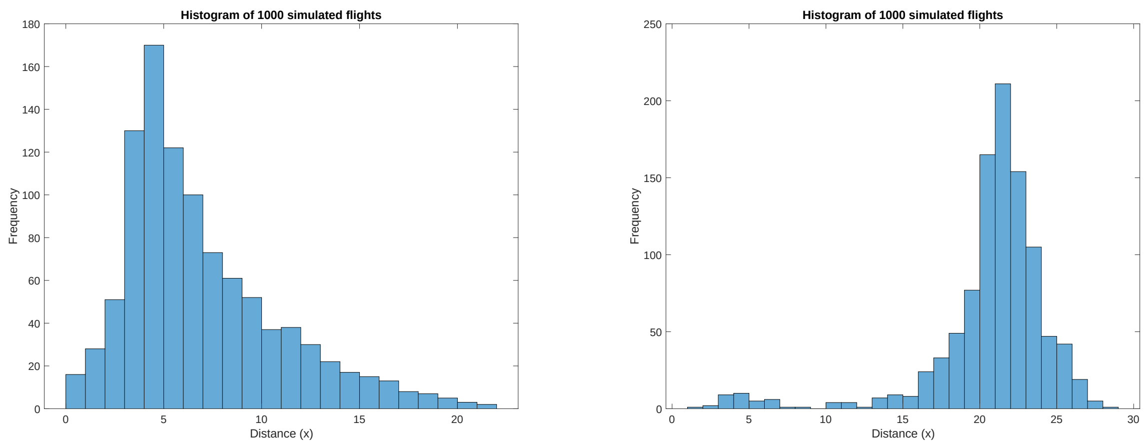
4.1 Uncontrolled

In the uncontrolled portion of the lab, a simulation was run "flights = 1000" iterations and a histogram, found in figure.1, was plotted. One may observe that the uncontrolled system did not have a very large glide distance and resulted in a mean of 6.9370 and median of 5.7886.

4.2 Controlled

For the controlled section of the lab, the simulation was run "flights = 1000" iterations and the histogram, which may be found in figure.2, was plotted. Unlike in the uncontrolled test, there was a clear effect of the controller in optimizing the range of the glider. With the mean of 20.7414 and median of 21.4106, this controller succeeded in meeting the verification of the set requirements in section (1.1).

4.3 Statistical Results



(a) Uncontrolled Simulation

(b) Controlled Simulation

Figure 1: Histogram of 1000 Flights

4.4 Conclusions

This simulation models a glider with sensors, $n = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$, and an actuator which controls the angular velocity, $\dot{\phi}$, of the elevators. The requirement, set in section (1.1), stipulated the glider must reach a glide distance of at least 20 meters at an accuracy of 75% after "flights = 1000" trials. In order to verify the requirements, the histogram must reflect said accuracy of 75% and contain a mean of no less than 18 and median of 20. After linearizing the non-linear ODE into its state space model, both a controller and observer were implemented in section (3). Two trials of the 1000 iterations were processed - uncontrolled and controlled - for comparison. For the controlled simulation, the mean $= 20.7414$, median $= 21.4106$, and accuracy $\approx 82.6\%$ and therefore passed the verification.