Linear Control Design For Insect Robot Flight Control

Sivakumar Balasubramanian
Department of Mechanical Engineering
University of Washington,
Seattle, USA

I. Introduction

A linear control system is designed for the flight control of an insect robot. Insect sized robots try to mimic biological insects thereby providing insights into their behavior [1]. The insects are able to perform high performance maneuvers like agile motions, make precise landings on ceilings etc. [2,3] . But they perform these operations with very small nervous systems. Thus by reverse engineering their techniques, a robust insect robotic design can be implemented.

In this project, a linear control system for the insect robot is formed. For simplicity the 2-D planar case is considered. We try to control the Roll angle and lateral velocity of the robot which are as shown in figure 1. We take the nonlinear ODE, linearize it around a fixed point and form a state space linear time invariant system (LTI), test its stability and assess controllability and observability. We also make various computations on the LTI system as explained in section IV and try applying the results obtained from the computations on the nonlinear system.

II. Model

The continuous nonlinear dynamics (CNL) for the 3-D case is as shown below

$$\dot{\varnothing} = W(\varnothing)\omega$$
$$J\dot{\omega} = \tau - \omega \times J\omega$$
$$M\dot{v} = f - \omega \times Mv$$

The quantity W(Ø) is a matrix that relates the angular velocity ω to the rate of change in Euler angles and J, v are the moment of inertia and velocity vectors. M is the mass of the robot.

The Diagram for the 2-D case is as shown in figure 1. A drag force f_d acts at a point above the centre of mass. The ODE after all approximations in drag force calculations is as shown.

$$\dot{\varnothing} = \omega$$

$$\dot{\omega} = -\frac{1}{J}(b_w r_w^2)\,\omega + \frac{1}{J}(b_w r_w)v$$

$$\dot{v} = -g\sin(\varnothing) + \frac{1}{M}((b_w r_w)\omega - \frac{1}{M} b_w v$$

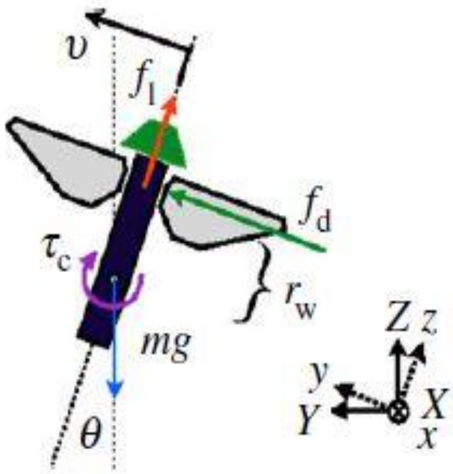

Figure 1: 2-D planar model from [1]

where, b_w is the drag factor , g is the acceleration due to gravity and r_w is the distance the drag force acts from the centre of mass and J is the Moment of Inertia around the x-axis. The values for all constants can be found in the appendix.

A. States

Ø - The Roll angle as shown in figure 1.
ω -The angular velocity
v -The lateral velocity of centre of mass.

B. Inputs

The inputs to the system are a control torque τ_c and a control force f_c that produced by the flapping of the wings. The Mechanism of flapping wings can be found in [4].

C. Outputs

The outputs from the system are the Roll angle Ø and angular velocity ω and lateral velocity v. The output function is of the form $\dot{Y}=h(x,u)$.

D. Assumptions.

The following assumptions have been made to simplify the system for making a linear analysis. The system is powered from an off board power source and the torques and forces due to the wirings are considered to be negligible. The wind that causes the drag force is always kept steady.

III. Analysis

A. Equillibrium

The equilibrium points for a system can be found by equating the ODE $\dot{X}$=$f(x_{eq}, u_{eq})$=0, where X is the state vector and u is the input vector.

The system in the project has various equilibrium points. One such point is [0,0,0], where the insect robot stays completely upright and without lateral shift at a position. Another interesting point would be [π,0,0] but we consider the first point as our point of interest because the desired action we want the insect to perform is to stay at the upright position. The inputs at equilibrium point is [0,0].

B. Linearization

The CNL can be linearized around the equilibrium point to make an analysis of the system. The continuous linear time invariant system is of the form as shown.

$$\dot{X} = \mathrm{A}X + \mathrm{B}u$$
$$\dot{Y} = \mathrm{C}X + \mathrm{D}u$$

Where the matrices A and B are obtained by taking the Jacobian of $f(x,u)$ with respect to X and u respectively and the matrices C and D are obtained by taking the Jacobian of $h(x,u)$ with respect to X and u respectively around the equilibrium points. The matrices at the equilibrium point for the project system are as shown.

$$\mathrm{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -11.4 & 1267.6 \\ -9.8 & 0.017 & -1.87 \end{bmatrix}$$

$$\mathrm{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathrm{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{D} = [0]$$

C. Stability Analysis

The stability of a CLTI system can be assessed using the eigenvalues of the A matrix. The eigenvalues of the A matrix for the project system are [7.6+19.41i, 7.6-19.41i, -28.5]. We see that the system has a complex conjugate pair of eigenvalues with positive real part and a negative real eigenvalue. The eigenvalue analysis suggests that the Roll angle and angular velocity spiral outwards and are unstable but the lateral velocity is stable. But the overall system is unstable as even a single unstable eigenvalue makes the overall system unstable [5].

D. Controllability and Observability

Since it is a time invariant system we can compute the Controllability and Observability matrices using the A, B and C matrices. We can conclude that the system is completely controllable and observable as the rank of both the matrices is equal to the size of the A matrix. The Controllabilility and Observability matrices are as shown.

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & -11.4 & 1267.6 \\ 1 & 0 & -11.4 & 1267.6 & 151.7 & -16853 \\ 0 & 1 & 0.017 & -1.89 & -10.02 & 25.1 \end{bmatrix}$$

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & -11.4 & 1267.6 \\ -9.8 & 0.017 & -1.89 \\ 0 & -11.4 & 1267.6 \\ -12423 & 151.7 & -16853 \\ 18.5 & -10.02 & 25.1 \end{bmatrix}$$

IV. Computations

The computations are performed by scaling down the values of the constants so that the higher powers of matrices don't have large values in their elements. The various computations performed are as follows.

A. Open Loop Control

An open loop controller is implemented for the LTI which steers the system states from a given initial condition to a final condition using its inputs. It is done by using the controllability matrix. Doing an SVD decomposition of the matrix makes it easy to invert it to get the inputs at each time step. The formula for calculating input at each time step t is as shown.

$$u(t) = V\,\Sigma^{+} U^{T}(\mathrm{Xf}\text{-}{A_d}^{t}\mathrm{X}(0))$$

where V, Σ^{+}, U^{T} are obtained by decomposing the controllability matrix, Xf, X(0) are the given final condition and initial condition respectively and $A_d = e^{(At)}$. For this project, the initial condition is taken as [π, π, 3] and the final condition is [0, 0, 0]. The plot for the trajectories of the states is as shown in Figure 2A, 2B, 2C.

B. Closed loop control

The closed loop control is established by giving an input that varies proportional to the state feedback. The input provided at each step is equal to $-KX$. The formula for the closed loop control is given as shown.

$$\dot{X} = (A - BK)X$$

K is designed in such a way to make matrix $(A - BK)$ to asymptotically stabilize the above equation. If the system is continuous, the eigenvalues of the matrix $(A - BK)$ should have negative real parts and if the system is discrete, it should have eigenvalues with real parts less than 1. The trajectories of each state are as shown in figure 2A, 2B, 2C.

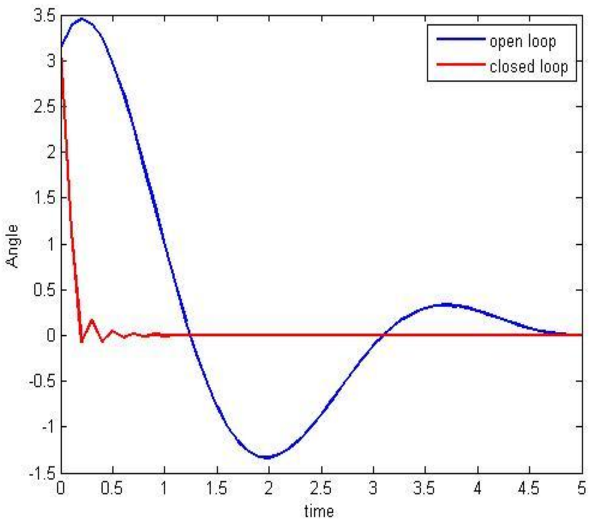

Figure 2A: Open and closed loop control of Angle (state1)

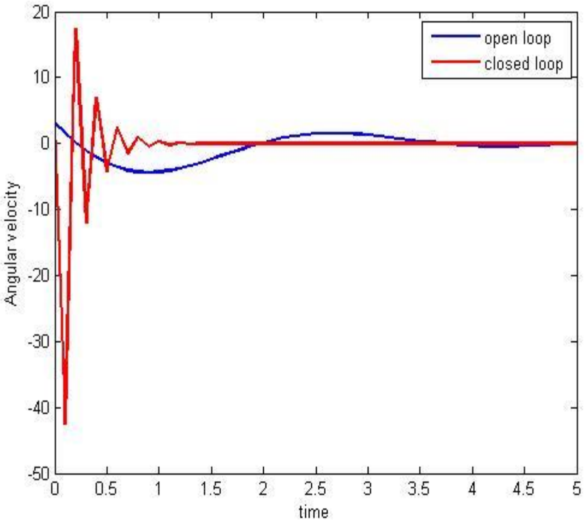

Figure 2B: Open and closed loop control of Angular velocity (state2)

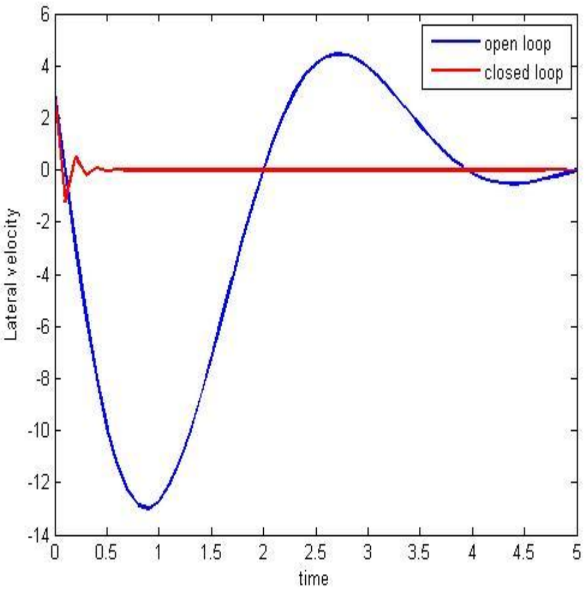

Figure 2C: Open and closed loop control of Lateral velocity (state3)

We see from figure 2A, 2B, 2C that the open loop drives the states to the desired end conditions and the closed loop asymptotically stabilizes the states and makes them go to zero. The closed loop is more robust in stabilizing the system during perturbations. In the project scenario the insect robot is more prone to such perturbations from sudden gust of wind and forces from the power line. Thus the closed loop control is more desirable.

C. State estimation

In section IV A and B, we had constructed controllers assuming that the states were fully observable. Now we can construct the estimate of our initial states using our output given by the formula below.

$$X(0) \ = \ V_o \, \Sigma_o{}^+ U_o{}^T Y$$

where X(0) is the estimate of initial states, $V_o \, , \Sigma_o{}^+, U_o{}^T$ are matrices obtained by doing an SVD of the observability matrix and Y is the output. The estimate of states at each time step can be found by making the simulation to run from that particular time step and using the above formula. There is an error that occurs in the estimation of state and this can be decreased by constructing the asymptotic state constructor as shown in the next section.

D. Asymptotic State Estimator

To make the state estimation process much more accurate, we construct the asymptotic state estimator using the following formulae.

$$e \ = \ X - \hat{X}$$

$$\dot{e} = (A - LC)e$$

$$\dot{\hat{X}} = A\hat{X} + Bu - L(\hat{Y} - Y)$$

where, e is the error in the estimation of state. $\hat{X}$ and $\hat{Y}$ are the estimates of the state and output respectively and L is designed in such a way to make matrix $(A - LC)$ to asymptotically stabilize the above equation. By making $\dot{e}$ go to zero we bring the estimated states equal to the actual states. The plots for the original and estimated states are as shown in Figure 3A, 3B, 3C.

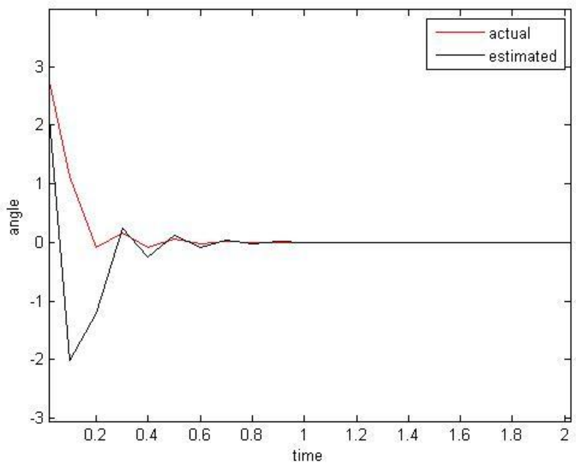

Figure 3A: original and estimated angle (state 1)

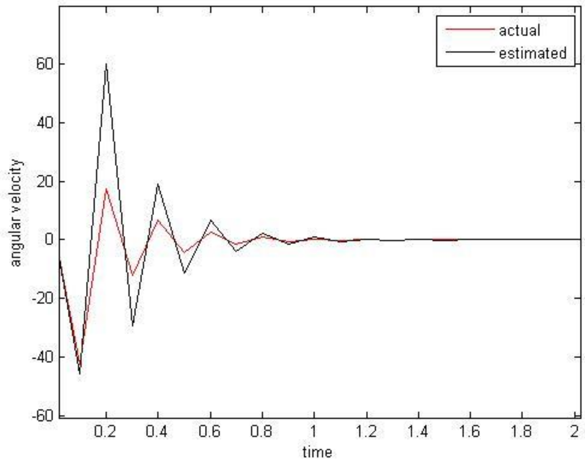

Figure 3B: original and estimated angular velocity (state 2)

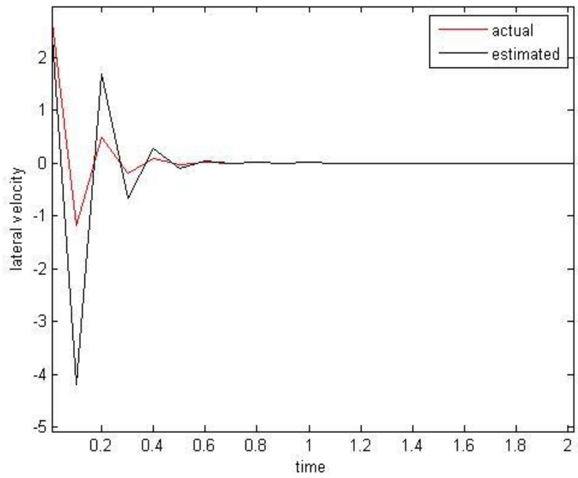

Figure 3C: original and estimated lateral velocity (state 3)

From the plots we see that the error in the estimated states go down to zero and make the estimated states coincide with the original state.

E. Closed Loop Estimator/Controller

We had stabilized the error between the estimated states and the original state in the previous sections. Now we can apply this to the equation in IV B and get a closed loop estimator/ controller. The equation for that would be as follows.

$$\dot{X} = (A - BK)\,\hat{X} + BKe$$

$$\dot{e} = (A - LC)e$$

The equation simultaneously stabilizes the error to zero and gets the state to its equilibrium point. The plots for the three states for the closed loop estimator/controller is as shown in Figure 4.

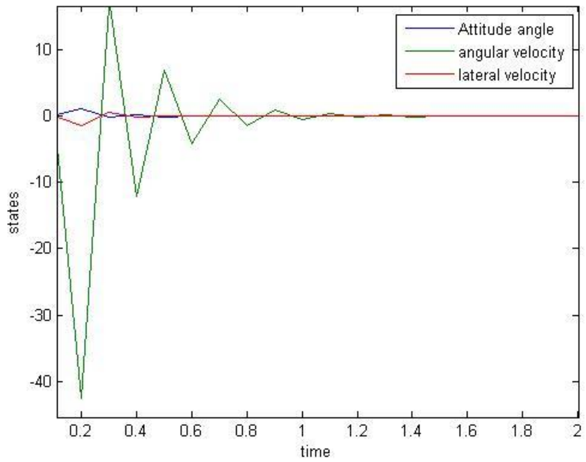

Figure 4: Closed Loop Estimator/Controller

From the plot we see that the system has lesser oscillations before stabilizing for states 1 and 3 compared to the closed loop control plots in Figure 2A, 2C.

V. APPLICATIONS

A. Nonlinear System

In section IV we had linearized our non-linear system and constructed a closed loop estimator/controller for it. Now we try to apply this estimator to our non-linear system and see if it still stabilizes the system. Thus we feed in the estimated states into the simulation of the non-linear dynamics. The plots for the trajectories obtained are as shown in Figure 5.

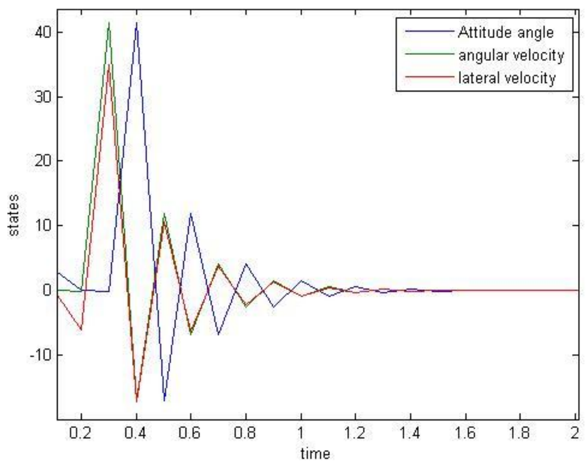

Figure 5: Estimator/controller applied to the nonlinear system

We see that the system stabilizes to the equilibrium point but it takes a longer time and oscillates more before it reaches the equilibrium.

B. Non-linear System with Disturbance

Now we introduce a parameter variation into the system. In real life scenarios the insect robot need not always have the same drag coefficient. It may vary according to the environmental conditions. Thus we try changing the value of the drag coefficient in the non-linear model and try testing the estimator/ controller. We get the plots shown in Figure 6A, 6B, 6C. We see that initially the simulations are different but they eventually coincide and stabilize to the equilibrium point.

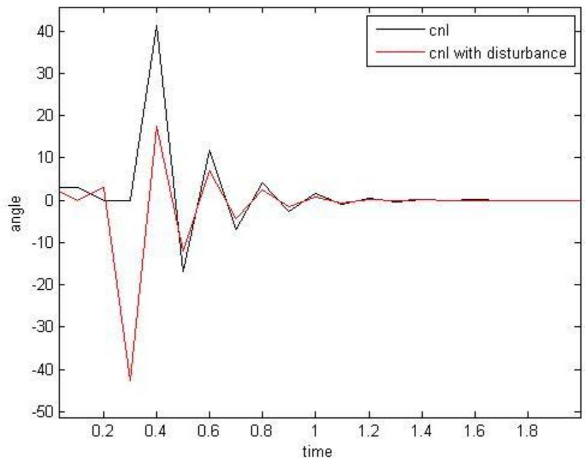

Figure 6A: Estimator for non-linear case with disturbance (state 1)

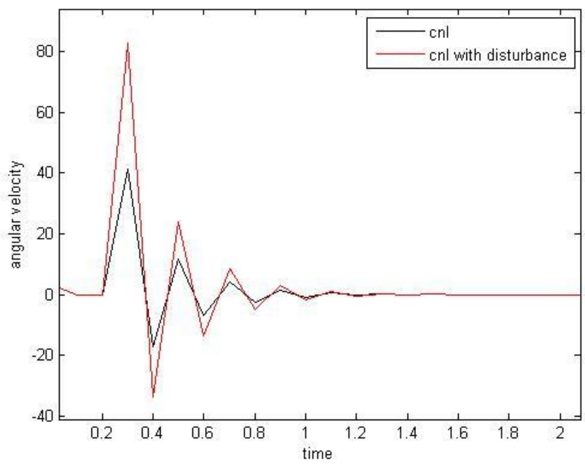

Figure 6B: Estimator for non-linear case with disturbance (state 2)

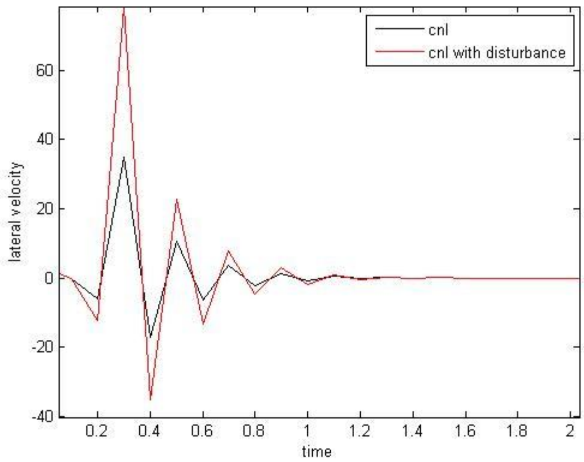

Figure 6C: Estimator for non-linear case with disturbance (state 3)

VI. Discussions

In the project we had considered the flight control of an insect robot and tried to develop a linear control for the 2-D case. We had linearized the system about an interesting equilibrium point. We found that the system was completely observable and controllable. We had then constructed a closed loop estimator/ controller for the linear case and found that it was more realistic and stabilized the system quicker than the open loop controller. We had also tested this on the non-linear dynamics and also by changing the drag parameter. The results showed that the estimator worked well in both cases.

In further work we can try to reduce the assumptions and consider the forces given by the wirings. We can also go towards making an optimal controller that would make the insect use much lesser energy in stabilizing. We could also try to access the system at other equilibrium points.

VII. Reference

1. Fuller SB, Karpelson M Censi A, Ma KY, Wood RJ. 2014 Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli. J. R. Soc.
2. Land MF, Collett TS. 1974 Chasing behaviour of houseflies. J. Comp. Physiol. A 89, 331–357. (doi:10.1007/BF00695351)
3. Dalton S. 1975 Borne on the wind. London, UK: Chatto & Windus.
4. Xinyan Deng LS, Wu WC, Sastry SS. 2006 Flapping flight for biomimetic robotic insects: part I. System modeling. IEEE Trans. Robot. 22, 776–788. (doi:10. 1109/TRO.2006.875480)
5. J.P. Hespana, “Linear systems theory”. Princeton, New Jersey:Princeton University Press, 2009.

VIII. Appendix

All computations and simulations were done in Matlab. Controllability and Observability were found by using inbuilt commands in Matlab. Other inbuilt commands such as SVD and Place were used for doing Computations.

The values of various constants involved are given in table 1.

Mass m	81 x10^{26} kg
Mass of ocelli	25 x10^{26} kg
Moment of inertia (J)	1.42 x 10^{29} kgm^2
Wing drag factor (b_w)	2.0 x10^{24} Ns m^2
Distance from CoM to wings (r_w)	9 x10^{23} m

Table 1: Values for constants