Analysis and Control of Single-Wheeled Mobile Bot

Daksh Dhingra

***Abstract*—The article highlights the control mechanism of a robot that is stable on a single wheel using the actuator mechanism based on the inverse mouse ball drive. The control system is designed and presented here to keep the ball-bot dynamically stable and to move it around when necessary.**

I. Introduction

Traditionally robots are equipped with atleast two driving wheels. However, the main drawbacks of these robot is they cannot move in every direction. The design that is presented here is omni-directional, that means the robot equipped with this design can move in every direction without changing the orientation of there wheels. This is achieved by a Single spherical wheel at the center of the robot, that is why this robot has very small footprint and one point contact with the ground.

These types of robots can specifically be used in a an environment like a hospital, where there is constant rush and they can make complex motions at high speeds to avoid bumping into people and obstacles. They can serve the purpose of transporting equipment, serving food to patients etc.

The drive system of an omni-directional wheel is mathematically modeled to give two non-linear equations. These equations are then linearized about the upright position to get the state space system. So, if you put a mass on a ball it would definitely fall. Thus the system is initially unstable. The system is then controlled by providing the right gain. A system estimator is designed to estimate the states that are required but cannot be measured in a physical setting. Modeling and simulation of the system is done non-linearly and it was concurred that the estimated poles are stable even for the non-linear system of equations.

II. System Model

Figure 1 [4] shows the simplified model of the ballbot. This figure is made in the sagittal plane of the system . Now while modeling the system three main elements were accounted.

1) The wheel of the actuator motor.
2) The Omni-directional ball.
3) The whole body of the system which is considered as an inverted pendulum and its mass concentrated at the center of the bob.

These elements are then transformed to the Free body diagram of each ball which resulted in the system equations.

$$J_0\ddot{\theta}_0 = \tau - b_e\dot{\theta}_0 - R_0 F_{R0} \quad (1)$$

$$\theta_0 = \frac{R_1}{R_0}\theta_1 \quad (2)$$

The two degree of freedom of ball bot are θ_1 and θ_2. J_0 is the motor's moment of Inertia and θ_0 is the rotation angle of the motor. Here, R_0 is the radius of the motor's wheel, b_e is the viscous coefficient of friction of the motor, τ is the input of the system that we apply to make the system stable,FR_0 is the friction between motor drive wheel and the ball and R_1 is the radius of the ball.

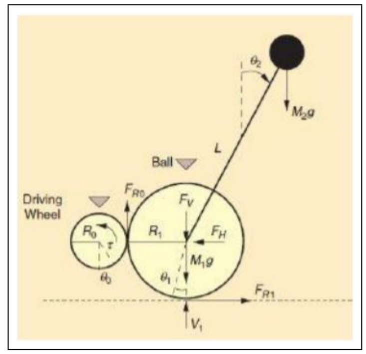

Fig. 1. Simplified model of Ballbot

Now, the second set of equations [4] define the rotational and transnational dynamics of the ball.

$$\dot{X}_1 = R_1\dot{\theta}_1 \quad (3)$$

$$M_1\ddot{X}_1 = F_{R1} - F_H \quad (4)$$

$$J_1\ddot{\theta}_1 = R_1 F_{R0} - R_1 F_{R1} \quad (5)$$

here X_1 denotes the horizontal position of the ball's center of mass measured from a non-inertial reference system, M_1 is the ball's mass, F_{R1} is the friction between the ground and the ball, F_H is the horizontal component of the force exerted by the body on the ball, and J_1 is the moment of inertia of the ball.

Equations for Translational and Rotational dynamics of body [4]:

$$M_2\ddot{X}_2 = F_H \quad (6)$$

$$M_2\ddot{Z}_2 = F_V - M_2 g \quad (7)$$

$$J_2\ddot{\theta}_2 = LF_V\sin\theta_2 - LF_H\cos\theta_2 + \tau \quad (8)$$

where X_2 and Z_2 are the horizontal and the vertical position of the body's center of mass respectively measured from the $0xz$ planar reference system, M_2 defines the mass of the body, F_V is the vertical component of the force exerted by the body, g is the force of gravity, θ_2 is the body rotation angle in the planar system, J_2 is the moment of inertia of the body and L is the half length of the body.
The relationship between the translational dynamics and variables of the ball and the body is given as[4].

$$X_2 = X_1 + L\sin\theta_2 \quad (9)$$

$$Z_2 = L\cos\theta_2 \tag{10}$$

The equations 1 to 10 and the constants given, are used to form the non-linear system dynamics. System is made stable using the non-linear dynamics.

A. State Equations

The input of the system here is τ which defines amount of torque that is applied from the motor to the ball, for the sake of simplification only one plane is considered to analysis so τ of only one motor is considered here. The four states are θ_1,$\dot{\theta}_1$,θ_2 and $\dot{\theta}_2$ which defines the angle of tilt of the body, angular velocity of the body, angle of tilt of the Ball and angular velocity of the ball respectively. The output of the system are two states θ_1 and θ_2. Only two outputs are take here because it was important to observe the system's tilt to physically interpret its stability. The state space model is present in figure 2.

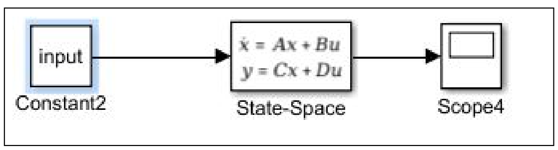

Fig. 2. State space system model

The equation 1 to 10 are solved to get the values of F_H, F_R1,F_R0 and F_V. Consequently, the following equations were obtained.

$$F_V = -M_2 + L\ddot{\theta}_2 sin\theta_2 - LM_2\dot{\theta}_2 cos\theta_2 + M_2 g \tag{11}$$

$$F_{R0} = -(J_0R_1\ddot{\theta}_1 + \tau R_0 - b_eR_1\dot{\theta}_1)/R_0^2 \tag{12}$$

$$F_H = M_2R_1\ddot{\theta}_1 + M_2L\dot{\theta}_2^2 sin\theta_2 + M_2L\ddot{\theta}_2 cos\theta_2 \tag{13}$$

$$F_{R1} = M_1R_1\ddot{\theta}_1 + F_H \tag{14}$$

These equations are then placed in equations 8 and 5 to get the final equations which are then simplified to get the continuous time non linear functions of $\ddot{\theta}_1$ and $\ddot{\theta}_2$. These equations will be our base equations for non linear computations. Following assumptions were made while computing the system dynamics [4]

1) No slippage is considered.
2) No deformation of the ball is taken into the account.
3) No friction is considered while modeling mathematically.
4) No steep inclinations are taken into account while modeling.

Small angle approximations are also considered in continuous time linear model evaluation[4].

III. Analysis

The system is at equilibrium at the point where all four states are equal to zero. This point is really interesting because it is unstable in nature it is like balancing a pen from its tip. Now if you apply a small displacement on the ball it will instantly go on falling due to its instability. This unstable equilibrium point where model is upright is really interesting and can be used to linearize the non-linear system. The model is given in Fig 1.

To liniearize the control system around the equilibrium point the small angle approximations [4] mentioned below are considered.

$$\theta_2^2 \approx 0 \tag{15}$$

$$cos\theta_2 \approx 1 \tag{16}$$

$$sin\theta_2 \approx 0 \tag{17}$$

$$\dot{\theta}_2^2\theta 2 \approx 0 \tag{18}$$

Using approximations the equations can be combined to form

$$h_1\ddot{\theta}_1 + h_2\ddot{\theta}_2 + h_3\dot{\theta}_1 = h_4\tau \tag{19}$$

$$h_2\ddot{\theta}_1 + h_5\ddot{\theta}_2 + h_6\theta_2 = \tau \tag{20}$$

These equations were simplified to make the main two equations for the $\ddot{\theta}_1$ and $\ddot{\theta}_2$ given below

$$\ddot{\theta}_1 = \frac{-h_3h_5\dot{\theta}_1 + h_6h_2\theta_2 - (h_2 - h_4h_5)\tau}{h_1h_5 - h_1^2} \tag{21}$$

$$\ddot{\theta}_2 = \frac{\tau}{h_5} + \frac{h_2h_3\dot{\theta}_1}{h_1h_5 - h_2^2} - \frac{h_6h_2\theta_2}{h_5(h_1h_5 - h_2^2)} - \frac{h_2^2\tau + h_4h_5h_2\tau}{h_1h_5^2 - h_2^2h_5} - \frac{h_6}{h_5}\theta_2 \tag{22}$$

The values of $h_1, h_2, h_3, h_4, h_5, h_6$ were found out using the given equations with the values of constants shown in the table 1.

$$h_1 = J_1 + J_0(R_1^2/R_0^2) + (M_1 + M_2)R_1^2 \tag{23}$$

$$h_2 = M_2LR_1 \tag{24}$$

$$h_3 = b_e(R_1/R_0)^2 \tag{25}$$

$$h_4 = R_1/R_0,\ h_5 = J_2 + M_2L^2 \ and \ h_6 = -M_2gL \tag{26}$$

Parameter	Description	Value
J_0	Moment of inertia of the motor	$4.5 \cdot 10^{-8}$
J_1	Moment of inertia of the ball	$6.7 * \cdot 10^{-3}$
J_2	Moment of inertia of the body	0.9375
R_0	Wheel radius	0.025
R_1	Radius of the ball	0.124
M_1	Mass of the ball	0.65
M_2	Mass of the body	30
L	Half-length of the body	0.5
b_e	The viscous friction coefficient of the body	$1.85 \cdot 10^{-3}$
g	The gravitational constant	9.81

Fig. 3. Parameters used for solution [4]

The linear system is written as:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Jacobian of the linearized equation is done to get the state matrices A,B,C,D which are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.6698 & -477.3945 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.1477 & 122.6790 & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ 69.7518 \\ 0 \\ -15.2579 \end{bmatrix}$$

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

A. Stability

To analyze the stability of the system the eigen values of A matrix are calculated. The eigen values are $\lambda_1 = 0$,$\lambda_2 = -11.3696$,$\lambda_3 = -0.0952$ and $\lambda_4 = 10.7949$. Out of which two eigen values are negative, one 0 and one positive. So because of the positive eigen value the system is unstable.

B. Controlability and Observability

To check the controlability the controlability matrix $\mathcal{C}$ is calculated. It is found to be

$$\mathcal{C} = \begin{bmatrix} 0 & 69.8 & -46.7 & 7315.3 \\ 69.8 & -46.7 & 7315.3 & -9816.7 \\ 0 & -15.3 & 10.3 & -1878.7 \\ -15.3 & 10.3 & -1878.7 & 2343.7 \end{bmatrix}$$

The rank of this matrix is 4 .Thus $\mathcal{C}$ have full row rank signifying that the system is controllable. To check the Observability of the system the observability matrix $\mathcal{O}$ is calculated and its rank was checked using Matlab.

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.6698 & -477.3945 & 0 \\ 0 & 0.1477 & 122.6790 & 0 \\ 0 & 0.4486 & 319.7646 & -477.3945 \\ 0 & -0.0989 & -70.4903 & 122.6790 \end{bmatrix}$$

The rank of the matrix is 4 that means $\mathcal{O}$ has full column rank. This tells us that the system is observable and its states can be estimated by using suitable state estimator.

IV. Computation

A. Open Loop Control

The open loop control system used the same state equations and some value of input is defined to make the system move from one state to another. The initial state here is $x_0 = [0, 0, 0.1, 0]^T$ and the final state where system has to reach is $x_1 = [0, 0, 0, 0]^T$. We are only giving a small displacement to the body because the system is linearized around an equilibrium point and it will not be able to stabilize if a large input is given which drives it away from the equilibrium zone. We are going to simulate the system from time 0 to 0.1 seconds and in the time steps of 0.0001 sec. Here the input is calculated using the controlability Gramian from the following equations [2].

$$W_c(t) = \int_0^t e^{-At} B B^T e^{-A^T \tau} d\tau \qquad (27)$$

W_c here is the controllability gramian which is used to compute the minimal input enegery u(t).

$$u(t) = -B^T \; e^{A^T(t_0 - t)} \; W_c^{-1}(t_1) \; [e^{At_1} x_0 - x_1] \qquad (28)$$

$u(t)$ is computed for each time step to find out the complete $u(t)$ needed to take the system from initial state to the desired state. after this we used Matlab command lsim and estimated the states of the system with the input calculated.

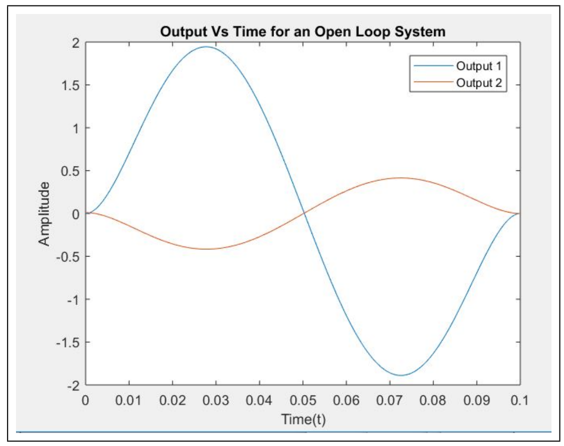

Fig. 4. Output when open loop input is applied

Note in the Fig 4 the system goes to the desired state in given time. The output 1 and output 2 are the 1st and 3rd state of the system. While it interesting to see the system converging, it will go unstable when there are modelling errors and disturbances. One interesting question here is why this specific time interval of 0 to 0.1 and time step of 0.0001 is taken. To illustrate this a time interval of 0 to 5 seconds and time step of 0.01 seconds is taken and following graph was obtained.

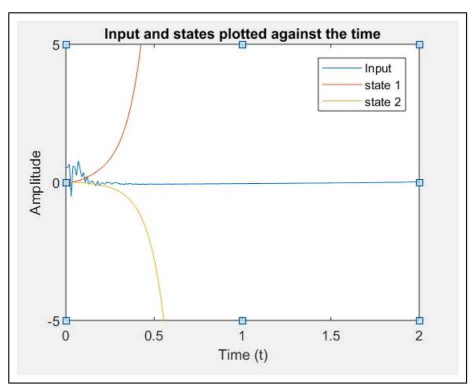

Fig. 5. Output when open loop input is applied

Here the states starts diverging even when the minimal energy input is applied. Notice the input goes zero after some time. This happens because W_c is very high in order of 10^{48} and the inverse of that is near to zero. So, as it keeps on decreasing Matlab takes it to be zero after around 0.5 seconds and when no input is applied the states go unstable and blow up.

B. Closed Loop State Feedback

A closed loop system is dependent upon the feedback to drive the system to stability. Now as Controllability gramian does not play a role here that means it doesn't really matter if it is small or more. Thus we define the time step here to be t= 0:0.01:2 seconds. The closed loop feedback law is

$$u(t) = r - k\ x \tag{29}$$

Now the input depends upon the state and is changed at each time instant. In this problem we calculated the input using the same process as in Open loop control but we can also use a step or a ramp input, the system will still remain stable in all cases. The feedback is calculated by placing the poles of the system in the desired value. The new poles of the system are chosen to be $[-17.003, -10.0708, -10, -3.1217]$. These are placed by using the place command in Matlab. The K matrix is calculated as.

$$K = \begin{bmatrix} -4.198 & -2.441 & -63.859 & -13.754 \end{bmatrix}$$

This K matrix is used to change the new A matrix of the system by A-BK. The figure 5 shows the plot with feed back. Note that the output is stable when you provide a feedback.

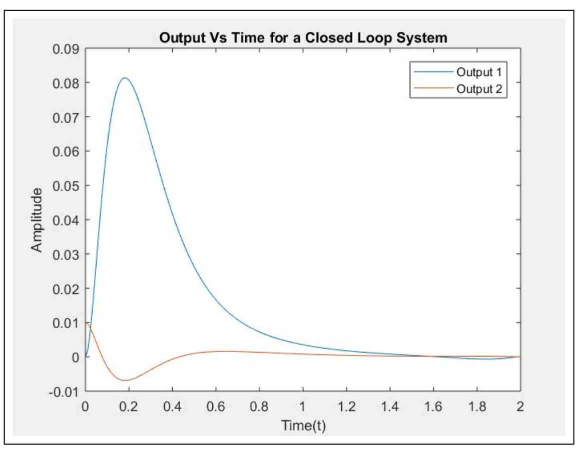

Fig. 6. Output when open loop input is applied

So, the closed loop system is better because it can compensate the noise. In our specific application domain it converges for higher times. Notice the inverse of controllability gramian for openloop system was going in order of 10^{-48} but for a closed loop system it doesn't play a major role. Moreover, this is supposed to converge faster that can help the system to go stabilize much earlier.

C. Open Loop Estimator

In order to use a proportional controller from last section we need to have state observer. Notice that the state feedback was done by assuming that all the state variables are available at the output. But in real-time we will only be measuring the body rotation angle θ_1 and ball rotation angle θ_2. Thus we need to make an observer that can accurately measure our state of the system. We first start with Open loop control, we are going to use the time= 0:0.0001:01 just to make the states converge. Now if we want to simulate that we can do it using convolution integral as given below 2.

$$y(t) = C[A_d^t x_(in) + \sum_{\tau=0}^{t-1} A_d^{t-\tau-1} B_d u(\tau)] \tag{30}$$

$$\mathcal{O}_{OL} x_{in} = \bar{y} \tag{31}$$

However, this all can be done in Matlab using lsim command. So, the input is calculated just similarly as done in open loop controller and fed to lsim with the state space system using C matrix as an identity matrix. The results are shown below

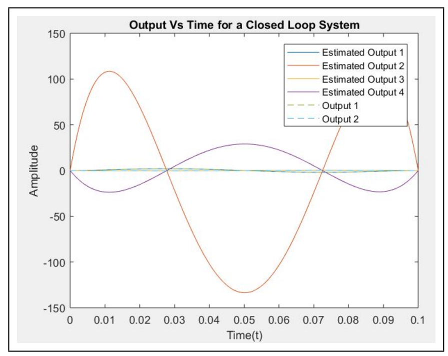

Fig. 7. Trajectory with open loop estimator. The estimated state is represented by solid line and actual states by dotted line

Notice the estimated states coincides with the output of the open loop system. But if any sensor noise or delay is induced in the system the estimation will tend to fail.

D. Closed Loop Estimator with Open Loop system

A closed loop estimator is considered more robust because it takes the output of the system in the account before estimating the states. The equation of the close loop estimator is [5]

$$\hat{x} = A\hat{x} + Bu - L(\hat{y} - y)$$

Now to start the system we need to put an initial state. So, any initial state is estimated because it doesn't matter what initial state you chose the error would slowly converge and the estimated state $\hat{x}$ is computed on every time instant. To plot the error we need to calculate [5]

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

and plot it. It must be highlighted here that if (A-LC) is stable the error would ultimately go to zero as shown in figure 7.

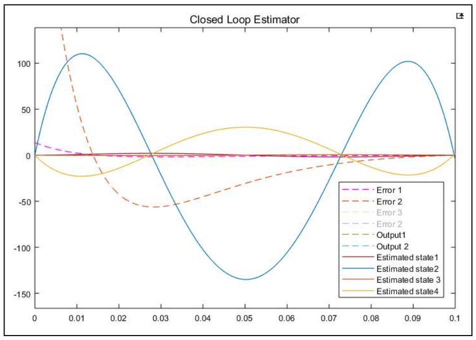

Fig. 8. The error for estimated state plotted with time

Note that the system here is driven from a 0.1 rad initial state so that the convergence of error can be observed . Here the value for L was calculated by placing the poles at $5 * [-17.003, -10.0708, -10, -3.1217]$. These poles are 5 times of the poles used in the state feedback system just to make the dynamic system converge faster. After using place command L was calculated as

$$L = \begin{bmatrix} 134.4 & 4177.96 & 4.92 & 259.12 \\ 3.7 & -293.72 & 65.86 & 916.43 \end{bmatrix}$$

It is interesting to see how the error starts with such a high value and then converges to 0. So, this system is better equipped to handle noise

E. Closed Loop Estimator with Closed Loop feedback system

Now the system has a feedback that makes it stable. We estimate the states of the stable system. It is more intuitive from the simulink model below.

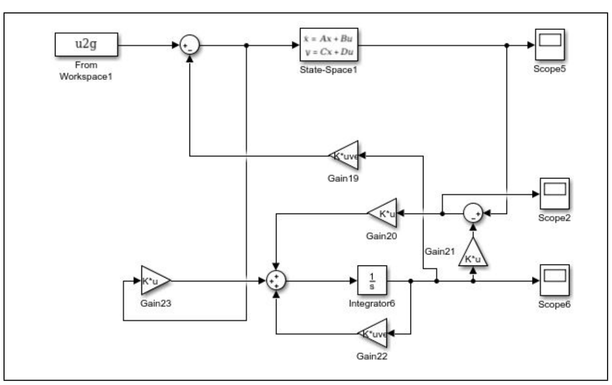

Fig. 9. Simulink Model of closed loop control with closed loop estimated system

Here the estimator is made same as before but the estimated state is fed into the closed loop feedback law. This makes the system dynamics converge faster, due to the closed loop estimator the estimator error converges and the system can withstand signal noise and superfluous vibrations easily.

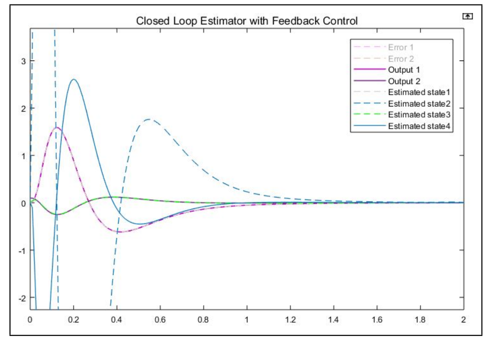

Fig. 10. States of a closed loop control with closed loop estimated system

Note here that the system converges to zero and it seems there is not much error in the estimation of the system.In error plot below you see that error is in the order of 10^{-1}.

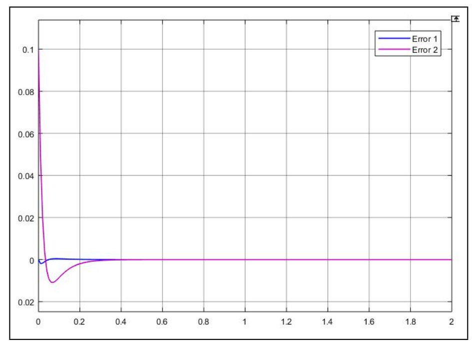

Fig. 11. Error plot of closed loop control with closed loop estimated system

V. Continuous Non Linear system

To simulate the continuous non linear system two files were made which contain the estimator function and the non linear function. These files were passed to the simulink and were iterated to evaluate the result. The solver used is ODE45 in simulink . The model is simulated for 2 seconds with timestep of 0.1 seconds. The results are as shown in the figure.

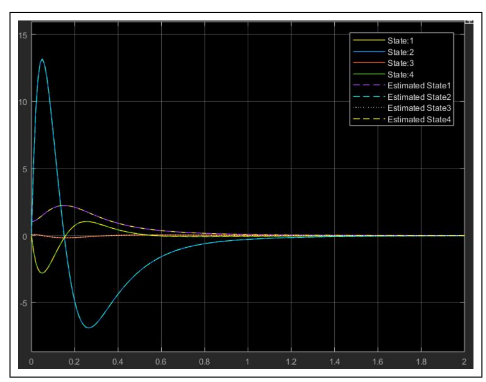

Fig. 12. Plot of estimated states and actual states of Continuous non Linear system

As, can be seen in the figure because the system is modelled without any assumptions for linearity it converges faster than the linearized system.

A. Noise Addition

The system is given some noise to check if the estimator can estimate those values. Though you would notice in Figure 13 that the estimated states coincide with the actual output but when it zoomed in, it is apparent that they are not on the same line.

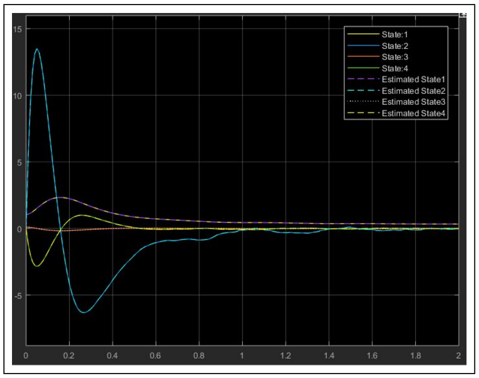

Fig. 13. Plot of estimated states and actual states of Continuous non Linear system

The trajectory of the system does change here but controller is able to minimize the vibrations and control the system. Practically these noise can be generated because of the sensor noise or system vibrations. However, both Linear and Non- Linear system with closed loop controller estimator are equipped to handle it.

VI. Discussion

In summary the non linear equations of the ballbot were generated. The ball bot was linearized around an interesting fixed point. The linearized model was simulated and it was found that the system is inherently unstable. Consequently the states were made stable by applying the feedback input with a gain after placing the poles. However when the small angle approximations were removed the system still acted stable even for large deviations in displacement. So, it means the gains calculated by linearized model are also applicable on the non-linear model. Practically the control of the robot would be non linear but with these results it can be concurred that the gains can be used to make the non linear system stable. When the closed loop estimator was applied to the system it was consistent in approximating the states of the system. Initially there was some error but in time it went on to zero.

Now, when the noise was there in the signal the estimator approximated the states pretty close to the actual values. For the practical application of the ball bot this estimator can be applied because the deviation that is noticed when the noise is there was negligible and can give us the actual system states. However, there is some scope of improvement by choosing the better pole values which are practically possible and can make the system more dynamically stable. A better pole placement of estimator can also make the estimated outputs closer to the real value and compensate more noise. Overall the system that is designed is suitable to apply on the real time robot and evaluate results.

References

[1] Anthony Anderson *Linear Analysis of a Cable Driven Device for Prosthesis Testing and Experimentation*. Mechanical Engineering, University of Washington.

[2] Ney Bruno, Jorge Silva Garcia and Paulo Cesar Pellanda *On the proof of the existence and uniqueness of minimum-energy control for linear time-invariant systems*. International Journal of Electrical Engineering Education .

[3] Peter Fankhauser,Corsin Gwerder *Modeling and Control of a Ballbot*.Autonomous Systems Lab ,ETH Zurich.

[4] A. Ravani *Analysis and Control of a Single-Wheeled Mobile Robot with Inverse Mouse-ball Drive-Ballbot*.

[5] Chi-Tsong Chen *Linear System Theory and Design*.