Solution to Homework Assignment II

Abdur Rafay, 6920683

I. Introduction

The task is to design a controller for a plant system. The state space representation of the system shows that the system has three states, three inputs and three outputs. The initial conditions of all the three states x1,x2 and x3 is set to be 1 and the desired final state of all the three is 0. Moreover C is an identity matrix and there is no feed forward D matrix so states are mapped to output by identity matrix. The aim is to design a feedback controller that gives fastest possible response, is economical and with minimum or no overshoots. In a real plant system, achieving all three goals is not completely possible so we will look for a trade between the above said attributes.

Firstly, we will check if the system is safe to experiment. We will check the stability of the system and for this we will check the eigen values of the system matrix A or the poles of the transfer function. It turns out that the all the real parts of the eigen values are negative. This shows that system is asymptotically stable and all the states can reach the origin value in a bounded time. Moreover, controllability of the system is also checked. The rank of the controllability matrix is also full rank so it means system is also controllable. The controllability of a closed loop feedback system is solely determined by plant characteristics. Both state and output feedback do not change the controllability of the plant system.

II. Analysis of Open loop system

The open system is designed on Simulink. The System knowledge for the design is enough. We use A,B and C matrices to design the system with a reference input of zero. We use the initial values of the states as [1 1 1].

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \qquad \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t). \tag{1}$$

We feed these state values in the integrator. The response shows that all the outputs y1 to y3 and the states x1 to x3 reaches to zero after some time. cf. Fig. 1. The system is asymptotically and BIBO stable. Moreover since 2 of the eigen values have imaginary components, that's why it shows and oscillating behavior. The third one is a real negative value so no oscillation is shown.

Now the task is to make the system more efficient by using different types of controllers. We will see the design and the tuning of these controllers and analyse the response.

III. Desgin and tuning of MIMO PI Controller

In order to design a Multiple Input Multiple Output (MIMO) controller, we first check the condition that rank of the gain matrix (Ks) should be at least equal to the dimension of output (y). In our plant model it is equal so integral gain exists.

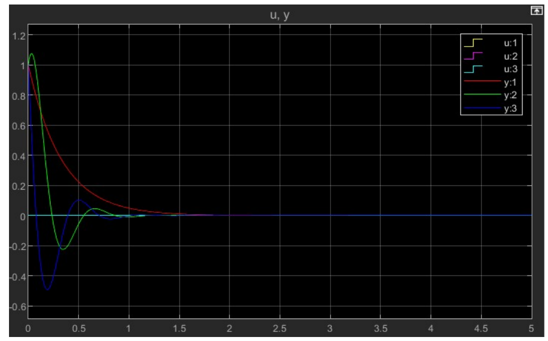

Fig. 1. Simulation results

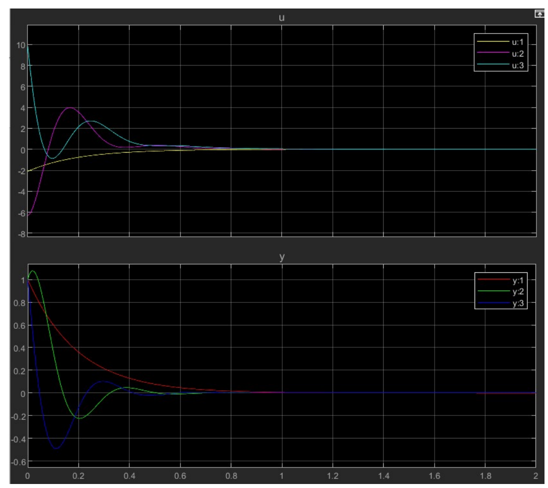

Fig. 2. Input and output simulation results of MIMO PI Controller

$$\boldsymbol{K_s} = -\boldsymbol{C} * inv(\boldsymbol{A}) * \boldsymbol{B} \tag{2}$$

We will design a MIMO PI feedback controller first with an integral feedback with a tuning factor and then to make it more efficient, we will then include a proportional gain with a tuning factor in it later on. The reason behind this is that integral controllers are somewhat slow with unstable output as its show slow response towards the produced error, while using a proportional controller in the system in combination reduces the disadvantage of integral control alone. Proportional controller reduces significantly the steady state error making the system more stable. Overshoots can be a minimized by carefully choosing the value of the tuning parameter.

A. Intergral Controller

In order to design an integral feedback, we first find out the gain matrix Ks and since it is a square matrix, feedback matrix (Ki) for integral control will be a simple inverse of this gain matrix. The design of Ki is such that the integrator's poles strictly tend towards negative real values. Starting with a sufficiently small value of (a) we do not need to consider the plant system poles for this first step since we already have a that calculated that they are asymptotically stable. Second, we slowly increase (a) and test the closed-loop control response after every modification step. This is an empirical trial-and-error procedure balancing the closed-loop response speed versus overshoots and oscillations. By aggressively tuning (a), response will be get more unstable and it will take more time to reach the desired state. So a low value of (a) is chosen to ensure quick and stable response.

$$\boldsymbol{Ki} = inv(\boldsymbol{Ks}) \tag{3}$$

B. Proportional Controller

In proportional design, we first find out the gain matrix Ks and since it is a square matrix.

$$\boldsymbol{Kp} = inv(\boldsymbol{Ks}) \tag{4}$$

Feedback matrix (Kp) for proportional control will be a simple inverse of this gain matrix. Then we tune (b), the tuning parameter, of the proportional control. Starting from zero we slowly increase the value of (b). We choose a slightly higher value of (b) as compared to (a) so that the desired results of quick response with less overshoots can be achieved.

The analyses of the system response shows that as compared to open loop system, the plant will now reach the steady state more quickly. The change in overall stability including overshoots doesn't change much. cf. Fig. 2.

IV. Design and tuning of LQR Controller

LQR is the optimal state-feedback controller that minimizes certain quadratic function. Before using an LQR controller, it has to be assumed that all the states of the system are measurable. If not, an observer that estimates the states by observing the measureable output is required.

In LQR controller design, state (Q) and control (R) weighting matrices are basic parameters of LQR which are tuned by designer using trial and error method. Through the plant description and some values of R and Q, we designed the feedback matrix using lqr function in matlab. We also design a working gain matrix (WW) to be used as a pre-filter to track the reference. Since in this plant system, the desired states are zero, pre filter effect is not that visible but once we change the reference from zero to some other value, we can clearly see the effect of using a pre filter. Although the LQR cost function regulates the states to the origin, a constant reference signal r can be considered without changing the optimal control framework

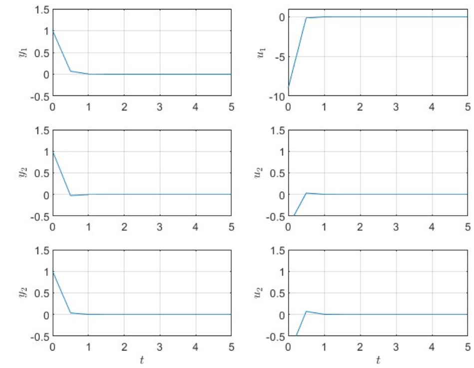

Fig. 3. LQR Controller input vs Output

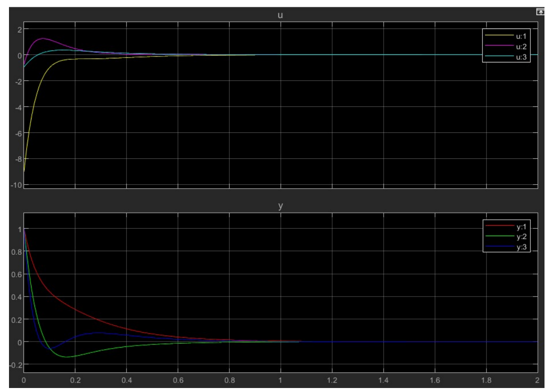

Fig. 4. Input and output simulation of LQR Controller

While setting the weights of the weighting matrices Q and R, certain constraints are kept in consideration. The designed problem has an input constraint such the all the inputs should be less than absolute value of 10. In this scenario, we have to assign values to weighting matrix in such a way that the system should follow the constraint and give the best possible response.

The Q matrix (symmetric and positive semi definite) penalizes state deviations from the origin. Is often a diagonal matrix, because then the individual states are evaluated independently, which facilitates the design. The control engineer decides which state should be regulated faster than others. While the R matrix (symmetric and positive definite) R penalizes the control effort which is particular useful if u(t) is associated with some kind of power or rare resource. Limiting the control actions leads to a trade-off decision compared to the controller dynamics. Also, a diagonal matrix can be applied to ease the design and interpretation. Generally a trade off is considered between time to reach the steady state and effort (input) applied to reach it. If anyone of the above is a constraint, then the designer will go for the best trade off. cf. Fig. 3.

The analyses of the system response shows that as compared to PI controllers, the plant will now reach the steady state a bit quickly but the main advantage is that system has now less overshoots and more stability. cf. Fig. 4.

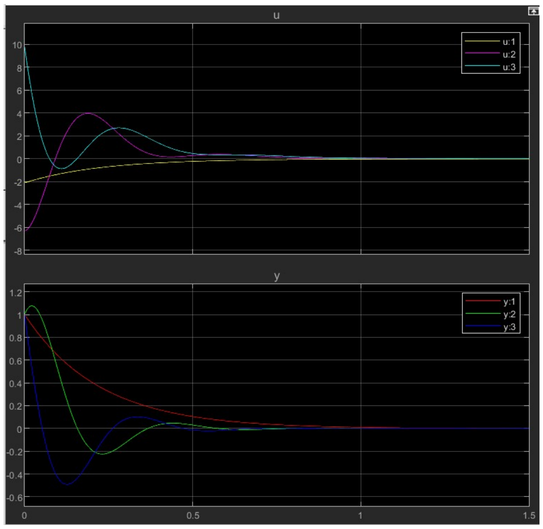

Fig. 5. Model Mismatch with PI controller

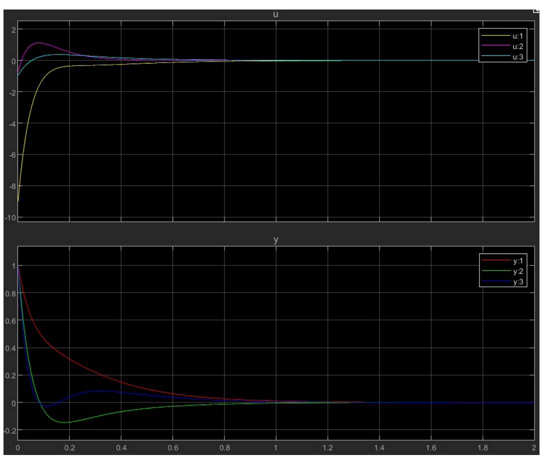

Fig. 6. Model Mismatch with LQR Controller

V. Comparison between the controllers keeping in view the Constraints

In the given plant model, input (less than or equal to absolute value of 10) is the constraint. We have designed a MIMO PI controller and plot its input and output response keeping in view the constraint. We also design an LQR regulator with the same approach and plot its input and output response too. Following are the key points of comparisons

- With the design of PI controller and after careful tuning, we can analyse that with in given input constraints, steady state can be achieve quickly but there will be obvious unstability and some overshoots in the plant system.
- With the design of LQR controller and after giving appropriate weights to the Q and R keeping in view the input constraints, we can analyse that overshoots has been significantly reduced and the system is more stable than that of the PI- controller but now it takes a little more time to reach the steady state.

VI. Model Mismatch

MIMO models will usually only are approximate descriptions of any real system. Thus the performance of the nominal control loop may significantly differ from the true or achieved performance. PI and LQR models are somehow more robust when subjected to external disturbances and model mismatches. It can be seen in our case too as well. After modifying the system matrix to 0.8A, and with the same feedback controllers, there is no significant difference in the system response. The response of PI controller hasn't change much as compared to the previous A matrix. It can be cf. Fig. 5. It is also visible by comparing Fig 4 and Fig 8.

Same is the case with the LQR Controller. The response of LQR controller hasn't change much as compared to the previous A matrix.cf. Fig. 6. It is also visible by comparing Fig 7 and Fig 9.