|  |
| --- |
| ME 389 – Spring 2016 |
| MEM 02 – Hydraulic Servomechanism |
| In this lab we will applied closed loop feedback control to a hydraulic system. This system transfer function was identified using a bode diagram. Using this transfer function, controllers were designed to regulate angular velocity and position of the servomotor. Controllers were tested experimentally to compare to our assumed system. |

|  |
| --- |
| Malcolm Erdongen, Brian Loughran  2-23-2016 |

**Introduction:**

Servomechanisms (or servos) are commonly used to provide position control. Servos operate on the concept of feedback, where a controlled input signal is compared to the true position of the system (generally measured by a transducer at the output). The difference between input and output is then used to drive the system in the direction needed to eliminate the error between the two. Servos can give either linear or angular output; in our case we will be working with an angular servomechanism.

In this lab, we will first identify the system using frequency response (bode diagram). Once this is complete, we will create a velocity PI controller for this transfer function, and compare the output for our theoretical and experimental transfer functions. Using our velocity transfer function, we will integrate and create a position transfer function. A PD controller will be designed to optimize performance, for which theoretical and experimental data will be again compared.

**Apparatus and Procedure**

The two most important components of the hydraulic servomechanism in terms of modeling the system are the servo-valve and the hydro-motor. For this system, an electric pump delivers oil from the reservoir through the hydro-motor. The hydro-motor spins at a rate proportional to the amount of oil flowing through the system. The hydro-motor works by transferring flow energy to angular velocity. A tachometer measures the angular velocity by changing the angular speed into a voltage proportionate to ω. Position error voltage or speed voltage are then inputs to the PC.

At this point, using our designed controller, we send a control signal to the servo-valve, therefore changing the flow speed through the hydro-motor. Controllers were be designed to optimize performance of this system.

**Control Design: Velocity**

For the given model of the experimental system, we considered a P controller. P controllers are designed to decrease steady state error, which for our system was about 15% without a controller. While a P controller cannot ever guarantee steady state error, it does decrease error substantially. P controllers also have the added benefit of speeding up the response of the system. A large gain of KP  = 100 will give less than 1% steady state error in simulation. Our system cannot be destabilized by any value for KP. A plot of the closed loop pole as a function of the gain of the P controller is shown below (Figure 1).

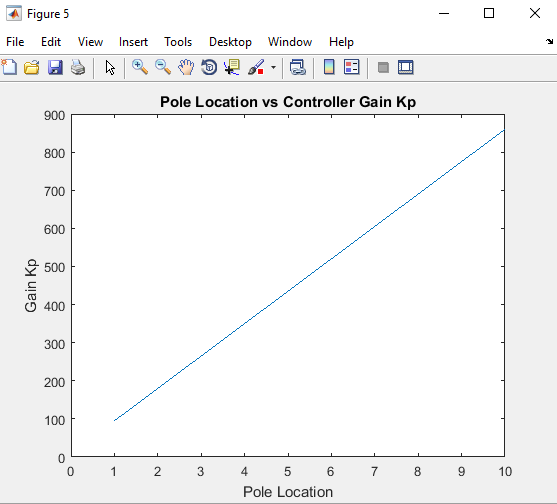


Figure 1 – Pole Location vs. Controller Gain (KP)

Unfortunately for us, our model saturates before we reach negligible overshoot. Our input voltage cannot exceed 5.6 volts due to physical limitations of the system. In our case steady state error is still 8.2% near saturation for a .5V step function. This can be improved with a PI controller.

Using a PI controller we can completely eliminate steady state error, as well as improve the response of our system. A PI controller comes in the following form (equation 1).

Values for KP and KI can be determined based on the desired response of the system. Necessary values this system had to meet were an overshoot less than 10% and a rise time of less than 0.5 s. Equations relating overshoot and settling time to damping and natural frequency (respectively) of the system are given below (equations 2 & 3), and will be used later to choose values for KP and KI.

Using given values for overshoot and rise time, we found that a system meeting these specifications would have a natural frequency (ωn) of 15.5 Hz, and damping of .591.

Closing the loop of our controller requires the closed loop transfer function equation. Generally, our model (or plant) is denoted G(s), and our controller is denoted H(s). This equation to close the feedback loop is given below (equation 4)

The left hand side of the equation represents the expected output voltage over the input voltage. The denominator of such a transfer function is expected to follow the same quadratic form shown in equation 5:

Following this process, a controller with the following values for KP and KI should give experimental results within the bounds given (equation 6).

Because of the integrator in the controller, our step function for our theoretical case saw no steady state error (unlike our P controller above). Overshoot was 7%, and settling time was 0.42 s, which was as expected, and within the specified range.

**Control Design: Position**

For position controller design, we were first asked to design a P controller to limit overshoot to less than 10%. Using equations 2, 4, and 5, we are able to find a value for KP to guarantee expected overshoot. Our value for KP was found to be .842. Closing the loop and plotting step response in MATLAB produced an overshoot of almost exactly 10%. A figure showing expected step response is shown below (Figure 2).

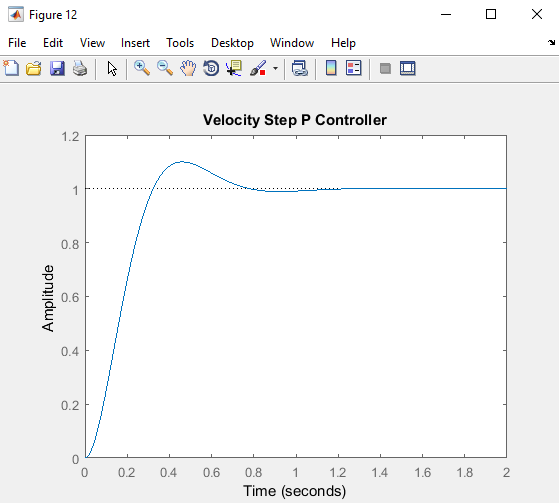


Figure 2 – Velocity step response using closed loop P controller

Performance can again be improved by adding tachometer feedback. This comes in the form of a PD controller. A generalized form of a PD controller is shown below (equation 7).

Using the PD controller, we can further limit overshoot, but at the cost of slightly increasing rise time. Considering our transfer function already has a relatively quick rise time (tr= 0.22 s) this is acceptable.

Our PD controller was found to have a value of 0.05. This gave a decreased overshoot value (now 7.3%), but slightly increased rise time (now 0.3 s). It is important to note that PD controllers do not guarantee zero steady state error. A plot showing step response to our PD controller is shown below (Figure 3).

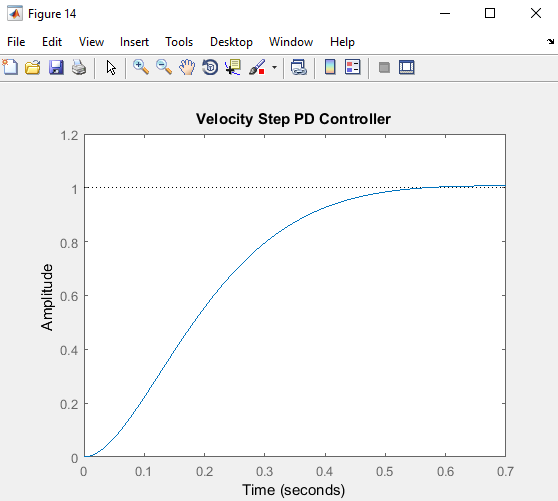


Figure 3 – Velocity step response using closed loop PD controller