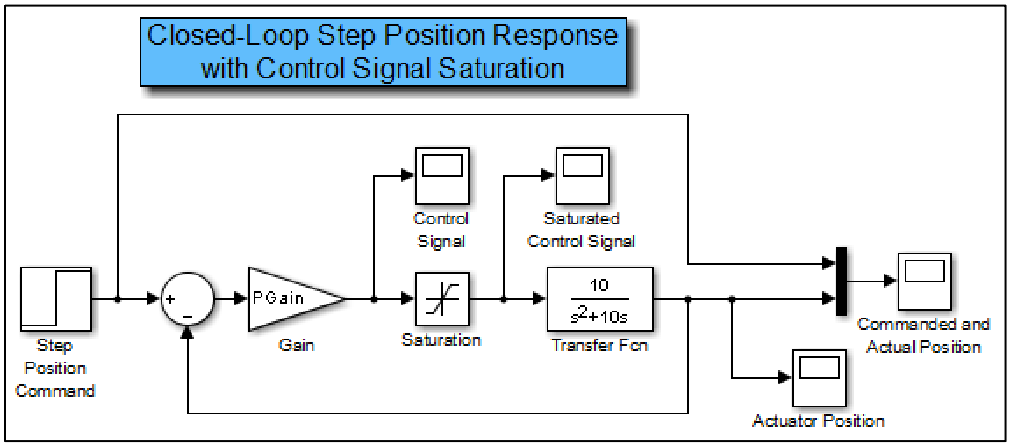
Introductory Motion and Control
Saturation, Pre-Filters, Motion Trajectories, and Feedforward

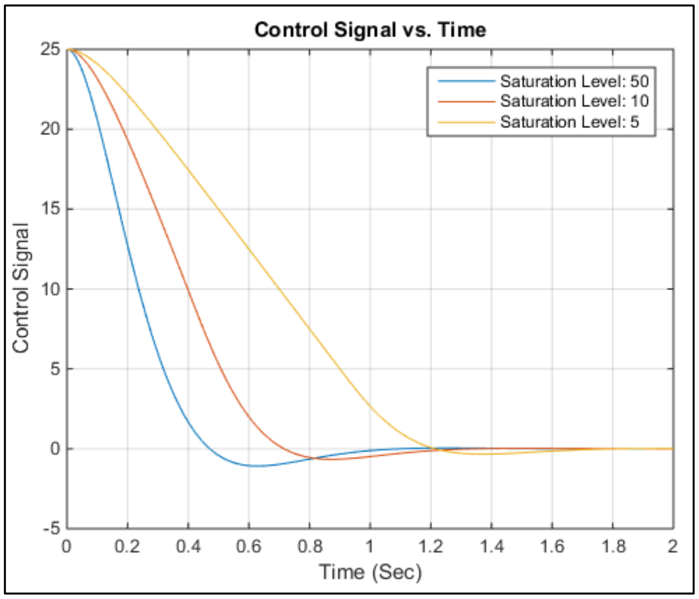
The notes that follow discuss the effects of component saturation and how pre-filters and trajectory planning can be used to avoid these effects. The concepts are applied to a simple actuation system as an illustration.

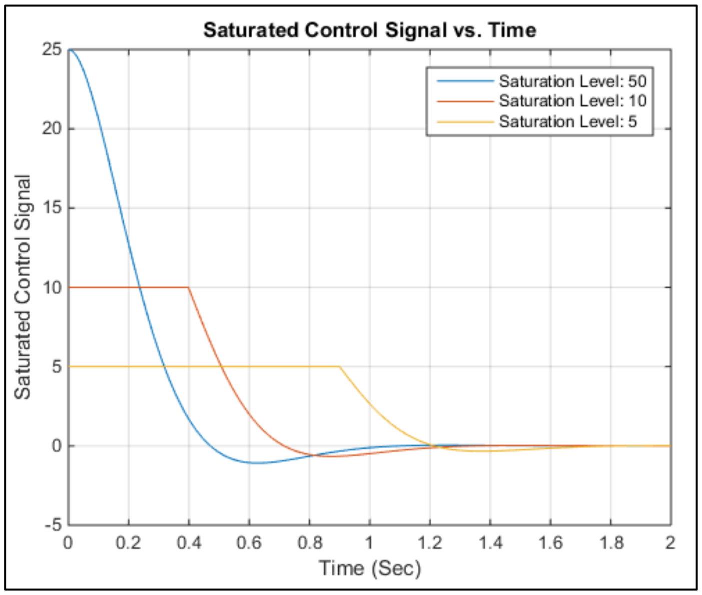
Effects of Actuator Saturation

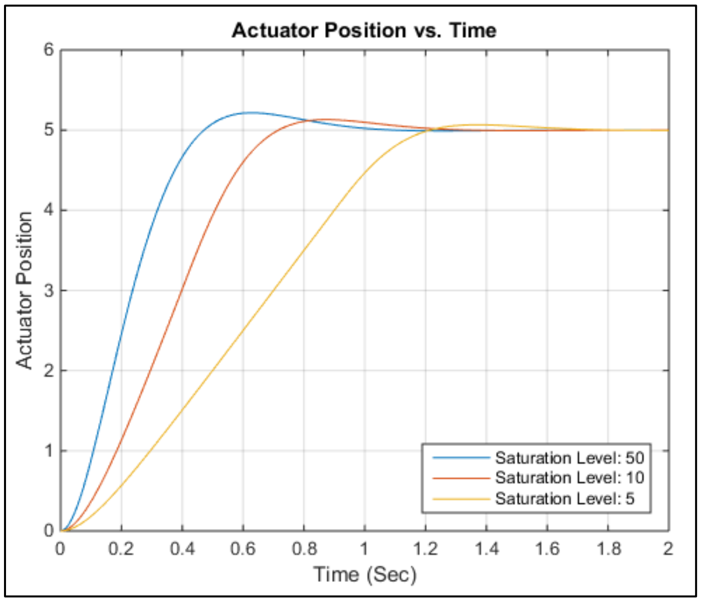
- ***Saturation*** occurs in a motion control system when any of its components reach their ***maximum capacity***.
  - ***Electro-hydraulic valves***, for example, often accept input of ± 10 volts to open their ports completely and develop maximum flow rates.
  - ***DC electric motors*** have similar voltage limitations.
  - ***Aerodynamic stall*** of a wing or flap occurs when the angle of attack is too large.
- ***Saturation*** of components is often ***undesirable***
  - Components are operating at maximum capacity
  - System behaves as an ***open-loop*** system, because the control signal does not vary with the feedback.
- ***Saturation*** often occurs in a control loop when a ***step*** position command is given due to the ***large error*** when the command is initiated.
- Saturation is usually ***not accounted for*** when the proportional control gain is determined. The Simulink model shown below is used to study how big an effect saturation will have on system performance.
- ***With saturation***, the system will generally be ***slower*** than the system without saturation.



- To illustrate the characteristics mentioned above, the Simulink model is executed with a ***proportional gain*** `PGain` = 5 and signal ***saturation levels*** of 50, 10, and 5. The results are presented in the ***three diagrams*** below for the ***control signal***, the ***saturated control signal***, and the ***actuator position***.
- With a saturation level of 50, the control signal and saturated control signals are identical. So, in this case, the saturation has no effect. At saturation levels of 10 and 5, the control signals are clearly altered. The ***lower*** the ***saturation level***, the more the ***maximum actuator speed*** is ***reduced***, and the ***longer*** it takes for the actuator to reach its final position.
- Note that because the actuator is a ***type one system***, the saturation ***does not alter*** its ***steady-state position***.

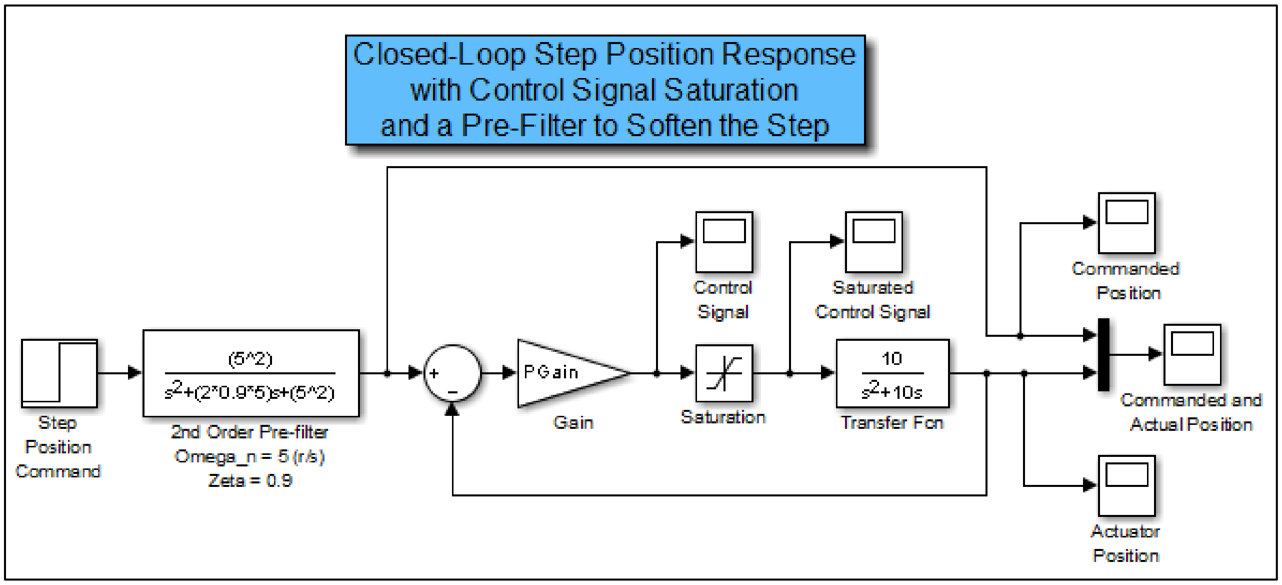




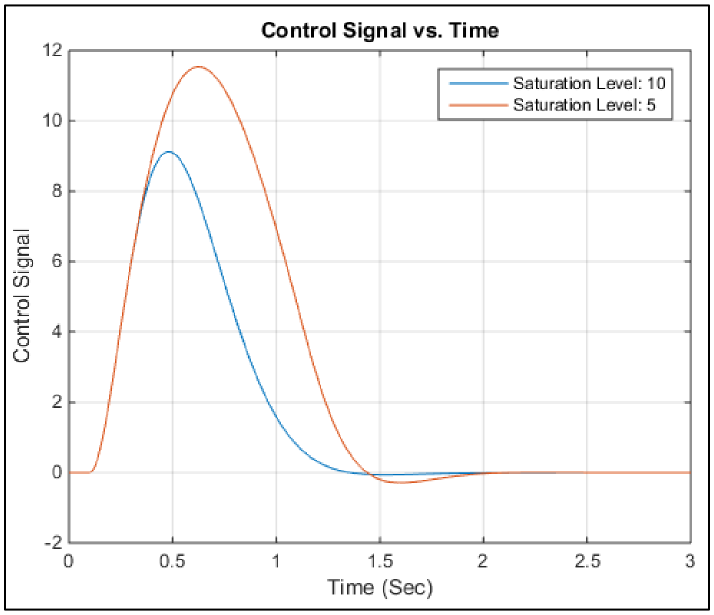


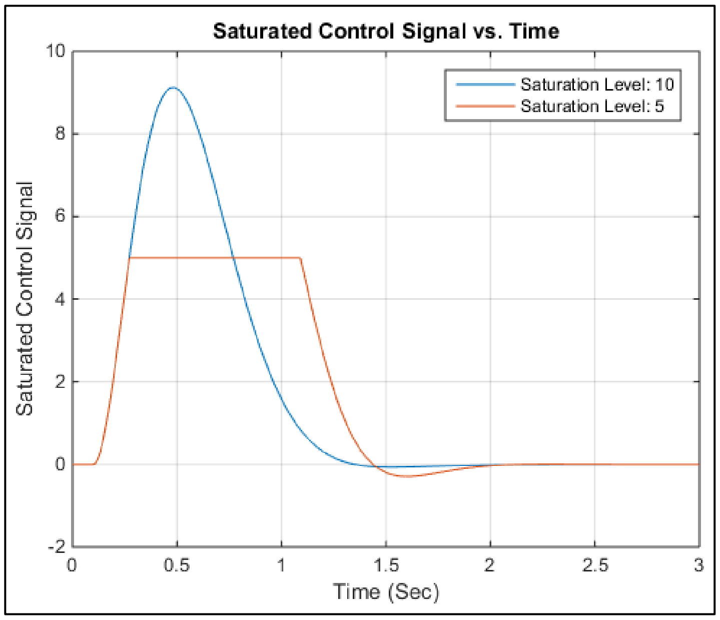
Use of Pre-Filters to Soften the Step Input

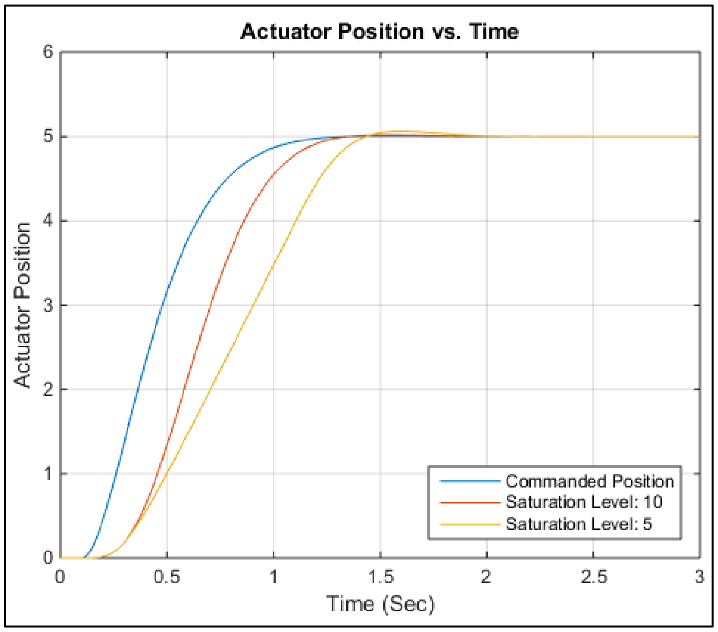
- The model below uses a ***normalized*** second-order ***pre-filter*** to make the input to the control loop ***rise more gradually*** and ***smoothly*** than the step command.
- The net effect of the pre-filter is to:
  - ***Lessen*** the ***position error*** at any time, and hence ***avoid saturation***
  - ***Slow down*** the system response



- To illustrate the characteristics mentioned above, the Simulink model is executed with a ***proportional gain*** $\texttt{PGain} = 5$ and signal ***saturation levels*** of 10 and 5. The results are presented in the ***three diagrams*** below for the ***control signal***, the ***saturated control signal***, and the ***commanded*** and ***actuator positions***.
- The use of a pre-filter has ***lowered*** the system's saturation level below 10 by ***softening*** the step command. As shown in the third plot, the filtered step command takes roughly 1.25 seconds to reach the final position. This clearly lowers the system's position error in the initial portion of its response.
- When the saturation level is lowered to 5, the control signal is clearly altered. As before, when the control signal is saturated, the actuator ***speed*** is ***reduced***, and the ***time*** required to reach the final position is ***increased***.

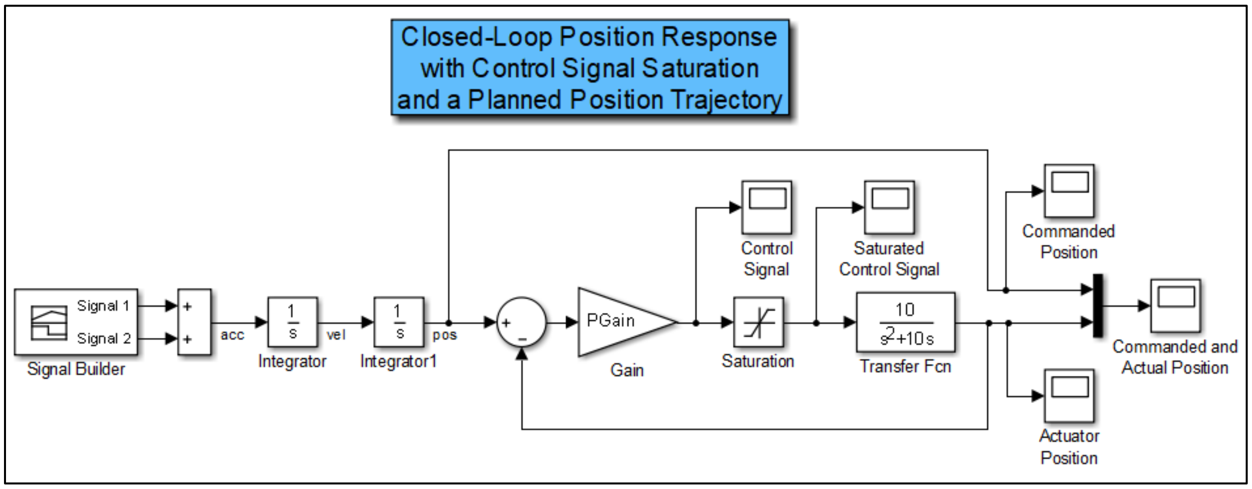


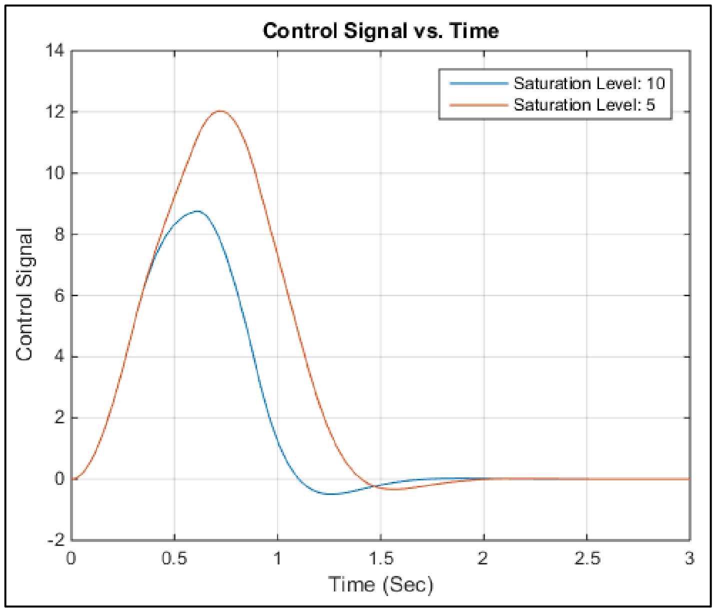


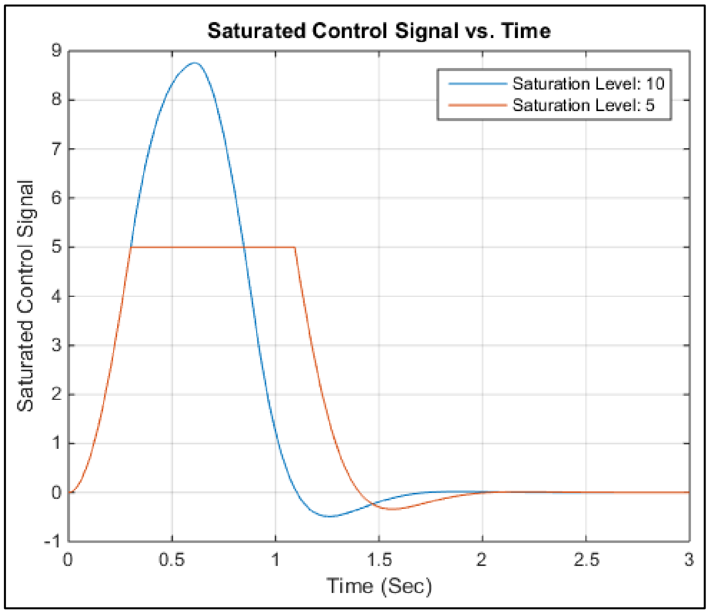


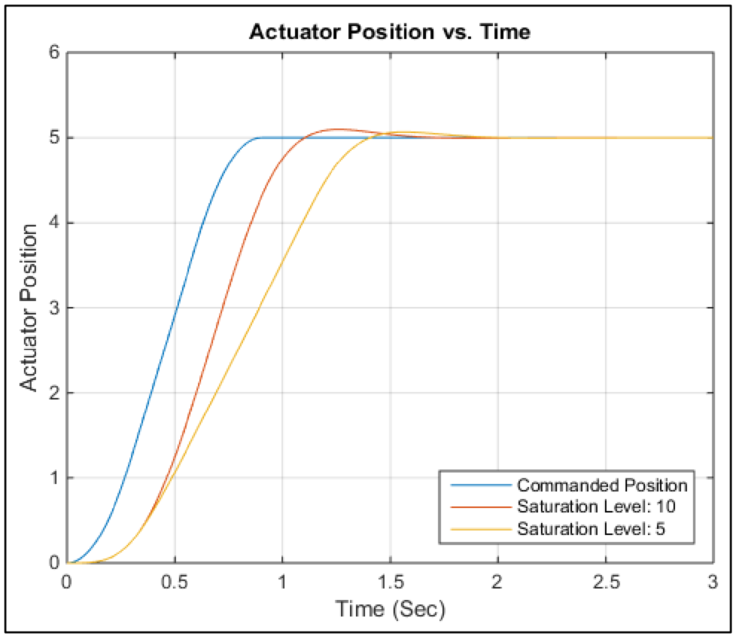
Use of a Motion Trajectory

- A ***motion trajectory*** is another method of supplying a softer, smoother input than the step function. As with the pre-filter, the ***system response is slower***.
- In the model below, a ***signal builder*** is used to generate a simple ***acceleration profile*** which is then ***integrated twice*** to generate the ***position command***. The motion profile commands the system to ***accelerate*** from rest at a ***constant rate***, then move at a ***constant velocity***, and finally ***decelerate*** at a ***constant rate*** before coming to rest in the final position.
- ***Similar results*** are obtained as with the pre-filter; however, the trajectory is calculated directly to produce a desirable motion profile for the system. As before, `PGain` $= 5$.


Closed-Loop Position Response with Control Signal Saturation and a Planned Position Trajectory
Signal 1
Signal 2
Signal Builder
acc
vel
pos
Integrator
Integrator1
PGain
Gain
Control Signal
Saturation
Saturated Control Signal
10
s^2+10s
Transfer Fcn
Commanded Position
Commanded and Actual Position
Actuator Position

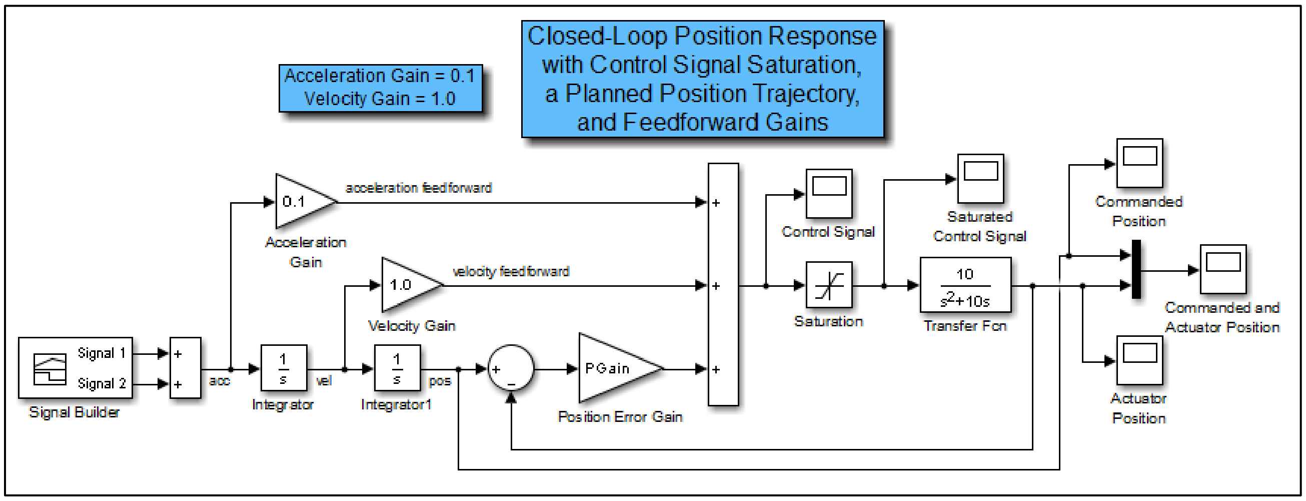

Control Signal vs. Time
Saturation Level: 10
Saturation Level: 5
Control Signal
Time (Sec)


Saturated Control Signal vs. Time
Saturation Level: 10
Saturation Level: 5
Saturated Control Signal
Time (Sec)


Actuator Position vs. Time
Commanded Position
Saturation Level: 10
Saturation Level: 5
Actuator Position
Time (Sec)

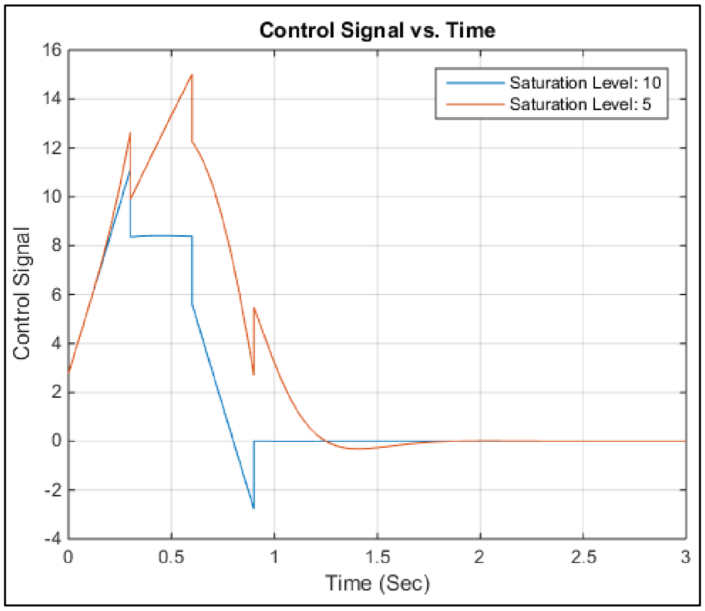
Use of a Motion Trajectory with Velocity and Acceleration Feedforward

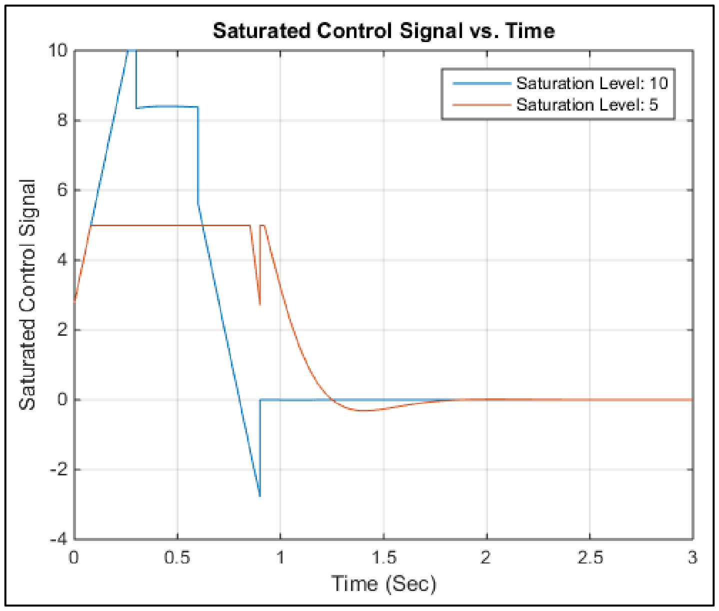
- Using a ***motion trajectory*** as described above allows access to not only the ***position*** commands, but also the corresponding ***velocities*** and ***accelerations***.
- Velocity and acceleration values can be ***fed forward*** into the control loop to ***anticipate*** the ensuing position commands.
- If carefully designed, the result is a system that ***follows*** the ***command trajectory*** with ***little error without saturating*** the system. The acceleration and velocity feedforward gains are found using trial and error.

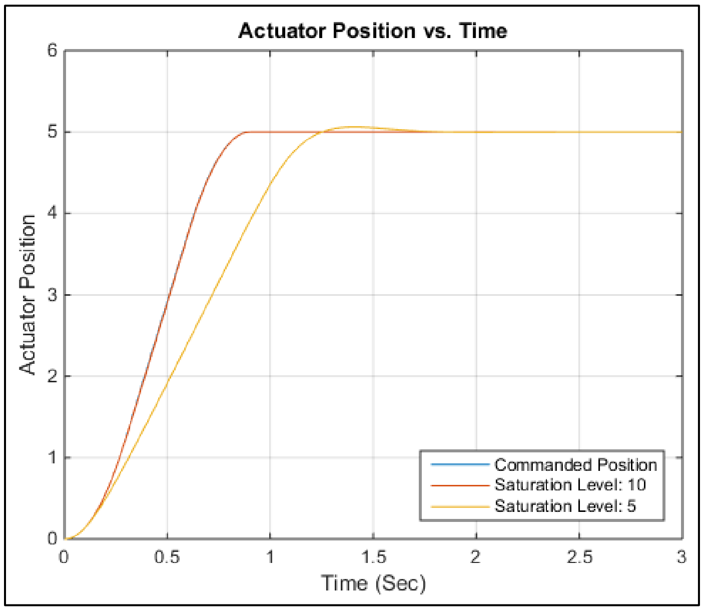


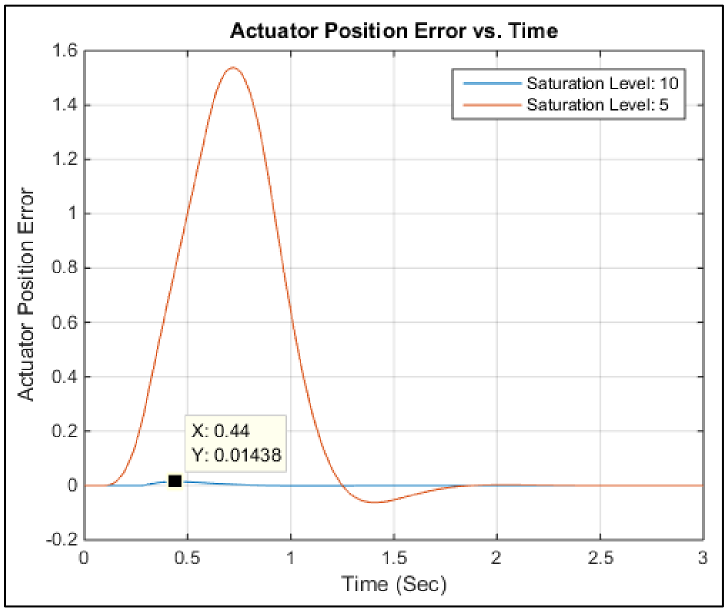
- The motion profile used in this model is the ***same*** as that used in the model above. The command is for a ***constant acceleration***, followed by a ***constant velocity*** (zero acceleration), and finally by a ***constant deceleration*** to the final position. Also, $\mathtt{PGain} = 5$.
- As illustrated by the top two plots below, the ***control signals*** exhibit ***step changes***, because of the ***discontinuities*** in the ***acceleration profile***.
- Note that at a saturation level of 10, the control signal has a ***small portion*** that is ***saturated*** at about $t = 0.25$ (sec), but this has little effect on the overall response. The saturation is much more significant a level of 5.
- The plot on the bottom left shows the commanded trajectory and the trajectory responses for saturation levels of 10 and 5. The ***trajectory response*** at a saturation level of 10 is ***nearly identical*** to the commanded trajectory.

- The plot on the bottom right shows that with a saturation level of 10, the ***maximum difference*** between the commanded and actual trajectories is a roughly 0.0144 at $t = 0.44$ (sec). Hence, the system is tracking the trajectory with very little error.


Control Signal vs. Time
Saturation Level: 10
Saturation Level: 5
Control Signal
Time (Sec)


Saturated Control Signal vs. Time
Saturation Level: 10
Saturation Level: 5
Saturated Control Signal
Time (Sec)


Actuator Position vs. Time
Commanded Position
Saturation Level: 10
Saturation Level: 5
Actuator Position
Time (Sec)


Actuator Position Error vs. Time
Saturation Level: 10
Saturation Level: 5
X: 0.44
Y: 0.01438
Actuator Position Error
Time (Sec)