Using an integrator or a lag compensator can lead to problems when the actuator saturates.

Consider:

Now, put into a real system:

Output of a saturation becomes constant if input exceeds \( u_{\text{max}} \) or \( u_{\text{min}} \).
What Happens?

Integrator builds up (stores) uncorrected error!

Integrator "unwinding" → generating error!

At this point, error is low enough, but integrator is "unwound"

Δ slope is fixed, since actuator is saturated "up"

Integrator sees larger error → builds up more "unsatisfied command"

Before - integrator builds up error, causing slight overshoot

Saturation

Saturation
Note: Although worst w/ a pure integrator, a slow pole can exhibit the same phenomenon.

Solution: Anti-Windup (from Franklin, Powell, & Emami-Naehi)

Unsaturated: \( u_{cc} = u_c \), no feedback

Saturated:

If \( K \) is large, this feedback drives \( u_{cc} \rightarrow u_c \) very quickly, \( e_i \rightarrow 0 \) I.O.W.I.S., integrator stops integrating
Equivalent Implementation:

```
+---------------------
|                      |
|   Kp                |
| +------------+      |
|     | e_r         |
|     +------------+      |
|                       |
|   Kf                |
| +---------------------
|      uc             |
| (servo)             |

"dead band" nonlinearity

\[ f / \Delta K \]
\[ -K \sqrt{u_{min}} \quad u_{max} \quad uc \]

does nothing until saturated, then cancels out error quickly

Easy to implement

In a digital system