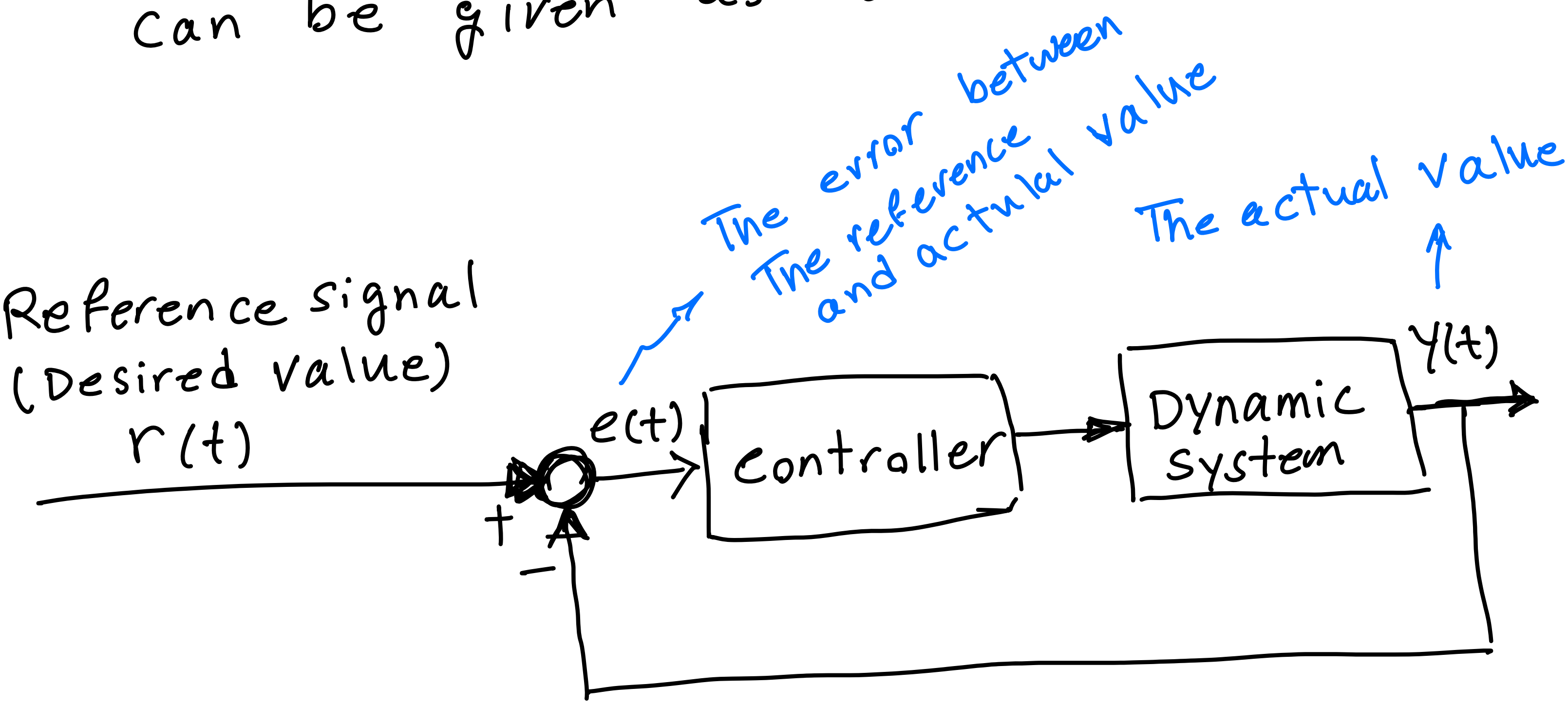
PID for Dynamic systems

A feedback control block diagram can be given as below.

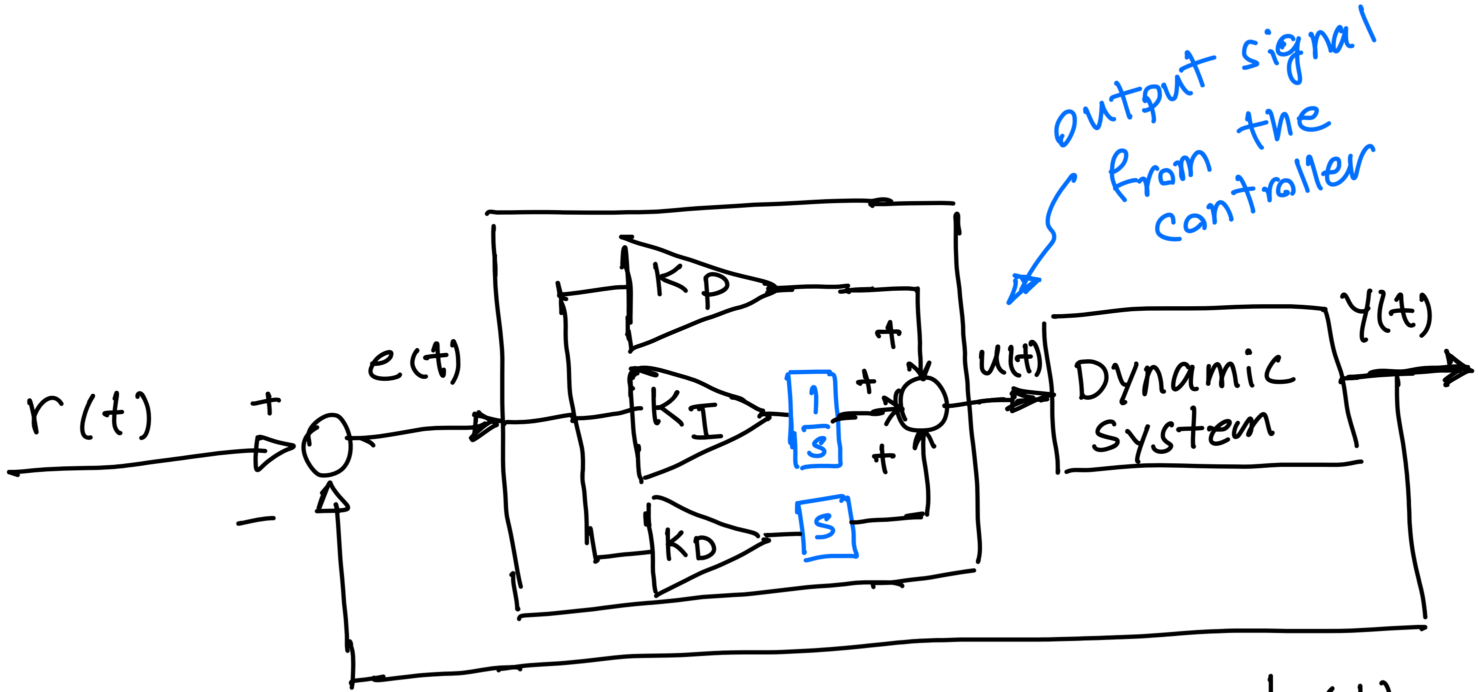


The error: $e(t) = r(t) - Y(t)$

$e(t)$: The error

$r(t)$: The reference or desired value

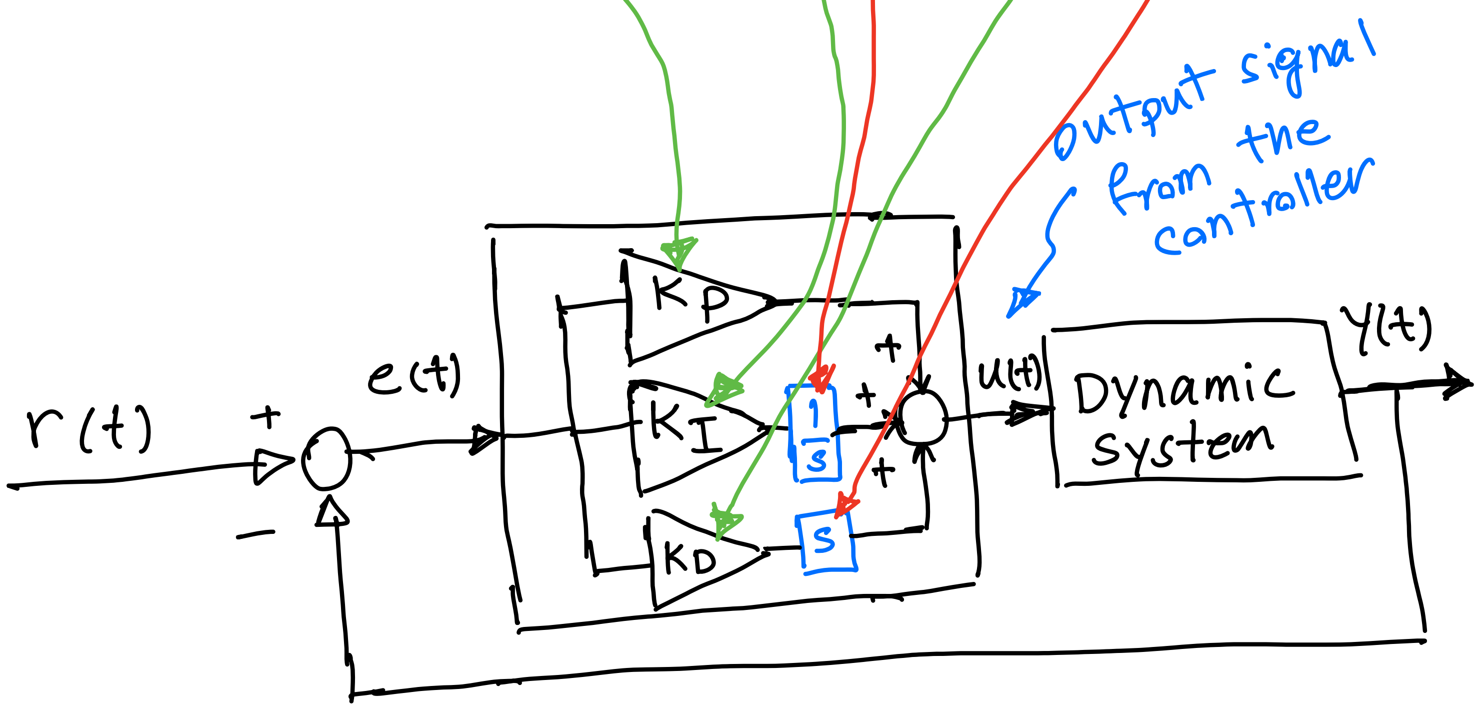
$Y(t)$: The actual value of the system



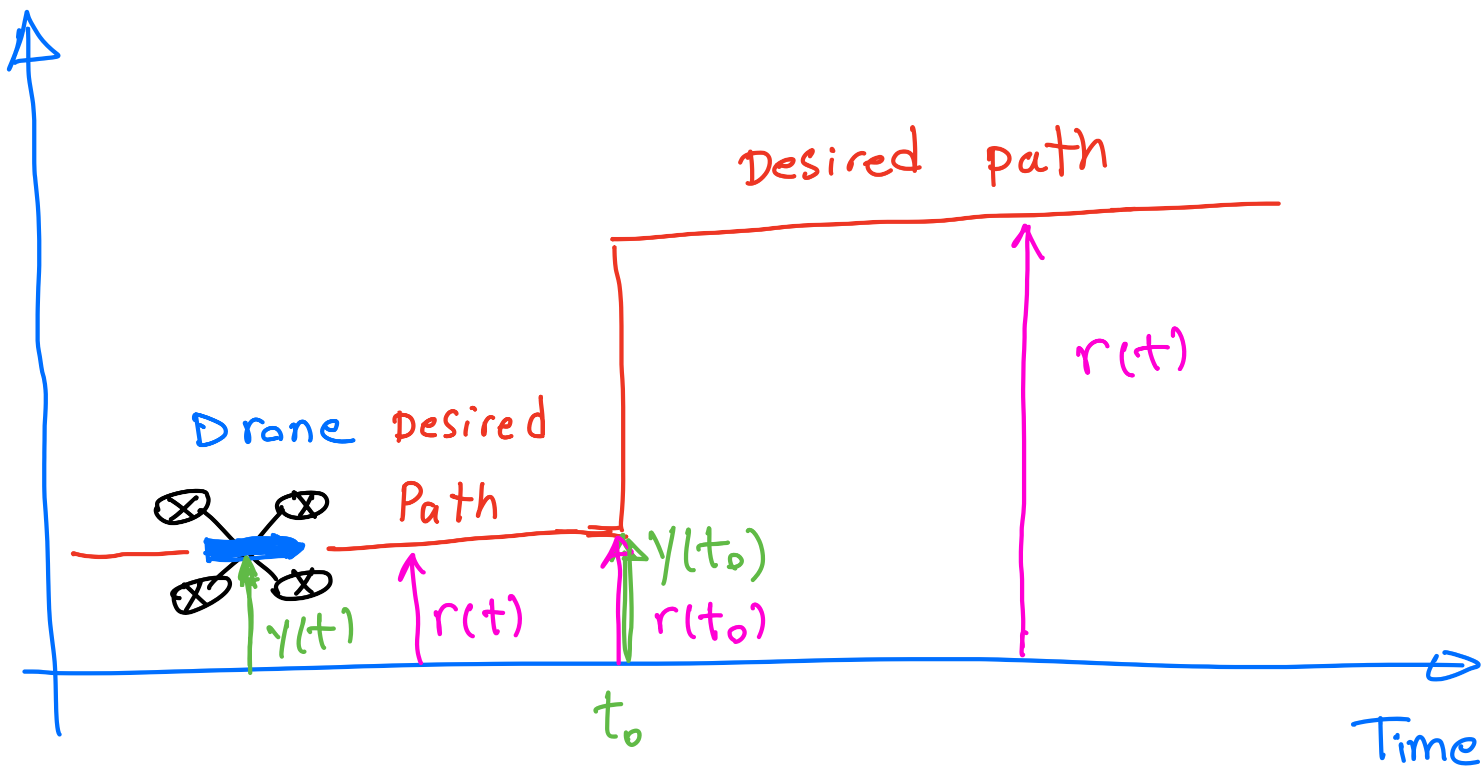
$$u(t) = \underbrace{K_p\, e(t)}_{\text{proportional controller with proportional gain of } K_p} + \underbrace{K_I \int e(t)\,dt}_{\text{Integral controller with the gain of } K_I} + \underbrace{K_D \frac{de(t)}{dt}}_{\text{Derivative controller with the gain of } K_D}$$

The Laplace representation of u(t) is:

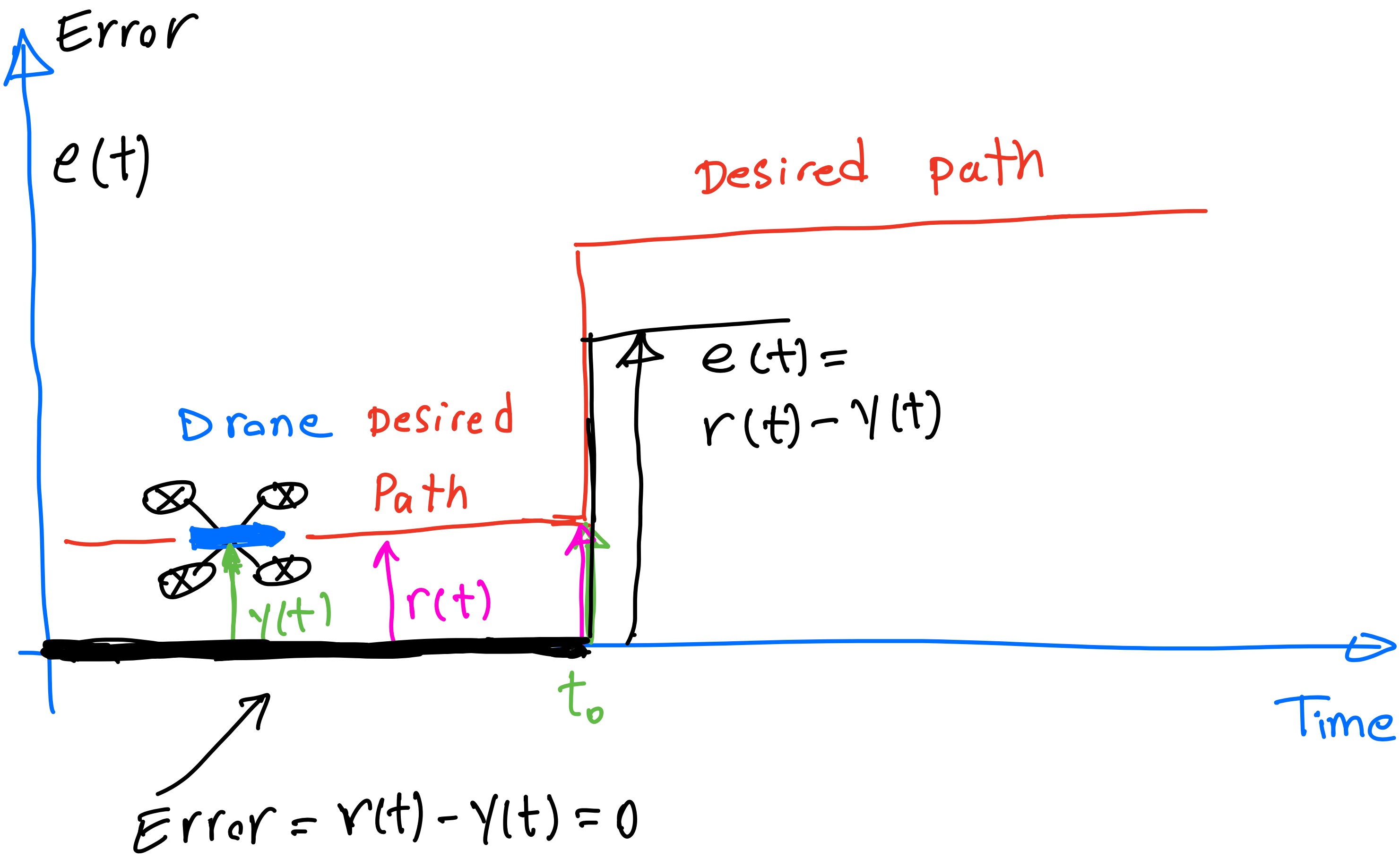
$$U(s) = K_p\, E(s) + K_I \frac{1}{s} E(s) + K_D\, s\, E(s)$$



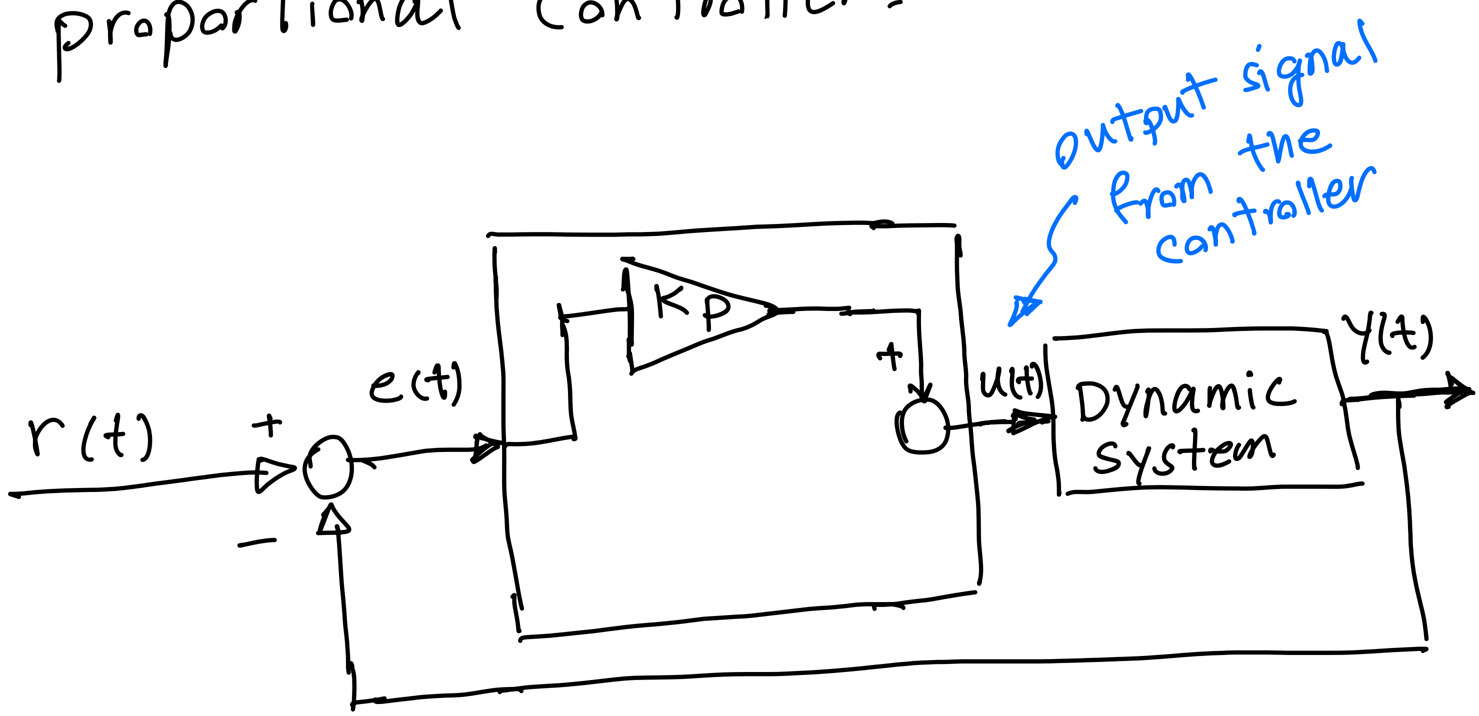
An example of a moving vehicle following a reference path (or desired path)



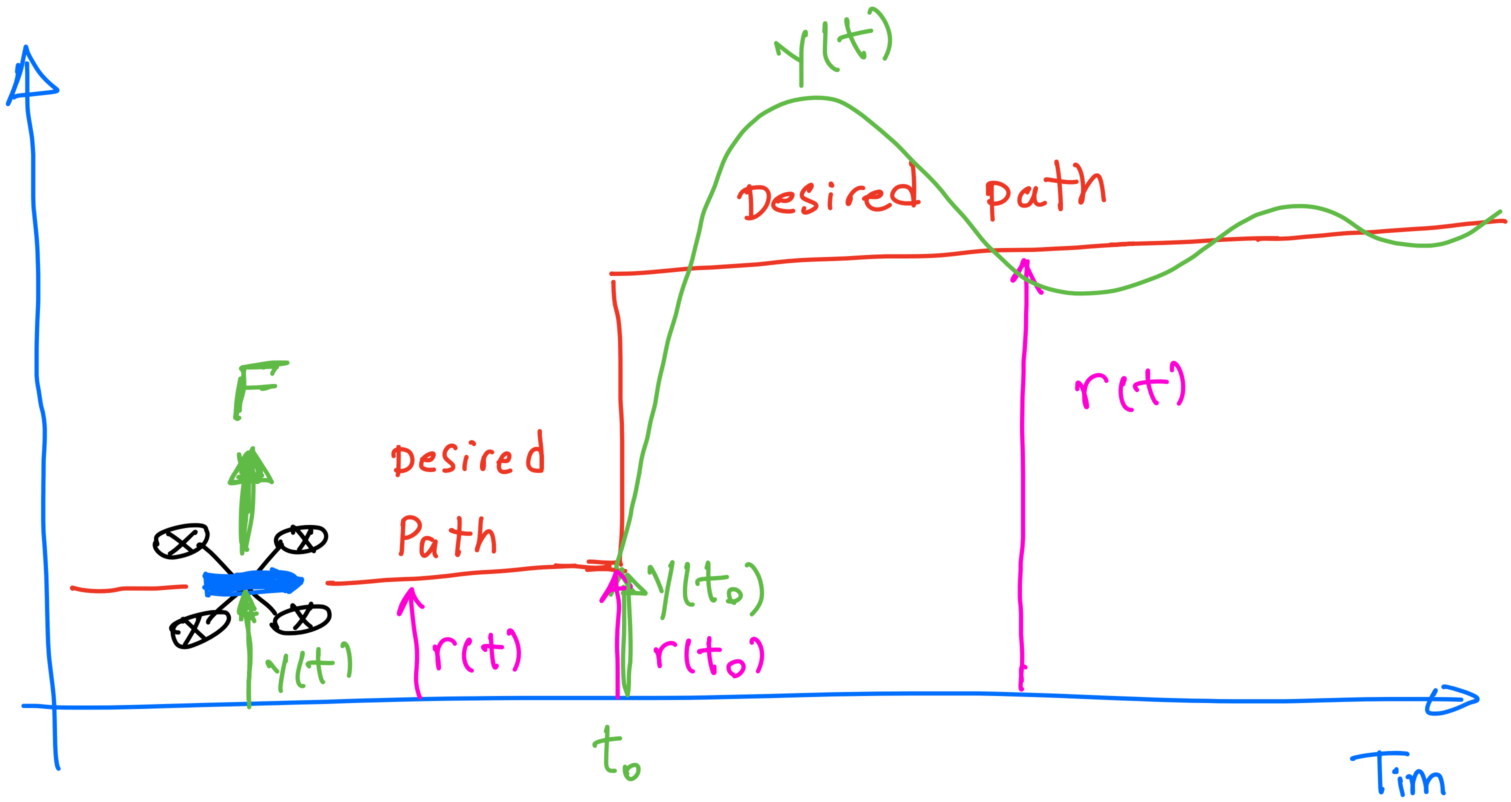
$$e(t_0) = r(t_0) - Y(t_0) = 0$$



No, let's use the controller to control the motion. Start with only a proportional controller.



A possible response can look like the path of motion as follows.



Force = mass x acceleration

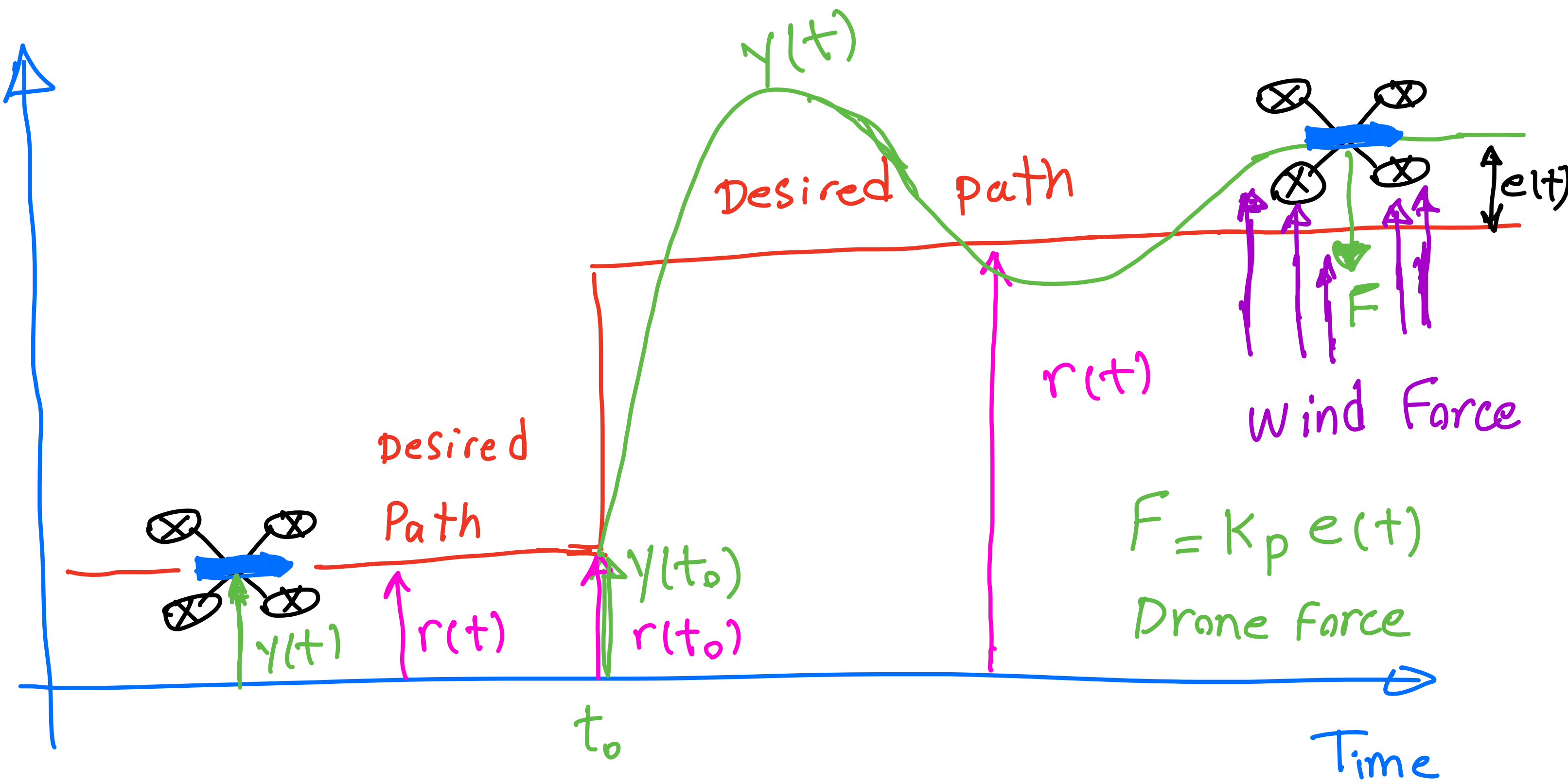
$\longrightarrow$ $F = ma$ or $F = m\ddot{y}(t)$

$F(s) = m s^2 Y(s)$

The transfer function for the drone $\longrightarrow$ $\dfrac{Y(s)}{F(s)} = \dfrac{1}{ms^2}$

Note: Increasing the K_p will make the drone response faster (However, too large value of K_p

can make the drone unstable which will be addressed in the derivative control section later)



The wind force is pushing the vehicle away from the desired path. Also, the error can be small.

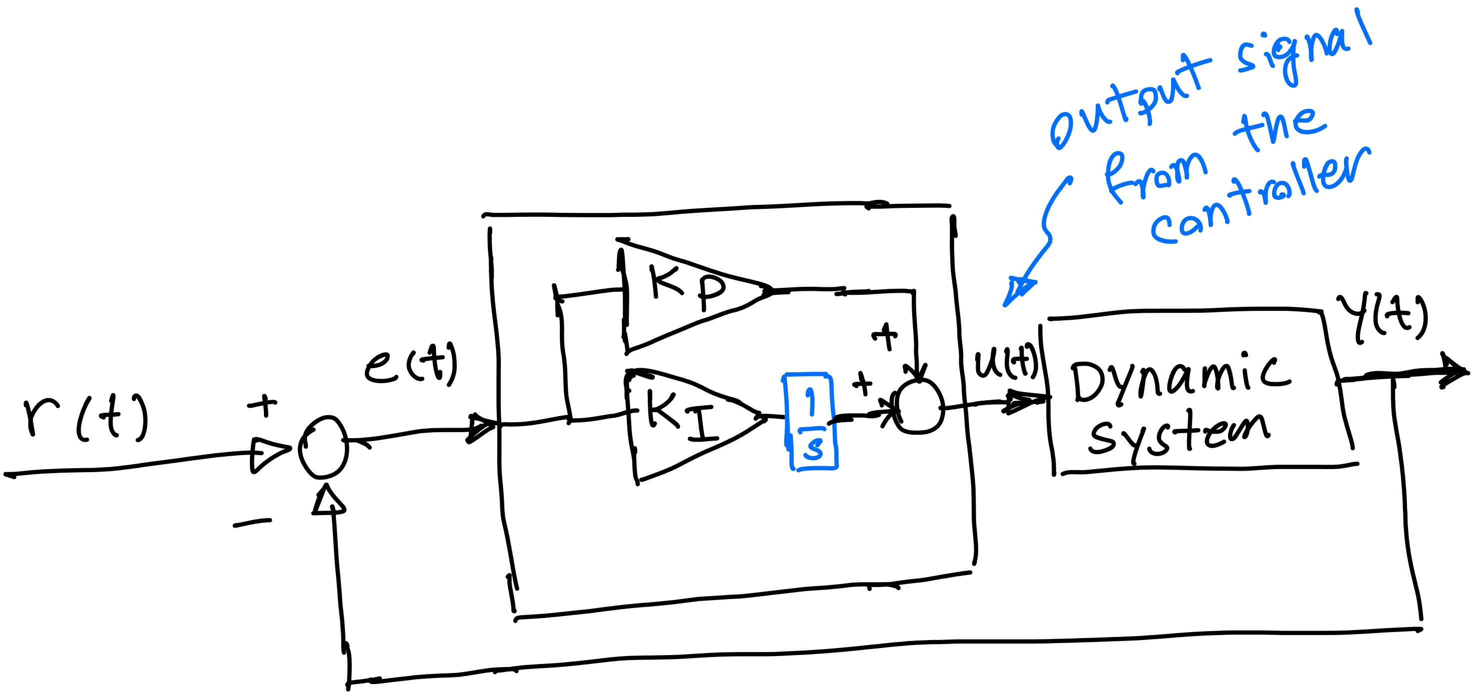
$$\underbrace{F}_{\text{Drone force}} = \underbrace{K_p}_{\text{proportional gain}} \underbrace{e(t)}_{\text{error}}$$

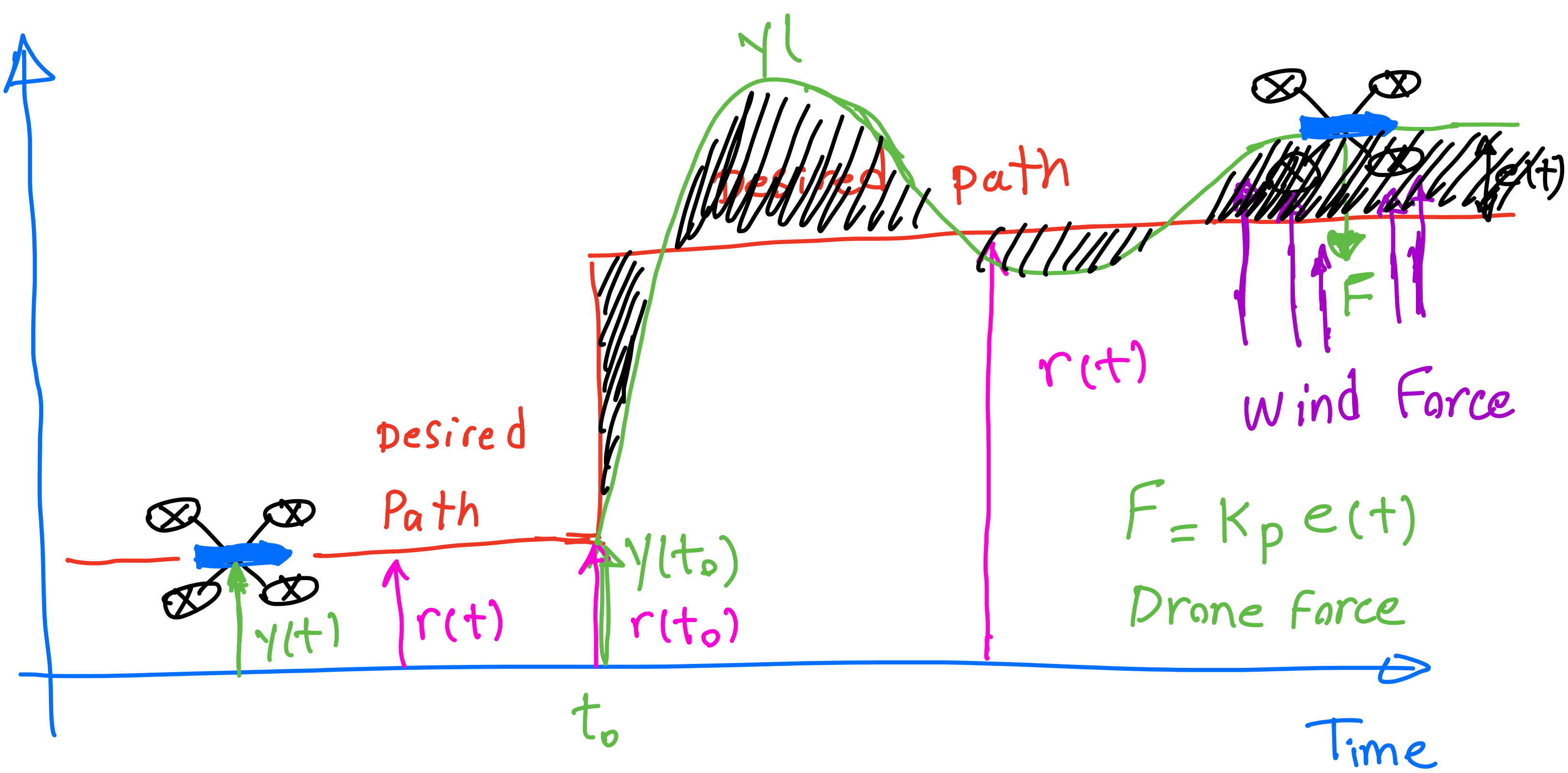
If the error ($e(t)$) is small, therefore the control signal is small, as it is proportional to the error ($K_p e(t)$). This small value can be smaller than the wind Force.

$$\text{wind Force} \gg \text{Drone force}$$

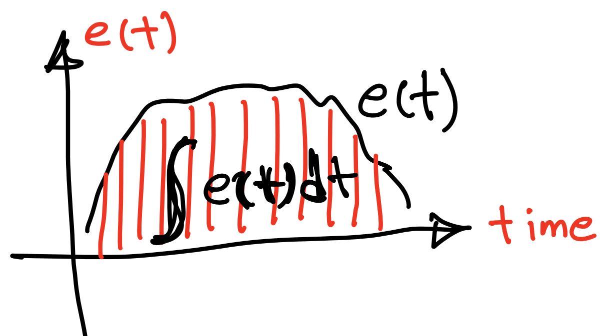
If wind Force = Drone force the drone is not able to compensate the error and there will be steady-state error in the path of motion.

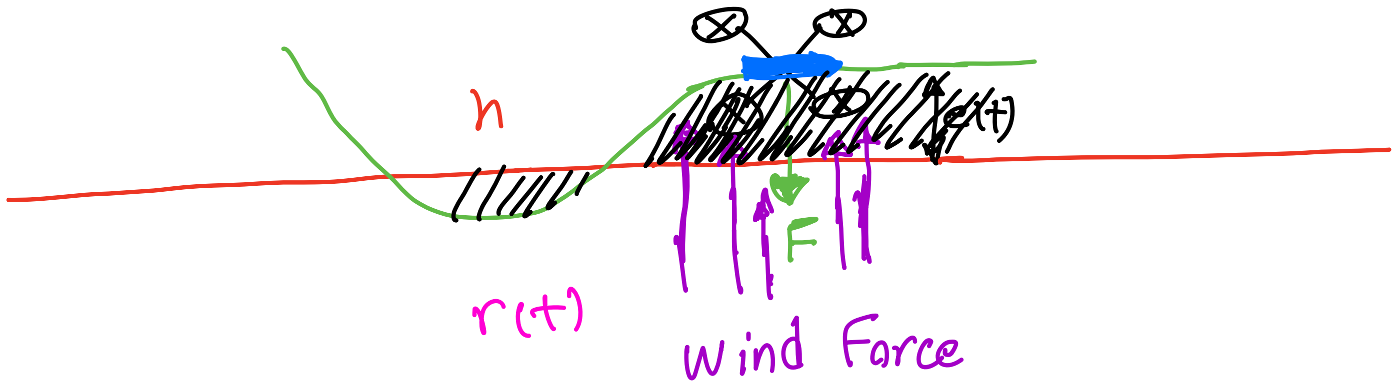
Let's add the integral controller.





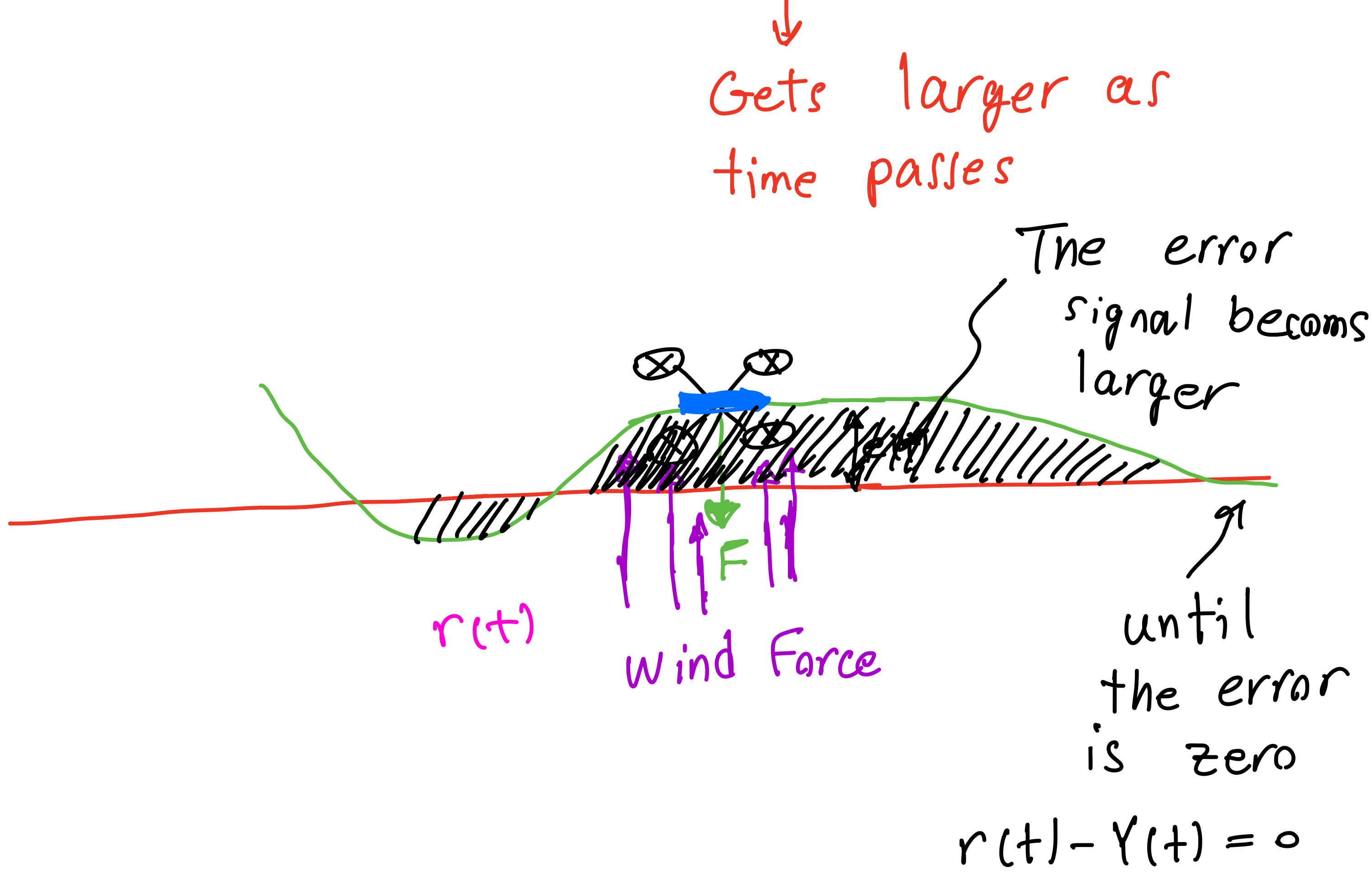
The integral of a function is the area under the curve:



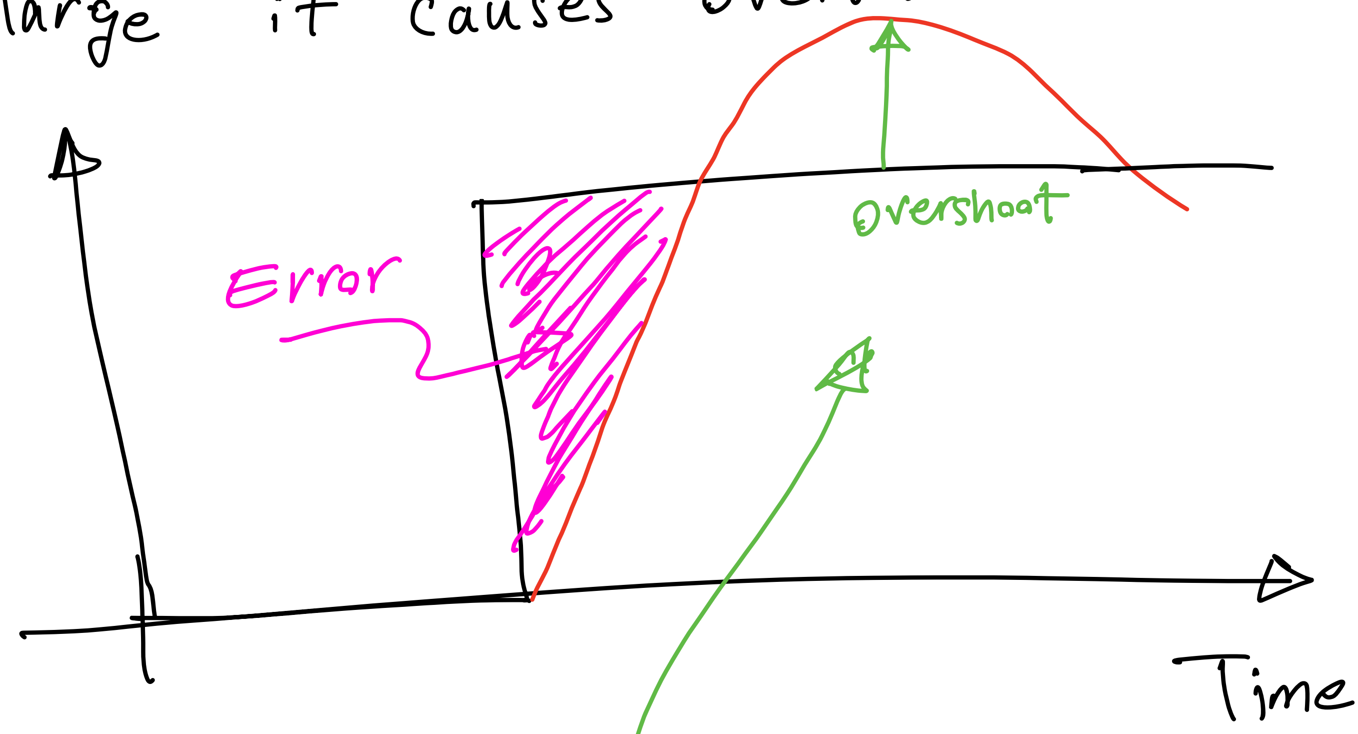


As the drone moves the integral controller adds the area under the error curve to the control signal. Therefore, the control signal becomes larger and larger until the drone force due to the larger control signal becomes larger than the wind force.

$$F = K_p e(t) + K_I \int e(t)\, dt$$



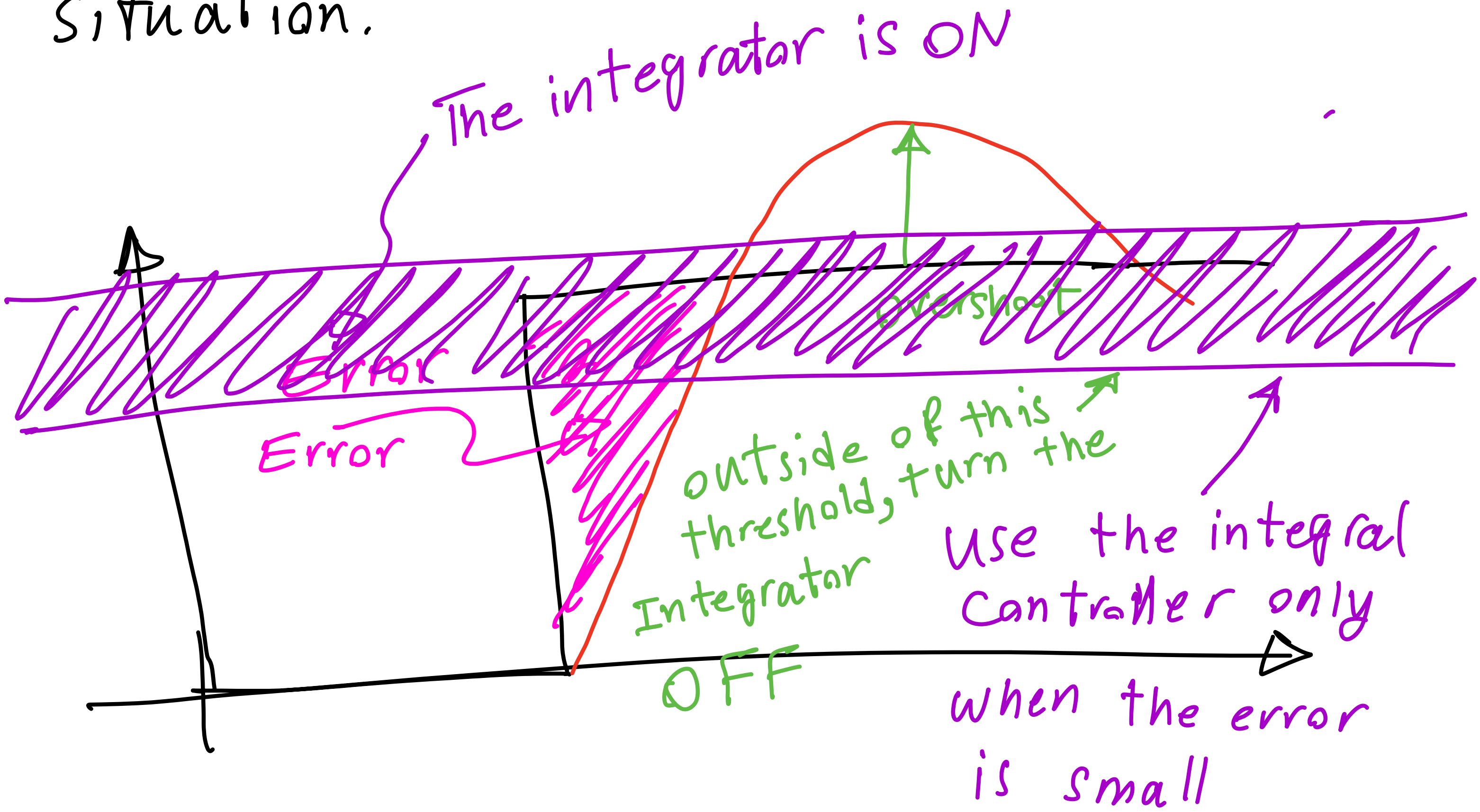
If the integrator term becames too large it causes overshoot.



Too much integrator will cause

overshoot.

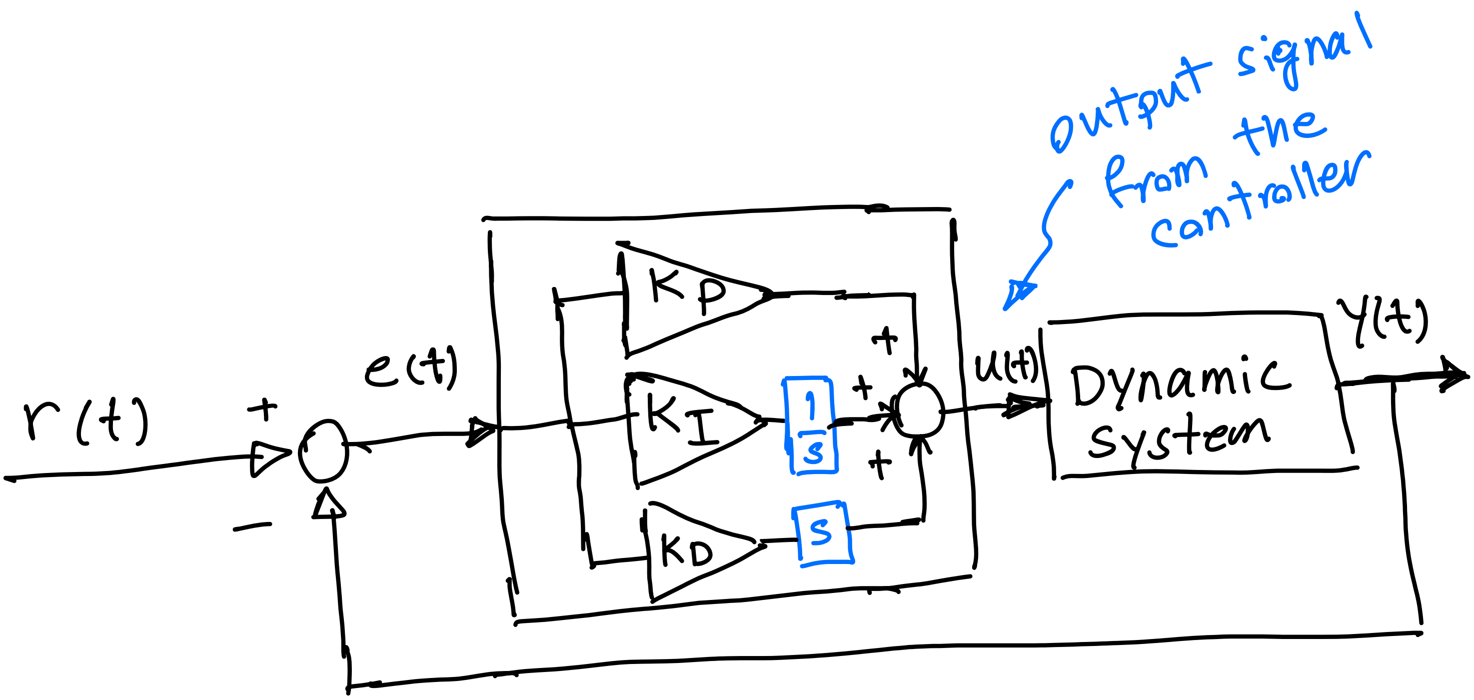
The problem was that the error was small, so we added the integrator controller for small error situation.



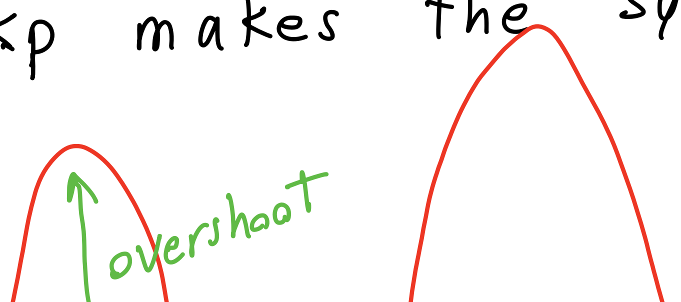
Therefore, turn on the integrator using a switch only when the error is small, and turn off the integrator for other error values.

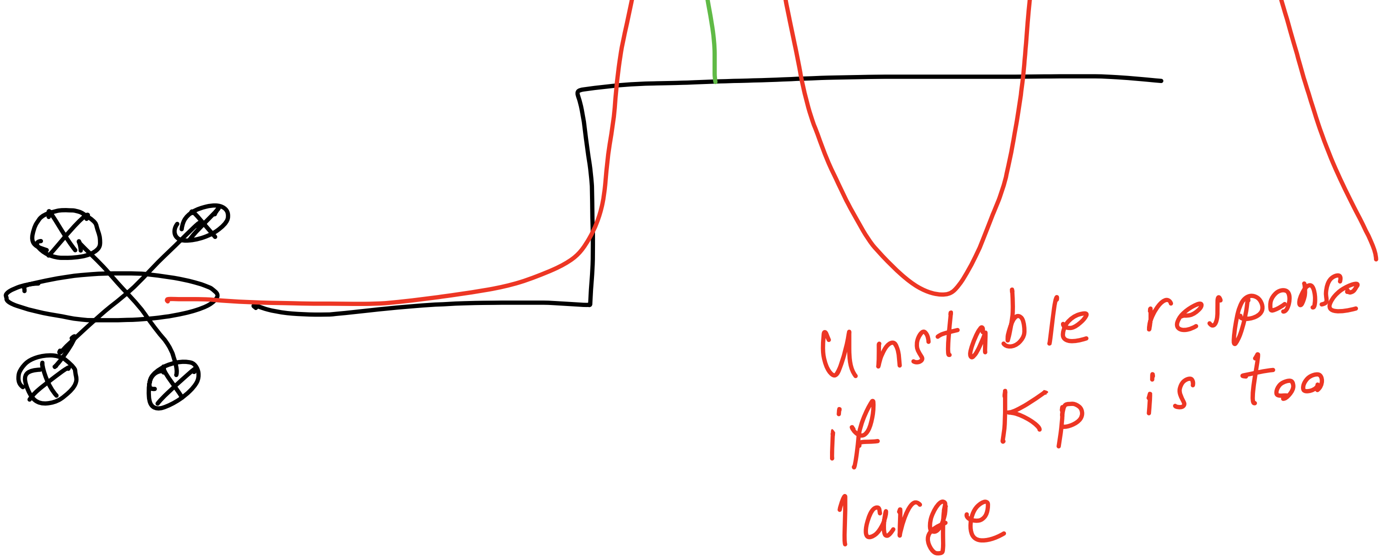
Note: when you use a switch, the system becomes nonlinear, and therefore we can not use the linear approaches such as root locus method anymore.

Now, add the derivative controller:



- Icreasing Kp makes the system faster but too much Kp makes the system unstable.





The derivative controller reduces overshoot and allows to increase K_p without making the system unstable.

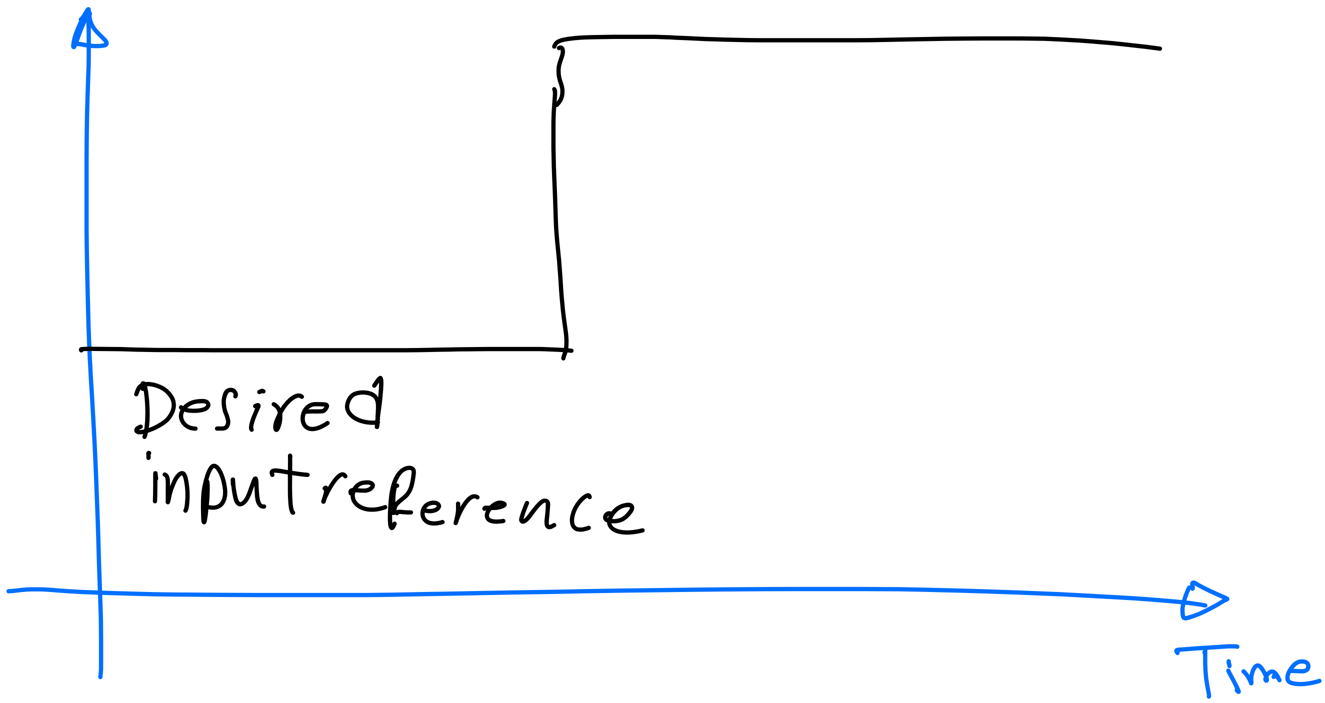
The derivative controller is $K_D \frac{de(t)}{dt}$

the $\frac{de(t)}{dt}$ term can be interpreted as the rate of change of the error, (or the velocity controll in a way).

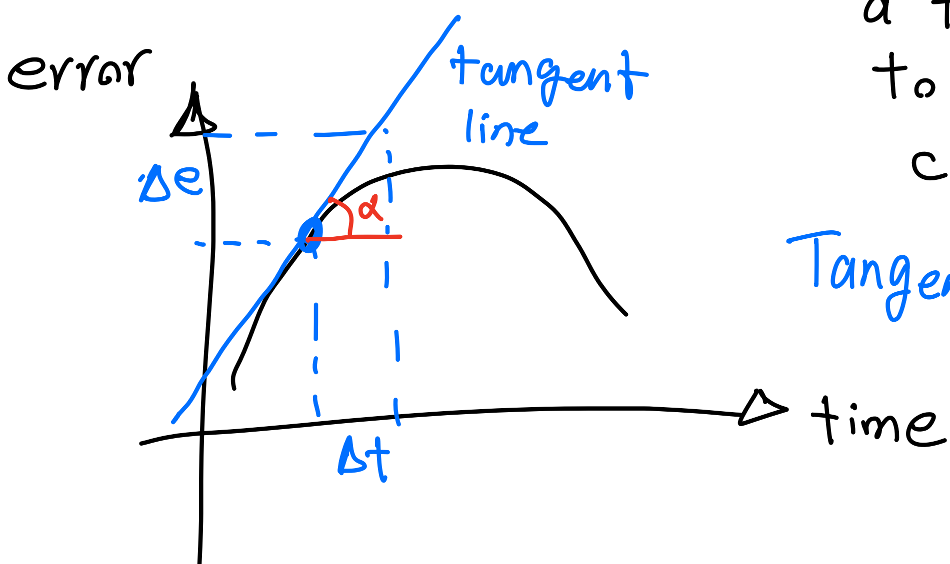
Also, can be interpreted as same sort of damping in the system.

(damping = damping constant × velocity)

The issue with the derivative is a sudden change in the desired reference input.



A derivative is $\frac{de(t)}{dt}$ → slope of a tangent line to the $e(t)$ curve

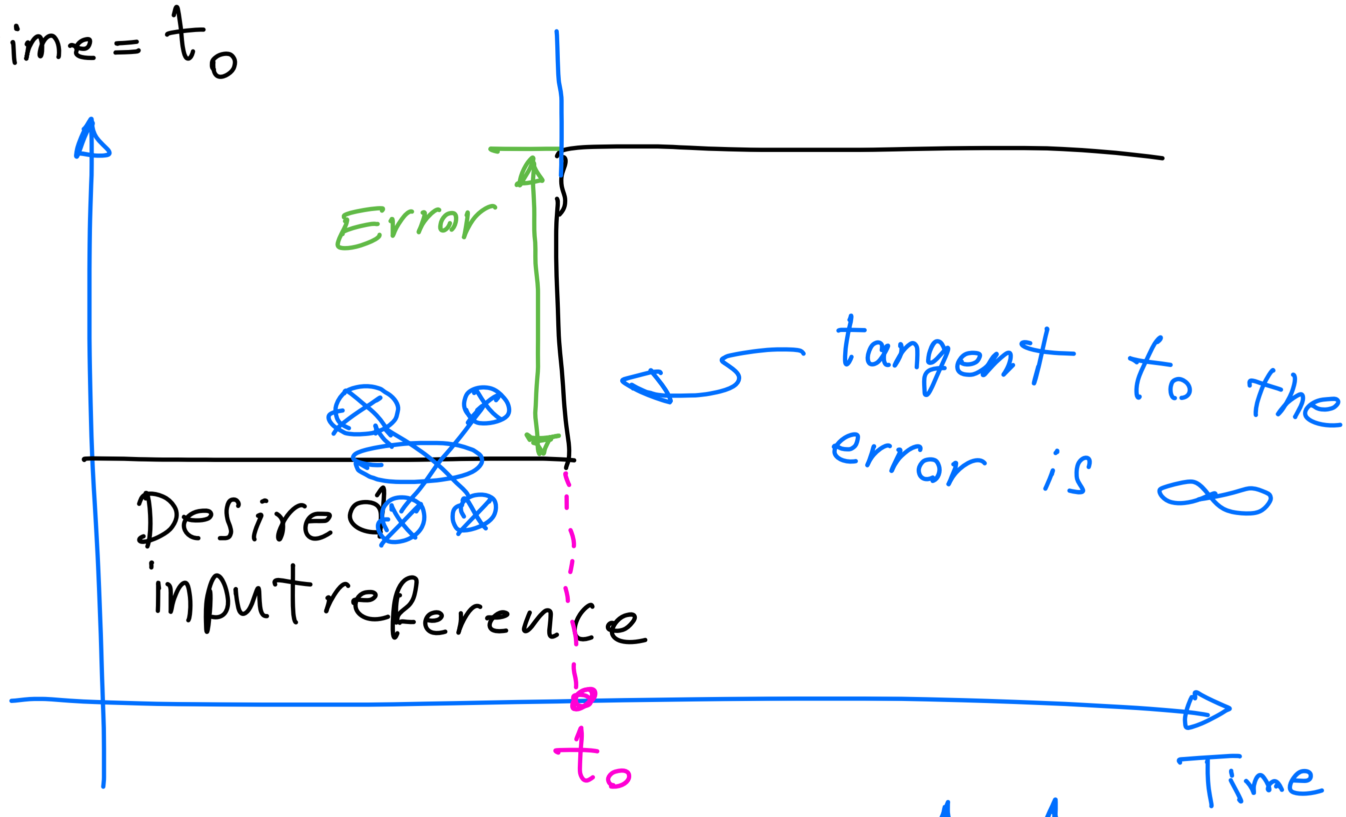


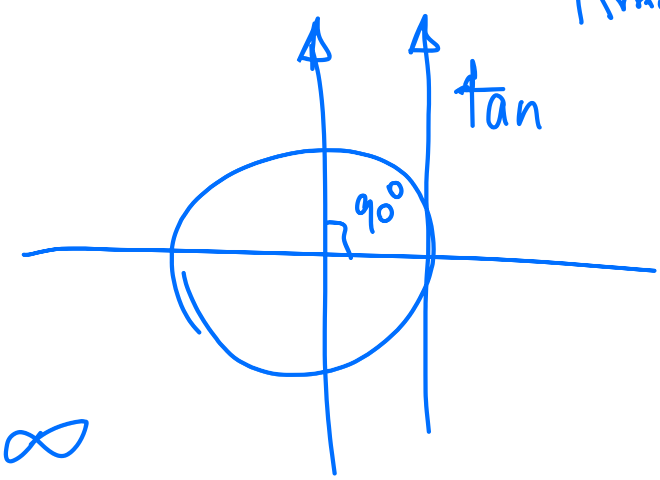
$$\text{Tangent} = \frac{\Delta e}{\Delta t}$$

$$\tan\alpha = \frac{\Delta e}{\Delta t}$$

If $\Delta t \to 0 \Rightarrow \lim_{\Delta t \to 0} \frac{\Delta e}{\Delta t} = \frac{de}{dt}$

what is the tangent to the error when time = t_0

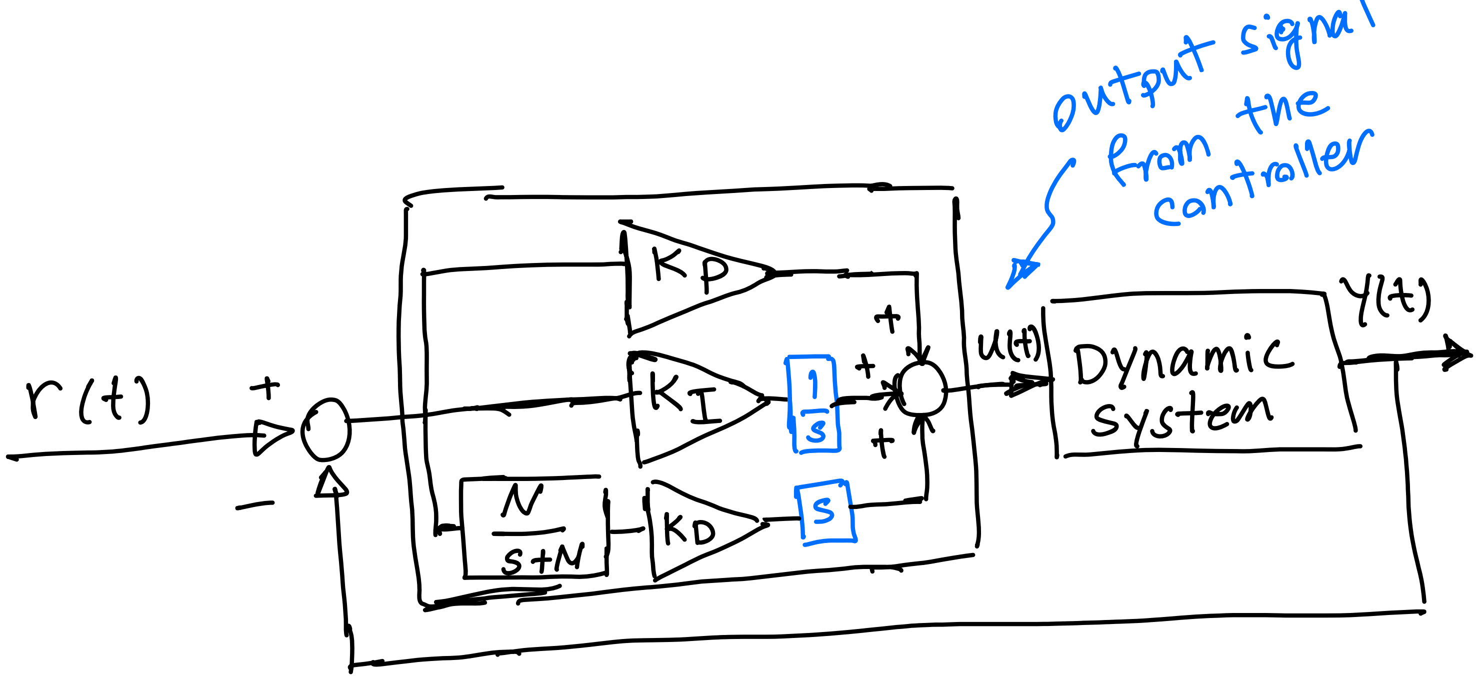


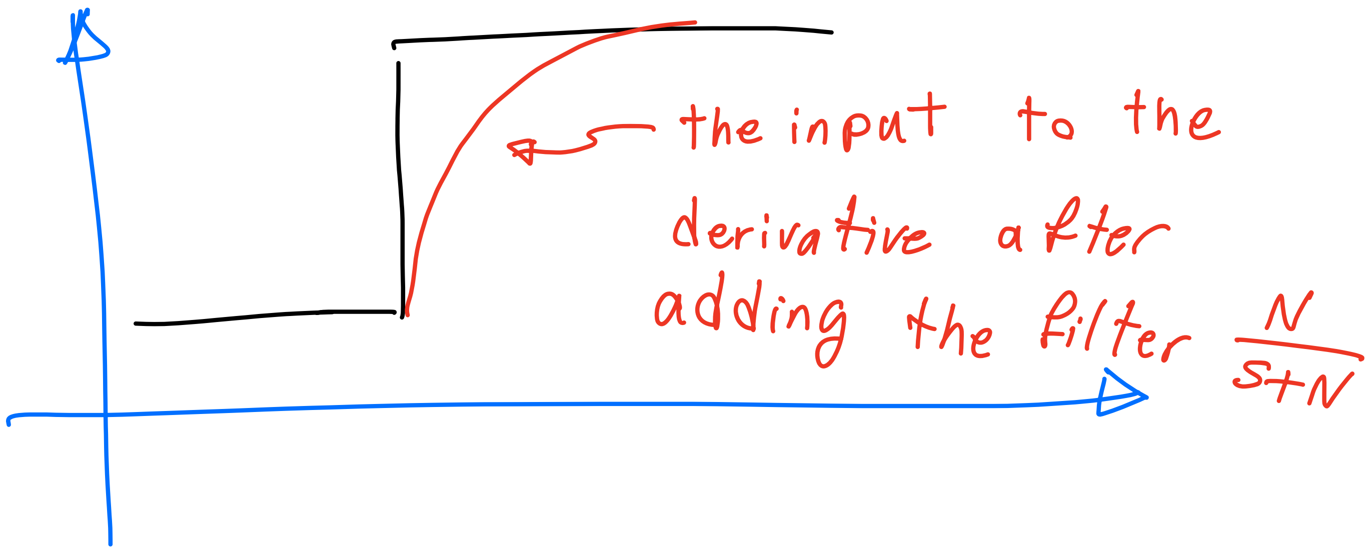


$$K_D \frac{de(t)}{dt} \rightarrow \infty$$

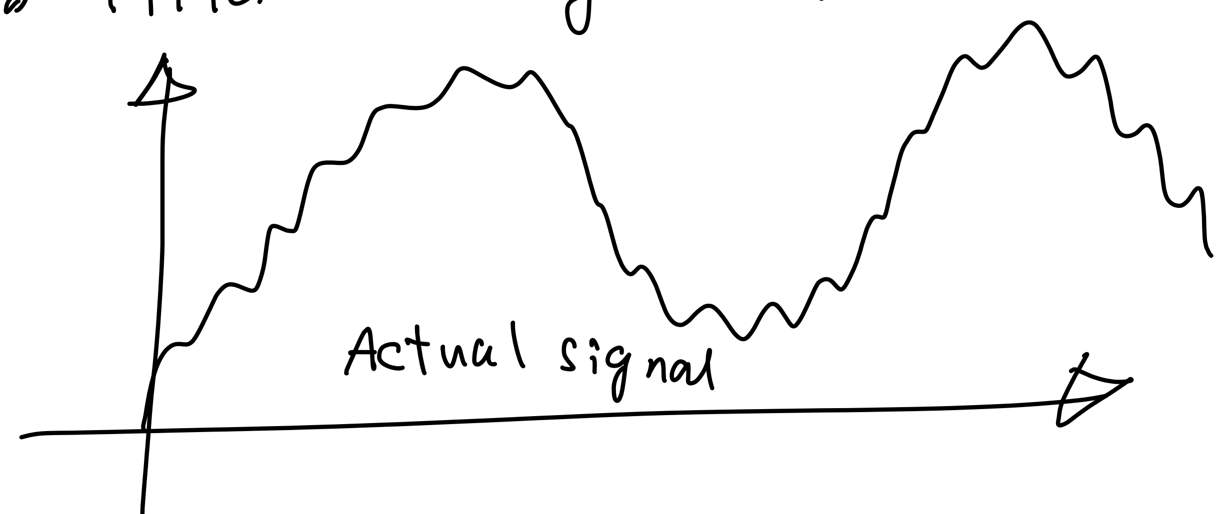
Infinity

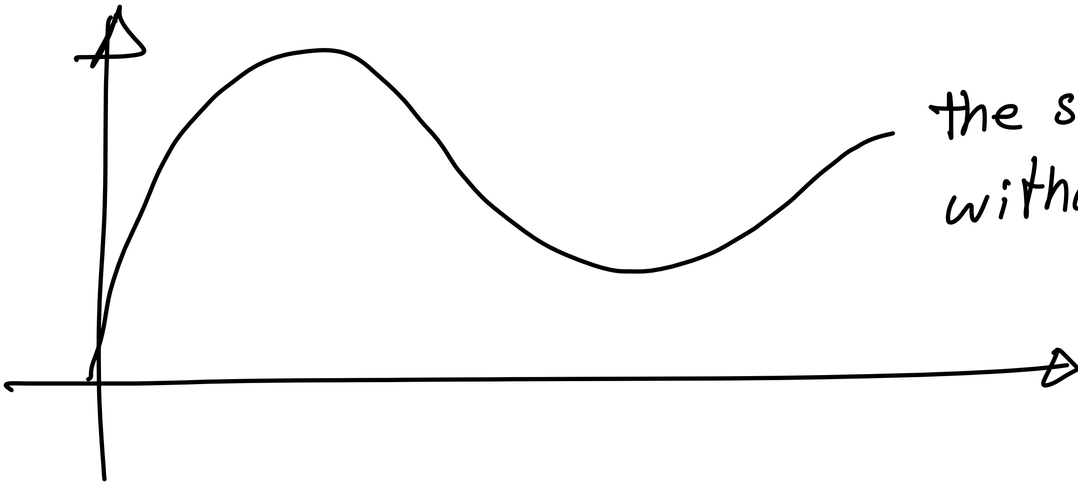
let's add a filter to the derivative controller to smooth the sudden change (and the problem with ∞)





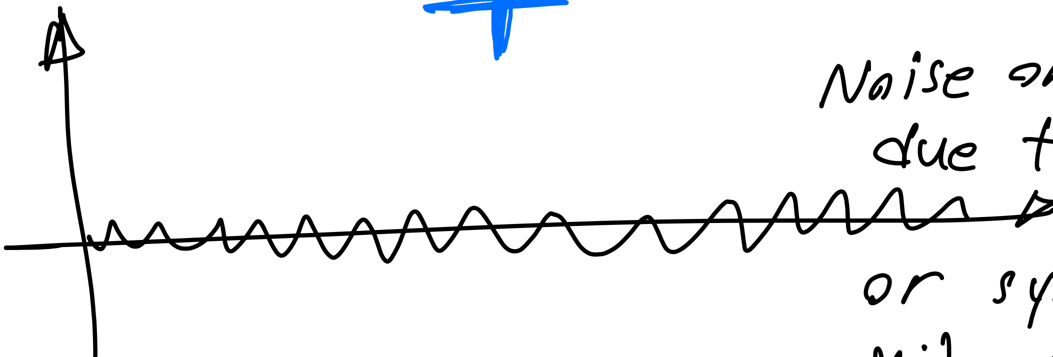
Another advantage of the filter is to filter the high frequency noise



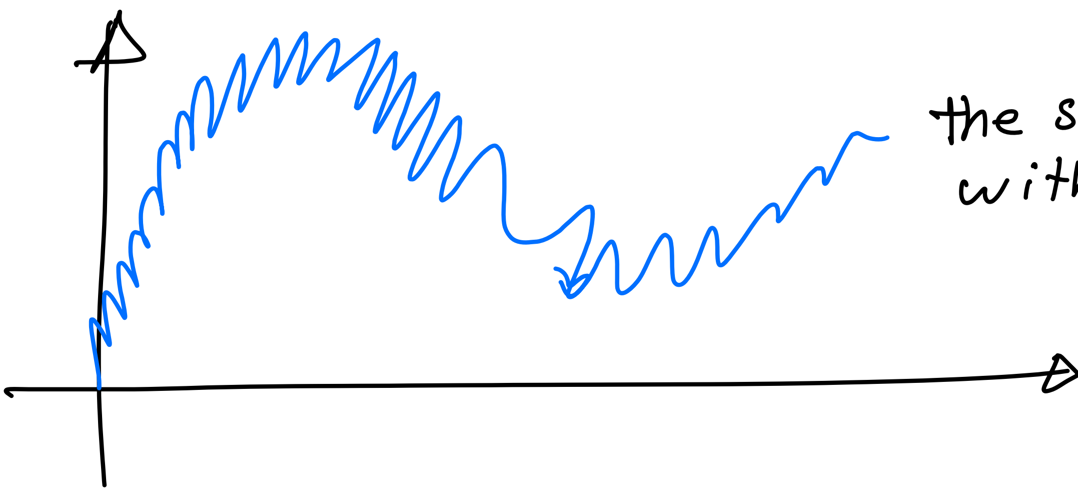


the signal
without noise

+



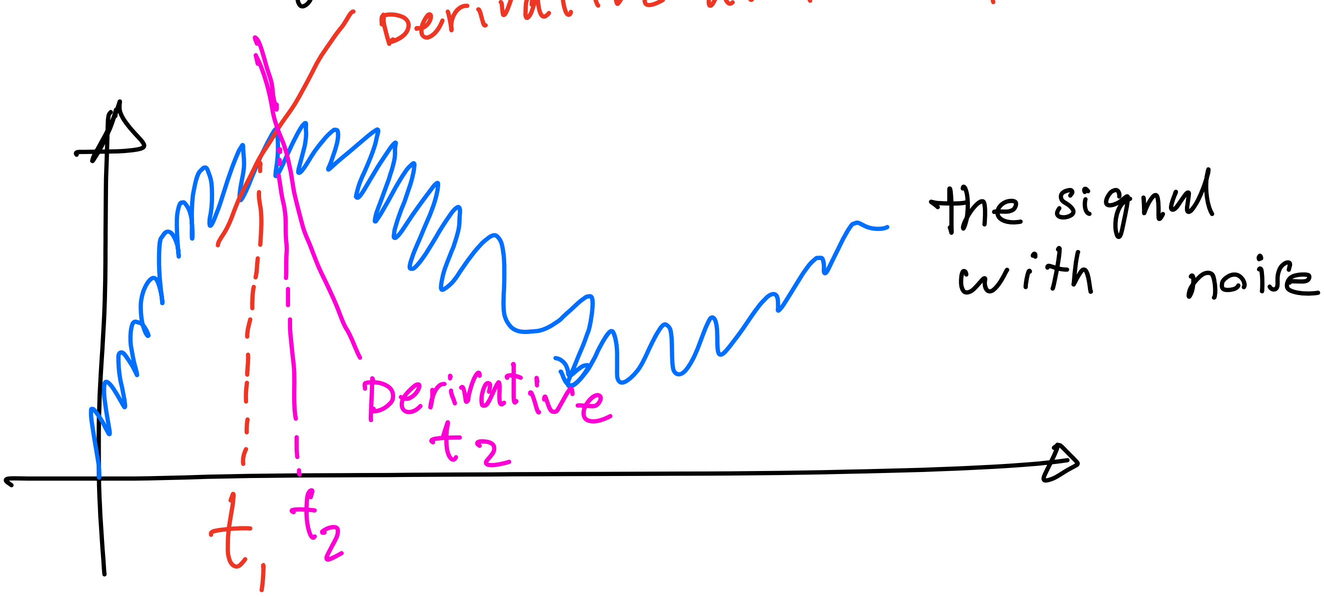
Noise only
due to sensor
or system
vibration,
or external input



the signal
with noise

If you use the derivative in this

noisy signal:

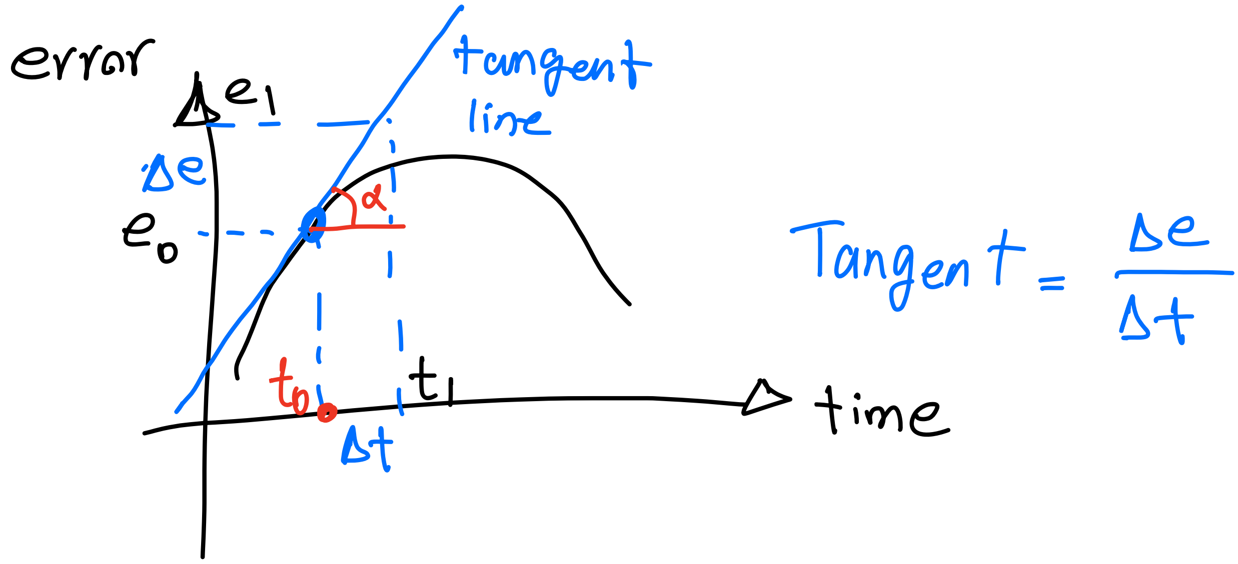


The derivative at t_1 is very different from the derivative at t_2.
Therefore, the noise is giving unacceptable control signal.

Note about the derivative control:

Derivative at time t_0 is defined as

$$\frac{de(t)}{dt} = \lim_{\Delta t \to 0} \frac{\Delta e}{\Delta t} \qquad \frac{\Delta e}{\Delta t} = \frac{e_1 - e_0}{t_1 - t_0}$$



The current time is t_0, therefore we do not have the information about e_1 yet. because t_1 is the next time step in futue. And we don't have the information about the future.

In another word we can not predict the future. So, in practice a derivative controller on it's own does not make sense. But a derivative with a filter is a good practical solution

$$K_D \frac{de(t)}{dt} \longrightarrow K_D\, s\, E(s)$$

Instead of $K_D\, s\, E(s)$ add the filter $\frac{N}{s+N}$ to the derivative controller.

Therefore, the practical derivative controller is

$$\frac{N}{s+N} K_D\, s\, E(s) \quad \text{or} \quad \frac{K_D\, s\, N}{s+N} E(s)$$