Ryerson University
Department of Electrical, Computer, and Biomedical Engineering
ELE709 - Real-Time Computer Control Systems

Project Answer Sheet 2

Name: Toni Pano #500822828 section 3

2. Consider the graph for Task 2.3 (Basic PID Controller). Is the response of the system satisfactory? If not, explain why.

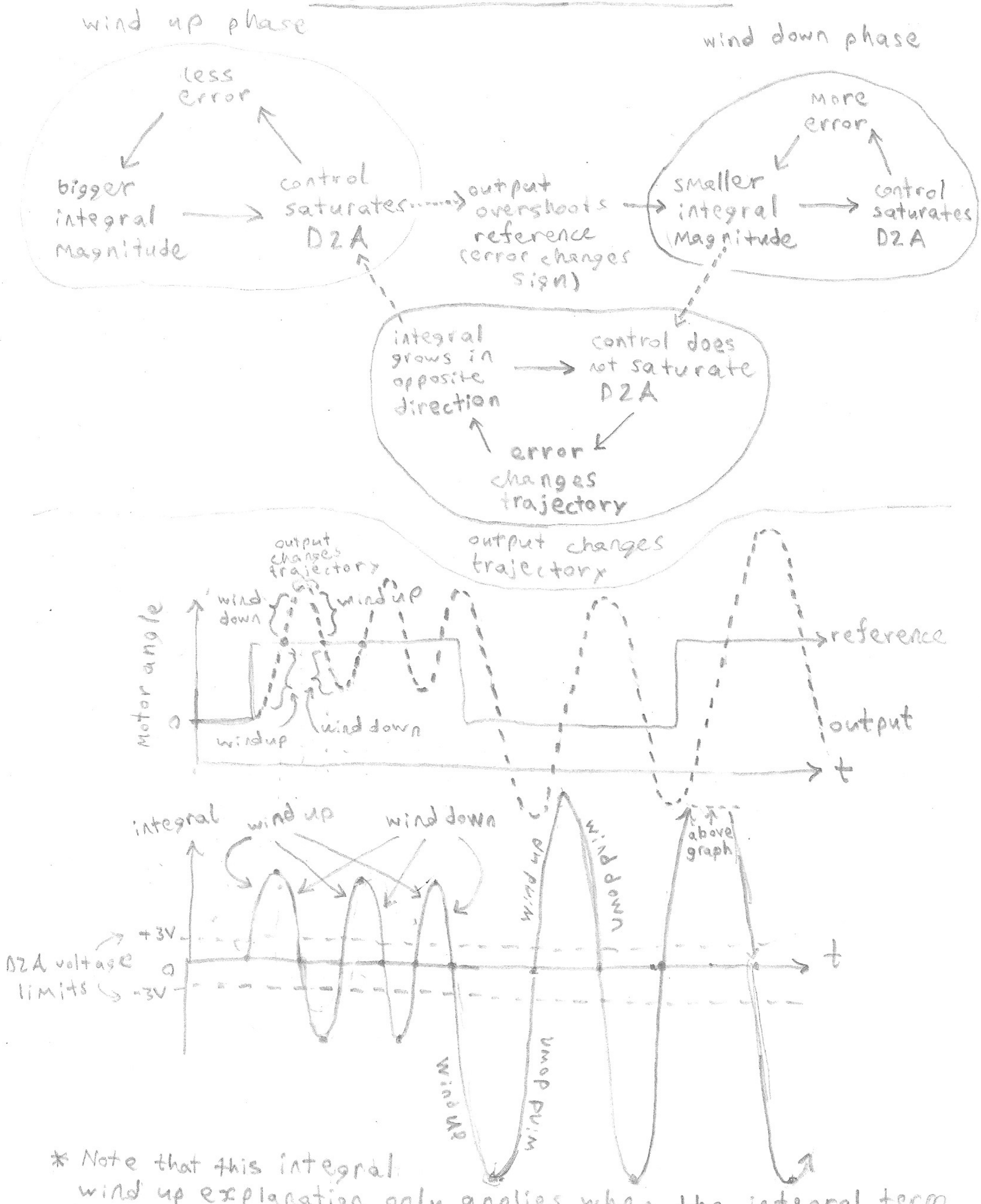
3. Derive the difference equation to calculate the control $u_k := u(kT)$ required to implement the Basic PID control in Task 2.3.

4. Derive the difference equation to calculate the control $u_k := u(kT)$ required to implement the Anti-Windup PID control in Task 2.4.

2. The controller output is not satisfactory because the integral term is large and makes the output oscillate around the reference signal with sharp turns (once the integral starts getting too small), like a triangle wave.

This behaviour happens because the D2A converter of the motor is limited to producing a voltage between -3V and +3V, as described by the saturation block in Figure 4 of the W2021 project manual. Should the controller demand a control signal that is too large in magnitude for the D2A converter to produce, the output will take a while to get closer to the reference signal, which means the error signal stays significantly large for a while, which means the integral of the error term will be very large. By the time the integral is small enough to make the control signal stay within range of the D2A converter limits, the motor output will have overshot the reference. This makes the error large, and repeats the above scenario. This behaviour is known as integral wind-up.

Diagram of Integral Wind-Up



* Note that this integral wind up explanation only applies when the integral term becomes large enough to make the proportional and derivative terms negligible.

Table of Contents for Questions 3 and 4

3. Derivation of the difference equation to calculate the control $u_k := u(kT)$ required to implement the Basic PID control in Task 2.3.

Splitting $U(s)$ into the proportional, integral, and derivative signal paths

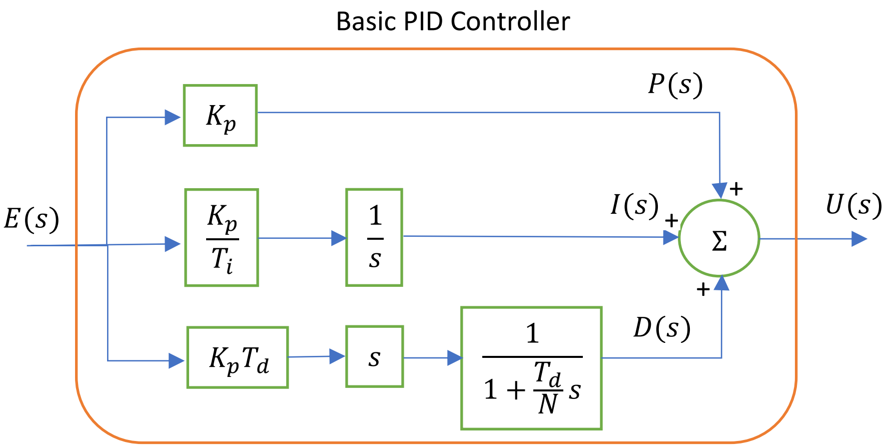


Figure 1. Transfer function block diagram of the Basic PID controller and all relevant signals from Task 2.3 of the W2021 Project Manual.

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}}\right)$$

$$G_c(s) = K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{1 + \frac{T_d s}{N}}$$

If $G_c(s)$ is the transfer function of the controller, then it must represent the ratio of the controller output signal $U(s)$ over the controller input error signal $E(s)$.

$$G_c(s) = \frac{U(s)}{E(s)}$$

$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{1 + \frac{T_d s}{N}}$$

$$U(s) = K_p E(s) + \frac{K_p}{T_i s} E(s) + \frac{K_p T_d s}{1 + \frac{T_d s}{N}} E(s)$$

Assume that the controller output signal $U(s)$ is the sum of the signals along the proportional control path $P(s)$, the integral control path $I(s)$, and the derivative control path $D(s)$.

$$U(s) = P(s) + I(s) + D(s)$$

The transfer functions for $P(s)$, $I(s)$ and $D(s)$ would then be:

$$P(s) = K_p E(s)$$

$$I(s) = \frac{K_p}{T_i s} E(s)$$

$$D(s) = \frac{K_p T_d s}{1 + \frac{T_d s}{N}} E(s)$$

Converting $U(s)$ into the difference equation u_k

The transfer function $U(s)$ can be converted into a continuous time domain signal by using the inverse Laplace transform $\mathcal{L}^{-1}\{\quad\}$.

$$\mathcal{L}^{-1}\{U(s)\} = \mathcal{L}^{-1}\{P(s) + I(s) + D(s)\}$$

$$\mathcal{L}^{-1}\{U(s)\} = \mathcal{L}^{-1}\{P(s)\} + \mathcal{L}^{-1}\{I(s)\} + \mathcal{L}^{-1}\{D(s)\}$$

$$u(t) = p(t) + i(t) + d(t)$$

We can create a discrete time difference equation $u(kT)$ that will match the behaviour of the continuous time control signal $u(t)$, by sampling $u(t)$ at every point in time when t is an integer multiple of a specified sampling period T (i.e. $t = kT$).

$$u(kT) = p(kT) + i(kT) + d(kT)$$

If the sampling period T is always a constant, then the subscript k can be used to denote the time when the equation is being sampled.

$$u_k = p_k + i_k + d_k$$

*The transfer functions $P(s)$, $I(s)$, and $D(s)$ can each be converted into their own discrete time difference equation p_k, i_k, and d_k, similar to how $U(s)$ was converted into the difference equation u_k.

Converting $P(s)$ into the difference equation p_k

$$P(s) = K_p E(s)$$

$$\mathcal{L}^{-1}\{P(s)\} = \mathcal{L}^{-1}\{K_p E(s)\}$$

$$\mathcal{L}^{-1}\{P(s)\} = K_p \mathcal{L}^{-1}\{E(s)\}$$

$$p(t) = K_p e(t)$$

$$p(kT) = K_p e(kT)$$

$$p_k = K_p e_k$$

Converting $I(s)$ into the difference equation i_k

$$I(s) = \frac{K_p}{T_i s} E(s)$$

$$\mathcal{L}^{-1}\{I(s)\} = \mathcal{L}^{-1}\left\{\frac{K_p}{T_i s} E(s)\right\}$$

$$\mathcal{L}^{-1}\{I(s)\} = \frac{K_p}{T_i} \mathcal{L}^{-1}\left\{\frac{E(s)}{s}\right\}$$

*Assuming 0 initial conditions, the inverse Laplace transform is:

$$i(t) = \frac{K_p}{T_i} \int_0^t e(t)dt$$

$$i(kT) = \frac{K_p}{T_i} \int_0^{kT} e(t)dt$$

$$i(kT) = \frac{K_p}{T_i}\left(\int_{(k-1)T}^{kT} e(t)dt + \int_0^{(k-1)T} e(t)dt \right)$$

$$i(kT) = \frac{K_p}{T_i} \int_{(k-1)T}^{kT} e(t)dt + \frac{K_p}{T_i} \int_0^{(k-1)T} e(t)dt$$

*the last term of this equation looks like the equation for $i(kT)$ when kT is replaced by $(k-1)T$

$$i(kT) = \frac{K_p}{T_i} \int_{(k-1)T}^{kT} e(t)dt + i((k-1)T)$$

*a forwards rectangle approximation of the integral is used to helps simplify the equation, as specified before Task 2.3 in the W2021 Project manual. An explanation of why this approximation is used can be seen in the approximation of the $u(t) - v(t)$ integral in Question 4. A comparison of the various integral approximation methods can be seen in Figure 2.

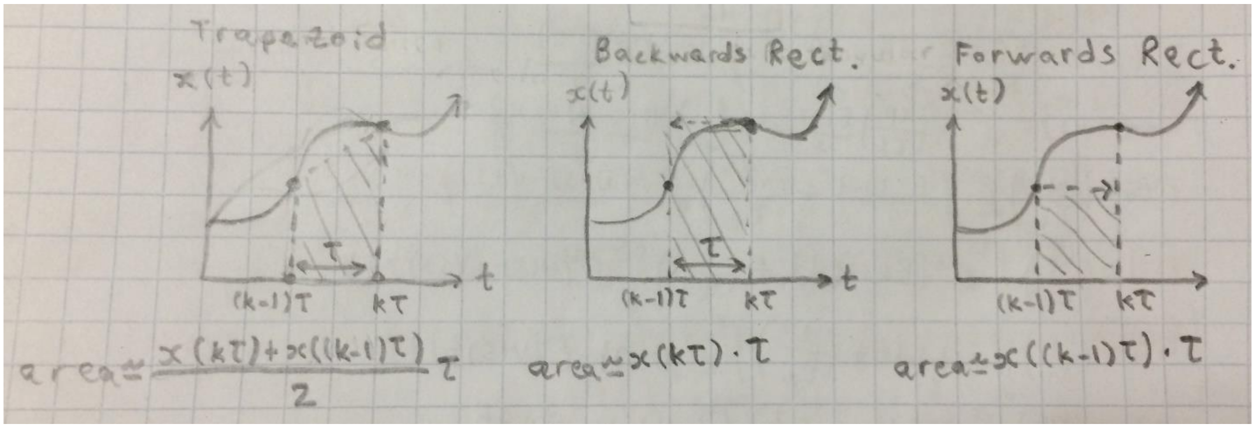


Figure 2. Diagrams of the trapezoidal, backwards rectangular, and forwards rectangular methods for approximating a continuous time integral as a discrete time difference equation.

$$i(kT) = \frac{K_p}{T_i}\left(e\left((k-1)T\right) * T\right) + i((k-1)T)$$

$$i_k = \frac{K_p}{T_i}(e_{k-1}T) + i_{k-1}$$

$$i_k = \frac{K_p T}{T_i} e_{k-1} + i_{k-1}$$

Converting $D(s)$ into the difference equation d_k

$$D(s) = \frac{K_p T_d s}{1 + \frac{T_d s}{N}} E(s)$$

$$\left(1 + \frac{T_d s}{N}\right) D(s) = K_p T_d s E(s)$$

$$D(s) + \frac{T_d s}{N} D(s) = K_p T_d s E(s)$$

$$\mathcal{L}^{-1}\left\{D(s) + \frac{T_d s}{N} D(s)\right\} = \mathcal{L}^{-1}\{K_p T_d s E(s)\}$$

$$\mathcal{L}^{-1}\{D(s)\} + \frac{T_d}{N}\mathcal{L}^{-1}\{sD(s)\} = K_p T_d \mathcal{L}^{-1}\{sE(s)\}$$

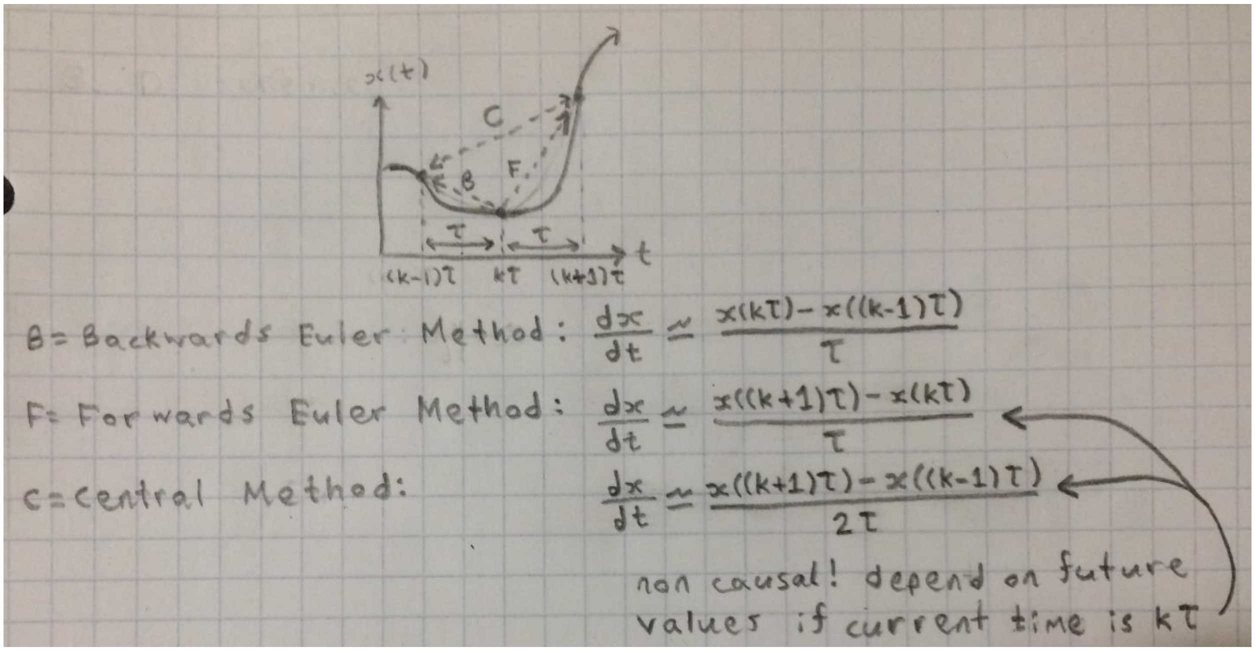
$$\mathcal{L}^{-1}\{D(s)\} + \frac{T_d}{N}\frac{d}{dt}\mathcal{L}^{-1}\{D(s)\} = K_p T_d \frac{d}{dt}\mathcal{L}^{-1}\{E(s)\}$$

$$d(t) + \frac{T_d}{N}\frac{d}{dt}d(t) = K_p T_d \frac{d}{dt}e(t)$$

$$Nd(t) + T_d\frac{d}{dt}d(t) = K_p N T_d \frac{d}{dt}e(t)$$

$$Nd(kT) + T_d\frac{d}{dt}d(kT) = K_p N T_d \frac{d}{dt}e(kT)$$

*a backwards difference rule (aka backwards Euler method) approximation of the derivative helps simplify the equation, as specified before Task 2.3 in the W2021 Project Manual. The only approximation which is causal is the Backwards Euler method, as seen in Figure 3.



B= Backwards Euler Method: $\frac{dx}{dt} \cong \frac{x(kT) - x((k-1)T)}{T}$

F= Forwards Euler Method: $\frac{dx}{dt} \cong \frac{x((k+1)T) - x(kT)}{T}$

C=Central Method: $\frac{dx}{dt} \cong \frac{x((k+1)T) - x((k-1)T)}{2T}$

non causal! depend on future values if current time is kT

Figure 3. Diagrams of the Backwards Euler, Forwards Euler, and Central methods for approximating a continuous time derivative as a discrete time difference equation.

$$Nd(kT) + T_d\left(\frac{d(kT) - d((k-1)T)}{T}\right) = K_p N T_d \left(\frac{e(kT) - e((k-1)T)}{T}\right)$$

$$Nd_k + T_d\left(\frac{d_k - d_{k-1}}{T}\right) = K_p N T_d \left(\frac{e_k - e_{k-1}}{T}\right)$$

$$NTd_k + T_d(d_k - d_{k-1}) = K_p N T_d (e_k - e_{k-1})$$

$$NTd_k + T_d d_k - T_d d_{k-1} = K_p N T_d (e_k - e_{k-1})$$

$$(NT + T_d) d_k - T_d d_{k-1} = K_p N T_d (e_k - e_{k-1})$$

$$(NT + T_d) d_k = K_p N T_d (e_k - e_{k-1}) + T_d d_{k-1}$$

$$(NT + T_d) d_k = T_d (K_p N (e_k - e_{k-1}) + d_{k-1})$$

$$d_k = \frac{T_d}{NT + T_d} (K_p N (e_k - e_{k-1}) + d_{k-1})$$

The full discrete time difference equation u_k for $u(kT)$ with Basic PID control

$$u_k = p_k + i_k + d_k$$

$$u_k = [K_p e_k] + \left[\frac{K_p T}{T_i} e_{k-1} + i_{k-1}\right] + \left[\frac{T_d}{NT + T_d} (K_p N (e_k - e_{k-1}) + d_{k-1})\right]$$

For implementation purposes u_k will not be simplified any further. In a computer program, it would be more convenient to calculate the 3 equations for p_k, i_k, and d_k separately before summing them as u_k, than it would be to calculate the fully simplified version of u_k on one line of code.

4. Derivation of the difference equation to calculate the control $u_k \coloneqq u(kT)$ required to implement the Anti-Windup PID control in Task 2.4.

Splitting $U(s)$ into the proportional, integral, and derivative signal paths

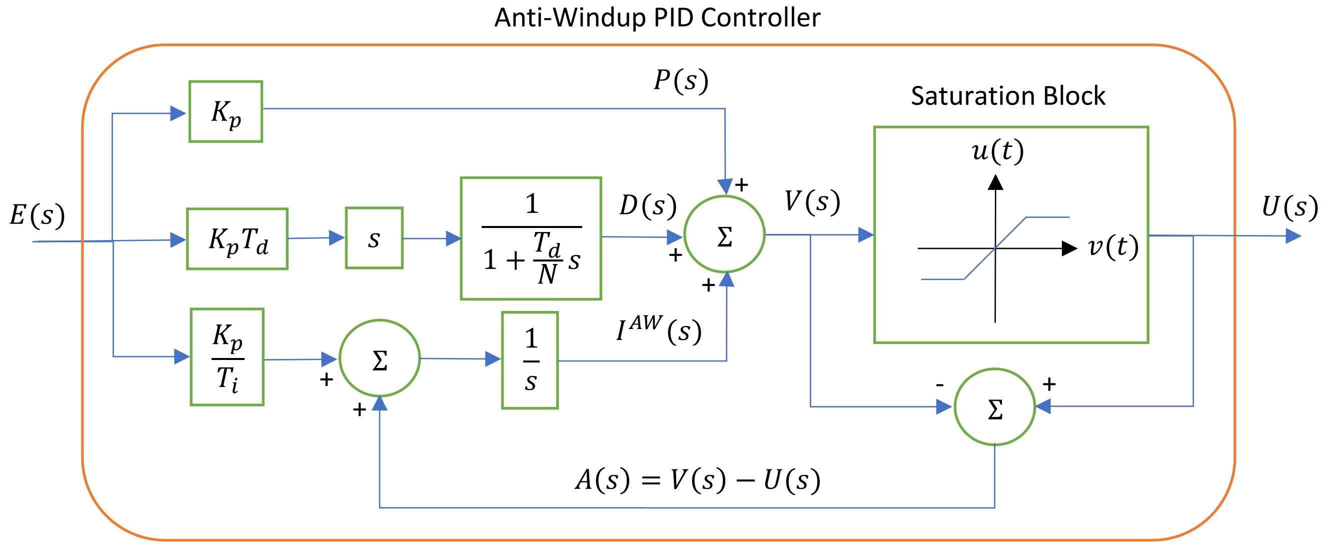


Figure 4. Transfer function block diagram of the Anti-Windup PID controller and all relevant signals from Figure 3 of the W2021 Project Manual.

The only difference in the controller is the integral path, which adds the result of the anti-windup signal to itself. Therefore, the proportional control path $P(s)$ and the derivative control path $D(s)$ should be the same as in Task 2.4. Therefore the transfer function of the control signal $U(s)$ should look like:

$$U(s) = P(s) + I^{AW}(s) + D(s)$$

where $I^{AW}(s)$ represents the transfer function of the integral control path with anti-windup.

$U(s)$ can then be converted into a discrete time difference equation similar to the method shown in Question 3. The difference equation u_k then becomes:

$$u_k = p_k + i_k^{aw} + d_k$$

$P(s)$, $I^{AW}(s)$, and $D(s)$ can each be converted into their own discrete time difference equation p_k, i_k^{aw}, and d_k, similar to how $U(s)$ was converted into u_k. The proofs for the transfer functions of $P(s)$ and $D(s)$ can be seen in Question 3.

Finding the transfer function $I^{AW}(s)$ for the integral path with anti-windup

The anti-windup signal path, which adds the difference between the control signal $u(t)$ and the saturated control signal $v(t)$ to the error signal $e(t)$, multiplied by their respective gains. The integral path integrates the combined signal:

$$\frac{K_p}{T_i}e(t) + \frac{1}{T_t}(u(t) - v(t))$$

$I^{AW}(s)$ represents the transfer function of the integral path with anti-windup. It integrates the above combined signal, which means $I^{AW}(s)$ is:

$$I^{AW}(s) = \frac{1}{s}\left(\frac{K_p}{T_i}E(s) + \frac{1}{T_t}(U(s) - V(s))\right)$$

Converting $I^{AW}(s)$ into the difference equation i_k^{aw}

$I^{AW}(s)$ can be converted into a discrete time difference equation i_k^{aw}, similar to how $U(s)$ was converted into u_k.

$$I^{AW}(s) = \frac{K_p E(s)}{T_i s} + \frac{(U(s) - V(s))}{T_t s}$$

$$\mathcal{L}^{-1}\{I^{AW}(s)\} = \mathcal{L}^{-1}\left\{\frac{K_p E(s)}{T_i s} + \frac{(U(s) - V(s))}{T_t s}\right\}$$

$$\mathcal{L}^{-1}\{I^{AW}(s)\} = \frac{K_p}{T_i}\mathcal{L}^{-1}\left\{\frac{E(s)}{s}\right\} + \frac{1}{T_t}\mathcal{L}^{-1}\left\{\frac{U(s) - V(s)}{s}\right\}$$

*Assuming 0 initial conditions, the inverse Laplace transform is:

$$\mathcal{L}^{-1}\{I^{AW}(s)\} = \frac{K_p}{T_i}\int_0^t \mathcal{L}^{-1}\{E(s)\}dt + \frac{1}{T_t}\int_0^t \mathcal{L}^{-1}\{U(s) - V(s)\}dt$$

$$i^{aw}(t) = \frac{K_p}{T_i}\int_{0}^{t} e(t)dt + \frac{1}{T_t}\int_{0}^{t} (u(t) - v(t))dt$$

$$i^{aw}(kT) = \frac{K_p}{T_i}\int_{0}^{kT} e(t)dt + \frac{1}{T_t}\int_{0}^{kT} \big(u(t) - v(t)\big)dt$$

$$i^{aw}(kT) = \frac{K_p}{T_i}\left(\int_{(k-1)T}^{kT} e(t)dt + \int_{0}^{(k-1)T} e(t)dt\right) + \frac{1}{T_t}\left(\int_{(k-1)T}^{kT} \big(u(t) - v(t)\big)dt + \int_{0}^{(k-1)T} \big(u(t) - v(t)\big)dt\right)$$

$$i^{aw}(kT) = \frac{K_p}{T_i}\int_{(k-1)T}^{kT} e(t)dt + \frac{K_p}{T_i}\int_{0}^{(k-1)T} e(t)dt + \frac{1}{T_t}\int_{(k-1)T}^{kT} \big(u(t) - v(t)\big)dt + \frac{1}{T_t}\int_{0}^{(k-1)T} \big(u(t) - v(t)\big)dt$$

$$i^{aw}(kT) = \frac{K_p}{T_i}\int_{(k-1)T}^{kT} e(t)dt + \frac{1}{T_t}\int_{(k-1)T}^{kT} \big(u(t) - v(t)\big)dt + \left(\frac{K_p}{T_i}\int_{0}^{(k-1)T} e(t)dt + \frac{1}{T_t}\int_{0}^{(k-1)T} \big(u(t) - v(t)\big)dt\right)$$

*the last two terms of this equation resemble the equation for $i(kT)$ when kT is replaced by $(k-1)T$

$$i^{aw}(kT) = \frac{K_p}{T_i}\int_{(k-1)T}^{kT} e(t)dt + \frac{1}{T_t}\int_{(k-1)T}^{kT} \big(u(t) - v(t)\big)dt + i((k-1)T)$$

*a forwards rectangle approximation of the integral is used to helps simplify the equation, as specified before Task 2.3 in the W2021 Project manual. Due to $u(kT)$ depending on $v(kT)$, and $v(kT)$ depending on $i^{aw}(kT)$, neither $u(kT)$ or $v(kT)$ can be known ahead of time before calculating $i^{aw}(kT)$. This means the backwards rectangle and trapezoidal integral approximations cannot be used. The only approximation which can be used is the forwards rectangle, since it depends on $u((k-1)T)$ and $v((k-1)T)$, which are both known ahead of time before calculating $i^{aw}(kT)$. A comparison of the various integral approximation methods can be seen in Figure 2.

$$i^{aw}(kT) = \frac{K_p}{T_i}\big(e\big((k-1)T\big) * T\big) + \frac{1}{T_t}\Big(\big(u\big((k-1)T\big) - v((k-1)T)\big) * T\Big) + i^{aw}((k-1)T)$$

$$i_k^{aw} = \frac{K_p}{T_i}(e_{k-1}T) + \frac{1}{T_t}\big((u_{k-1} - v_{k-1})T\big) + i_{k-1}^{aw}$$

$$i_k^{aw} = T\left(\frac{K_p}{T_i}e_{k-1} + \frac{1}{T_t}(u_{k-1} - v_{k-1})\right) + i_{k-1}^{aw}$$

The full discrete time difference equation u_k for $u(kT)$ with Anti-Windup PID control

$$u_k = p_k + i_k^{aw} + d_k$$

*Note that the equations for p_k and d_k have been copied from Question 3, since the transfer functions of $P(s)$ and $D(s)$ are also the same in Question 3.

$$u_k = \left[K_p e_k\right] + \left[T\left(\frac{K_p}{T_i}e_{k-1} + \frac{1}{T_t}(u_{k-1} - v_{k-1})\right) + i_{k-1}^{aw}\right] + \left[\frac{T_d}{NT + T_d}(K_p N(e_k - e_{k-1}) + d_{k-1})\right]$$

For implementation purposes u_k will not be simplified any further. In a computer program, it would be more convenient to calculate the 3 equations for p_k, i_k^{aw}, and d_k separately before summing them as u_k, than it would be to calculate the fully simplified version of u_k on one line of code.