Handout
Testing PID Controller
HL

To test the PID controller in your design, please follow the procedure below:

1. Use input data sequence {x(n) | n =0, 1, …, 9}, where x(n) = 1, which is the sequence from the actual motion trajectory, e.g., it is coming from ADC output. See Table 1 for the tabulation of this data.
2. Assume the vehicle output Contl(n) = 0 for n=0, e.g., right at the beginning the vehicle position is 0.
3. Use your program to compute error Error(n).
4. Assign K_p value, the gain of the proportional controller.
5. Compute derivative of E(n) by central difference, e.g, compute cDError(n). Assume the inti
6. Assign K_d value, the gain of the derivative controller.
7. Compute integral of error, iDError(n), by summation of the past q unit, assuming q = 4, compute summation of Error(n) + Error(n-1) + Error(n-2) + Error(n-3). If in the initial condition, there is no previous value of Error(n), then assume Error(n) = 0 for n < 0.
8. Assign K_i value, the gain of the integral controller.
9. Assign weighting factor alpha_p, alpha_i, alpha_d to 1 initally.
10. Compute the sum of the PID controller, e.g., Sum_pid = alpha_p * K_p * Error(n) + alpha_i * K_i * iDError(n) + alpha_d * K_d * cDError(n);
11. Assume PWM operating frequency f_pwm = 500 Hz.
12. Compute the steering angle, Ang(n), based on the N_pwm(n), e.g., number of PWM pulses equal to angle value (and/or frequency for servo motor). Note assume two different motor options, stepper motor and servo motor. Let's use stepper motor as an example, you will need one GPIO pin to control the direction of the rotation of the motor, and assum f_pwm = 500 (Hz), count the number of pulse (each pulse for a step, a half step, a ¼ step or 1/8 step, depending on the configuration of the motor controller, assume initially 1 step, if there is a need to do refinement in the future, you can always change to micro steps).
13. Assume the angle range 500(pulses) * micro-step (1/8 of a full step).
14. Assume the vehicle moving at 5 Km per hour, e.g., speedVe(n) = 5000;
15. Compute the lateral displacement of the vehicle, disVe(n), based on the speed speedVe(n) and Ang(n);
16. Assign Contl(n) = disVe(n);
17. Compute the sensor output h(n), which in reality is the sensor senses the lateral displacement disVe(n), in this simulation, you can just assume h(n) is the data after disVe(n) is going through embedded system ADC and producing digital data h(n).
18. In simulation, assume ideal case, h(n) = Ang(n) + Noise(n), where is Noise(n) is a random noise, assume Noise(n) is white additive noise.
19. Compute kernel(n) for noise removal, assume using Gaussian kernel.
20. Compute 1D convolution hPrime(n) = kernel(n) (*) h(n).
21. Compute E(n+1).

Table 1. The tabulation of the computation result for PID controller testing.

n (Time)	0	1	2	3	4	5	6	7	8	9
x(n)	1	1	1	1	1	1	1	1	1	1
Contl(n)	0									

Error(n)	1									
K_p										
cDError(n)										
K_d										
q										
iDError(n)										
alpha_p	1									
alpha_i	1									
alpha_d	1									
Sum_pid(n)										
f_pwm	500									
N_pwm(n)										
Ang(n)										
SpeedVe(n)	5000									
disVe(n)										
h(n)										
Noise(n)										
Gaussian(n)										
hPrime(n)										
e(n+1)										

Use excel program to plot the data and observer the change of the controlled output Contl(n), e.g., disVe(n), and discuss fine tuning of your design.

(End)