Water Level Controlling System Using PID Controller

AIM (problem statement):

In certain applications such as chemical and industrial processes, it is important to keep the level of water or any other liquid in a tank or similar container at a certain desired level.

In this project, we present PID based controller system where the level of water is controlled by adjusting the rate of the incoming water flow to the container by varying the speed of a DC motor water pump that is filling the container.

The accuracy of the PID based controlling is demonstrated using the MATLAB and Simulink Software simulations.

COMPONENTS REQUIRED:

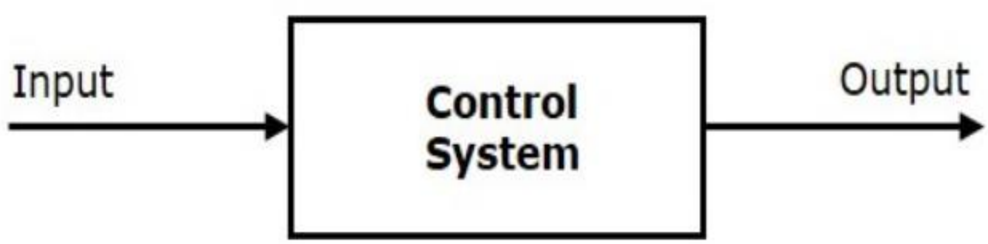
DC motor water pump, PID controller, water tank, pipes, inductors, resistance.

Platform used: MATLAB and Simulink.

THEORY:

Control Systems:

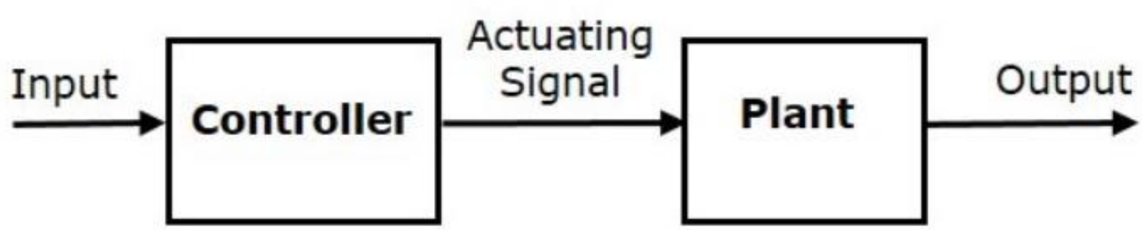
A control system is a system, which provides the desired response by controlling the output. The output is controlled by varying the input. It can range from a single home heating controller using a thermostat controlling a domestic boiler to large Industrial control systems which are used for controlling processes or machines.



Control Systems can be classified as **open loop control systems and closed loop** control systems based on the **feedback path**.

Open Loop Control Systems:

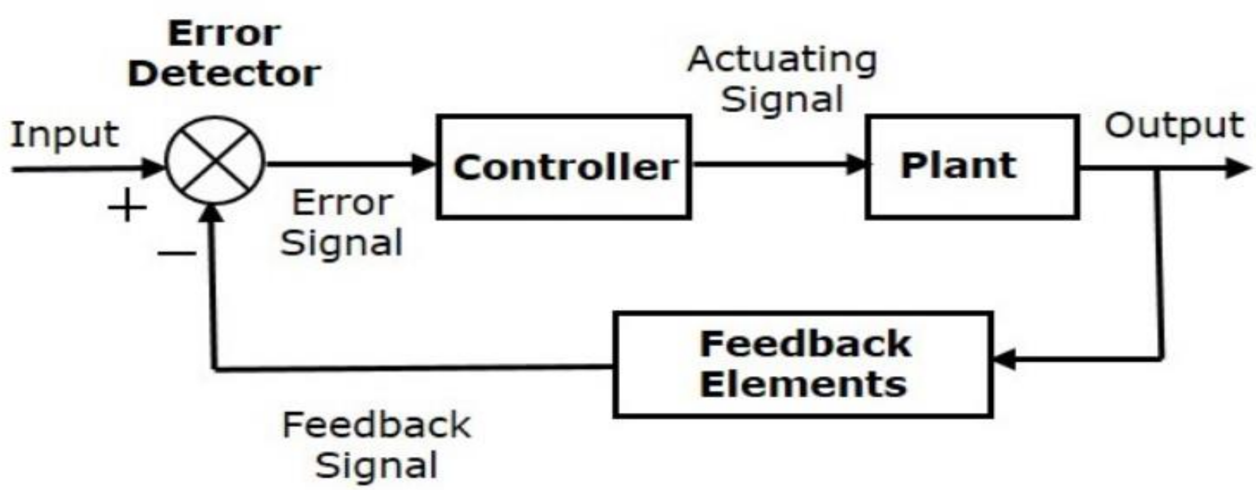
- In **open loop control systems**, output is not fed-back to the input. So, the control action is independent of the desired output.



- Here, an input is applied to a controller and it produces an actuating signal or controlling signal. This signal is given as an input to a plant or process which is to be controlled. So, the plant produces an output, which is controlled.

Closed Loop Control Systems:

- In closed loop control systems, output is fed back to the input. So, the control action is dependent on the desired output.



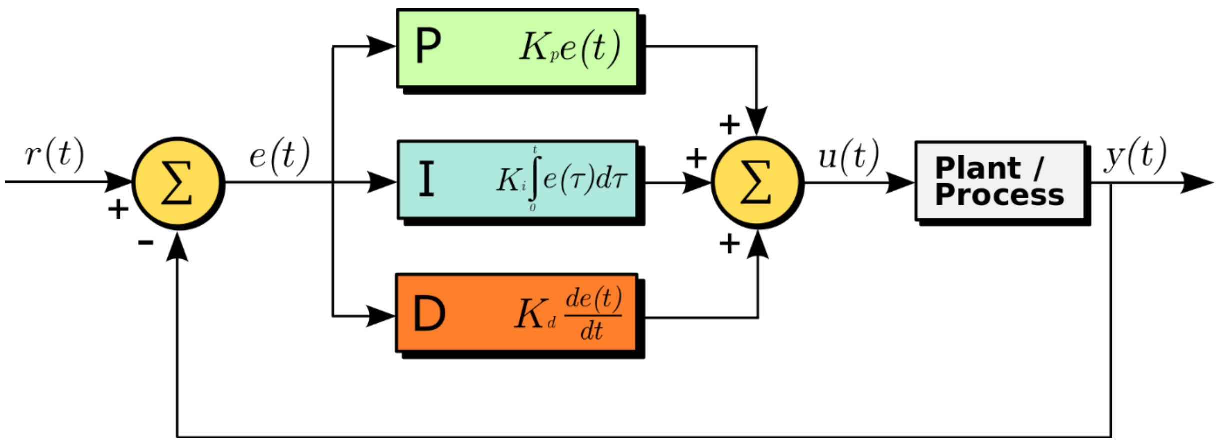
- The error detector produces an error signal, which is the difference between the input and the feedback signal. Instead of the direct input, the error signal is applied as an input to a controller.

- So, the controller produces an actuating signal which controls the plant. In this combination, the output of the control system is adjusted automatically till we get the desired response. Hence, the closed loop control systems are also called the automatic control systems

PID Controller:

A PID controller continuously calculates an error value as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

A closed-loop system like a PID controller includes a feedback control system. This system evaluates the feedback variable using a fixed point to generate an error signal. Based on that, it alters the system output. This procedure will continue till the error reaches Zero otherwise the value of the feedback variable becomes equivalent to a fixed point.



Mathematical theory:

The overall control function is:

$$u(t) = K_\text{p} e(t) + K_\text{i} \int_0^t e(t')\, dt' + K_\text{d} \frac{de(t)}{dt},$$

where

K_p is the proportional gain, a tuning parameter,

K_i is the integral gain, a tuning parameter,

K_d is the derivative gain, a tuning parameter,

$e(t) = \text{SP} - \text{PV}(t)$ is the error (SP is the setpoint, and PV(t) is the process variable),

t is the time or instantaneous time (the present),

τ is the variable of integration (takes on values from time 0 to the present t).

Equivalently, the transfer function in the Laplace domain of the PID controller is:

$$L(s) = K_p + K_i/s + K_d s,$$

Proportional term:

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant. It provides stable operation but always maintains the steady-state error. The speed of the response is increased when the proportional constant Kp increases.

The proportional term is given by:

$$P_{\mathrm{out}} = K_p e(t).$$

Steady-state error:

The steady-state error is the difference between the desired final output and the actual one. Because a non-zero error is required to drive it, a proportional controller generally operates with a steady-state error. Steady-state error can be corrected dynamically by adding an integral term.

Integral term:

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.

The integral term is given by:

$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau.$$

The integral term accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the setpoint value. So, there is a need to add a derivative term to control the overshooting.

Derivative term:

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd.

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Derivative action predicts system behavior and thus improves settling time and stability of the system.

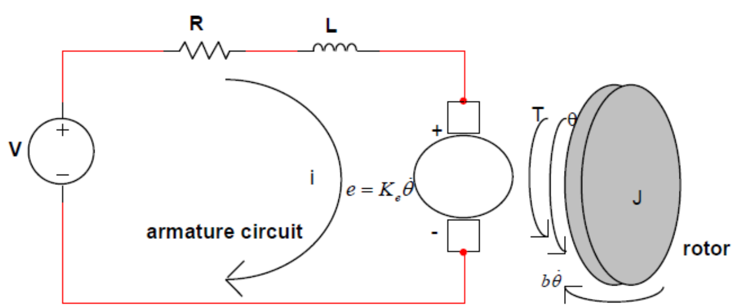
The derivative term is given by:

$$D_{\mathrm{out}} = K_{\mathrm{d}} \frac{de(t)}{dt}.$$

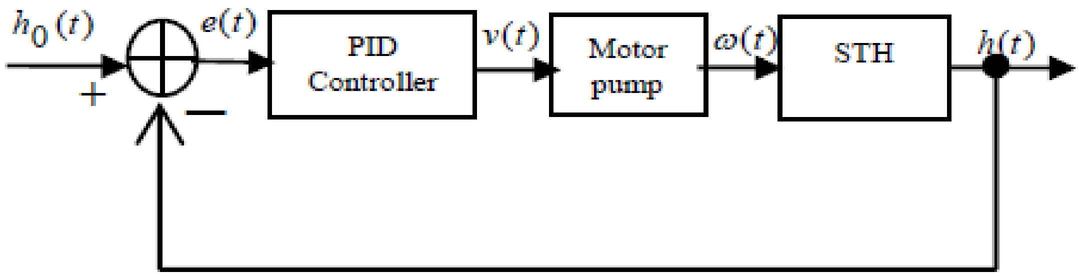
Increasing the *derivative time (T_d)* parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (T_d), because the Derivative Response is highly sensitive to noise in the process variable signal.

Circuit Diagrams:

1)Dc motor:



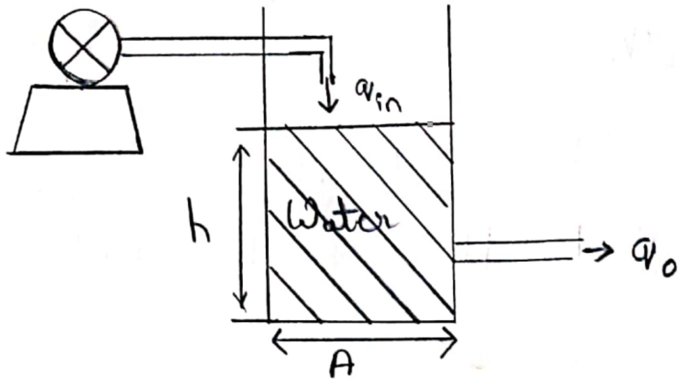
2)System model:



Procedure:

1) Based on the system model, the transfer functions of the PID controller, Motor Pump, Speed to height transformation block and the overall system must be derived.
2) Derivations are as follows:

→ Derivation:



- Continuity equation:

Let the motor pump water at a rate of q_{in} (m^3/s).

- Water is leaving the container at a rate of q_o (m^3/s)

$$\Rightarrow \quad q_{in} - q_o = A\frac{dh}{dt}$$

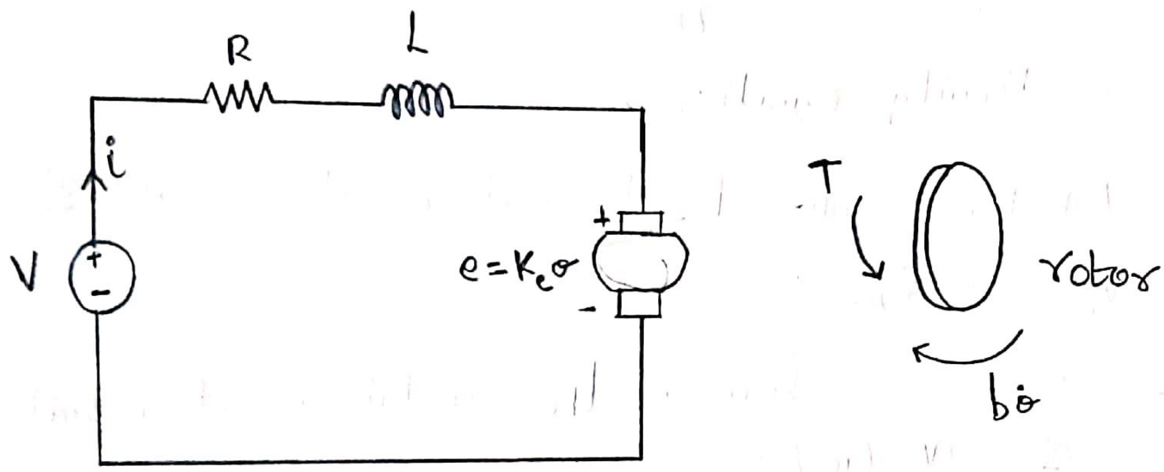
- For the sake of simplicity, lets assume linear resistance to flow.

$q_o = \frac{h}{R_f}$, where R_f (s/m^2) is the flow resistance.

- Similarly, let us assume a linear relationship between the speed and the incoming flow rate.

$$\omega(t) = K_f \, q_{in}$$

→ Now lets determine the impulse response for each of the blocks in the system model.



The input to the motor is a voltage source applied to the motor's armature. The output is the rotational speed of the shaft $\omega(t) = \frac{d\theta}{dt}$.

- We further assume that the frictional torque is proportional to shaft angular velocity.

Physical parameters of the motor:

1) J : moment of inertia of the rotor ($kg.m^2$)

2) b : motor viscous friction constant (Nms)

3) K_e : electromotive force constant ($V/rad.s$)

4) K_t : motor torque constant ($N.m/A$)

5) R : electrical resistance (in Ω).

6) L : electrical inductance (in H).

- Assuming armature controlled motor, torque $T = K_t i$ + the emf is proportional to the angular velocity of the shaft, $e = K_e \omega$

→ Net torque:

$$J\ddot{\theta} = K_t i - b\dot{\theta} \quad \text{--- } ①$$

→ KVL:

$$V - L\frac{di}{dt} - iR - e = 0 \quad \Rightarrow L\frac{di}{dt} + iR = V - K_e\dot{\theta} \quad \text{--- } ②$$

Applying Laplace on ① + ②

$$\Rightarrow s^2 J\theta(s) + sb\theta(s) = K_t I(s) \quad —③$$

$$sLI(s) + RI(s) = V(s) - sK_e\theta(s) \quad —④$$

$$\omega(s) = s\theta(s)$$

$$\Rightarrow sJ\omega(s) + b\omega(s) = K_t I(s)$$

$$sLI(s) + RI(s) = V(s) - K_e\omega(s)$$

eliminating $I(s)$, we get the frequency response as:

$$P_m(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{s^2JL + s(JR + bL) + Rb + K_tK_e} \quad —⑤$$

→ the PID controls the applied voltage V on the basis of the error signal.

$$V(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt}$$

- Frequency response is

$$C(s) = \frac{V(s)}{e(s)} = K_p + \frac{K_i}{s} + sK_d \quad —⑥$$

→ Angular velocity to height conversion

$$q_{in} - q_o = A\frac{dh}{dt} \quad —⑦$$

$$q_o = \frac{h}{R_f} \quad —⑧$$

$$\frac{\omega(t)}{K_f} = q_{in} \quad —⑨$$

Using 7, 8, 9

$$\frac{w(t)}{K_f} - \frac{h}{R_f} = \frac{A\,dh}{dt}$$

Applying Laplace, we get

$$R_f\, w(s) - K_f\, h(s) = K_f R_f A\, s\, h(s)$$

$\Rightarrow$ frequency response transfer function is,

$$G(s) = \frac{h(s)}{w(s)} = \frac{R_f}{K_f + sK_f R_f A} \quad \text{— (10)}$$

→ Overall Transfer function :

$$C(s)\, P_m(s)\, G(s) = \frac{h(s)}{e(s)} = F(s) \quad \{\text{say}\}$$

we need $\frac{h(s)}{h_o(s)}$.

$$e(t) = h_o(t) - h(t)$$

$$\Rightarrow e(s) = h_o(s) - h(s)$$

$$\Rightarrow F(s) = \frac{h(s)}{e(s)} = \frac{h(s)}{h_o(s) - h(s)}$$

$$\Rightarrow \frac{h(s)}{h_o(s)} = \frac{F(s)}{1+F(s)}$$ { System transfer function }

$$= G_{sys}(s)$$

Calculation:

$$G_{sys}(s) = \frac{As^2 + Bs + C}{Es^4 + Fs^3 + Gs^2 + Hs + I}$$

where,

$A = K_d \cdot K_t \cdot R_f$ $B = K_t \cdot K_p \cdot R_f$ $C = K_i \cdot K_t \cdot R_f$

$E = K_f \cdot R_f \cdot Ar \cdot J \cdot L$

$F = J \cdot L \cdot K_f + (J \cdot R + b \cdot L) K_f \cdot R_f \cdot Ar$

$G = (J \cdot R + b \cdot L) \cdot K_f + (R \cdot b + K_t \cdot K_e) K_f \cdot R_f \cdot Ar + K_d \cdot K_t \cdot R_f$

$H = K_f \cdot (R \cdot b + K_t \cdot K_e) + K_t \cdot K_p \cdot R_f$

$I = K_i \cdot K_t \cdot R_f$

→ To obtain the system without PID,

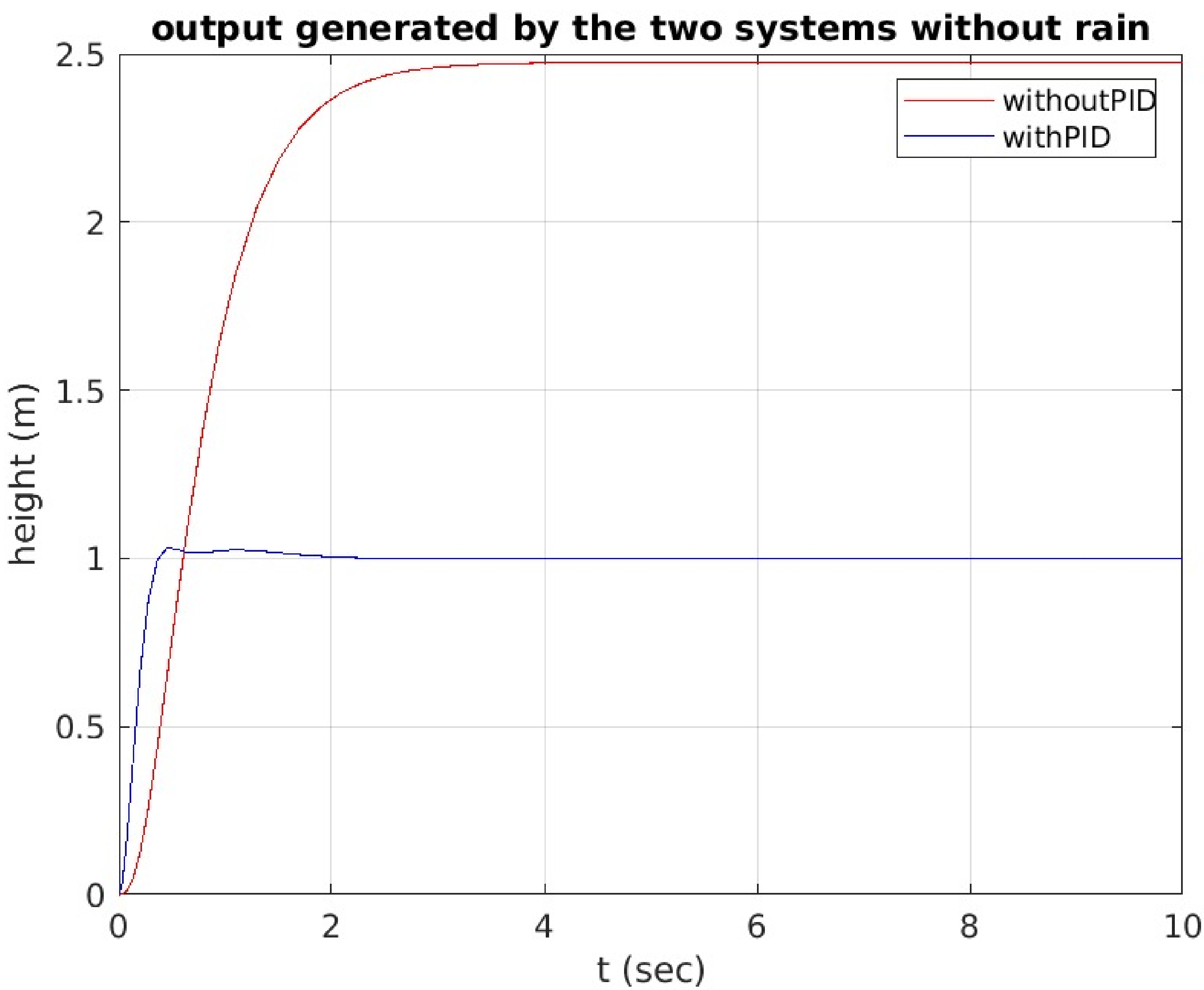
we can remove the feedback & instead of a PID we can have a height to voltage converter.

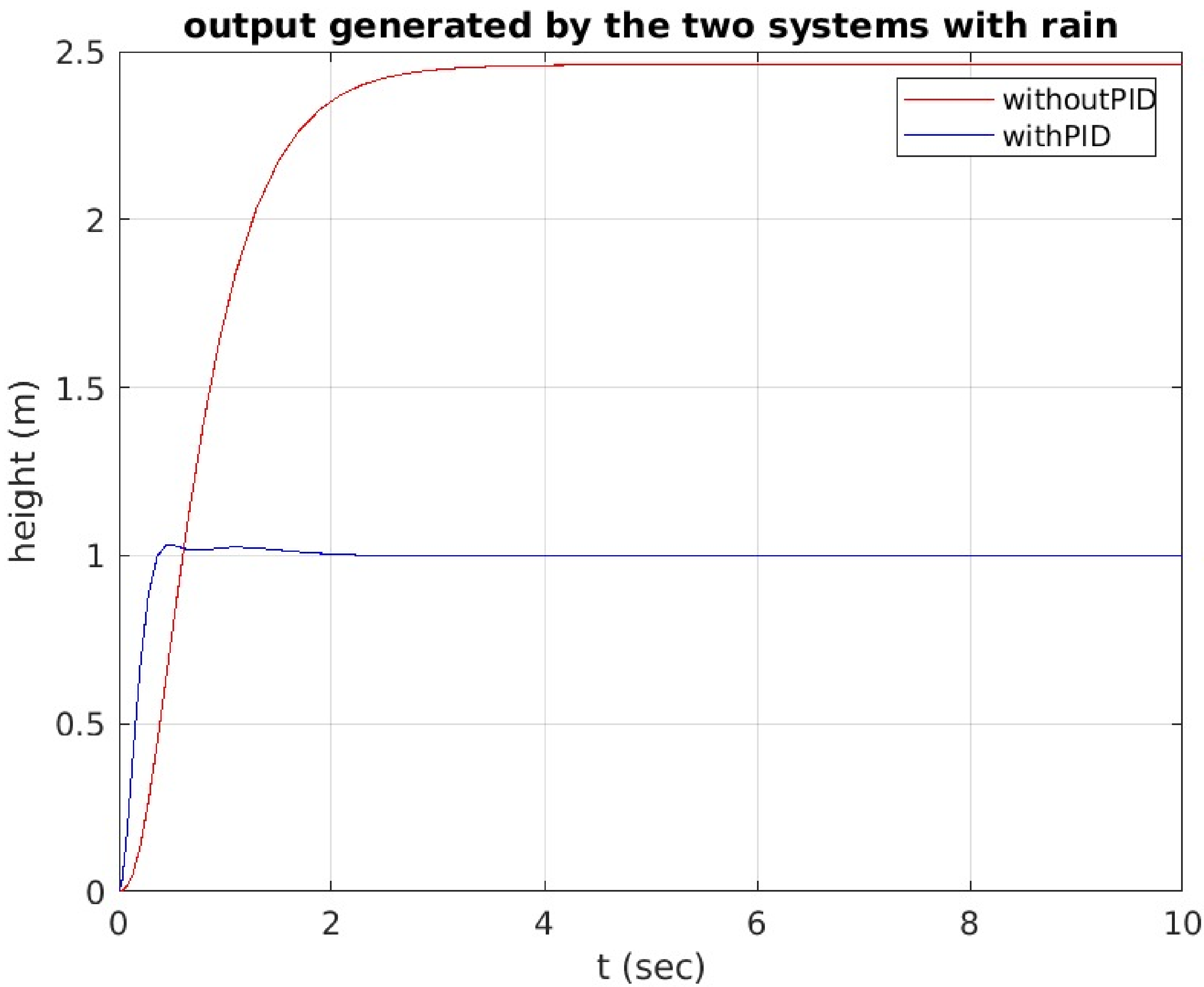
3)After deriving the final transfer function Gsys(s), we need to make the circuit in the Simulink similar to the system model and input the numerator and denominator coefficients derived above in the transfer function block.

4)Then we can plot the graphs using matlab and see whether the final water level is equal to the desired set point level.

Observations:

Here h0=1 is the desired set point.





Results:

We can see that with the usage of pid controller we are efficiently able to control the water level to the desired set point.