Task 1

Analysis of non-linear and linearized water tank model with and without PID control.

Contents

Model and linearization

Differential equation describing the tank water level:

$$\frac{d}{dt}H = \frac{bV - a\sqrt{H}}{A}$$

This ode is non-linear in $\sqrt{H}$. However, we can approximate this by a first order taylor expansion/linearization in a neighbourhood $H_0 + \hat{H}$ around the stationary point H_0:

$$\sqrt{H_0 + \hat{H}} \approx \sqrt{H_0} + \frac{1}{2\sqrt{H_0}} \cdot (H - H_o)$$

With this linearization we arrive at the linear ODE:

$$\frac{dH}{dt} = \frac{b}{A}V - \frac{a}{2A}\sqrt{H_0}(H - H_0) - \frac{a\sqrt{H_0}}{A}$$

State Space Representation

We can find the state space representation of this system (ignoring the non-homogeneous part):

$$\frac{d}{dt}H = \left[\, -\tfrac{a}{2A\sqrt{H_0}} \,\right] H + \left[\, \tfrac{b}{A} \,\right] V$$

Alternatively, with our constants and linearization point:

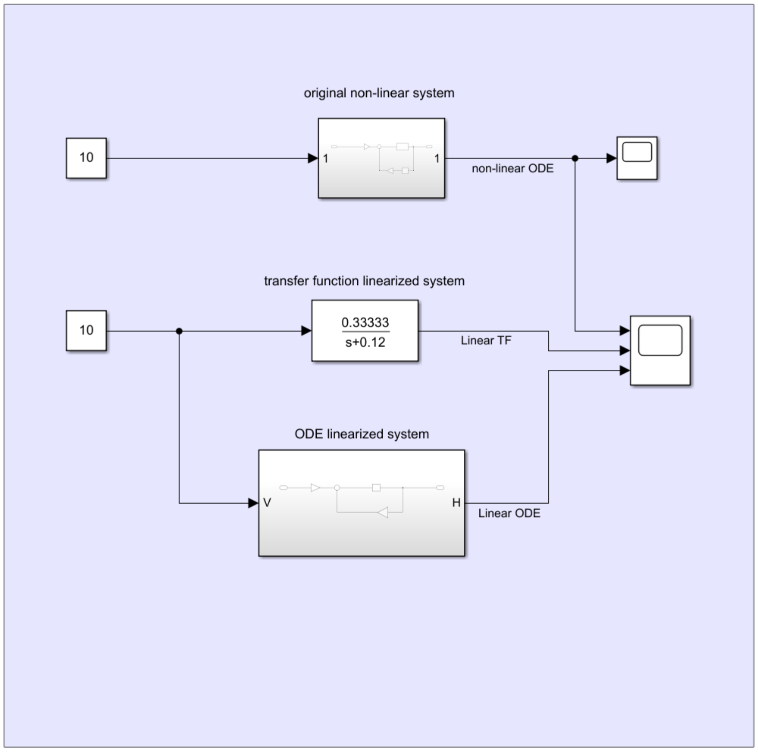
$$\dot{H} = \left[\, -\tfrac{3\sqrt{10}}{80} \,\right] H + \left[\, \tfrac{1}{3} \,\right] V$$

Our state space consists of a one-dimensional state vector H and a one-dimensional control vector V

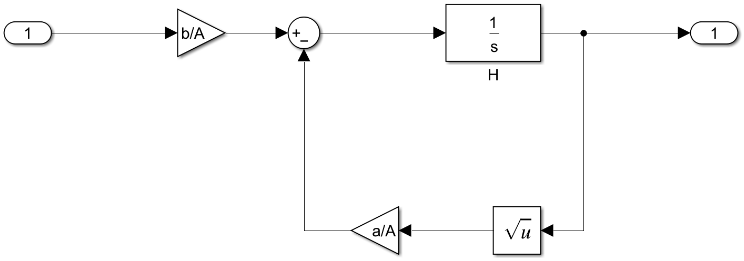
Simulink implementation

High-level overview of non-linear system, linear TF system and linear ODE system.

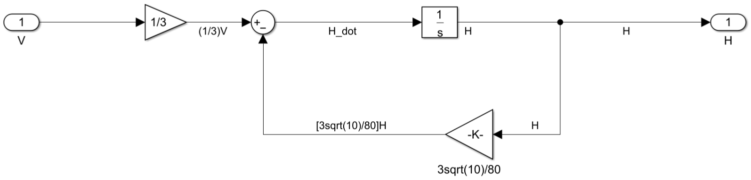
Without control



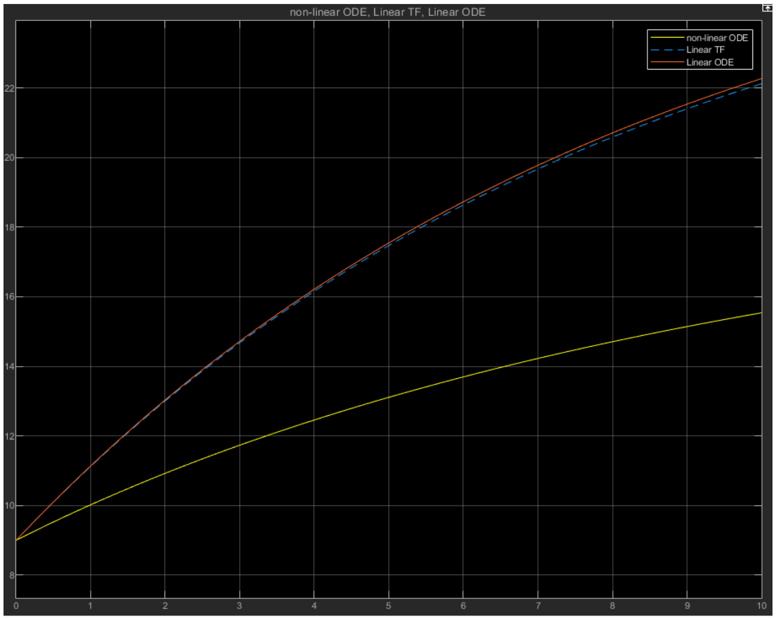
Open-loop non-linear system:



Open-loop linear system:



Plotting the responses of each system (input 10, initial output 9):



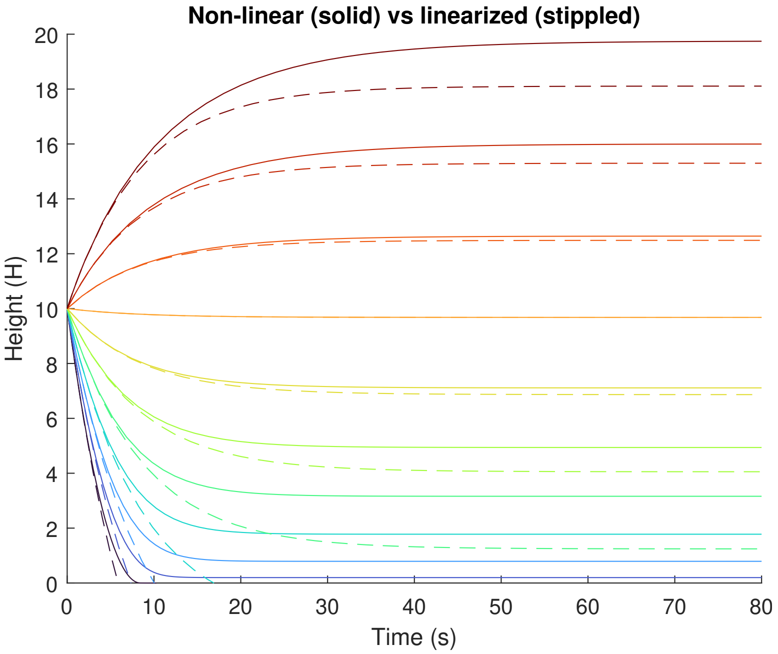
As can be seen, the TF and ODE implementation are identical (up to number rounding), while the linear approximations become less and less accurate relative to the true non-linear system.

Simulation of open loop system

For fun, we can simulate the non-linear and linearized differential equations to inspect the accuracy of the linear approximation. To do this I use matlab's ode45 which can simulate most ordinary differential equations The non-linear and linear ODE's are defined in tank_system_linear.m and tank_system_nonlinear.m, respectively.

```
A = 24; b = 8; a = 18; %water tank parameters

timespan = [0, 80]; %simulaion interval
H0 = 10; %initial level
colormap = turbo(11);

% Solve the and plot the ODE's
figure;
hold on;
for V = 0:10
    [t, H] = ode45(@(t, H) tank_system_nonlinear(t, H, a, b, A, V), timespan, H0);
    plot(t, H, 'Color', colormap(V+1, :));

    [t, H] = ode45(@(t, H) tank_system_linear(t, H, a, b, A, V, H0), timespan, H0);
    plot(t, H,'--','Color', colormap(V+1, :));
end

xlim(timespan);       ylim([0,20]);
xlabel('Time (s)');   ylabel('Height (H)');
title('Non-linear (solid) vs linearized (stippled)');
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```



Transfer function response

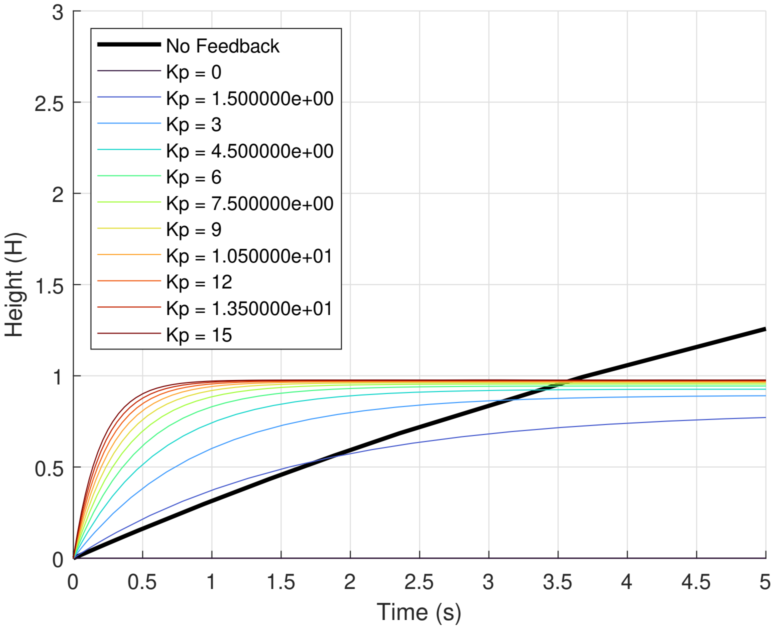
The transfer function is determined by taking the laplace transform of the homogeneous linear ODE, which after rounding gives:

$$\frac{H(s)}{V(s)} = \frac{\frac{b}{A}}{s + 0.16\frac{a}{A}}$$

Here, I simulate and plot the closd loop step responses for varying proportional gains:

```
close;

H0 = 0; %initial tank level
V = 1; %step response no feedback
H_desired = 1; %setpoint
colormap = turbo(11);

step_timespan = [0, 5];
figure;
grid on;
hold on;
```

```
%without feedback:
Kp = 0;
[t, H] = ode45(@(t, H) tank_system_step(t, H, a, b, A, H_desired, H0, V), timespan, H0);
plot(t, H, 'Color', 'k', 'LineWidth', 2, 'DisplayName', 'No Feedback');

for i = 0:10
    Kp = i*1.5;
    [t, H] = ode45(@(t, H) tank_system_step_feedback(t, H, a, b, A, H_desired, H0, Kp), time
    plot(t, H, 'Color', colormap(i+1, :), 'DisplayName', sprintf('Kp = %d', Kp));
end

xlim(step_timespan);       ylim([0,3]);
xlabel('Time (s)');   ylabel('Height (H)');
legend('Location', 'northwest');
```



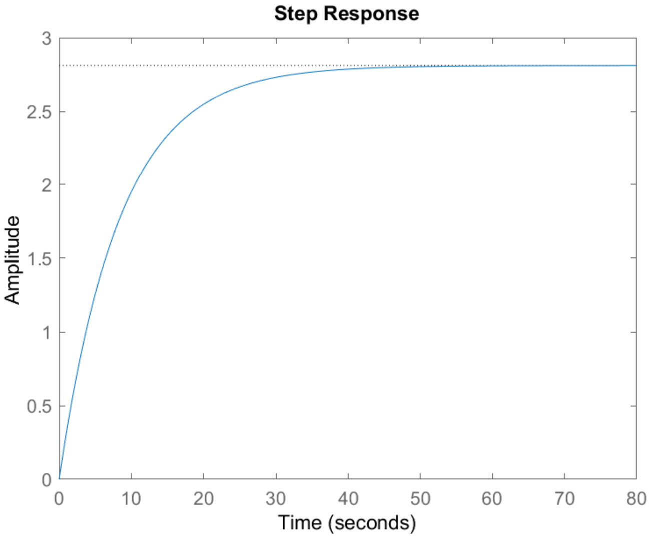
Since the system is a first-order linear system without disturbances, it behaves nicely and doesn't reach any oscillations.

Step response with control

Now we develop a complete PID controller for our plant using the transfer function of the linearized model.

```
close;
```

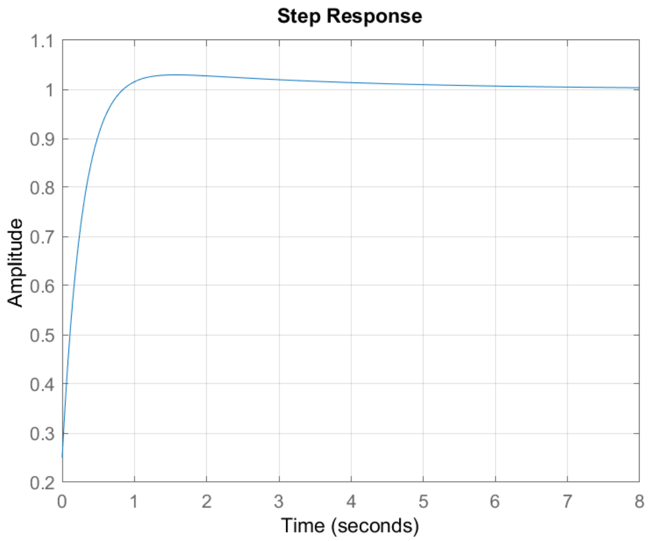
```
s = tf('s');

A = 24; b = 8; a = 18; %water tank parameters
tf_linear = (b/A)/(s+1/(2*sqrt(10))*(a/A)); %defining transfer function as given in the task
step(tf_linear);
```



Now we tune all the gains to achieve the desired response

```
Kp = 15;
Ki = 5;
Kd = 1;

H = tf_linear; %plant transfer function
C = Kp + Ki/s + Kd*s; %PID transfer function

CH = C*H; %closed loop transfer function
closed_tf = CH/(CH+1);
step(closed_tf);

info = stepinfo(closed_tf);
overshoot     = info.Overshoot;
rise_time     = info.RiseTime;
settling_time = info.SettlingTime;

% Display results
fprintf('Overshoot: %.2f%%\n', overshoot);
fprintf('Rise Time: %.2f s\n', rise_time);
fprintf('Settling Time: %.2f s\n', settling_time);
grid on;
```

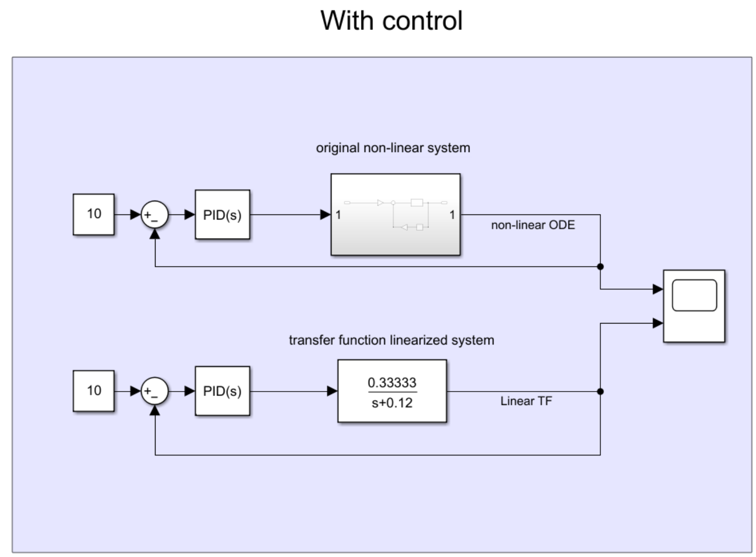
```
Overshoot: 2.95%
Rise Time: 0.51 s
Settling Time: 3.76 s
```

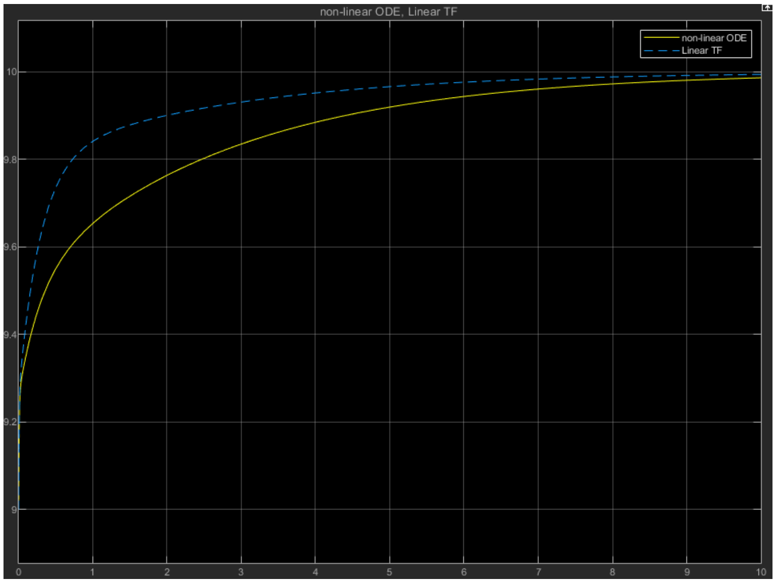


Applying the control to the non-linear and linearized plants in simulink

With our conservative gains [Kp, Ki, Kd] = [15, 5, 1] We achieved a nice response.

Here I plot the response for a setpoint of 10 an an initial tank level of 9, for both the non-linear and linear systems.





The response of the linear system here is not just a scaled version of the closed_loop unit step response, since the setpoint and initial conditions arent just scaled versions of the unit step (theres a translation in the setpoint). If we instead plotted the response of a setpoint of 10 and an initial condition of 0, the response of the linearized system would be identical to a scaled version of the unit step response. We also see again that the linear approximation starts out good, but diverges from the true state of the system as we get further away from the linearization point.