Lab 1 Coupled drives

December 23, 2020

Jaya Parmar

1 Introduction

Our aim is to study the step response on a coupled drive system in three different scenarios.

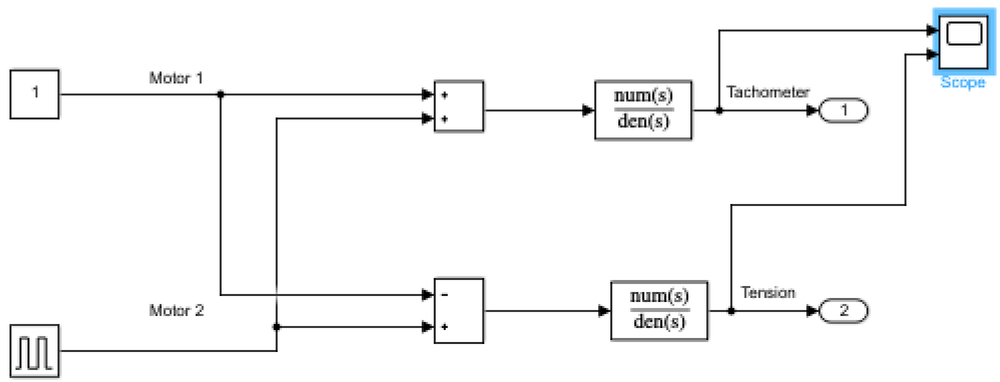
- First as the coupled system.
- Second as the decoupled system.
- Third as the decoupled system with feedback control.

In the third case, we test with different control parameters and finalize a suitable setting for controller gain values.

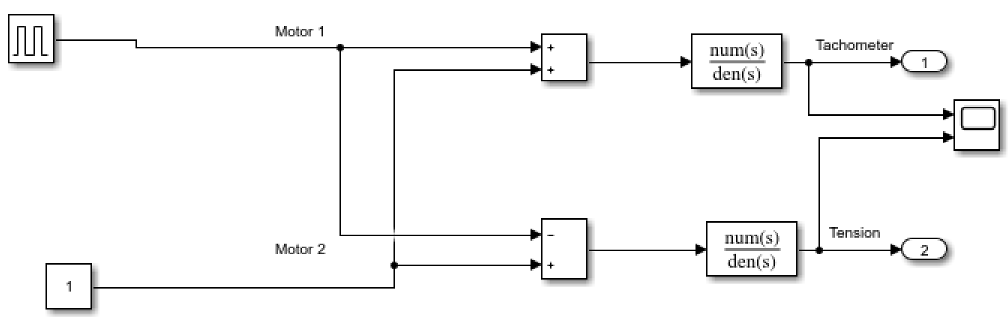
2 Method

In this section, we will show the block diagram of every system one by one. The scope measurements will be shown and discussed in the results sections.

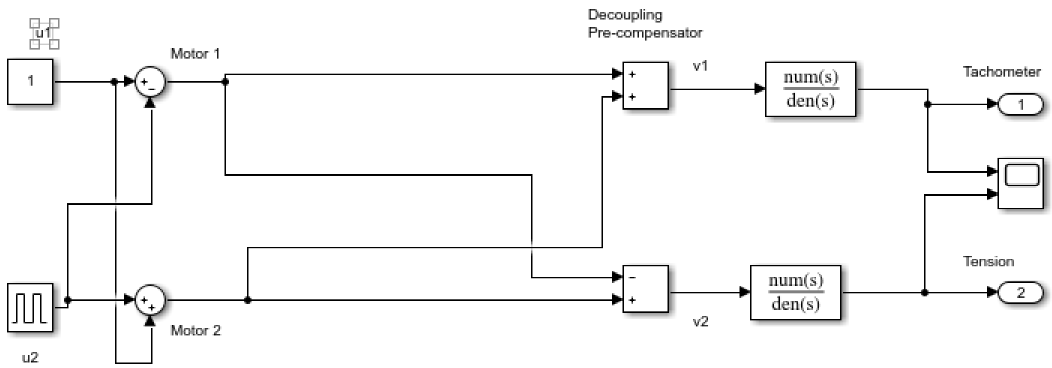
1. Coupled drives with v1=1 and v2=pulse



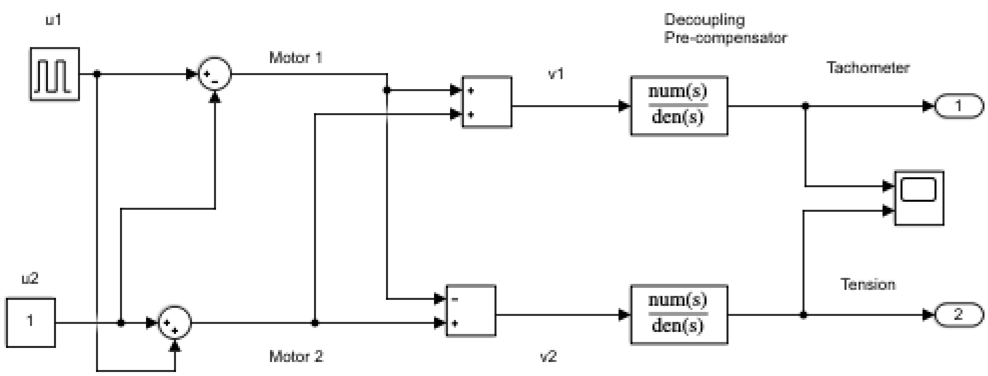
2. Coupled drives with v1=pulse and v2=1



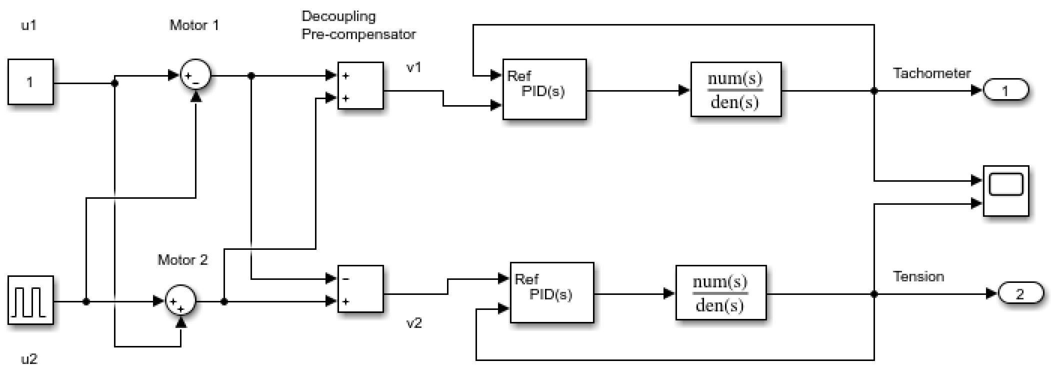
3. Decoupled drives with u1=1 and u2=pulse



4. Decoupled drives with u1 = pulse and u2=1



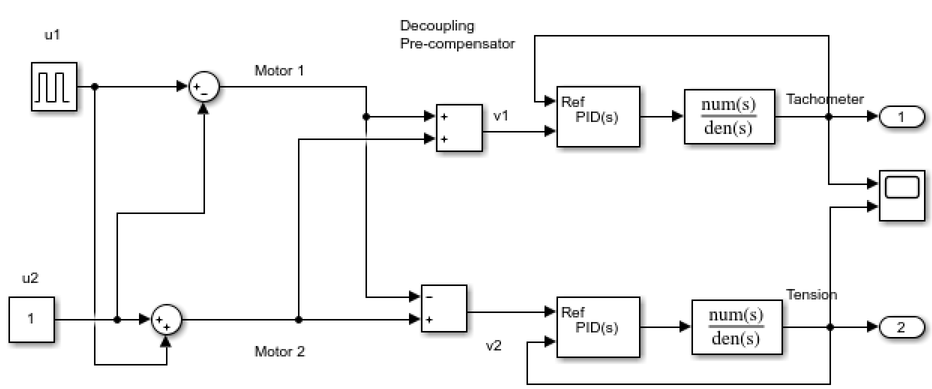
5. PID control with u1 =1 and u2 = pulse



- Ideally, we will see output of the same shape as input i.e. y1 constant and y2 pulsating in this case. It can be achieved by putting a PID controller in the feedback loop of tachometer and tension control.

- In the tachometer loop, transfer function of the system is $\frac{0.87/2}{0.66s+1}$. Pole of the system is $\frac{-1}{0.66} = -1.51515$. Using $k_{p1} = -1.51515$; $k_{i1} = -1.51515$ and $k_{d1} = +1.51515$ we get the output shown in figure 5.3
- In the tension loop, PID controller has proportional gain $k_{p2} = 0.01$; integral gain $k_{i2} = 5$ and $k_{d2} = 0.04$.

6. PID control with u1 = pulse and u2 = 1



- We want y1 pulsating and y2 constant in this case. We put PI controllers each in the feedback loop of tachometer and tension control.

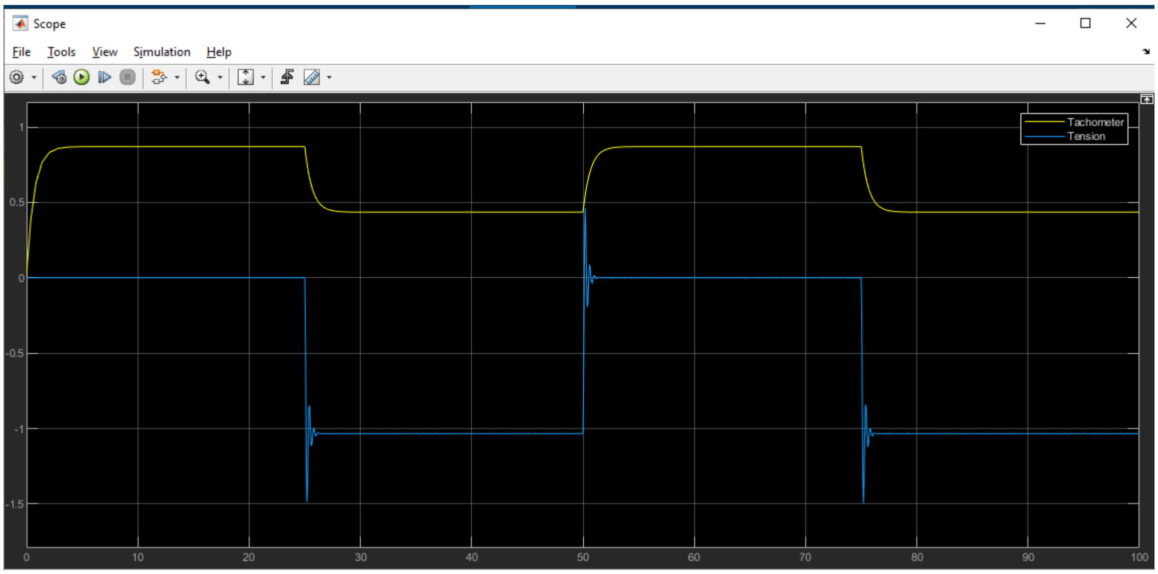
- In the tachometer loop, $k_{p1} = -1.51515$; $k_{i2} = -3.1$ and $k_{d2} = 0.0001$.

- In the tension loop, proportional gain $k_{p2} = 1$; integral gain $k_{i2} = 1000$ and $k_{d2} = 10$.

3 Results

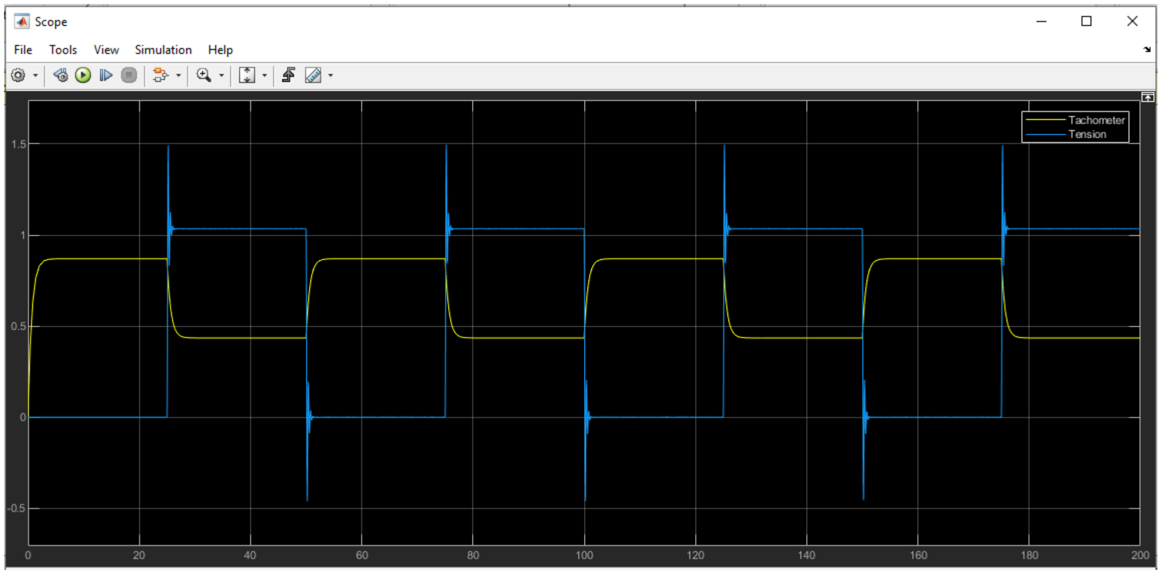
y1 is the tachometer ouput in yellow and y2 is the tension ouput in blue

1. Figure 1- Coupled drive with v1 constant and v2 pulse



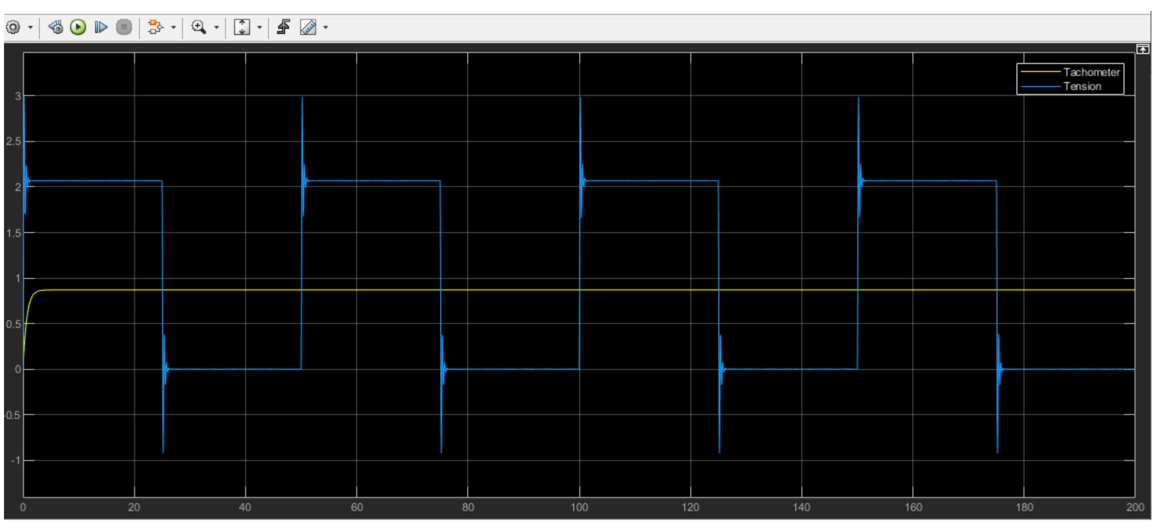
   y2 pulsates between values 0 and -1. Since the pulse width is 50, the positive pulse drops at 25 and rises again at 50.
   y1 with a constant input 1 shows a pulsating output too. It does not reach an output of 1 instead keeps pulsating between 0.9 and 0.45 (approx values). y1 stays high for the first 25 steps and drops when y2 drops.
   Note that coupled input to y2 is v2 - v1 which leads to pulsation between 0 and -1 instead of 0 and +1.

2. Figure 2- Coupled drive with v1 pulse and v2 constant



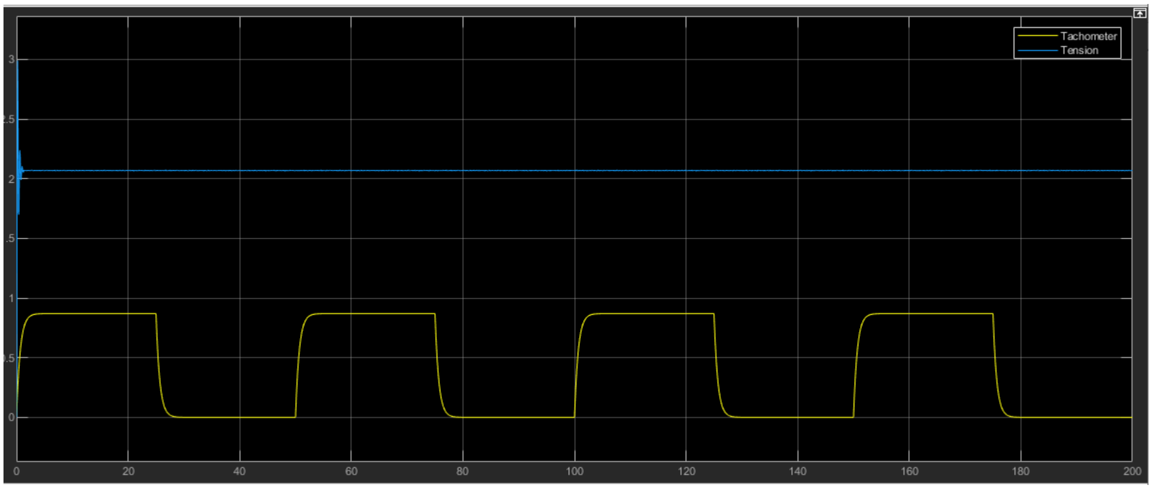
Between steps 0 and 25, v1 = 1 and v2 = 1. This leads to y2 = 0. Between steps 26 and 50, v1=0 and v2 = 1 making y2 = 1.
y1 = v1 + v2. Between steps 0 and 25, y1 should be 1 (cannot overshoot input 1). Between steps 26 and 50, y1 should be 1. Due to the coupling effect, output y1 varies between 0.9 and 0.45.

3. Figure 3- Decoupled drive with u1 constant and u2 pulse



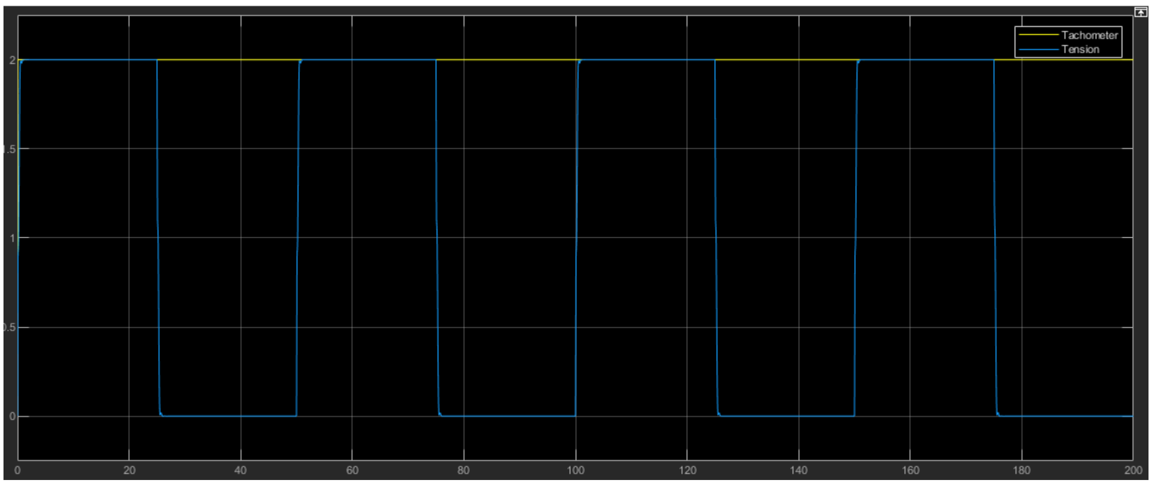
Note that y1 (yellow line) is now constant around 1 and y2 (blue line) is pulsating. Compare with Figure 1, the two ouputs are decoupled from each other.

4. Figure 4- Decoupled drive with u1 pulse and u2 constant



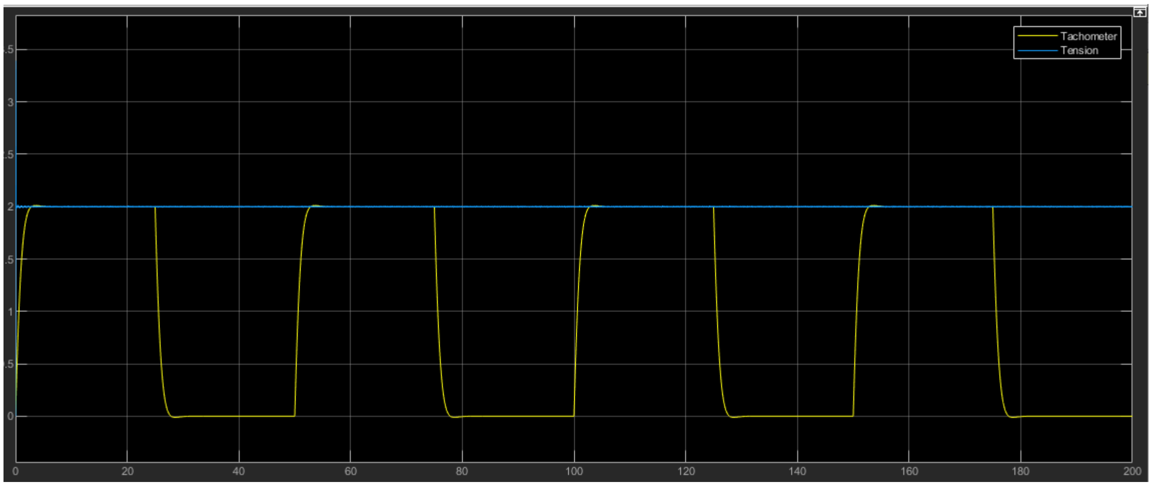
The decoupling is prominently seen in this figure (compare with figure 2). y1 pulsates between 0 and 1 whereas y2 = 2 .

5. Figure 5- PID control with u1 constant and u2 pulse



Now the output y1 stays constant and y2 pulsates which is what we wanted.

6. Figure 6- PID controller decoupled drive with u1 pulse and u2 constant



y1 pulsates and y2 stays constant.

4 Discussion

1. **Decoupling**
We introduce two new input variables u1 and u2 from which v1 and v2 are constructed so that the transfer function, u1 − y1 and u2 − y2 are decoupled. This is done by using a Decoupling Pre-compensator with inputs u1 and u2 and outputs v1 and v2 in both tachometer and tension control.

2. **PID controllers**
We controlled the tachometer loop by adjusting the controller gain with pole values of transfer function.
The tension loop has a second order transfer function. We should have used $k_{p2} = -p_1 - p2$ and $k_{i2} = p_1 * p_2$. But the transfer function had complex poles at -4.0250 +15.7098i and -4.0250 -15.7098i hence we could not use them to model the controller gains. With trail and error, we finalized the proportional gain value as 1 since it would contribute to the bulk of output change, which in our case is 1. We choose integral gain to accelerate the movement of the process towards setpoint. The overshoot from the setpoint value is adjusted using derivative gain.