Project 2 Report – Elevator Control System

David Eelman & Thomas Abdallah

2016-12-09

Contents

[1 – Motor Modelling & Simulation 2](#_Toc468959460)

[1.0 DC Motor Model 2](#_Toc468959461)

[1.1 Tachometer Model 2](#_Toc468959462)

[1.2 Servo-Amplifier Model 2](#_Toc468959463)

[1.3 Complete Motor Model 3](#_Toc468959464)

[1.4 Simulation vs Real-World Results 3](#_Toc468959465)

[2 – Motor Speed Control with Op-Amps 5](#_Toc468959466)

[2.0 Controller Targets 5](#_Toc468959467)

[2.1 Controller Development 5](#_Toc468959468)

[2.2 Op-Amp Controller Design & Implementation 6](#_Toc468959469)

[2.2 Simulation vs Real-World Results 7](#_Toc468959470)

[3 – Elevator System Modelling 8](#_Toc468959471)

[3.0 Modelling Elevator Dynamics 8](#_Toc468959472)

[3.1 Complete System Model 8](#_Toc468959473)

[3.2 Simulation Results 9](#_Toc468959474)

[4 – Elevator System Control 10](#_Toc468959475)

[4.0 Controller Targets 10](#_Toc468959476)

[4.1 Controller Development 10](#_Toc468959477)

[4.2 Controller Implementation 10](#_Toc468959478)

[4.3 Simulation vs Real-World Results 11](#_Toc468959479)

# 1 – Motor Modelling & Simulation

## 1.0 DC Motor Model

Simulink was used to create a model for the DPP240-29V48 motor (Figure 1). Parameters for the motor were modelled based on values from the manufacturer datasheet. The model input is voltage and output is speed in rad/s.

Voltage 
Ke 
CEMF 
Ke 
CEM F 
To Workspace2 
netVoltage 
Add 
netTorque 
Addl 
Resistance 
Inductance 
Inertia 
b 
Damping 
d/dt0) 
d2/d2(tneta) 
current 
Integrator 1 
Integrator 
current 
To Workspace 
sped 
Speed 
To Workspace 1 

Figure - DC Motor Model

## 1.1 Tachometer Model

The tachometer included with the DPP240-29V48 motor was modelled with speed (rad/s) as its input and voltage as its output (Figure 2). The model parameter for speed to voltage of the tachometer was pulled from the manufacturer datasheet.

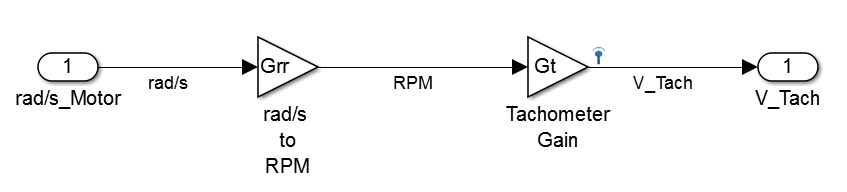


Figure - Tachometer Model

## 1.2 Servo-Amplifier Model

The servo amplifier was modelled with set point voltage as its input and applied voltage to the motor as its output (Figure 3). Gain is configurable as a MATLAB variable.

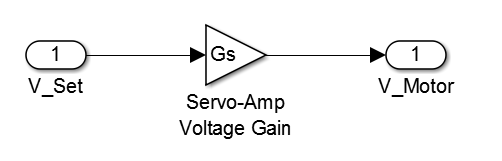


Figure - Servo Amplifier Model

## 1.3 Complete Motor Model

The complete motor model is a cascaded system including the servo amplifier, DC motor and tachometer (Figure 4). The model input is the set point voltage and the output is tachometer voltage.

V Set 
Step 
V Motor 
Voltage 
Speed 
DC Motor 
rad's Motor 
V Tach 
Scope 
Servo-Amp 
Tachometer 

Figure - Complete Motor Model

The transfer function for the complete DC motor model was derived for controller development and is shown below with parameters from Table 1.

Table - Elevator System Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Symbol** | **Value** |
| Servo Amplifier Gain | Gs | User configurable |
| Tachometer Gain | Gt | 14V / 1000 rpm |
| Rad/s to RPM | Grr | 9.5494 rpm/rad/s |
| Motor Torque Constant | Kt | 0.1 N\*m/A |
| Motor Voltage Constant | Ke | 0..1 V/rad/s |
| Motor Inductance | L | 5.27 mH |
| Motor Resistance | R | 4 Ohms |
| Rotor Damping | J | 2.32x10-4 N\*m\*s |
| Rotor Inertia | b | 2.5x10-5 Kg\*m2 |

## 1.4 Simulation vs Real-World Results

Figure 5 shows the step-response of the DC motor model. Figure 6 shows the real-world step response of the DC motor when driven with a 10V step input as captured on an oscilloscope.

Time Series Plot:speed 
70 
10 
0.01 
0.02 
0.03 
0.04 
Time (seconds) 
0.05 
0.06 
0.07 
0.08 

Figure - Motor Model Step Response

2.00 v 
Rise Time 
Value 
16.76ms 
Mean 
16.76m 
10.0ms 
16.76m 
100kS/s 
10k oints 
Max 
16.76m 
4.32 v 
Std Dev 
0.000 
18 Nov 2016 

Figure - Real-world DC Motor Step Response (Dark Blue = Tachometer, Light Blue = Input Signal)

The raw data from the oscilloscope was imported to MATLAB and the tachometer signal was filtered to eliminate the high frequency noise. Figure 7 shows the real-world step-response together with the modelled step-response. The overall rise-time of the real-world DC motor system closely matches the model.

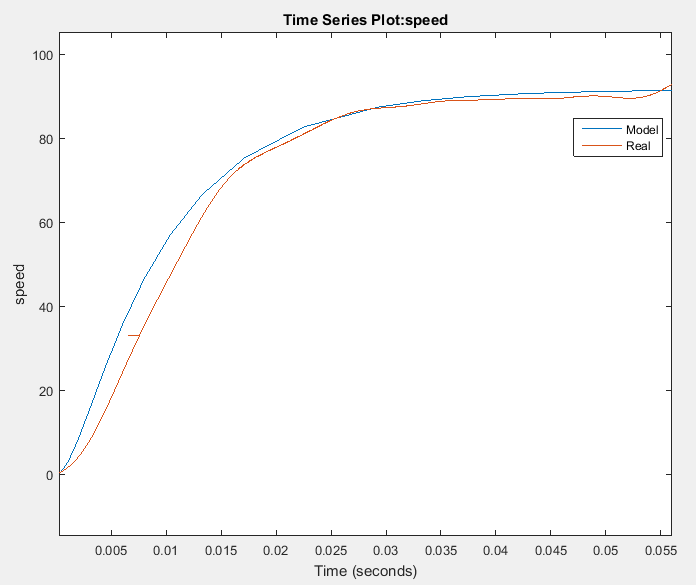


Figure - DC Motor Step Response (Blue = Model, Red = Real-world)

# 2 – Motor Speed Control with Op-Amps

## 2.0 Controller Targets

A speed controller was to be developed with the following step-response characteristics:

* 90% of final value in 10ms
* Close to critically damped

## 2.1 Controller Development

MATLABs *pidtool* function was used to tune a PI controller for the motor model shown in Figure 4. An integral only controller with a gain of 16.456 was found to meet our performance targets (Figure 8).

1: Parallel 
Options 
ENTROLLER 
erence tracking 
0.9 
0.8 
0.7 
0.6 
0.5 
0.4 
0.3 
0.2 
0.1 
= O, Ki=1646 
Domain: 
Time 
DESIGN 
Slower 
Aggressive 
Resoonse Time (seconds) 
Transient Behavior 
.1006 
Faster 
Robust 
TUNING TOOLS 
Step Plot: Reference tracking 
System. Tuned response 
I/O: u to ufb 
Time (seconds): 0.1 
Amplitude: 0.899 
Des "n 
Para meters 
System: Tuned response 
I/O: u to ufb 
Time (seconds): 0.258 
Amplitude: 0.998 
0.3 
response 
0.05 
0.1 
0.15 
02 
Time (seconds) 
0.25 
0.35 
0.4 
Controller Parameters: KP 

Figure - pidtool Controller Design

## 2.2 Op-Amp Controller Design & Implementation

The op-amp controller design calculations for resistor and capacitors values are as follows:

We chose to design with a capacitor value of 0.1µF, allowing a simple calculation for the input resistor.

Figure 9 shows the op-amp speed controller circuit as-built.

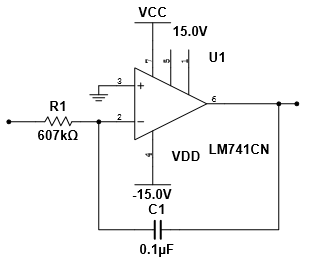


Figure - PI Op-amp Controller

The differential input op-amp circuit shown below in Figure 10 produces an error signal that is the difference between Vin and Vtach. That error signal is then fed into the PI controller to produce the desired controlled output.

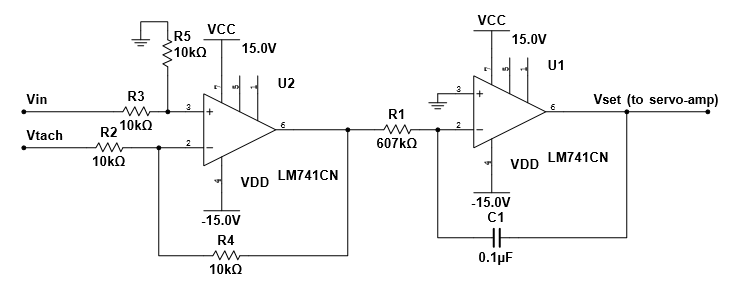


Figure - Op-amp Controller Circuit

## 2.2 Simulation vs Real-World Results

The op-amp controller circuit was tested with a 2.5V square wave input at 0.5 Hz and the results are shown in the oscilloscope capture (Figure 11). At the transient from 0 to 2.5V, the error signal increases and corrects for the change in output as we would expect for the controlled system.

20. Oms 
AFG 
5.00 v 
5.00 v 
S uare 
2.00 v 
500.00mHz 
2.5000 v 
SO 
10k ints 
23 Nov 2016 
1.80 v 

Figure - DC Motor Response (Dark Blue = Input, Light Blue = Error, Pink = Controller Output, Green = Tach Output)

# 3 – Elevator System Modelling

## 3.0 Modelling Elevator Dynamics

The existing motor model was expanded to include the effective inertia of the elevator cars, and the output of the model was changed from speed to distance. The effective inertia of the elevator cars is calculated to be:

Where, Rp = Pitch Radius of Drive Pulley (m) = 0.0145542m

Mc = Mass of each elevator car (kg) = 0.4kg

The effective inertia of the entire system can be calculated as the sum of elevator inertia and motor + tachometer inertia.

Jeq = Jm + Jc = 1.480441x10-3kgm2

## 3.1 Complete System Model

The complete elevator system model is similar to the original motor model with the addition of appropriate gain terms to convert rotational speed into elevator car position (Figure 12). The model input is the set point voltage and the output is elevator position in metres.

V Set 
Step 
V Motor 
Voltage 
Speed 
DC Motor 
rad's 
Motor 
V Tach 
Scope 
Servo-Amp 
Tachometer 
Speed Position 
Motor Speed to Car Position 

Figure - Elevator System Model

The transfer function for the complete elevator system model was derived and is shown below with parameters from Table 1.

## 3.2 Simulation Results

The step-response of the elevator system model is shown in Figure 13. As expected, the position increases linearly after the initial transients.

1.4 
1.2 
0.8 
0.6 
0.4 
0.2 
-02 
Step Response 
x: 10.01 
Y: 1.261 
Time (seconds) 

Figure - Step-Response of Elevator System Model

# 4 – Elevator System Control

## 4.0 Controller Targets

A controller for elevator car position was to be developed with the following criteria:

* Controller effort limited to +/- 2V
* Close to critically damped
* Step-response as fast as possible with servo-amp gain of 4

## 4.1 Controller Development

MATLABs *pidtool* function was used to tune a PI controller for the elevator system model shown in Figure 12Figure 4. A proportional only controller with a gain of 1.992 was found to meet our performance targets (Figure 14Figure 8).

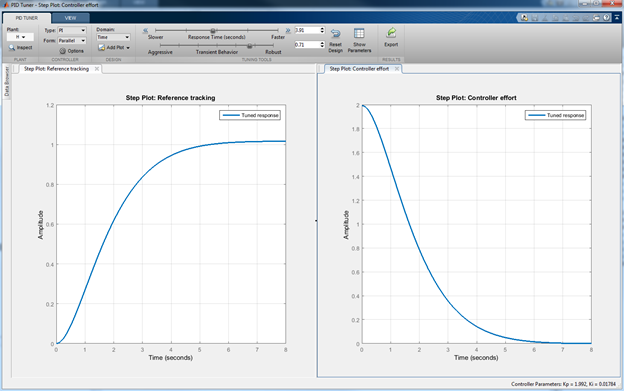


Figure - pidtool Step-Response and Controller Effort for position controller

## 4.2 Controller Implementation

The position controller was implemented in Simulink on the classroom CLIP connected to the RAPCON controller (Figure 15). The controller gain was adjusted by a factor of 1000 as the RAPCON position reading is in millimeters instead of meters.

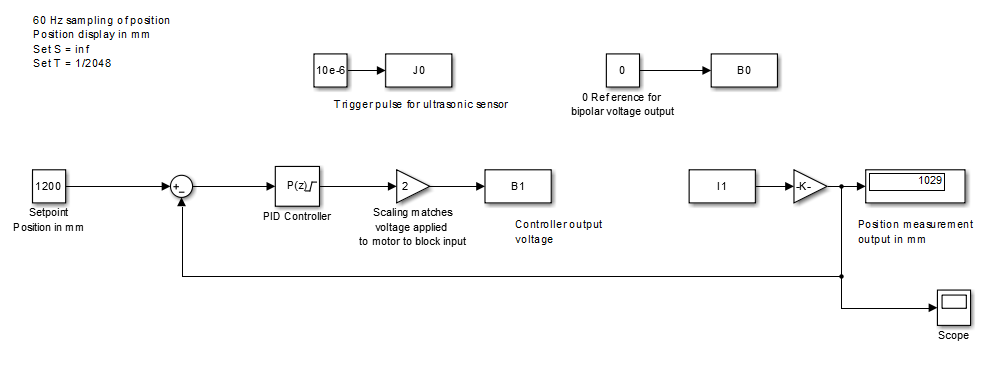


Figure - Position controller implemented in Simulink

## 4.3 Simulation vs Real-World Results

The response of the elevator to a 1 meter step input is shown in Figure 16. The 63% rise-time of the elevator is observed to be approximately 2.0 seconds (Tau = 2). The 10-90 rise-time of the system is calculated to be 4.4 seconds (2.2 \* Tau) which is slightly higher than the rise-time calculated by *pidtool*. The steady-state error of the system is also larger than predicted by *pidtool*. Both of these differences can be attributed to the fact that our model does not account for friction which is present in the physical system, as well as the quantization error of the position signal in the real-world.

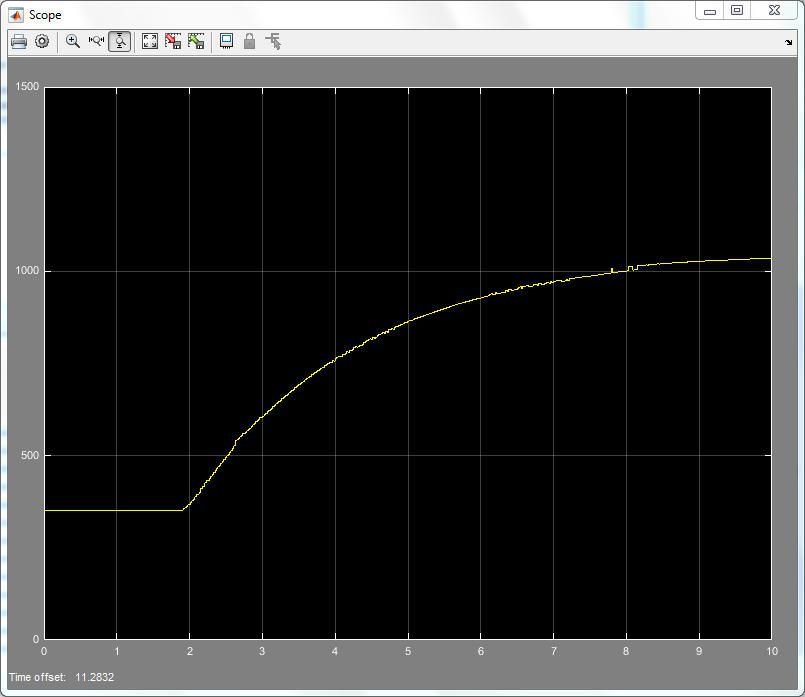


Figure - Position response of elevator - 1 meter step input