Project 2 Report – Elevator Control System

David Eelman & Thomas Abdallah

2016-12-09

Contents

[1 – Introduction 2](#_Toc469054297)

[2 – Motor Modelling & Simulation 2](#_Toc469054298)

[2.0 DC Motor Model 2](#_Toc469054299)

[2.1 Tachometer Model 3](#_Toc469054300)

[2.2 Servo-Amplifier Model 3](#_Toc469054301)

[2.3 Complete Motor Model 3](#_Toc469054302)

[2.4 Simulation vs Real-World Results 4](#_Toc469054303)

[3 – Motor Speed Control with Op-Amps 5](#_Toc469054304)

[3.0 Controller Targets 5](#_Toc469054305)

[3.1 Controller Development 6](#_Toc469054306)

[3.2 Op-Amp Controller Design & Implementation 6](#_Toc469054307)

[3.3 Simulation vs Real-World Results 7](#_Toc469054308)

[4 – Elevator System Modelling 9](#_Toc469054309)

[4.0 Modelling Elevator Dynamics 9](#_Toc469054310)

[4.1 Complete Elevator System Model 9](#_Toc469054311)

[4.2 Simulation Results 10](#_Toc469054312)

[5 – Elevator System Control 11](#_Toc469054313)

[5.0 Controller Targets 11](#_Toc469054314)

[5.1 Controller Development 11](#_Toc469054315)

[5.2 Controller Implementation 13](#_Toc469054316)

[5.3 Simulation vs Real-World Results 13](#_Toc469054317)

[6 – Elevator System Control 15](#_Toc469054318)

# 1 – Introduction

This report will discuss the Semester 5 elevator control system project including modelling & simulation of a DC motor and servo-amplifier, the development of an op-amp based speed control system, the modelling and simulation of the complete elevator system, and the development of a position controller for the elevator system.

# 2 – Motor Modelling & Simulation

## 2.0 DC Motor Model

Simulink was used to create a model for the DPP240-29V48 motor (Figure 1) based on the differential equation shown in (Figure 2). 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. Model parameters and units can be found in Table 1. Figure 3 shows the initial transfer function of the DC motor model.

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 1 - DC Motor Model

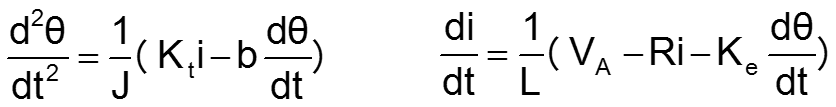


Figure 2 - Differential Equations for Mechanical & Electrical DC Motor Model

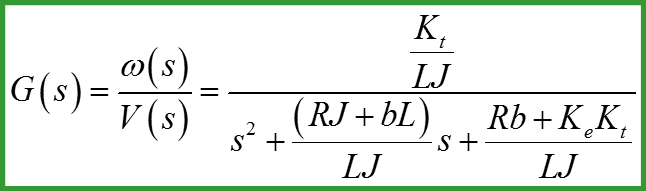


Figure 3 - DC Motor Model

## 2.1 Tachometer Model

The tachometer included with the DPP240-29V48 motor was modelled as shown in Figure 4. The model parameter for speed to voltage of the tachometer was pulled from the manufacturer datasheet.

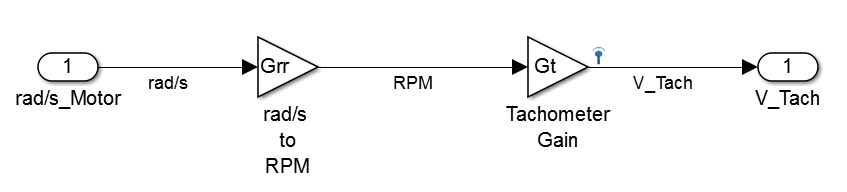


Figure 4 - Tachometer Model

## 2.2 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 5). Gain is configurable through a MATLAB variable.

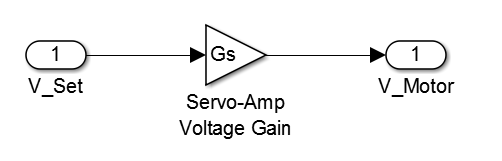


Figure 5 - Servo Amplifier Model

## 2.3 Complete Motor Model

The complete motor model is a cascaded system including the servo amplifier, DC motor and tachometer (Figure 6). 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 6 - 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 1 - Elevator System Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter (Units)** | **Symbol** | **Manufacturer Value** | **Measured Value** |
| Servo Amplifier Gain | Gs | User configurable | User configurable |
| Tachometer Gain (V/1000rpm) | Gt | 14 | NA |
| Rad/s to RPM (rpm/rad/s) | Grr | NA | 9.5494 |
| Motor Torque Constant (N\*m/A) | Kt | 0.1038 | 0.1 |
| Motor Voltage Constant (V/rad/s) | Ke | 0.104087 | 0.1 |
| Motor Inductance (mH) | L | 4.8 | 5.27 |
| Motor Resistance (Ohms) | R | 3.2 | 4 |
| Rotor Damping (N\*m\*s) | J | NA (estimate of 0.1) | 2.32x10-4 |
| Rotor Inertia (Kg\*m2) | b | 3.8136x10-5 | 2.5x10-5 |

## 2.4 Simulation vs Real-World Results

Figure 7 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 7 - 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 8 - 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 9 shows the real-world step-response together with the modelled step-response. Real-world measurements of the DC motor parameters were taken to improve the correlation of the model with the real-world response of the system. The real-world measurements are included in Table 1.

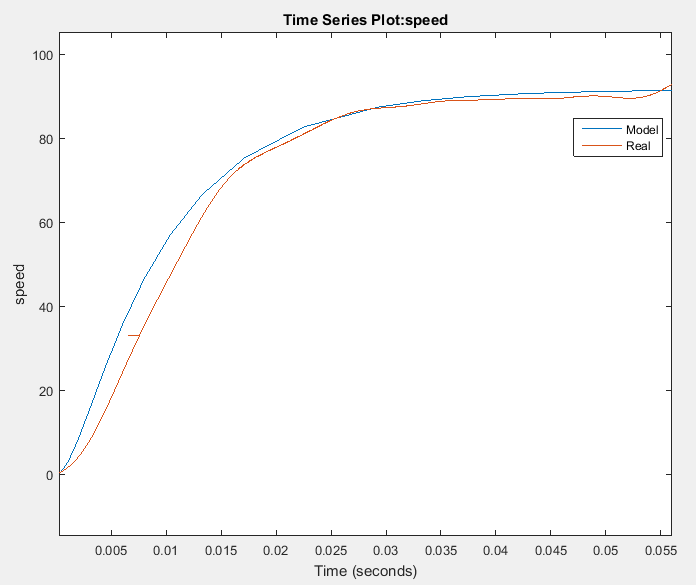


Figure 9 - DC Motor Step Response (Blue = Model, Red = Real-world)

# 3 – Motor Speed Control with Op-Amps

## 3.0 Controller Targets

A speed controller was developed with the following step-response design targets:

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

## 3.1 Controller Development

MATLABs *pidtool* function was used to tune a PI controller for the motor model shown in Figure 6. An integral only controller with a gain of 16.456 was found to meet our performance targets (Figure 10).

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 10 - pidtool Controller Design

## 3.2 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 11 shows the op-amp speed controller circuit as-built.

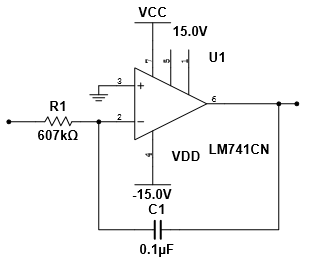


Figure 11 - PI Op-amp Controller

The schematic in Figure 12 shows the complete op-amp controller circuit fed by a differential amplifier. The controller output is fed to the servo-amp input which drives the DC motor. The tachometer output signal is fed back into the differential amplifier input.

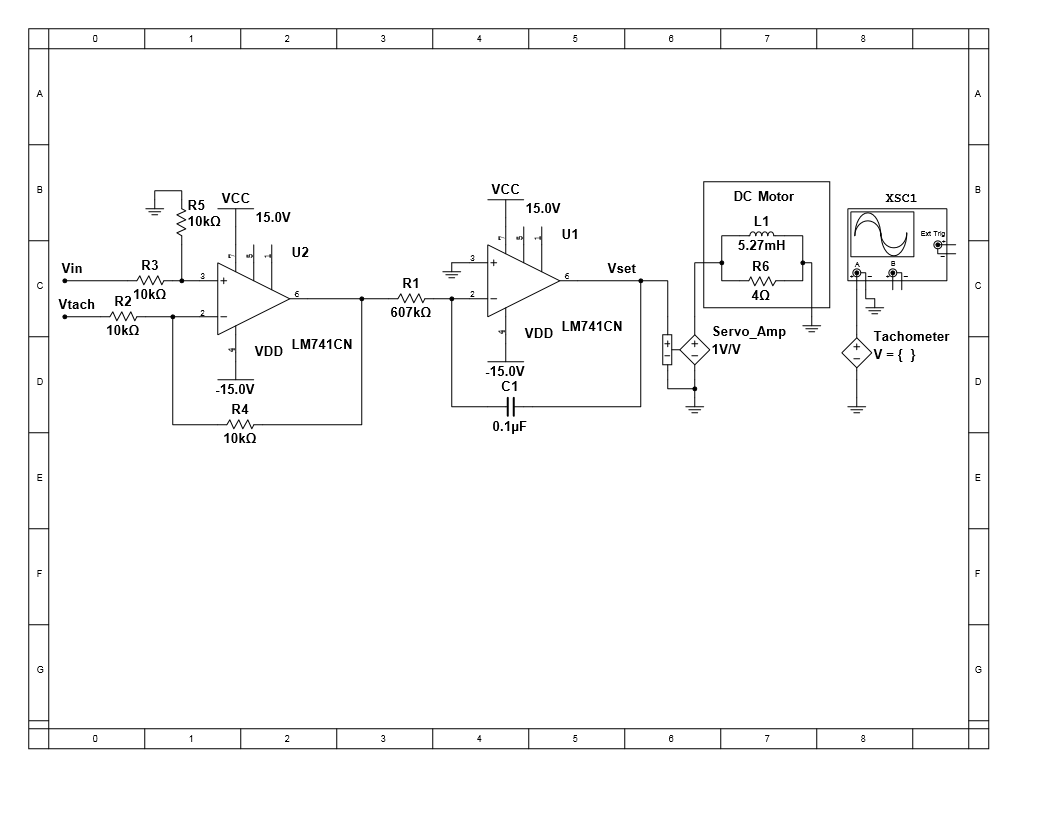


Figure 12 - Op-amp Controller Circuit

## 3.3 Simulation vs Real-World Results

The op-amp controller circuit was driven with a 0.5Hz 2.5V square wave input. Figure 13 shows the response of the system. As the input steps from 0 to 2.5V the system responds by increasing the controller output until error is reduced to zero. The steady-state error of the system is essentially zero.

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

Figure 14 shows the controller response to an external load. The controller increases the output voltage until error is reduced to zero.

Rôll 
5.00 v 
2.00 v 
400ms 
2.00 v 
23 Nov 2016 
10k ints 
1.80 v 
AFG Ram 
500.oomHZ 
4,0000 v 

Figure 14 - Controller Response to External Load

Figure 15 shows the response of the controller as it tracks a ramp input. The controller output and tachometer reading closely tracks the input ramp waveform.

2.S0kS/s 
10k ints 
AFG 
5.00 v 
5.00 v 
Ram 
400ms 
2.00 v 
500.00mHz 
4,0000 v 
23 Nov 2016 
1.80 V 

Figure 15 - Controller Tracking a Ramp Input

In conclusion our implementation of an op-amp based DC motor speed controller performs as expected.

# 4 – Elevator System Modelling

## 4.0 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

## 4.1 Complete Elevator 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 16). 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 16 - Elevator System Model

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

## 4.2 Simulation Results

The step-response of the elevator system model is shown in Figure 17 with position amplitude as the output on the y-axis. As expected, the position increases linearly after the initial transients are overcome.

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

Figure 17 - Step-Response of Elevator System Model

# 5 – Elevator System Control

## 5.0 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

## 5.1 Controller Development

MATLABs *pidtool* function was used to tune a PI controller for the elevator system model shown in Figure 16Figure 6. A proportional only controller with a gain of 1.992 was found to meet our performance targets (Figure 18). The compensator output is shown in Figure 19. Figure 20 shows the step response of the elevator system model with the compensator applied. The step response meets our design targets.

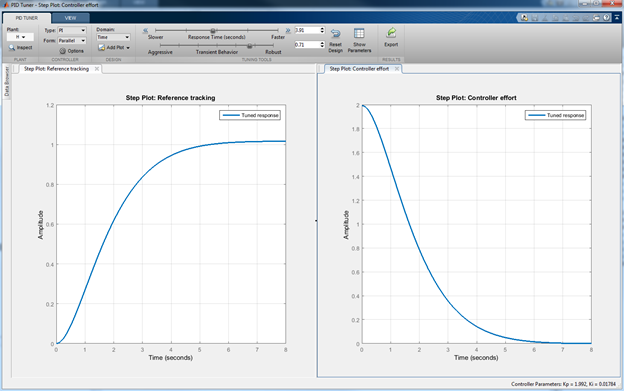


Figure 18 - pidtool Step-Response and Controller Effort for position controller

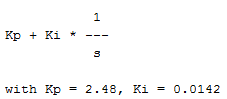


Figure 19 - Compensator Output

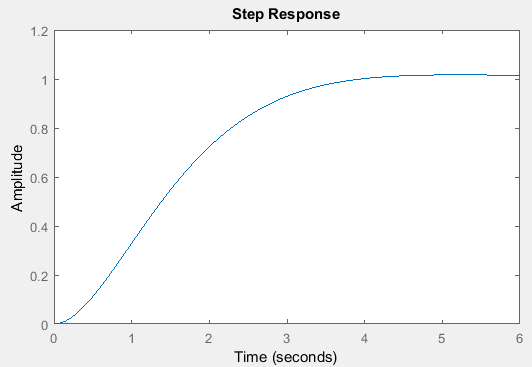


Figure 20 - Modelled Step Response with Compensator Applied

## 5.2 Controller Implementation

The position controller was implemented in Simulink on the classroom CLIP connected to the RAPCON controller (Figure 21). The controller gain was adjusted by a factor of 1000 as the RAPCON position reading is in millimeters instead of meters.

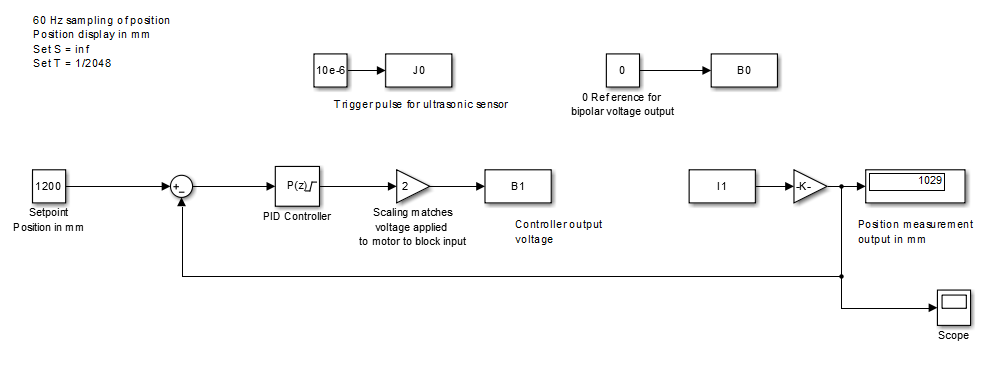


Figure 21 - Position controller implemented in Simulink

## 5.3 Simulation vs Real-World Results

The response of the elevator to a 1 meter step input is shown in Figure 22. 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 22 and Figure 23 show the step-response of the controlled elevator system.

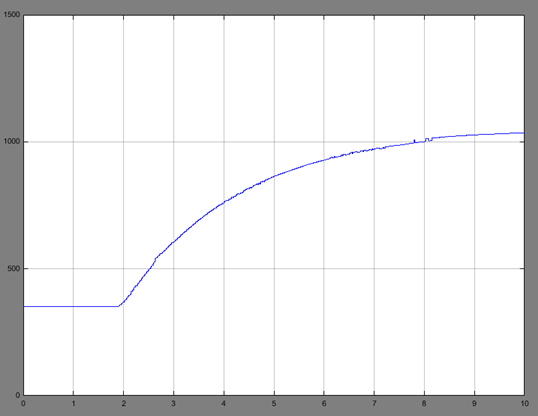


Figure 22 - Position response of elevator - 1 meter step input up

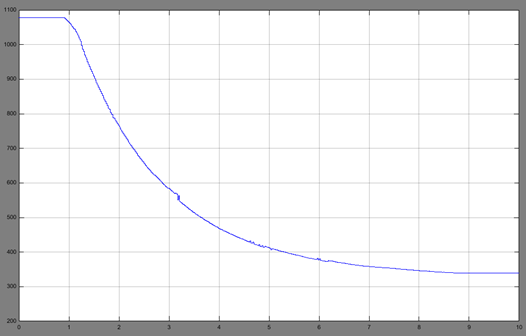


Figure 23 - Position response of elevator - 1 meter step input down

# 6 – Elevator System Control

In conclusion, using a model-based design approach resulted in the successful implementation of a speed controller for a DC motor and a position controller for an elevator system powered by the DC motor. The model-based design process gave us confidence in our controller design by allowing us to verify the controller performance through simulation before implementation in the real-world. The performance of our controllers in the real-world meets expectations given the limitations of our system model which does not include friction.