Finite Element Analysis and a Model-Free Control of Tension and Speed for a Flexible Conveyor System

A Thesis Presented
By

Zhan Zhang

To
Department of Mechanical and Industrial Engineering
In Partial Fulfillment for the Requirements
For the Degree of

Master of Science

In the field of

Mechanical Engineering

NORTHEASTERN UNIVERSITY
Boston, Massachusetts

Copyright May 2012 ©
NORTHEASTERN UNIVERSITY
Graduate School of Engineering
Thesis Title: **Finite element analysis and a model-free control of tension and speed for a flexible conveyor system**

Author: Zhan Zhang

Program: Mechanical Engineering

Approved for: Master of Science in Mechanical Engineering

Thesis Advisor: **Abe Zeid**

Thesis Advisor: **Rifat Sipahi**

Chairman of Department:

Graduate School Notified of Acceptance:

Associate Dean, Graduate and Research Programs:

Date

Copy deposited in library:

Reference Librarian
Abstract

In today's life, there is a wide variety of conveyor structures, from large heavy load conveyor in modern mining industry to the conveyor belt with small and simple structure in magnetic tape. Since its structure is simple, reliable and easy to deploy, it has become one of the basic mechanical drive methods. But almost every drive system suffers from fatigue failure, particularly the conveyor belt itself. So how to analyze the stress in conveyor structure has become a problem, which is worthy to study.

This thesis has two goals. First, we analyze the stresses in the conveyor belt system to determine the maximum stresses in the belt to ensure system safety during its operation. We build a 3D model of the system in Solidworks CAD system. Then we build a finite element model of the system using tetrahedral elements. We utilize nonlinear finite element analysis to calculate the stresses. The analysis considers the material nonlinearity due to the belt rubber material.

Second, we derive a control algorithm to allow us to control the belt stress and speed. We have used Simulink model of Matlab to build control a system. We utilize the Kelvin-Voigt model, the Maxwell model and the Dzierzek model in our simulation.

The results of both the computational finite element analysis and the control simulation compare well with closed-form solutions. The maximum stress from the finite element analysis is 16% off compared with the simple stress calculations. This is attributed to the fact that we modeled the entire conveyor belt system as assembly. The controller system results are within our expectations. The speed and stress in the belt meet our desired goal of keeping them constant via the control system.
Acknowledgements

First and foremost, I would like to express my most sincere gratitude to my advisors Dr. Abe Zeid and Dr. Rifat Sipahi for the guidance and patience. I thank them for teaching me all the skills necessary to complete this journey.

I would like to thank my parents for their endless support and patience. Without them I couldn’t have come this far. I thank them for always being there for me.
# Table of Contents

Acknowledgements  
List of Figureures  
List of tables  

Introduction  
  1.1 Overview  
  1.2 Introduction to Simulink  
  1.3 CAD system  
  1.4 Components of conveyor system  
  1.5 Purpose and scope of research and research objectives  

Chapter 2 Analysis of tape drive system  
  2.1 Overview of belt drive system  
  2.2 Belt drive analysis  
  2.3 Effective tension $F$  
  2.4 Slipping  
  2.5 Euler's formula  
  2.6 Stress analysis of belt drive  

Chapter 3 – 3D Modeling of tape drive system  
  3.1 Components of tape drive system  
  3.2 Assembly method  
  3.3 Computational stress analysis of belt system  
  3.4 Analysis of simulation results  

Chapter 4 – Control of tape drive system  
  4.1 Introduction to traditional PID control method  
  4.2 Introduction to ADRC  
  4.3 ADRC structure  
  4.4 Decoupled method for controlling the conveyor belt  
  4.5 Choices of modules  

Chapter 5 – Simulation of tape drive system  
  5.1 Material model of conveyor system  
  5.2 Building ADRC in Simulink  
  5.3 Simulation Results  
  5.4 Putting the whole system together  
  5.5 Analysis of result  

Reference
List of Figures

Figure 1.1 Simulink window
Figure 1.2 flat belt conveyor1
Figure 1.3 flat belt conveyor2
Figure 1.4 experiment model
Figure 2.1 conveyor structure
Figure 2.2 tension in the belt
Figure 2.3 tension in the belt work condition
Figure 2.4 centrifugal stress
Figure 2.5 bending stress
Figure 2.6 total stress
Figure 3.1 sketch of pulley
Figure 3.2 extrude pulley
Figure 3.3 chamfer
Figure 3.4 keyway
Figure 3.5 fillet1
Figure 3.6 fillet 2
Figure 3.7 sketch of hole
Figure 3.8 extrude cut hole
Figure 3.9 pattern the holes
Figure 3.10 pulley
Figure 3.11 sketch of base
Figure 3.12 extrude base
Figure 3.13 sketch of axis
Figure 3.14 extrude axis
Figure 3.15 components
Figure 3.16 pulley1 coincident mate1
Figure 3.17 pulley1 face coincident mate2
Figure 3.18 complete assembly
Figure 3.19 assembly features
Figure 3.20 create belt
Figure 3.21 conveyor assembly
Figure 3.22 feature selection
Figure 3.23 sketch selection
Figure 3.24 modifying status
Figure 3.25 extrude parameters
Figure 3.26 create belt
Figure 3.27 solid belt
Figure 3.28 open belt part
Figure 3.29 belt part
Figure 3.30 sketch on the belt
Figure 3.31 rectangle sketch
Figure 3.32 extrude cut
Figure 3.33 half belt
Figure 3.34 simulation study window
Figure 3.35 half belt assembly model
Figure 3.36 fixture on flat belt surface
Figure 3.37 fixture on circular arc belt surface
Figure 3.38 fixture on pulley
Figure 3.39 component contact
Figure 3.40 torque on belt contact surface
Figure 3.41 forces on the belt
Figure 3.42 torque on pulley
Figure 3.43 centrifugal force
Figure 3.44 mesh model
Figure 3.45 material window
Figure 3.46 simulation result
Figure 3.47 tight side stress
Figure 3.48 loose side stress
Figure 3.49 stress in the pulley
Figure 3.50 tight side stress
Figure 3.51 loose side stress
Figure 3.52 stress in the pulley
Figure 3.53 tight side stress
Figure 3.54 loose side stress
Figure 3.55 stress in the pulley
Figure 3.56 tight side stress
Figure 3.57 loose side stress
Figure 3.58 stress in the pulley
Figure 3.59 tight side stress
Figure 3.60 loose side stress
Figure 3.61 stress in the pulley
Figure 4.1 PID structure
Figure 4.2 global control
Figure 4.3 ADRC structure
Figure 4.4 poles’ location
Figure 5.1 Kelvin-Voigt model
Figure 5.2 Maxwell model
Figure 5.3 Dzierzek model
Figure 5.4 speed estimator structure
Figure 5.5 nonlinear combination
Figure 5.6 ADRC speed controller
Figure 5.7 tension estimator structure
Figure 5.8 nonlinear combination
Figure 5.9 ADRC tension controller
Figure 5.10 controller subsystems
Figure 5.11 tension test state-space window
Figure 5.12 tension controller test
Figure 5.13 tension controller test result
Figure 5.14 speed test state-space window
Figure 5.15 speed controller test
Figure 5.16 speed controller test result
Figure 5.17 Kelvin-Voigt model test
Figure 5.18 test result without filter
Figure 5.19 test result with filter
Figure 5.20 Maxwell model test
Figure 5.21 test result without noise
Figure 5.22 subsystem of Maxwell model
Figure 5.23 Maxwell model with noise
Figure 5.24 test result with noise
Figure 5.25 Dzierzek model
Figure 5.26 Dzierzek subsystem model
Figure 5.27 Dzierzek system with noise
Figure 5.28 Dzierzek system test result
Figure 5.29 Dzierzek system with filters
Figure 5.30 test result
Figure 5.31 Dzierzek with friction block
Figure 5.32 test result
Figure 5.33 Dzierzek with noise and friction blocks
Figure 5.34 test result
Figure 5.35 Maxwell model test result
Figure 5.36 Dzierzek model test result
Figure 5.37 Dzierzek model test result
Figure 5.38 Maxwell model test result
Figure 5.39 Maxwell model test result
Figure 5.40 Maxwell model test result
## List of tables

Table 3.1 tight side and loose side forces
Introduction

1.1 Overview

In today’s world, industry has become the foundation of society. It is important to keep industry improving. In the industry field, there are several kinds of Transmission mechanisms, such as gears, cam, worm gears and conveyors. Since the conveyor is suitable for the transmission over large distance between two shafts centers, and it has a good flexibility, when there is impact and vibrations on the conveyor, it could reduce the effect of the impact and vibration [1]. Also when it is over-loaded, the slipping effect will occur, and in this way, the other parts will be protected. Considering the prices of all transmission mechanisms, the conveyor is a cheap and reliable structure. With all above advantages, the conveyor has become a widely used transmission method in all fields of industry. So it is quite necessary for us to keep improving the methods of analyzing the force in the belt and the way to control the conveyor system.

Since there are varieties of conveyors in practice, the large conveyor could be used in mining industry, and the small one could be used in magnetic tapes. As a result we need a general model to analyze the system. Also we could use this general model to develop various complex models, which will be closer to real working conditions. This project
consists of two parts; one is using active disturbance rejection control method to control speed and tension of the belt. The other part uses Solidworks to build 3D model and apply finite element analysis on this model.

In this thesis, based on a general flat belt conveyor system model [2-4], in the control system design; the belt was treated as a thin belt, which means only the tension was considered. In the 3D finite element model, the belt could be analyzed with different materials and boundary conditions to see the changes of stress and strain, so that we could judge the best material and operating conditions for the conveyor and build a control system to control it.

1.2 Introduction to Simulink

Simulink is an important component of Matlab; it provides an environment of dynamics system modeling, simulation and a comprehensive analysis. It is a visual simulation tool in Matlab, and also a package based on Matlab block diagram design environment. Simulink is widely used in the modeling and simulation of linear and nonlinear systems, digital control and digital signal processing. In Simulink, the sample time could be defined as discrete sampling time or continuous sampling time. It also supports multi-rate system, where different parts have different sampling time. To build a dynamic system, Simulink offers
a graphical user interface (GUI) to model the block diagram; the process could be done by simply clicking and dragging the mouse. It provides a fast, straightforward manner, and the user can immediately see the simulation results of the system.

The Simulink has following characteristics: rich expansion predefined module library; interactive graphics editor to combine and intuitive manage module diagram.

There are two ways to run the Simulink:

1. Type Simulink in the MATLAB command window

There will be a window called Simulink Library Browser (Figure 1.1); all the blocks are listed in this window.
2. Click the quick link button on MATLAB main window

There are 8 Sub-libraries, classified by their functions, in Simulink: Continuous, Discrete, Function & Tables, Math, Non-linear, Signals & Systems, Sinks, Sources. We could choose necessary blocks to build the desired system.

1.3 Solidworks CAD system

Solidworks is mechanical design software which contains varieties of components and functions. It is also the first three-dimensional CAD software which is developed based on Windows system. For the users who are familiar with Microsoft’s Windows system, basically they can use Solidworks to begin designing jobs.

Here are some key advantages of Solidworks:

- Solidworks provides a complete set of dynamic interface, and drag-mouse control method. The “full dynamic” user interface reduces design steps, and improves the designer experience.

- The intelligent assembly technology [5] which is called capturing mate is used in Solidworks to speed up the general assembling process. The intelligent assembly technology means that Solidworks could automatically capture and define assembly relationships.

- With Solidworks, we could view all the dynamic movements of moving parts in the assembly. We could also make dynamic interference checking
and gap detection.

With all the advantages, Solidworks is the best option for us to perform analysis of the conveyor system.

1.4 Components of conveyor system

The conveyor system is always divided into two parts: pulleys connected by traction pieces. This kind of conveys always include traction pieces, carrier components, drive units, tensioning device, redirecting means, support member, etc.

Traction pieces can be used to transfer traction, they can be conveyor belt, traction chain or wire rope. Carrier components are used to hold the materials we want to transfer; they could be hopper, bracket or spreader. The drive unit is used to transmit power; it is usually consisted of electric motor, reducer and brake. The tensioning device is generally divided into two kinds, one is screw type and the other is heavy hammer type; it is used to maintain a certain tension in the belt. So the conveyor could transmit power at maximum efficiency. The support member is used for supporting traction members; it can be roller or pulley.

The structure characters of a conveyor device having a traction member is that, the material need to be transported is mounted on the supporting members which are linked with the traction member, or directly mounted on the traction member (such as conveyors). Traction members bypass
connect to both of the respective rollers or sprockets, including the formation of transportation of materials with a closed loop of the load branch and no-load transportation of materials branch. The structures are shown in Figure 1.2 and Figure 1.3

![Figure 1.2 flat belt conveyor1](image1)

![Figure 1.3 flat belt conveyor2](image2)

The conveyor system is too complex to study, so we ignore some parts in our study; we only focus on the belt and the pulleys, shown in Figure 1.4.
We could make the model more complex according to different working conditions and different types of conveyors by changing the dimensions of pulleys and motors. Also we could change the math model of conveyor to make it become more realistic.

1.5 Purpose and scope of research and research objectives

Our purpose is to design a control system with the help of Simulink to control the speed and tension in the conveyor belt. Another goal is to build 3D model in Solidworks to apply finite element analysis on the model to check the stress and strain in the belt and make comparison with different loading conditions.
Chapter 2 Analysis of tape drive system

2.1 Overview of belt drive system

Belt drive system consists of two pulleys and a belt tightly wrapped around them. Relying on the friction between the contact surfaces of the belt and pulley to transmit motion and power, it is a flexible frictional transmission as shown in Figure 2.1.

![Figure 2.1 conveyor structure](image1)

2.2 Belt drive analysis

Figure 2.2 shows the belt forces on both sides of the belt in static conditions, i.e. the belt is not working, when the belt moves, it has a tight side and a loose side. The tight side tension $F_1$ increasing from $F_0$ to $F_1$; The loose side tension is $F_2$ decreasing from $F_0$ to $F_2$

![Figure 2.2 tension in the belt](image2)
Assuming the total length of the belt will not be changed, the tight side tension incremental $F_1 - F_0$ should be equal to the amount of decrease of the slack side tension $F_0 - F_2$.

### 2.3 Effective tension $F$

Effective tension $F$ is the effective circumferential force the belt drive can transfer effectively. It is not the concentrated force acting in a fixed point, but the sum of the friction generated by the contact surface of the belt and pulley, i.e.

$$F_0 = \frac{1}{2}(F_1 + F_2)$$

$$F = F_1 - F_2$$

$$F_1 = F + \frac{F}{2}, \quad F_2 = F - \frac{F}{2}$$

The relationship between the circumferential force $F$ (N), the belt speed $v$ (m/s) and the transmission power $P$ (KW):

$$P = \frac{Fv}{1000}$$

### 2.4 Slipping

When the desired delivery circumferential force exceeds the friction force between the belt and the tread, sliding between the belt and the pulley occurs, a phenomenon called slip. The wear of the belt will often aggravate slipping and reduce the transmission efficiency, resulting in
drive failure.

2.5 Euler's formula

The relationship between the tension forces $F_1$ and $F_2$ on both sides of the belt are given by the following Euler’s formula:

$$F_1 = F_2 e^{f\alpha}$$  \hspace{1cm} (2.3)

Where: 
- $f$ is the coefficient of friction between the belt and the wheel surface; 
- $\alpha$ is the Wrap angle (rad); 
- $e$ is the natural logarithms ($e \approx 2.718$).

The above equation is the basic flexible friction formula, known as Euler’s formula. The belt forces are shown in Figure 2.3.

![Figure 2.3 tension in the belt work condition](image)

2.6 Stress analysis of belt drive

We use flat belts as an example. When the belt is moving, the stress in the belt is composed of the following three parts:

(1) Tight side and loose side tensions generate tensile stresses

Tight side tensile stress:

$$\sigma_1 = \frac{F_1}{A}$$ \hspace{1cm} (2.4)
Loose side tensile stress:

\[ \sigma_2 = \frac{F_2}{A} \]  \hspace{1cm} (2.5)

A is the cross-sectional area of the belt (mm\(^2\)).

(2) Tensile stress caused by Centrifugal force

Centrifugal tensile stress is given by Equation (2.6), and shown in figure 2.4.

\[ \sigma_c = \frac{F_c}{A} = \frac{qv^2}{A} \]  \hspace{1cm} (2.6)

Figure 2.4 centrifugal stress

(2) Bending stress, shown in figure 2.5. The stress is given by Equation (2.7)

\[ \sigma_b = \frac{2yE}{D} \]  \hspace{1cm} (2.7)
Figure 2.5 bending stress

Where: \( y \) is the vertical distance (mm) of neutral layer of the tape to the outermost layer; \( E \) is the modulus of elasticity of the belt material (MPa); \( d \) is the pulley diameter (for the V-belt pulley, \( d \) is the standard diameter.)

Figure 2.6 total stress

Based on the above analysis, while belt is transferring forces, there are three kinds of stresses in the belt: tensile stress, Centrifugal tensile stress, and bending stress. As shown in figure 2.6, the maximum stress occurs around the tight side at the small pulley. The maximum stress value is:
\[ t = \sigma_1 + \sigma_b + \sigma_c \] (2.8)

So while the belt drive system is running, the maximum location stress varies. This means the belt works under varying stress. When the number of stress cycles reaches a certain value, the fatigue failure will occur.
Chapter 3 – 3D Modeling of tape drive system

3.1 Components of tape drive system

3.1.1 Design of pulley

Since the model is based on the flat band, to make the study more realistic, the 3D model is drawn according to the design process of flat band pulley.

Step 1, according to relative calculation and background; draw the sketch, shown in Figure 3.1.

Figure 3.1 sketch of pulley
Step 2, revolved boss to 360 degree

Step 3, chamfer the angle as 45 degree on the selected sides.
Step 4, make extrude cut to create the keyway

Figure 3.4 keyway

Step 5, fillet the selected sides

Figure 3.5 fillet1
Figure 3.6 fillet 2

Step 6, draw the sketch on the face of pulley

Figure 3.7 sketch of hole
Step 7, extrude cut, through all bodies

Figure 3.8 extrude cut hole

Step 8, circular pattern as shown below

Figure 3.9 pattern the holes
Step 9, review the pulley 3D CAD model

Figure 3.10 pulley

3.1.2 Base design

The function of base is to make sure the pulleys move in the same way as they did under real operating conditions. So the structure is simple. And when we create simulation, we could exclude the base from the study.

Step 1, Draw the base sketch

Figure 3.11 sketch of base
Step 2, Extrude the sketch

Figure 3.12 extrude base

Step 3, Draw the circle sketch on the base face

Figure 3.13 sketch of axis

Step 4, Extrude the sketch a distance of 15mm

Figure 3.14 extrude axis
3.2 Assembly method

(1) Insert 2 pulleys and the base in the assembly model

Figure 3.15 assembly components

(2) Mate the two selected faces as coincident

Figure 3.16 pulley1 coincident mate1
(3) Make the two faces coincident

![Figure 3.17 pulley1 face coincident mate2](image)

(4) Repeat the same step for the other pulley

![Figure 3.18 complete assembly](image)

(5) Add relationship between two pulleys. Since it is belt drive system, we need to choose Belt/Chain option
(6) Choose parameters of belt, such as the face the belt is mounted on

(7) Add a thickness of 1mm to the belt
In this assemble method; the pulleys could be organized as the work condition. And the assembly feature is set as belt, so when one pulley rotates, the other pulley will follow its rotation.

Since there is no real belt in the assemble drawing, we need to create the belt:

(1) Click the Belt2, select belt2-2^Assem1 option. And choose edit part option

![Feature selection](image_url)
(2) Select Sketch2

Figure 3.23 sketch selection

(3) Extrude the sketch. The parameters are set as shown

Figure 3.24 modifying status

Figure 3.25 extrude parameters
(4) Exit the edit model, and belt is created as shown.

We could use this model in the motion study. But if we want to apply FEM, we need to create half belt.

(1) select the belt2, click open part option
(2) Sketch rectangle on the selected face to cut the belt by half to enable us to perform stress analysis using the finite element method.
(3) Draw rectangle

Figure 3.31 rectangle sketch

(4) Make extrude cut as shown

Figure 3.32 extrude cut

(5) The half belt is shown, save as half-belt.

Figure 3.33 half belt
3.3 Computational stress analysis of belt system

Since the two pulleys are the same size, as created in Chapter 2, we only need to focus on half of the model (figures 3.34 and 3.35). So we could exclude the base and the other pulley in the simulation study.

In order to simulate the operating condition, it is necessary to define the fixture [6]. Since it is a static study, there should be tension in the belt, torque and centrifugal force on the contact surface between the pulley and
the belt. Since the pulley could rotate around the axis, and the belt will be tight, so the fixture could be defined as:

（1）The bottom side of the belt is set as roller/slider fixture.

(2) The contact surface between the pulley and the belt is set as fixed geometry.

(3) Since the pulley could rotate around the axis, the pulley is set as fixed on the cylindrical faces, and circumferential sub option, but the study is
static, so the rotation angle is set as zero.

Figure 3.38 fixture on pulley

(4) We also need to define the component contact to make sure there would be contact fore between the pulley and the belt.

Figure 3.39 component contact

Next step, we need to apply external loads on the model.

(1) Apply 2 NM torque on the contact surface
(2) Apply two forces on the cross section on the belt. The value of forces are calculated from Chapter 2

(3) Apply a torque on the surface of pulley, but the direction is opposite with the torque on the contact surface.
(4) Apply centrifugal force on the inter surface. Suppose the angular velocity is 2 rad/s.

(5) Create the finite element mesh. The mesh method is selected as Jacobian 4 points model.
The last step is to define different materials of different parts in the model.

(1) Define the belt as rubber, and define the pulley as alloy steel. We could choose the materials from Solidworks material list.
(2) Run the simulation

![Simulation Result](image)

Figure 3.46 simulation result

### 3.4 Analysis of simulation results

We use the equations shown in Chapter 2. Suppose the tight side tension is 2N, we could calculate the loose side force:

$$F_1 = F_2 e^{f\alpha}$$  

(3.1)

In this model the wrap angle is 180°, which is described as π in rad. And according to related information, the friction coefficient $f$ is 1. So here is the calculation result:

<table>
<thead>
<tr>
<th>F1(N)</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2(N)</td>
<td>0.08</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
</tbody>
</table>

It is obvious that the loose side tension is very small compared with tight side tension. Since the pulleys are rotating, there should be centrifugal force in the belt. The friction force appears as torque on the contact point.
surface.

(1) Condition: $F_1 = 2$ N, $F_2 = 0.08$N. Angular speed: 2 rad/s, Torque1=2 N·M. Torque2= - 2 N·M, belt material: rubber, pulley material: alloy steel. Figure 3.47-3.49 show the Von Mises stress result.

Figure 3.47 tight side stress

Figure 3.48 loose side stress
According to the Figures, the belt max stress occurs on the the tight side of the belt where the belt wraps around the pulley.

According to Equation (2.6) in Chapter 2, the max stress is:

\[ \sigma_{\text{max}} = \sigma_1 + \sigma_b + \sigma_c . \]

So we could get the value of max stress that:

\[
\begin{align*}
\sigma_1 &= \frac{F}{A} = \frac{2}{8} = 0.25 \text{Mpa} \\
\sigma_b &= \frac{2Ey_0}{d} = \frac{2 \times 0.5 \times E}{10} = 0.61 \text{Mpa} \\
\sigma_c &= \frac{qv^2}{A} = \frac{4 \times 0.01}{8} = 0.004 \text{Mpa} \\
\sigma_{\text{max}} &= 0.864 \text{Mpa}
\end{align*}
\]

The max stress is about 1Mpa, according to Figure. 3.47. Comparing the manual calculation result and the simulation result, we get the error:

\[
\text{error} = \frac{\sigma_{\text{max}}^{\text{fem}} - \sigma_{\text{max}}^{\text{manual}}}{\sigma_{\text{max}}^{\text{max}}} \times 100\% = \frac{0.864 - 0.411}{0.864} \times 100\% = 52.4\%
\]
(2) Condition: \( F_1 = 5 \text{ N}, F_2 = 0.2 \text{ N}, \) angular speed: \( 2 \text{ rad/s}, \) torque1 = 5 \( \text{N} \cdot \text{M}, \) torque2= -5 \( \text{N} \cdot \text{M}, \) belt material: rubber, pulley material: alloy steel. The stress result are shown in figure (3.50-3.52).

Figure 3.50 tight side stress

Figure 3.51 loose side stress
The max stress still obeys the rule, and the value increases significantly. According to max stress Equation (2.6) in Chapter 2, we get the following result:

\[
\sigma_1 = \frac{F}{A} = \frac{5}{8} = 1.25\text{Mpa}
\]

\[
\sigma_b = \frac{2E_{y0}}{d} = \frac{2 \times 0.5 \times E}{10} = 0.61\text{Mpa}
\]

\[
\sigma_c = \frac{qv^2}{A} = \frac{4 \times 0.01}{8} = 0.004\text{Mpa}
\]

\[
\sigma_{\text{max}} = 1.239\text{Mpa}
\]

The max stress is about 1Mpa, according to Figure. 3.50. Comparing the manual calculation result and the simulation result, we could get the error:

\[
\text{error} = \frac{\sigma_{\text{max}} - \sigma_{\text{max}}}{\sigma_{\text{max}}} \times 100\% = \frac{1.239 - 1.05}{1.239} \times 100\% = 15.3\%
\]

Figure 3.52 stress in the pulley
(3) Condition: \( F_1 = 10 \text{ N}, \ F_2 = 0.4 \text{N}, \) angular speed: 2 rad/s, torque1 = 10 N\(\cdot\)M, torque2 = -10 N\(\cdot\)M, belt material: rubber, pulley material: alloy steel. Stress results are shown in figure (3.53-3.55).

Figure 3.53 tight side stress

Figure 3.54 loose side stress
The max stress still obeys the rule, and the value increases significantly. According to max stress Equation (2.6) in Chapter 2, we get the following result:

\[
\sigma_1 = \frac{F}{A} = \frac{10}{8} = 1.25\text{Mpa}
\]
\[
\sigma_b = \frac{2Ey_0}{d} = \frac{2 \times 0.5 \times E}{10} = 0.61\text{Mpa}
\]
\[
\sigma_c = \frac{qv^2}{A} = \frac{4 \times 0.01}{8} = 0.004\text{Mpa}
\]
\[
\sigma_{\text{max}} = 1.864\text{Mpa}
\]

According to Figure 3.53, the max stress in the belt is about 2Mpa, so we get the error:

\[
\text{error} = \frac{\sigma_{\text{max}}^{\text{fem}} - \sigma_{\text{max}}^{\text{manual}}}{\sigma_{\text{max}}} \times 100\% = \frac{2 - 1.864}{1.864} \times 100\% = 7.3\%
\]
(4) Condition: F1=15 N, F2=0.6N, angular speed: 2 rad/s, torque1=15 N·M, torque2= - 15 N·M, belt material: rubber. Pulley material: alloy steel. The stress results are shown in figure(3.56-3.58).
According to max stress Equation (2.6) in Chapter 2, we get the following result:

\[
\sigma_1 = \frac{F}{A} = \frac{15}{8} = 1.875 \text{MPa}
\]

\[
\sigma_b = \frac{2Ey_0}{d} = \frac{2 \times 0.5 \times E}{10} = 0.61 \text{MPa}
\]

\[
\sigma_c = \frac{qv^2}{A} = \frac{4 \times 0.01}{8} = 0.004 \text{MPa}
\]

\[
\sigma_{\text{max}} = 2.489 \text{MPa}
\]

According to the simulation result, the max stress in the belt is about 2.9 MPa, so there will be an error:

\[
\text{error} = \frac{\sigma_{\text{max}} - \sigma'_{\text{max}}}{\sigma_{\text{max}}} \times 100\% = \frac{2.9 - 2.489}{2.489} \times 100\% = 16.5\%
\]

(5) Condition: F1=20 N, F2=0.8N, angular speed: 2 rad/s, torque1=20 N·M, torque2=-20 N·M, belt material: rubber. Pulley material: alloy steel. The stress results are shown in figure(3.59-3.61).
According to max stress Equation (2.6) in Chapter 2, we get the following result:
\[ \sigma_{i} = \frac{F}{A} = \frac{20}{8} = 2.5\text{Mpa} \]

\[ \sigma_{b} = \frac{2Ey_{0}}{d} = \frac{2 \times 0.5 \times E}{10} = 0.61\text{Mpa} \]  \hspace{1cm} (3.6)

\[ \sigma_{c} = \frac{qv^{2}}{A} = \frac{4 \times 0.01}{8} = 0.004\text{Mpa} \]

\[ \sigma_{\max} = 3.1\text{Mpa} \]

In the simulation result, the max stress in the belt is around 3.3Mpa, so we could get the error:

\[
\text{error} = \frac{\sigma_{\text{max}}^{\text{fem}} - \sigma_{\text{max}}^{\text{manual}}}{\sigma_{\text{max}}} \times 100\% = \frac{3.8 - 3.1}{3.1} \times 100\% = 20\%
\]
Chapter 4 – Control of tape drive system

4.1 Introduction to traditional PID control method

Since PID theory is developed long time ago, it is limited by the technology level at that time [7-9]; it could not be applied with advanced digital signal processing technology. But the PID controller is able to handle most problems, and as a main theory in the engineering control field.

The basic structure of PID [10] is shown in Figure 4.1.

![Figure 4.1 PID structure](image)

There are several incompletions [11-13] of PID control:

(1) The error is generated by \( e = v - y \) directly [10]. Our control goal is that \( v \) could be drastically changed, but the output of object \( y \) has inertia, so \( y \) could not be drastically changed, but slowly. In this way, applying the slowly changed variable \( y \) to track drastically changed variable \( v \) is not rational.

(2) There is no good way to generate differential signal of error \( \frac{de}{dt} \).
since differentiation could be obtained only with approximation
\[ y = \frac{v(t) - v(t - \tau)}{\tau} \] where tau can be seen as the sampling period.

Differentiation could also cause problems with noisy signals. When the input signal \( v(t) \) is affected by noise, the approximated differential \( y = \frac{v(t) - v(t - \tau)}{\tau} \), in the output \( y \) will be affected by the noise component as \( \frac{n(t)}{\tau} \), which amplifies as \( \tau \) goes to a small number. To prevent this, a low pass filter could be used, rendering the derivative operation transfer function as
\[ y = \frac{s}{\tau s + 1} v_o, \] where \( \tau > 0 \) is the time constant of the filter producing a cut-off frequency at \( \frac{1}{\tau} \) in frequency domain. Although the output \( y \) can be made less affected from noise, the value of \( \frac{1}{\tau} \) determines the bandwidth, and some sacrifice is needed in system’s bandwidth to be able to obtain a differential of a noisy signal with minimal effects due to noise.

(3) There are several disadvantages to applying the feedback of integration of error signal in the form of \( \int_0^x e(\tau) d\tau \). Feedback law utilizing the signal feedback make is in general known to make the closed loop response slower and prone to oscillations due to induced low damping ratio.
4.2 Introduction to ADRC

Active Disturbance Rejection Control (ADRC) [14] is a new method to build a control system around a plant, model of which is unknown. This method is also based on the error feedback, which can be seamed together with a PID controller.

In general, control systems have a structure as shown in Fig 4.2, where \( v \) is the reference, and the controller uses both the reference and the output of the plant, to produce the control action \( u \) to be applied on the plant. Importantly, here the mathematical model of the plant is known, so that a proper controller can be designed for it.

![Figure 4.2 global control](image)

In this traditional approach, the control parameters are selected based on the open loop differential equation, e.g. given by \( \dot{x} = f(x, \frac{dx}{dt}) \), where the function \( f \) is known, and the state variables are \( x \) and \( \frac{dx}{dt} \).
So if we want to control the system described above, we need get the information about the open loop dynamics \( f(x, \frac{dx}{dt}) \) and the state variables \( x, \frac{dx}{dt} \). But actually this could not be done, for the reason that in real world, there is seldom information about the open loop dynamics, and in some cases the open loop is too complex that its modeling is prohibitive. Therefore, in such situations, the classical control design approached cannot be directly applied to control the open loop system.

In fact, to achieve the control goal does not mean we need to know the open loop dynamics of the system, as argued by Han [14] in his work on ADRC. For example, consider the second order dynamics given by:

\[
\begin{align*}
\dot{x} &= f(x, \frac{dx}{dt}) + u \\
y &= x
\end{align*}
\] (4.1)

Where \( u \) is the input, function \( f(x, \frac{dx}{dt}) \) is unknown, state variables are \( x, \frac{dx}{dt} \), and \( y \) is the measurement.

The main idea behind model-free control for achieving the control goal is to reduce the error between the desired value \( v(t) \) and the output of the system \( y \), by regulating the input \( u(t) \) into the system. So what we need to know is the specific value of the open loop dynamics in the control process, but not necessarily the explicit definition of the function \( f \). We suppose this value is

\[
a(t) = f(x(t), \frac{dx}{dt})
\] (4.2)
According to Equation (4.1), \( a(t) \) could be expressed as

\[
a(t) = \dot{x} - u
\]

(4.3)

So if we know input \( u \) and output \( y = x \), we could calculate \( a(t) \) from Equation (4.3) by carefully differentiating \( x \) twice. In this way, once \( a(t) \) is at hand, the control signal \( u(t) \) can be adjusted as:

\[
u(t) = -a(t) + u_0
\]

(4.4)

After this basic transformation, the differential equation of the closed loop system becomes a linear process:

\[
\frac{d^2 y}{dt^2} = u_0(t)
\]

(4.5)

Then the control signal \( u_0(t) \) can be designed using a PID-like block on the error \( e \) and differential error \( \dot{e} = v - y \) as:

\[
u_0(t) = -a_1 e - a_0 \frac{de}{dt} + \frac{d^2 e}{dt^2}
\]

(4.6)

Finally, the closed loop equation becomes:

\[
\frac{d^2 e}{dt^2} = -a_1 e - a_0 \frac{de}{dt}
\]

(4.7)

Obviously, this differential equation is stable as long as \( a_0 \) and \( a_1 \) are positive quantities, \( e(t) \Rightarrow 0 \), so we could achieve the control goal \( y(t) \Rightarrow v(t) \).

This control method approach that underlies the main idea behind model-free control is different from the classical control design approach.
as it does not rely on mathematical model of the open loop system. It is mainly based on signal processing of input-output signals, and estimating the “dynamic load” \( a(t) \), to produce control actions using \( a(t) \) to be able to attain the control objective. It is important to note here that the estimation of \( a(t) \) should be instantaneous, and preferably noise free, and the open loop system, either linear or nonlinear, should have some smoothness properties [10].

4.3 ADRC structure

Here \( y \) which is measurable is the output to be controlled, \( u \) is the input, and open loop nonlinear system is described by an unknown function \( f(x_1, x_2, w(t), t) \), which is in general a function of states, external disturbances and time.

Suppose the system is given by:

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 \\
y &= x_1 \\
\frac{dx_2}{dt} &= f(x_1, x_2, w(t), t) + bu
\end{align*}
\]

(4.8)

According to ADRC theory, we do not need to know \( f(x_1, x_2, w(t), t) \) specifically, and the open loop system can still be controlled as we summarize next. First sets \( f(x_1, x_2, w(t), t) \) as \( x_3 \), and \( \frac{df(x_1, x_2, w(t), t)}{dt} = G(t) \), so the whole system could be described as:
\[ \frac{dx_1}{dt} = x_2 \]
\[ \frac{dx_2}{dt} = x_3 + bu \]
\[ \frac{dx_3}{dt} = G(t) \]
\[ y = x_1 \]

Then constructs a state observer, known as the extended state observer (ESO).

\[ e = z_1 - y \]
\[ \frac{dz_1}{dt} = z_2 - \beta_{01} e \]
\[ \frac{dz_2}{dt} = z_3 + bu - \beta_{02} e \]
\[ \frac{dz_3}{dt} = \beta_{03} e \]

To predict the numerical value of \( x_3 = F(t) = f(x_1, x_2, w(t), t) \). In this ESO, the input is the system output \( y \) and control signal \( u \).

So the system equations become:

\[ e = z_1 - y \]
\[ \frac{dx_1}{dt} = x_2 \]
\[ \frac{dx_2}{dt} = u_0 \]
\[ y = x_1 \]

Figure 4.3 shows the ADRC architecture.
Figure 4.3 ADRC structure

Since the control goal is for instance to maintain a constant force in the tension of the conveyor belt, and a constant speed of conveyor belt, we could let the two control signals $V_1$ and $V_2$ as $\frac{dv_1}{dt} = v_2$. See Fig 4.3.

### 4.4 Decoupled method for controlling the conveyor belt

From figure 1.4, the equations of motion of the conveyor belt system can be developed as:

\[
J \frac{d^2\theta_1}{dt^2} = -T_e r + K_m i_1 \tag{4.12}
\]

\[
J \frac{d^2\theta_2}{dt^2} = -T_e r + K_m i_2 \tag{4.13}
\]

\[
T_e = k(x_2 - x_1) + b\left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right) \tag{4.14}
\]

\[
x_3 = \frac{(x_1 + x_2)}{2} \tag{4.15}
\]

What we need to control is the tension $T_e$ and the velocity $\frac{dx_3}{dt}$, by applying control signals to the two motors. Assuming the above model is actually not available, we will apply the ADRC scheme to the control.
problem described above. For this, we will demonstrate how ADRC applies to the system in the decoupled form. We know that the plant at hand could be uncoupled and treated as two independent second-order system. One is for \(X_3\) and the other one is for \(T_e\).

To accomplish the uncoupling, note equal commands on the two current inputs. One will cause the tension, but no change in \(X_3\), the other will cause a change in \(X_3\), but no effect on tension \(T_e\). So suppose:

\[
U_d = \begin{bmatrix} u_3 \\ u_r \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}
\]  
(4.16)

We could get the transform matrix:

\[
T_u = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}
\]  
(4.17)

We also need to define a new state. What we need is \(X_3\) and tension \(T_e\), so the new state should include \(X_3\), tension \(T_e\) and its derivatives, i.e.

\[
X_d = \begin{bmatrix} X_3 \\ \frac{dX_3}{dt} \\ T_e \\ \frac{dT_e}{dt} \end{bmatrix}
\]  
(4.18)

We already know the system matrix:

\[
X_d = \begin{bmatrix} 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 10 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \omega_1 \\ \omega_2 \end{bmatrix}
\]
\[
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
8.533 & 0 \\
0 & 8.533
\end{bmatrix} + \begin{bmatrix}
i_1 \\
i_2
\end{bmatrix}
\tag{4.19}
\]

And the state matrix before the decoupling modification is given by:

\[
x = \begin{bmatrix}
x_3 \\
T_e
\end{bmatrix} = \begin{bmatrix}
0.5 & 0.5 & 0 & 0 \\
-2.113 & 2.113 & 0.375 & 0.375
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
\omega_1 \\
\omega_2
\end{bmatrix}
\tag{4.20}
\]

In which, we define two transformation matrices:

\[
[H_s] = \begin{bmatrix}
0.5 & 0.5 & 0 & 0
\end{bmatrix}
\tag{4.21}
\]

\[
[H_t] = \begin{bmatrix}
-2.113 & 2.113 & 0.375 & 0.375
\end{bmatrix}
\tag{4.22}
\]

So the new state could be expressed as:

\[
x_d = \begin{bmatrix}
H_3 \\
H_3F \\
H_1 \\
H_1F
\end{bmatrix} x = T_d x
\tag{4.23}
\]

So we can write the new state-space equation:

\[
\frac{dx_d}{dt} = F_d x_d + G_d U_d
\tag{4.24}
\]

Where

\[
F_d = T_d F T_d^{-1} \quad \text{and} \quad G_d = T_d G T_u^{-1}
\tag{4.25}
\]

As we run the calculations in Matlab, and we get:
\[
G_d = \begin{bmatrix}
0 & 0 \\
42.7 & 0 \\
0 & 3.2 \\
0 & 176.5
\end{bmatrix} \quad \text{and} \quad F_d = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & -66.3 & -1.18
\end{bmatrix}
\]

(4.26)

In this way, the system has been decoupled into two separate systems.

\[
\begin{aligned}
\frac{d}{dt} x_3 &= 0 \begin{bmatrix} x_3 \\ \frac{d}{dt} x_3 \end{bmatrix} + 0 \begin{bmatrix} 42.7 \end{bmatrix} u_3 \\
\frac{d^2}{dt^2} x_3 &= 0 \begin{bmatrix} x_3 \\ \frac{d}{dt} x_3 \end{bmatrix}
\end{aligned}
\]

(4.27)

And

\[
\begin{aligned}
\frac{d}{dt} T_e &= 0 \begin{bmatrix} T_e \\ \frac{d}{dt} T_e \end{bmatrix} + 0 \begin{bmatrix} 3.2 \end{bmatrix} u_6 \\
\frac{d^2}{dt^2} T_e &= -66.3 \begin{bmatrix} T_e \\ \frac{d}{dt} T_e \end{bmatrix}
\end{aligned}
\]

(4.28)

The two system equations could be used to design a control system as if the plant is composed of two decoupled single input single output dynamics.

**4.5 Choices of modules**

In this section, we discuss how to choose the parameters in ADRC controllers.

**4.5.1 K_p, K_i, K_d controller gains in the tension controller**

According to ADRC theory, the tension system could be described as:
\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 + b_1u \\
\frac{dx_2}{dt} &= x_3 + b_2u \\
\frac{dx_3}{dt} &= G
\end{align*}
\] (4.29)

\(x_3\) is the state we do not know, and we set \(u = \frac{u_0 - x_3}{b_2}\), so the system equation becomes:

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 + \frac{b_1}{b_2}(u_0 - z_3) \\
\frac{dx_2}{dt} &= u_0
\end{align*}
\] (4.30)

Notice that only \(X_1\) and \(X_2\) are the variables we want to control. And the input \(U_0\) should be based on the error of estimated value and desired value. So we define \(U_0\) as:

\[
u_0 = K_p (x_1 - z_1) + \frac{K_i (x_1 - z_1)}{s} + K_d (x_2 - z_2)\] (4.31)

Since \(z_1, z_2, z_3\) are the disturbances, we could ignore them in the stability calculation. Taking \(U_0\) into the system equation, and applying the values \(b_1=3.2\) and \(b_2=176.5\), we could get the following characteristic equation:

\[
\begin{align*}
0.02K_d + s^2 + \frac{K_p + 0.02K_i}{0.02K_d - 1}s + \frac{K_i}{0.02K_d - 1} &= 0
\end{align*}
\] (4.32)

Since there are three variables in the equation, it is difficult to apply Routh’s stability criterion. Root locus could be easier to implement for studying stability of Equation (4.32). As we know, the settling time is
based on dominant poles is $4/\omega_n$, which we can use to make sure the system response is fast and has sufficient damping.

For the 3rd system in Equation (4.32), we need to choose 3 poles. We start the pole locations at $-3, -5 \pm 3i$, and use them in Equation (4.32) to find out the controller gains, which are found as $k_d=100, k_i=102$. We next relax the $K_p$ gain to explore if we can improve the system performance determined by the dominant poles. For this, we re-write Equation (4.32) in root locus form with respect to the parameter $K_p$

$$\frac{K_p (0.02s^2 + s)}{s^3 + 100s^2 + 2.04s + 102} = -1$$

(4.33)

We then plot the root locus in Matlab:

![Figure 4.4 poles’ locations of the root locus form in Equation (4.33)](image)

According to the Figure 4.4, it is obvious the poles are all on the left side of the Imaginary axis, which means the system is stable. We could choose a better $K_p$ on the root locus by dragging the poles location. In this case,
we choose $K_p = 2930$.

$$K_p = 2930$$

Finally, the parameters should be: $K_i = 100$

$$K_i = 100$$

$K_d = 102$

$$K_d = 102$$

Which locates the closed loop poles at: $-49.98 \pm 17.81, -0.04$

### 4.5.2 Kp Ki Kd selection speed controller

Compared with tension controller, speed control is relatively easier. The system equation is:

$$\frac{dx}{dt} = x_2 + bu$$

$$y = x_i$$

$$\frac{dx_2}{dt} = G$$

(4.34)

Assume the control signal is:

$$u_0 = (K_p + \frac{K_i}{s} + K_d s)(x_1 - z_1)$$

(4.35)

Take $U_0$ into system equation, we get:

$$(K_d - 1)s^2 + K_p s + K_i = 0$$

(4.36)

Suppose the poles are at $-4 \pm 4i$, then we get:

$$K_p = 1$$

$$K_i = 4$$

$$K_d = 1.125$$

### 4.5.3 Controller design of the Estimator in tension controller

Bet4 bet5 bet6 determine the stability and speed of the estimator, which directly affects the success of control. In the estimator, the system equations are given by:
\[
\begin{align*}
\frac{dz_1}{dt} &= z_2 + bu - \beta_4 e \\
\frac{dz_2}{dt} &= z_3 - \beta_5 e \\
\frac{dz_3}{dt} &= -\beta_6 e
\end{align*}
\] (4.37)

The states that need to be stabilized are \(z_1\), \(z_2\) and \(z_3\), so we need to treat control signal \(u\) and output \(y\) as disturbance. Transform the system equations into matrix form:

\[
\begin{bmatrix}
\frac{dz_1}{dt} \\
\frac{dz_2}{dt} \\
\frac{dz_3}{dt}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2 \\
z_3
\end{bmatrix} +
\begin{bmatrix}
-\beta_4 \\
-\beta_5 \\
-\beta_6
\end{bmatrix}
(z_1 - y) + \text{Disturbance} \quad (4.38)
\]

Next exclude the output \(y\) as it is the non-homogeneous part of Equation (4.38), and apply Laplace transformation on the homogeneous terms to get the eigenvalue problem:

\[
[sI - M] =
\begin{bmatrix}
s + \beta_4 & -1 & 0 \\
-\beta_5 & s & -1 \\
-\beta_6 & 0 & s
\end{bmatrix}
\] (4.39)

Which leads to the following characteristic equation:

\[
s^3 + \beta_4 s^2 + \beta_5 s + \beta_6 = 0 \quad (4.40)
\]

To make sure this system is stable, suppose the poles are all -10, take the poles into equation, to calculate the beta values by comparing Equation (4.40) with the following equation:
\[ s^3 + 30s^2 + 300s + 1000 = 0 \]  

(4.41)

\[ \beta_4 = 30 \]

From which we get:  
\[ \beta_5 = 300 \]
\[ \beta_6 = 1000 \]

4.5.4 Controller gains in the Estimator for speed controller

In the speed control problem, the estimator is a 1\textsuperscript{st} order system,

\[
\frac{dz_1}{dt} = z_2 + bu - \beta_1 e \\
\frac{dz_2}{dt} = -\beta_2 e
\]

(4.42)

Treat output y and control signal u as disturbance, and transform the system equation into matrix form:

\[
\begin{bmatrix}
\frac{dz_1}{dt} \\
\frac{dz_2}{dt}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix} +
\begin{bmatrix}
-\beta_1 \\
-\beta_2
\end{bmatrix} z_1 + \text{Disturbance}
\]

(4.43)

Make Laplace transformation on the homogeneous terms only to get the following eigenvalue problem:

\[
[sI - M] =
\begin{bmatrix}
s + \beta_1 & -1 \\
\beta_2 & s
\end{bmatrix}
\]

(4.44)

Then we suppose that two poles of the estimate are both at -1. Which re-writes as:

\[ s^2 + 2s + 1 = 0 \]

(4.45)

From which we get:
\[ \beta_1 = 2 \]
\[ \beta_2 = 1 \]

After all the parameters are chosen, the two ADRC controllers should be
stable. Since the controllers are independent from the plant, we could apply the controllers to control different plants, while keeping the system stable. For simulation purposes however we still need to use the plant model, nevertheless the controller around this plant does not use the plant model in formation, to be able to control the plant.
Chapter 5 – Simulation of tape drive system

5.1 Material model of conveyor system

In the simulation field, the microscopic properties of the rubber-like materials is not perfect [15], so are the test methods in establishing the characteristics of the rubber semi-empirical model. To get the model parameters, we need to take advantage of the test data. So we could form the equivalent model. This has become the main method of studying Rubber mechanical models.

But facing varieties of semi-empirical math models, how to choose the perfect and suitable model for a specific object of analysis, has become one of the most important topics. While choosing the math model, we need to consider the accuracy of the model, identification methods, as well as the test workload,

For this reason, we will compare several different and classic rubber semi-empirical mechanical models, and make a comprehensive analysis of their characteristics

5.1.1 Kelvin-Voigt model

Kelvin-Voigt model [16] is a simplified linear viscoelastic model. It was used in the early study of rubber mechanical characteristics. The Kelvin-Voigt model consists of an elastic element and a viscous
component. The two parts are organized in parallel. As shown in Figure 5.1.

![Kelvin-Voigt model](image)

Figure 5.1 Kelvin-Voigt model

Kelvin-Voigt model is the basic model we used in Chapter 4. Its motion equation is:

\[
F = k_1 x + b_1 \frac{dx}{dt}
\]  

(5.1)

Which is a linear model and can be expressed in our state-space model.

### 5.1.2 Three-parameter Maxwell model

The three-parameter model consists of a Maxwell model [17] and an elastic spring. Figure 5.2 shows the model.

![Maxwell model](image)

Figure 5.2 Maxwell model

The motion equation of the model is
\[ F + \frac{b_2}{k_2} \frac{dF}{dt} = k_1 x + (k_1 + k_2) \frac{b_2}{k_2} \frac{dx}{dt} \]  

(5.2)

This is time-variant model, as the tension \( F \) is a function of its rate of change in time. This model can easily be modeled in Simulink.

### 5.1.3 Dzierzek model

Among all the nonlinear factors that affect the dynamic characters of rubber, the large displacement nonlinear damping of elastic parts reflects frequency effect on rubber. Following is the Dzierzek model [18]:

![Dzierzek model diagram](image)

**Figure 5.3 Dzierzek model**

This model could well simulate damping and stiffness of rubber as discussed in [18]. The force consists of \( F_e, F_v, F_{m1}, \) and \( F_{m2} \). Here \( F_e \) is a nonlinear restoring force; and \( F_v, F_{m1}, \) and \( F_{m2} \) are forces of visco-elastic members. All the parameters of the model can be found in [18].

### 5.2 Building ADRC in Simulink

The original system can be decoupled into two independent systems, i.e. the tension control system and speed control system. Thus we need to
build two controllers, one for tension control, the other for speed control. Notice however that the speed characteristics of the tension and speed variables within ADRC design are based on the decoupled system, and hence, when we merge these systems together, we expect the system outputs to still remain stable but speed/performance characteristics could be affected due to coupling.

There are several ways to construct the ADRC controller here. The most common method is to write a Matlab code to perform the calculations, and use of s-function to define different parts of ADRC controller [19-20].

5.2.1 Speed controller

The control goal is to maintain the belt speed at a constant value. So we set the control signal as constant speed and zero acceleration. According to the decoupled system:

\[
\begin{bmatrix}
\frac{dx_3}{dt} \\
\frac{d^2x_3}{dt^2}
\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ \frac{dx_3}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ 42.7 \end{bmatrix} u_3 \tag{5.4}
\]

We set matrix \( A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \), \( B = \begin{bmatrix} 0 \\ 42.7 \end{bmatrix} \), but we do not need to express \( A \) in the controller. The main mission is to build ESO in speed controller, as shown in figure 5.4. According to ADRC theory, the ESO could be built as:
And since we only need to control the speed and the system is a 1st order system, there is only one constant input signal, and the nonlinear sum could be a PID controller. The derivative part is defined as $\frac{s}{s+1}$, as shown in figure 5.5.

Finally, combining the two parts together in figure 5.6, we get the speed controller, based on the ADRC theory.
5.2.2 Tension controller

The control goal is to keep tension at a constant value, and achieve the control process in a relative short time. The tension system is given by:

\[
\begin{bmatrix}
\frac{dT_e}{dt} \\
\frac{d^2T_e}{dt^2}
\end{bmatrix} = 
\begin{bmatrix}
0 & 1 \\
-66.3 & -1.18
\end{bmatrix}
\begin{bmatrix}
\frac{T_e}{dt} \\
\frac{dT_e}{dt}
\end{bmatrix} + 
\begin{bmatrix}
3.2 \\
17.5
\end{bmatrix} u_i \tag{5.5}
\]

Set matrix \( A = \begin{bmatrix} 0 & 1 \\ -66.3 & -1.18 \end{bmatrix} \), \( B = \begin{bmatrix} 3.2 \\ 176.5 \end{bmatrix} \), so the system is rewritten as:

\[
\begin{aligned}
\frac{dx}{dt} &= AX + Bu \\
Y &= CX
\end{aligned} \tag{5.6}
\]

According to ADRC theory, we need to keep the estimated value \( z_1 \) as close as possible to output \( y \). Based on this, we build the ESO in tension controller as shown in figure 5.7:
Figure 5.7 tension estimator structure

Since we have a $2^{nd}$ order system, and the tension must be kept constant, we have that its time derivative must be zero. The weighted sum could be organized as:

Figure 5.8 weighted sum of individual control actions

When we combine the two parts together, as shown figure 5.9, we get the tension controller:
All the modules are chosen, so we define all the gains at certain values. Since the controller is a little complex, and inconvenient to be applied to a system, we need to create a subsystem, as shown in figure 5.10.

**5.3 Simulation Results.**

Since the system is decoupled, if we want the system to be stable, we need to make sure the two controllers are stable. In this method, we build the two systems independently and to see whether the system is stable

**5.3.1 Tension controller test**

According to the decoupled system matrix, we use state-space block to
describe the plant, as shown in figure 5.11.

The whole system is shown in figure 5.12.

When we run the system, we could get the following plot, which indicate that the system is stable after 2 seconds, so the settling time is good, and
controller is stable, as shown in figure 5.13.

![Figure 5.13 tension controller test result](image)

**5.3.2 Speed controller test**

We treat the speed controller in the same way as the tension controller.

Figure 5.14 shows the controller setup in Simulink.

![Figure 5.14 speed test state-space window](image)
As shown in figure 5.15, first of all, we create a subsystem and use state-space block to describe the plant.

Figure 5.15 speed controller test

Figure 5.16 shows the speed control plot; the settling time is good, which means the controller is stable.

Figure 5.16 speed controller test result
5.4 Putting the whole system together

5.4.1 Kelvin-Voigt model

Kelvin-Voigt model is the basic model we use, and the whole plant could be described in state-space blocks. We set:

\[
[A] = \begin{bmatrix}
0 & 0 & -10 & 0 \\
0 & 0 & 0 & 10 \\
3.315 & -3.315 & -0.5882 & -0.5882 \\
3.315 & -3.315 & -0.5882 & -0.5882 \\
\end{bmatrix}, \quad [B] = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
8.533 & 0 \\
0 & 8.533 \\
\end{bmatrix}
\]

Also we need to couple the control signal together, so set:

\[
[T_u]^{-1} = \begin{bmatrix}
-0.5 & 0.5 \\
0.5 & 0.5 \\
\end{bmatrix}
\]

Matrixes \([A]\) and \([B]\) are used in state-space block to describe the plant in Simulink. And matrix \([T_u]^{-1}\) is used in gain block to decouple the control signal.

Actually, there will be noise in the real experiment, so we use white noise block to simulate noise. Finally the whole system could be organized as shown in figure 5.17.
There are two filters in the system to weaken the noise effect. According to the system matrix given by Equation (4.19), the output is position and tension, if we want to control speed, we also need to get derivative of position, so the filter1 is a combination of filter and derivative.

When we run the simulation, we get the plots shown in figures 5.18 and 5.19, the yellow line is speed, and the purple line is tension, we could see the filter works obviously in the system.

According to the plots, the filter reduce the effect of noise, but it also slow down the whole system, so we need to choose a suitable filter depending on our need.
5.4.2 Three-parameter Maxwell model

Since in the three-parameter Maxwell model, the force is changing with time, so it is difficult to apply state-space to describe the plant. To organize the plant correctly, add a derivative block and a gain to build a closed loop feedback. In this way, we could describe the tension.
In this model, desired velocity is 3 m/s, desired tension is 1 N. The tension and velocity should become stable as quickly as possible. Figure 5.20 shows Maxwell model.

![Figure 5.20 Maxwell model test](image)

Before simulation, we need to set simulation time to 10 sec. And in the configuration parameters option, set Type as “Fixed-step”, Fixed-step size as 1e-3. This will speed up the simulation. We run the simulation to generate figure 5.21:
Since the plant is quite complex, create a subsystem of Maxwell model to simplify this plant. Select the plant blocks; add one input block, named “current”; add one output block, and add the tension and velocity outputs together. In this way, the whole system changes to figure 5.22:

Also, by noticing that, in the real experiment, there should be noise existing. So we add white noise block to the plant:
Set noise power as 0.00001 and simulation time as 10 seconds. Run the simulation, we obtain figure 5.24.

According to the plot, the max tension is almost 12 N. Although the system is stable, the plot is bad; we need to choose another group of parameters’ values in the controllers to make a better plot.
5.4.3 Dzierzek model

Since in the Dzierzek model, there are several forces, such as $F_{m1}$ and $F_{m2}$, changing with time, it is more difficult to apply state-space to describe the plant. To organize the plant correctly, we add derivative blocks and corresponding gains to build closed loop feedbacks. In this way, we could describe the whole system.

The desired velocity is 5 m/s, desired tension is 2 N. The final goal is to keep velocity and tension constant and stable. Figure 5.25 shows the model.

A complex plant will always affect us to modify or improve the system. Again select all the plant, add the outputs together, add input and output blocks, and create the subsystem, as shown in figure 5.26.
The whole system is simplified. Next step, we add white noise block to the system, the noise power is 0.00001, as shown in figure 5.27.

Before simulation, we need to set simulation time as 10 sec. And in the configure ration parameters option, set Type as “Fixed-step”, Fixed-step size as 1e-3. This will speed up the simulation. When we run the simulation, we will get figure 5.28:
The noise makes the plot unclear. To get a better plot, we add two filters into the system. Since the output is position, the filter for velocity should also include the derivative function, as shown in figure 5.29.

Figure 5.28 Dzierzek system test result

Figure 5.29 Dzierzek system with filters
The noise effect is obviously reduced, but the filter also effects the settling time and vibration of plot, as shown in figure 5.30.

To make this model more close to real conditions, we add a Viscous Friction block to the tension output, as shown in figure 5.31. Exclude the effect of noise at first.
Compared with the former result, there is obvious that the plot is changed, as shown in figure 5.32.

Next step, add the noise and filters. We Notice that the filters may cause the system to become unstable. So we need to consider both response time and stability at the same time, as shown in figure 5.33.

Set the parameters in Viscous Friction block, suppose the coulomb friction value is 1N, and coefficient of viscous friction is 0.5. Still add the
noise, filters and run the simulation to generate plot shown in figure 5.34.

Figure 5.34 test result

5.5 Analysis of result

5.5.1 Effects of poles’ locations

According to the plot of Maxwell model and Celvin-Vigot model, when the simulation began, there are lots of damping in tension curves, which means the tension estimator could not keep up with the output. So suppose the estimator is too slow to changing with output. Considering the parameters effects estimator are bet4, bet5 and bet6, it is necessary to choose better poles’ locations. The original poles’ locations are:

\((-10,0),(-10,0),(-10,0)\)

The poles do not have a good damping ability, so we choose new poles:
Calculating with Matlab, we got the equation:

\[ s^3 + 160s^2 + 8725s + 163500 = 0 \]

So the corresponding parameters’ values are:

\[
\begin{align*}
    b_{et_4} &= 160 \\
    b_{et_5} &= 8725 \\
    b_{et_6} &= 163500 
\end{align*}
\]

When we run the simulation to test the result, we can see the result clearly, excluding the effect of noise at first, as shown in figure 5.35 and 5.36.
It is clear that the damping at beginning still looks bad. So suppose another group of poles:

\((-10,0),(-1+i,0),(-1-i,0)\)

With the help of Matlab, we get the equation:

\[s^3 + 12s^2 + 22s + 20 = 0\]

So the corresponding parameters’ values are:

\[\begin{align*}
    \text{bet}_4 &= 12 \\
    \text{bet}_5 &= 22 \\
    \text{bet}_6 &= 20
\end{align*}\]

We run the simulation
Figure 5.37 Dzierzek model test result

Figure 5.37 shows the damping at beginning was reduced obviously, but the settling time is longer. Since the dominate poles are \((-1+i, 0), (-1-i, 0)\), so this result reflects the effects of poles’ locations.

Figure 5.38 Maxwell model test result

Figure 5.38 shows the damping part is also reduced. And there is a little
delay in settling time. This result shows that the dominate pole’s location
effect both settling time and damping ability of the system. And a better
response time often means a bad effect on damping ability.

5.5.2 Effect of filter

The time coefficient $\tau$ in the denominator of filter always creates a delay
in the system. Also different value of $\tau$ will affect the filter’s ability to
reduce effects of noise.

Here take Maxwell model as an example, choose $\tau_1 = 12, \tau_2 = 4$, and run
the simulation. The result is shown in Figure 5.39.

Choose $\tau_1 = 4, \tau_2 = 2$, and run the simulation.
Compare the two results shown in figure 5.39 and 5.40, we could see the settling time is much reduced, but the plots of velocity become very rough. So the time coefficient is chosen by the priority between settling time and noise reducing.
Reference

Tanaka,” A method for auto-tuning of PID control parameters”.


[18] DZIERZEK S. “Experiment-based modeling of cylindrical rubber
