Optimisation of Motion Profiles to Reduce Energy in Motor-Driven Systems

by

Marcel Afunyah

u2015484

Supervised By: **Mark Dooner**

3rd Year Project [CS351]

Department of Computer Science

May 2023

Abstract

Modern industries rely heavily on the use of robotics and automation. Running such machinery accounts for a large amount of energy consumption, leading to increased bills and carbon footprint. The aim of this project is to quantify energy consumption of industrial robotics and develop suitable software modification techniques to increase energy efficiency. These optimisations are developed for use in conveyor belt systems, a very commonly used system. The main outcome of this project is an application which can successfully calculate the optimal values for crucial parameters of the system, which lead to lower energy consumption. These modifications do not require changes in hardware to be made, making it suitable for already existing systems.

Keywords: Motion Profile, Motor, Conveyor, PID Controller, Genetic Algorithm, Energy, Load

Table of Contents

List of Figures

List of Tables

List of Equations

1 Introduction

1.1 Background

The gradual shift of production and manufacturing facilities from their dependence on manual labour to robotics and automation (Atkinson, 2019) has pushed the topic of energy consumption into the light. Whereas energy consumption of factories was not a talking point for processes that predominantly relied on manpower, it becomes a main concern in the planning, design and maintenance of factories which are fully or substantially automated.

Automated industrial processes are often comprised of repetitive tasks; conveyor belts, palletizing robots, CNC machines, assembly robot arms and so on are common examples of machines that work repetitively in one motion along a set path. The energy usage of a single unit does not raise an issue; however, such machines are usually deployed in large numbers forming part of production lines and are likely to be operated for several hours at a time.

According to the International Federation of Robotics (IFR), the active working stock of industrial robot installations was computed at 3,014,879 units in 2020; seeing a 10% increase from the previous year. Asia alone contributed to 71% of newly deployed robot installations, with China and Japan leading the advancement. The largest customer industries for automated robotic systems were the electrical/electronics sector (29%) and the automotive sector (21%) in 2020. (IFR, 2021)

In addition to the increasing dependence on industrial robots and the steady annual growth of units deployed, energy costs are also seeing a significant rise.

The Department for Business, Energy & Industrial Strategy (BEIS) published that the average cost of non-domestic (transportation, industrial and commercial) electricity in the UK has increased by 45% from April 2021 to June 2022 alone (BEIS, 2022).

The U.S. Energy Information Administration (EIA) also released data that indicates the energy consumption by industrial processes accounted for 32.79% of the overall energy consumption in 2018 (EIA, 2022). In the same year, the industry sector in the UK accounted for 17% of the total energy consumption (Waters, 2018).

As energy consumption increases, the question of environmental concern and responsibility also rises. Considering the role large industries play in current affairs involving climate change, in terms of pollution (Howell, 2022), it is in the best interest of the public and industrial corporations to find ways to efficiently reduce energy usage. Nowadays, attention to cleaner energy policies and lower energy usage is a good indicator of ‘environmental friendliness’ which tends to work in favour of industrial corporations through reduced running costs and swaying public opinion about their products.

This project focuses on the analysis of the operation of robotic systems in order to effectively optimise energy usage. The techniques employed are purely software based. By doing so, existing machinery can be optimised without altering fixed hardware. The systems studied in this paper will be similar to a simplified version of industrial robotics.

In order to understand the aim of the project, it is important to understand motion profiles. A motion profile of a system describes the behaviour of a motor during movement of a load. The position, velocity and acceleration of the load are specified in the motion profile. There are several different motion profiles, some examples are trapezoidal, s-curve, cycloid and sinusoidal. Each profile has its own advantages and disadvantages.

For the target system, conveyor belts were chosen. Conveyor belts are a common installation in many industrial settings. Since they operate repetitively for long durations of time, small optimisations in energy consumption accumulates to significant savings. This, in combination with software-only changes, would be very useful for real-life applications.

1.2 Relevant Literature

Carabin et al (2019) investigated optimising the trapezoidal motion profile for conveyor systems, similar to the aim of this project. The main outcome of this research showed that it is possible to reduce the energy consumption of industrial robots without altering their hardware. Using mathematical models, it was demonstrated that changing the behaviour of a motor during the movement of a load can affect and optimise the energy consumed.

However, only the trapezoidal motion profile was explored alongside constraints such as having equal acceleration and deceleration times. As well as optimising only one variable; either the total time taken or the acceleration time.

Carabin et al (2020) further extended the research to include the cycloidal motion profile and physical test rigs. The tests were conducted on a linear axis of a Cartesian robot and on two coupled servomotors. Both tests verified the initial propositions made, thereby expanding the generalisability of the research to more types of machines.
However, just as the previous research, there is a lack of optimisable parameters; either the total time taken or the acceleration/deceleration time can be optimised.

Assad et al (2018) investigated optimising the energy consumption of the s-curve trajectory. In this research, a simple mass and spring system was used. Interestingly, a Particle Swarm Optimisation algorithm was used. The results proved, yet again, that the energy consumption of the system was optimisable.
Although a rather simple system was used in this case, it provides the grounds to explore more complex systems with multivariable optimisation techniques.

1.3 Objectives

The aim of this project is to optimise the energy consumption of conveyor belt systems. The differences between this project and the previous work discussed are as follows:

- The focus of this project is solely conveyor belt systems, so a lot of physical modelling concerning the conveyor system and the driving motor is explored.
- Great detail about the setup of the mathematical models and simulation models have been provided.
- A hybrid genetic algorithm is used to handle multivariable optimisation.
- A complementary application has been developed to combine all the findings and results of the research

That being said, the results of previous work are also confirmed in this project.

The ideal workflow for the project can be described as listed below:

- Modelling the conveyor system. Firstly, through mathematical modelling. This step will consist of representing the system as a set of mathematical equations. The equations will feature parameters such as motor current and voltage, various torques, inertia and total energy usage. The next step will be to model the system in simulation software such as Simulink. By doing so, the behaviour of the system can be studied by altering parameters used in mathematical modelling.
- Defining a trajectory. For example, the trajectory can be described by setting the desired distance moved and the time taken to move that distance.
- Establishing the motion profile of the system. For the defined trajectory, choose a suitable motion profile shape (trapezoidal, s-curve, cycloid etc.) in terms of properties such as jerk.
- Obtaining baseline data for the system. As energy consumption is the target parameter, it is necessary to obtain data for the pre-optimisation state of the system in order to compare and justify the post-optimisation state of the system. This can be done by observing simulation data.
- Optimising the motion profile. An appropriate optimisation technique should be used. The simplest forms of optimisation are based on finding local minima by differentiation. However, more advanced techniques such as genetic algorithms can be applied. The expected energy reduction for a single movement cycle will be small; however, the accumulated energy reduction over several cycles should be significant.

2 System Design and Modelling

2.1 System Overview

The main components of the modelled system are the conveyor, the DC motor and the motion profile controller. Measurements such as velocity of the load and energy values are taken during operation.

The controller controls the motor by varying the input voltage, allowing different torques to be applied. The applied torque moves the load through a distance and the velocity of the load increases. The velocity of the load is fed back as an input to the controller.

Based on the velocity of the load, the controller adjusts the motor voltage to achieve the desired motion as specified by the motion profile. The basis of this control is a PID controller, linking the motor control to the motion profile.

Figure 1 illustrates a condensed block diagram which shows a simplified overview of the whole system. The control parameters consist of user defined variables such as the total time of the motion and length of the trajectory. Optimised parameters (if available) are also included in the control parameters.

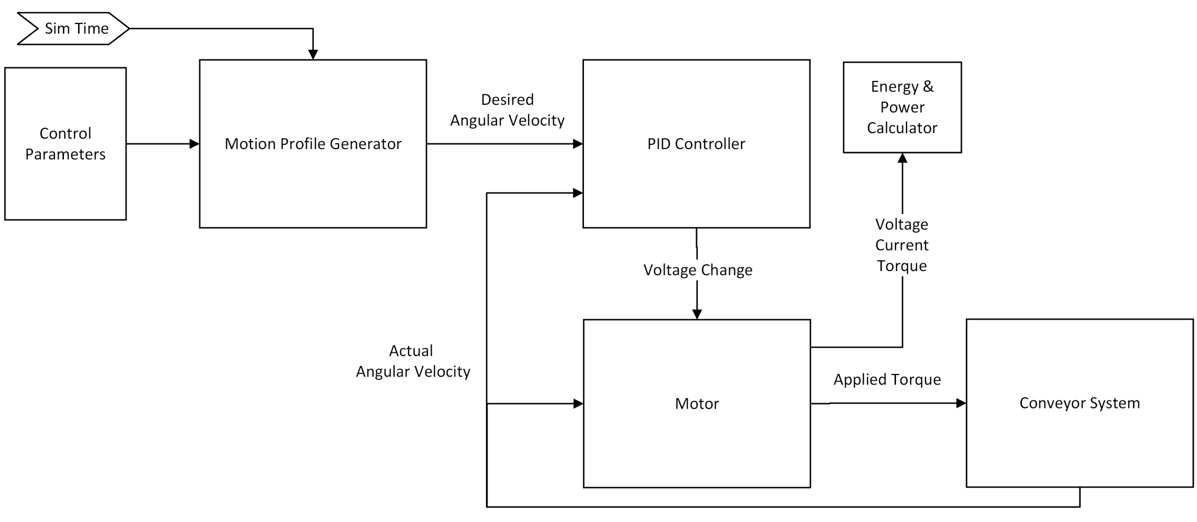

Figure 1 Condensed Block Diagram of System Overview

2.2 System Modelling

2.2.1 Conveyor Configuration

The conveyor belt system consists of a motor, shafts, pulleys, a gearbox, a belt and a load (mass or object to be transported).

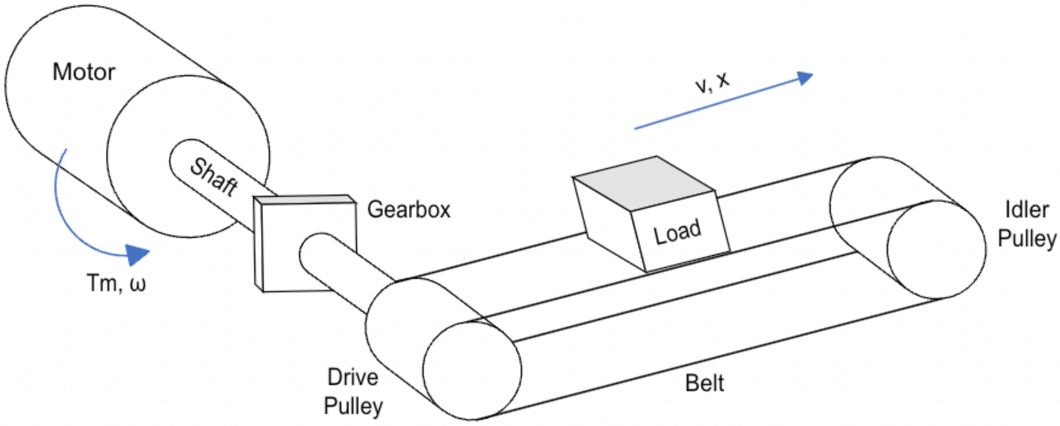

Figure 2 Conveyor System Diagram

Conveyor belt systems have many configurations, such as the roller-bed and modular belt conveyors. The most appropriate configuration in this case is the slider-bed conveyor belt. In this setup, the bed is a slab placed inside the belt loop. The topside of the conveyor belt lies on the bed. Ideally, a material for the bed should be chosen such that there is a low coefficient of friction between the belt and the bed.

During operation, the belt slides over the bed. The bed is used as support for heavy loads, preventing the pulleys from being overloaded.

This configuration was chosen because it is very common and can be used for a large variety of purposes, as opposed to configurations which are specific to just one product.

The Model 108 from Titan Conveyors is an example of a slider-bed conveyor system. (Titan Conveyors, n.d)

Figure 3 Model 108 Conveyor from Titan Conveyors

2.2.2 Conveyor Modelling

The shafts and pulleys are modelled as solid cylinders whereas the belt, load and gearbox are modelled as point masses. The motor can also be considered as a point mass, though datasheets typically provide inertia values.

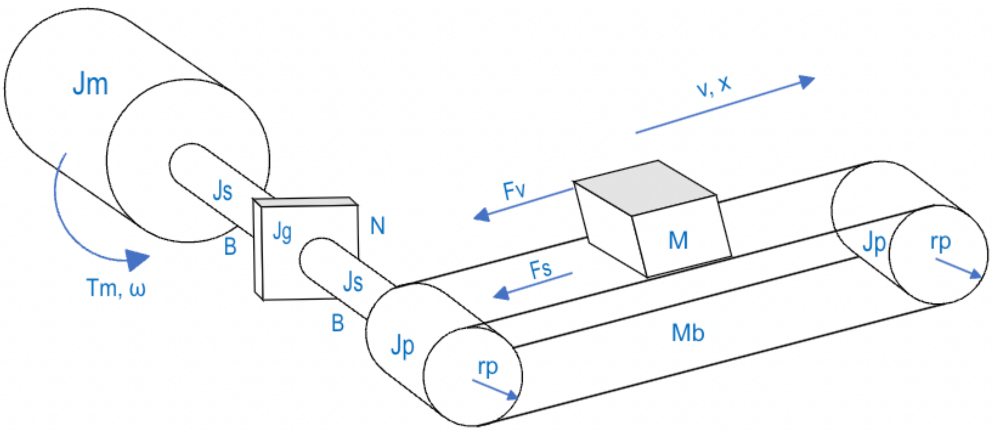

Figure 4 Annotated Conveyor System Diagram

The gearbox has an efficiency value which is also included in the system model, making the model more realistic and accurate.

It can be assumed that the pulleys are identical, and the shafts are also identical.

The angle of inclination between the conveyor and the horizontal is given as θ, measured in radians.

Inertia

Inertia is the property of a mass which concerns its reluctance to move from rest or come to a stop during motion. This directly affects the acceleration (and deceleration) of the system. The inertias of the components are related to their shapes and masses.

The unit of measurement used for inertia is kgm^2 (SI unit). The inertias of the shafts, pulleys, load and belt are given as:

$$J_S = \frac{1}{2} m_S {r_S}^2$$

$$J_P = \frac{1}{2} m_P {r_P}^2$$

$$J_L = m_L {r_P}^2$$

$$J_B = m_B {r_P}^2$$

Where J is inertia, m is mass (in kilograms) and r is radius (in metres). The subscripts S, P, L and B stand for shaft, pulley, load and belt respectively. So J_P represents the inertia of the pulley, m_L represents the mass of the load, and so on.

The inertia of the motor is J_M and the inertia of the gearbox is J_G. These values are typically provided in the datasheets of the motor and gearbox.

The total inertia of the system, J_T, can be written as:

$$J_T = J_M + J_G + J_S + \frac{1}{N^2 \eta_g} \cdot (2J_p + J_L + J_B + J_S)$$

$$J_T = J_M + J_G + \frac{1}{2} m_S {r_s}^2 + \frac{1}{N^2 \eta_g} \cdot \left({r_p}^2 (m_p + m_L + m_B) + \frac{1}{2} m_S {r_s}^2 \right)$$

The dividing factor $N^2 \eta_g$ arises from the fact that the two pulleys, the load, the belt and the second shaft need to be referred back to the motor across the gearbox. As such, N is the gear ratio and η_g is the efficiency of the gearbox.

Damping

Shafts introduce damping into the system, due to them being connective components affected by vibrations. The motor also has internal damping, which is included in the system. This

value is typically provided in the datasheet. Damping is related to vibrations, and so it is directly affected by the speed of the system.
The unit of measurement used for damping is Nms/rad. The damping equations are given as:

$$B_T = B_M + B_s + \frac{B_s}{N^2}$$

Where B_T is the total damping of the system, B_M is the internal damping of the motor and B_s is the shaft damping (with the second shaft referred back to the motor across the gearbox).

Force
The force of gravity (or weight) due to the load and belt must also be overcome by the motor. The expression for weight is given as:

$$F_G = (m_B + m_L)\ g \cdot sin(\theta)$$

Where F_G is the resultant force of gravity, measured in Newtons (N). g is the acceleration due to gravity, given as 9.81m/s^2. The angle of elevation above the horizontal, θ, is also accounted for.

Friction is generally a very complex force to model. In this system, the most important form of friction is the static friction between the belt and the bed. This can be modelled as follows:

$$F_f = \mu_B(m_B + m_L)\ g \cdot cos(\theta)$$

Where F_f is the frictional force, measured in Newtons (N), and μ_B is the coefficient of friction between the bed and the belt.

Torque
The resultant forces can be converted to the torque load which must be overcome by the motor. The torque load, T_L is given as:

$$T_L = \frac{r_P}{N\eta_g}(F_G + F_f)$$

$$T_L = \frac{r_P(m_B + m_L)\, g}{N\eta_g}(\mu_B cos(\theta)\; + \; sin(\theta))$$

The torque load is referred back to the motor across the gearbox. The unit of measurement used for torque is Nm (SI units).

The overall inertia, damping and torque load contribute to the total torque which the motor must overcome in order to move the load. The system equation can be written as:

$$T_M = J_T\ddot{\theta} + B_T\dot{\theta} + T_L$$

$$T_M = \left(J_M + J_G + J_S + \frac{1}{N^2\eta_g} \cdot \left(2J_p + J_L + J_B + J_S\right)\right)\ddot{\theta} + \left(B_M + B_s + \frac{B_s}{N^2}\right)\dot{\theta} + \frac{r_P}{N\eta_g}(F_G + F_f)$$

$$T_M = A\ddot{\theta} + B\dot{\theta} + D \qquad (1)$$

Equation 1 Motor Torque Derivation

Where $A = J_T,\; B = B_T,\; D = T_L$

The motor torque, T_M, is the torque which the motor must produce to move the load.

$\ddot{\theta}$ represents the acceleration of the load, measured in m/s^2, and $\dot{\theta}$ represents the velocity of the load measured in m/s.

2.2.3 Electric Motor Modelling

The motor can be modelled as a simple electrical circuit consisting of a resistor, an inductor and the back-emf produced by the motor to apply a torque. This is shown in Figure 5.

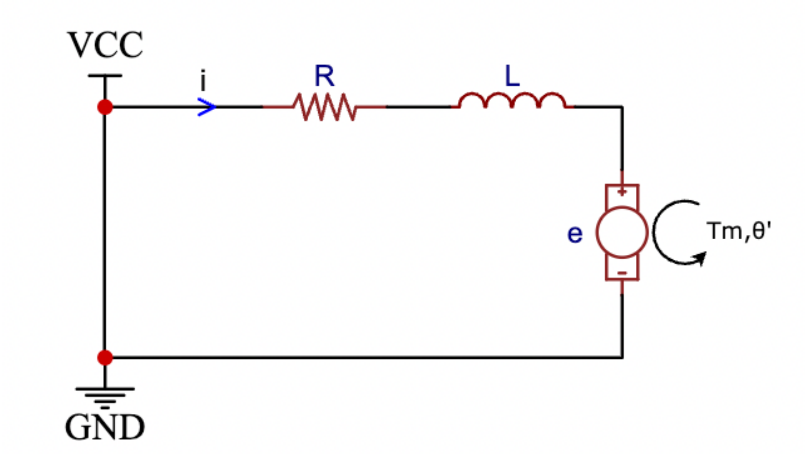

Figure 5 Electrical Circuit for Simple Motor Model

The electrical system equation for the motor is a first order differential equation and is given by:

$$V = L\frac{di}{dt} + iR + K_e\dot{\theta} \qquad (2)$$

Equation 2 Electrical System Equation

Where V (in volts, V) is the input voltage, i (in amperes, A) is the current produced, R (in ohms, Ω) and L (in Henry, H) are the resistance and the inductance of the motor windings, respectively.

The torque constant, K_e, is a constant of proportionality which defines the relationship between the motor torque and the current. Mathematically,

$$T_M = K_e i \qquad (3)$$

Equation 3 Torque Constant Equation

Where K_e is measured in mNm/A.

2.2.4 Combined System Model

The system can be represented mathematically as coupled differential equations. From equations 1 and 3, the combined system equation is given as:

$$K_e i = A\ddot{\theta} + B\dot{\theta} + D \tag{4}$$

Equation 4 Combined System Equation

This can be implemented in Simulink by rearranging the equation 2 and equation 3,

$$\ddot{\theta} = \frac{K_e i - B\dot{\theta} - D}{A}$$

$$\frac{di}{dt} = \frac{V - iR - K_e\dot{\theta}}{L}$$

The derived models are implemented in Simulink as shown in Figure 6 and Figure 7.

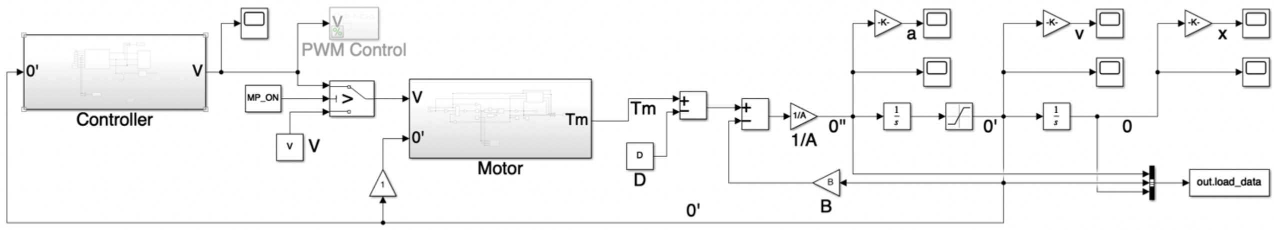

Figure 6 Top View of Full Simulink Model

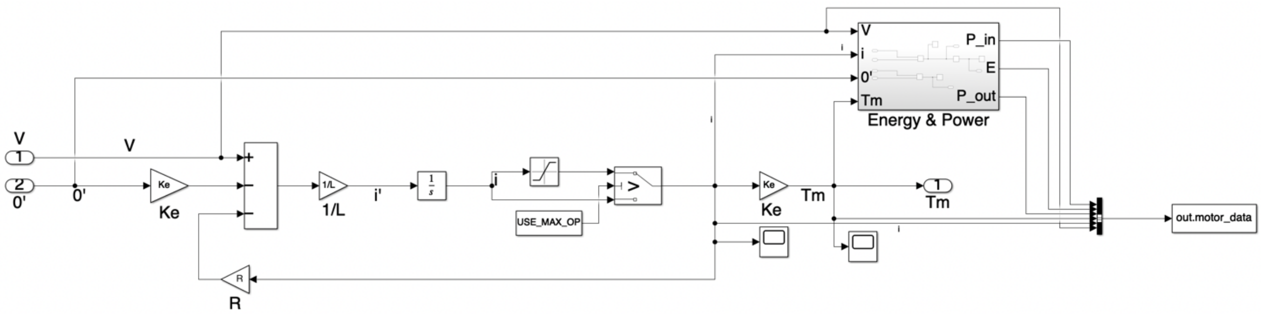

Figure 7 Motor Simulink Model

2.2.5 System Model Verification

To verify and test the feasibility of the system, the motor model was tuned to the specifications of real motors. Information regarding motors can be found in datasheets (accessible online), as they are made publicly available by manufacturers.
Using real motor data provides a foundation to confirm whether the derived system model is feasible or not.

The selected motors used were the *A-max 26, 7 Watt* and the *RE 65, 250 Watt* from Maxon. (Maxon, 2023)
Two motors with a large difference in specifications were chosen to ensure the model was credible.
The datasheet typically provides values of the motor at no-load condition and at stall conditions.
The no-load condition occurs when the motor is left to spin freely, without being connected to a shaft and a load. This can be attained within the model by setting all inertia, damping and torque terms (related to the load) to 0.
The stall condition occurs when the torque produced by the motor is insufficient to move a given load. In other words, when the operator attempts to push the motor past its limits. This can be attained within the model by setting the mass of the load to an unrealistically high value.

Table 1 is an extract from the RE 65 motor datasheet.

VALUES AT NOMINAL VOLTAGE	
Nominal voltage	24 V
No load speed	4090 rpm
No load current	697 mA
Nominal speed	3810 rpm
Nominal torque (max. continuous torque)	501 mNm
Nominal current (max. continuous current)	10 A
Stall torque	15700 mNm
Stall current	292 A

CHARACTERISTICS	
Terminal resistance	0.0821 Ω
Terminal inductance	0.0308 mH
Torque constant	53.7 mNm/A
Rotor inertia	1290 gcm²

Table 1 Maxon RE 65 Motor Parameters

The model parameters were adjusted to emulate the no-load and stall conditions. The results obtained are shown in Table 2.

No load speed	4257 rpm
No load current	735 mA
Stall torque	15700 mNm
Stall current	292.3 A

Table 2 Maxon RE 65 Simulation Results

From the results obtained, it is sufficient to conclude that the model is verified and accurate enough for the scope of this project.

2.3 Motion Profile Design

2.3.1 Motion Profile Theory

A motion profile describes the behaviour of a motor during the movement of a load. The motion profile is concerned with certain properties of the load motion such as: how fast it should go, how quickly it should it set off and how quickly it should stop.
Formally, the properties of a motion profile are:

- Position
- Velocity
- Acceleration
- Jerk
- Total time
- Distance or length of the trajectory

Motion profiles are an important aspect of systems which move loads. The motion profile designer takes many factors into account when designing a system. Factors such as the weight and the fragility of the load affect the choice of motion profiles.

Different motion profiles have different properties. For example, a motion profile may be configured to prioritise jerk reduction over max velocity of the load. This will be more suitable for moving fragile loads, as there will be less quick and sudden sharp movements. There are several different motion profiles, some examples are trapezoidal, s-curve, cycloid and sinusoidal. Each profile has its own advantages and disadvantages.

The properties of motion profiles are defined in terms of mathematical equations. The graphs of a trapezoidal motion profile are shown below, in Figure 8, and its corresponding equations are shown in Figure 9.

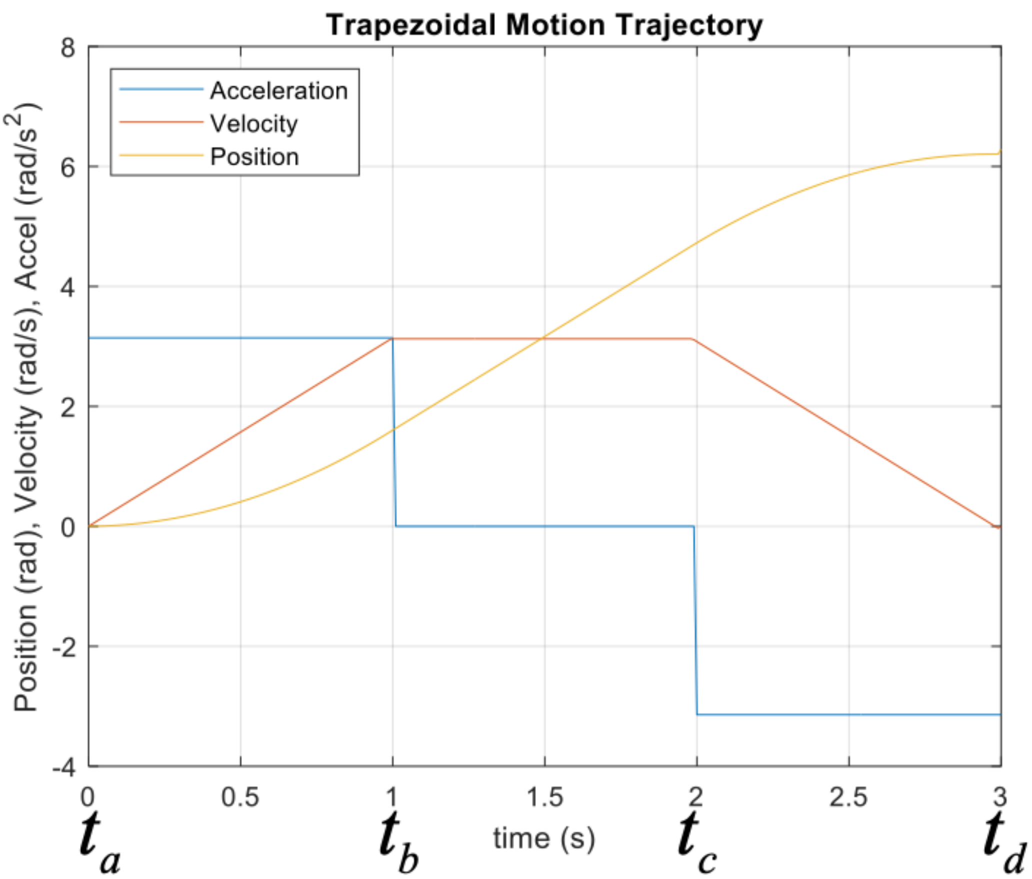

Figure 8 Trapezoidal Motion Profile Graphs

$$\ddot{\theta}_{trap}(t)=\begin{cases}\alpha & t\in[t_a,t_b]\\ 0 & t\in(t_b,t_c]\\ -\alpha & t\in(t_c,t_d]\end{cases}\qquad \dot{\theta}_{trap}(t)=\begin{cases}\alpha t & t\in[t_a,t_b]\\ \alpha t_b & t\in(t_b,t_c]\\ \alpha t_b-\alpha(t-t_c) & t\in(t_c,t_d]\end{cases}$$

$$\theta_{trap}(t)=\begin{cases}\frac{1}{2}\alpha t^2 & t\in[t_a,t_b]\\ \frac{1}{2}\alpha t_b^2+\alpha t_b(t-t_b) & t\in(t_b,t_c]\\ \frac{1}{2}\alpha t_b^2+\alpha t_b(t_c-t_b)+\alpha t_b(t-t_c)+\alpha t_c(t-t_c)-\frac{1}{2}\alpha(t^2-t_c^2) & t\in(t_c,t_d]\end{cases}$$

Figure 9 Trapezoidal Motion Profile Equations

$\ddot{\theta}, \dot{\theta}, \theta$ represent functions of acceleration, velocity and position respectively (with respect to time). The constant α represents the maximum acceleration.

The main graph of motion profiles is the velocity graph, in this case the trapezoidal motion profile is named as such because the velocity graph has a trapezoidal shape.

Typically, a motion profile consists of three phases: the acceleration phase, the constant velocity phase and the deceleration phase. Each phase has an active time frame. Using the trapezoidal motion profile graphs from Figure 8 as an example, the time frame of the acceleration phase is $[t_a, t_b]$, $(t_b, t_c]$ for the constant velocity phase and $(t_c, t_d]$ for the deceleration phase.

The time frames of each phase are used to derive the equations of motion, since the velocity graph can be split into distinct shapes. For example, the acceleration phase in the time frame $[t_a, t_b]$ is triangular, so its enclosing area can be calculated as

$$\frac{1}{2} * base * height$$

The geometric formulas at each phase can be manipulated to obtain the equations of motion.

The main motion profile properties which interact with the system model are the position, velocity and acceleration. Jerk is the first derivative of acceleration (or the third derivative of position) but is of less importance in this use case. The shape of the acceleration graph is enough to infer whether the jerk is suitable for a particular case.

2.3.1.1 Slew Rate

For a given motion profile, an additional property known as the slew rate can be derived from the total time taken, the acceleration and the deceleration time.
The slew rate is defined as the fraction of the total time where the velocity of the load is constant. Various time periods of the motion can then be written in terms of the slew rate.

$$t_S = kt_M$$

$$k = \frac{t_M - t_{ACC} - t_{DEC}}{t_M}$$

$$t_a = 0$$

$$t_b = t_a + \frac{1}{2}\left(t_M - kt_M\right) = t_{ACC}$$

$$t_c = t_b + kt_M$$

$$t_d = t_c + t_b = t_M$$

Figure 10 Equations for Time Periods and Slew Rate

The slew rate equation condenses the total time taken, the acceleration and the deceleration time into one parameter. This allows for simpler equations to be used during the verification of the Energy Optimisation.

2.3.2 Motion Profile Controller

Motion profiles are expressed as a set of equations which can be implemented in code.
For a given motion profile, a MATLAB function is created and added to the Simulink model.

For this project, the trapezoidal, cosine and cubic motion profiles were implemented. This allows an operator to select a suitable motion profile based on their needs.
A switch-case mechanism in Simulink controls the motion profile which is used during operation.

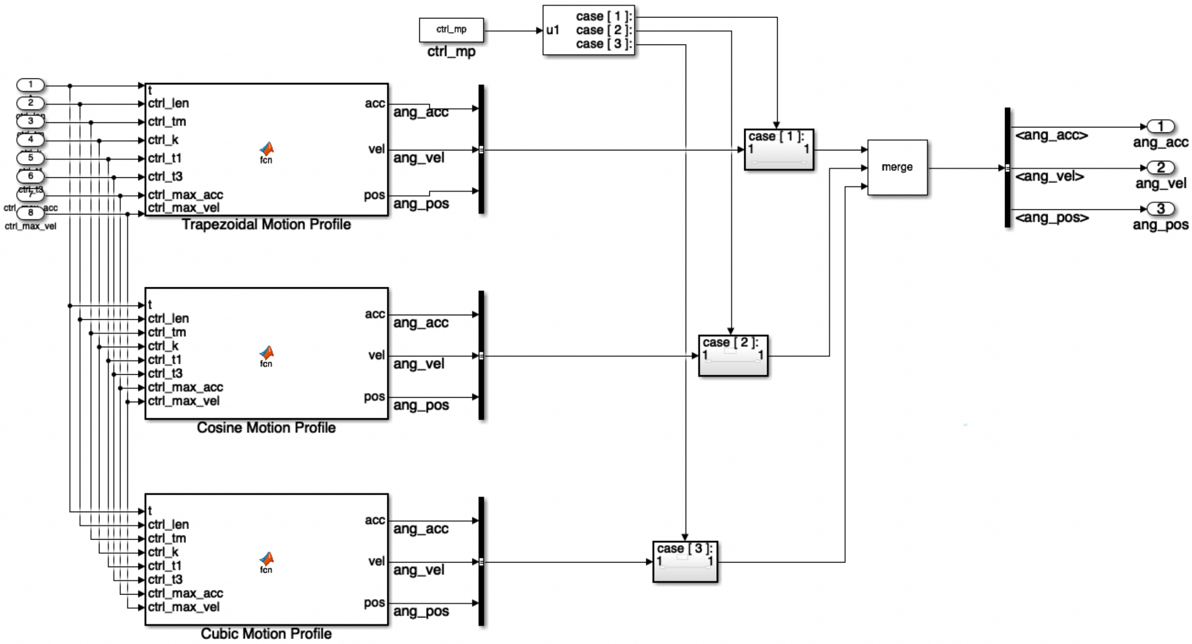

Figure 11 Motion Profile Selector in Simulink

The motion profile functions require the following inputs:

- Total time
- Slew rate
- Acceleration time
- Deceleration time
- Maximum velocity

The outputs of the motion profile functions are angular position, angular velocity and angular acceleration.

To link the motion profile to the control of the motor, a PID (Proportional-Integral-Derivative) controller is used.

Given a reference point, a PID controller applies a corrective signal to an input signal – with the aim of making the input signal reach the reference signal.

In this case, the reference signal or desired value is the velocity calculated by the motion profile function.

The input signal (feedback) to be modified is the current velocity of the load.

The PID controller applies a corrective signal based on the error (motion profile velocity – current velocity).

The corrective signal is used to control the motor by adjusting the input voltage, since the applied motor torque controls the velocity of the load. That is, the greater the torque, the faster the load moves.

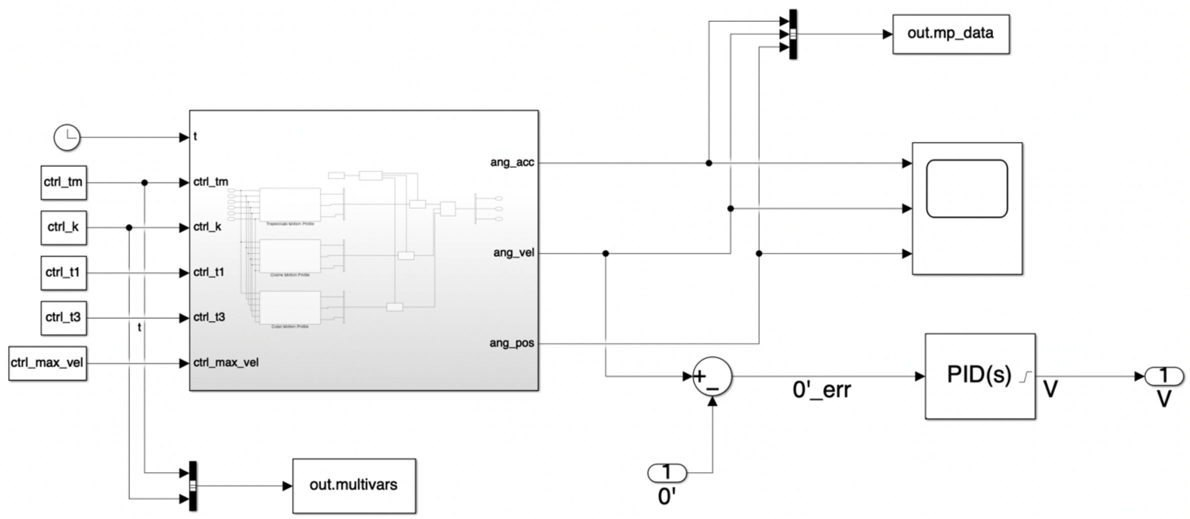

Figure 12 Motion Profile Controller

2.3.3 Motion Profile Verification

In order to verify the implementation of the motion profile controller, the following scenario was given: *moving a 25kg load through 30 radians in 10 seconds with k = 0.55*
With such a simple case, the only inputs needed to verify the system are the total time taken, total distance travelled and the slew rate. The trapezoidal motion profile was used.

The following graphs are produced:

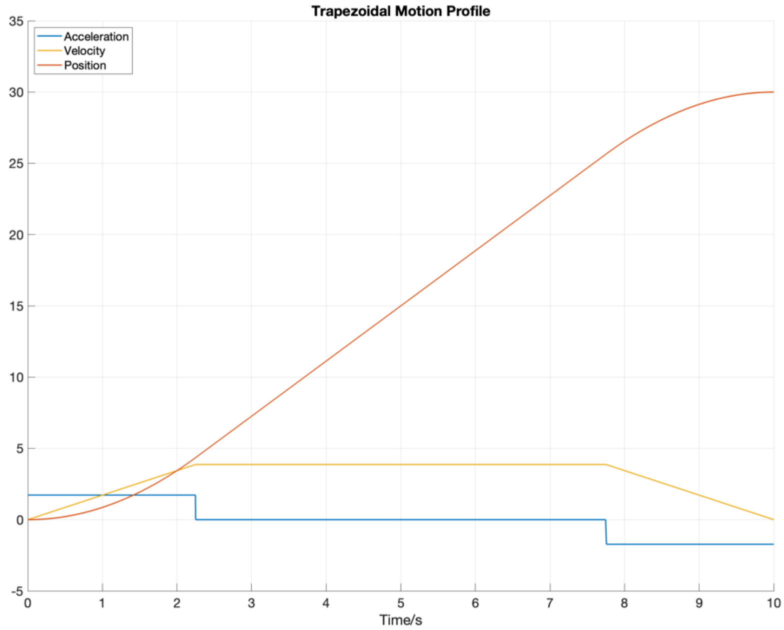

Figure 13 Generated Motion Profile

The graph in Figure 13 shows the motion profile generated by the trapezoidal function. This is the ideal trajectory that the motor will attempt to achieve.

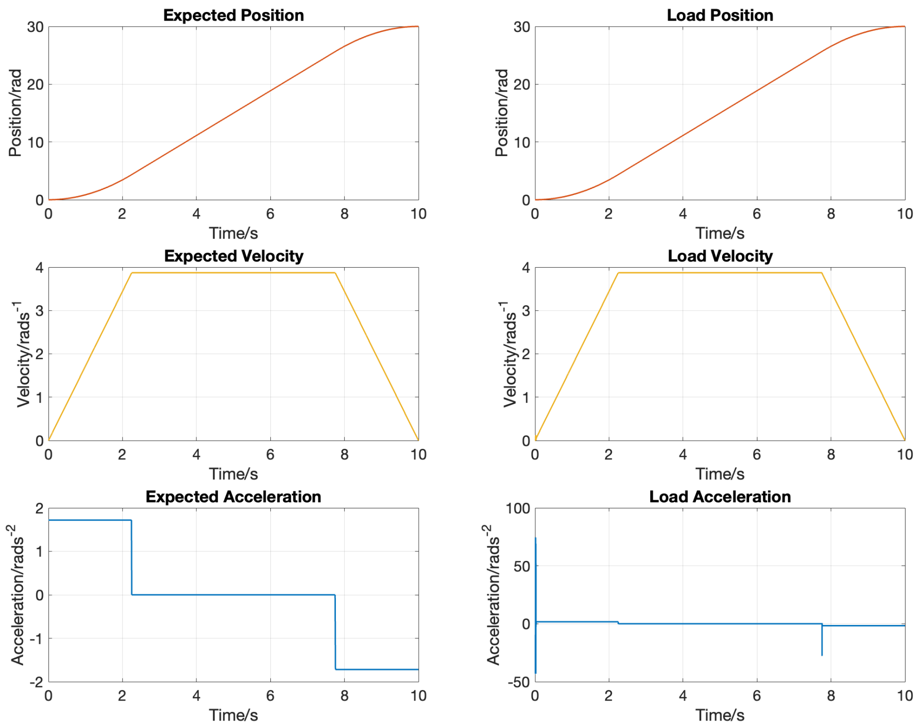

Figure 14 Comparison of Expected Graphs to Load Graphs

The graphs in Figure 14 show the expected graphs on the left and the actual graphs produced by the load during simulation. The difference between the velocity and position graphs cannot be seen from this scale. However, there is a significant difference in the acceleration graphs.

There is the spike at the beginning of the motion where $t = 0$, and there is another spike in the deceleration phase. The initial spike is due to the motor having 'infinite torque' when it first starts with zero velocity. The PID controller does not handle these instantaneous changes very well, also observed in the second spike. This occurs for less than 0.028 seconds.

The position and velocity are not affected as much, as it is impossible to reach infinite velocity and position within the confines of the physical conveyor in such a small amount of time.

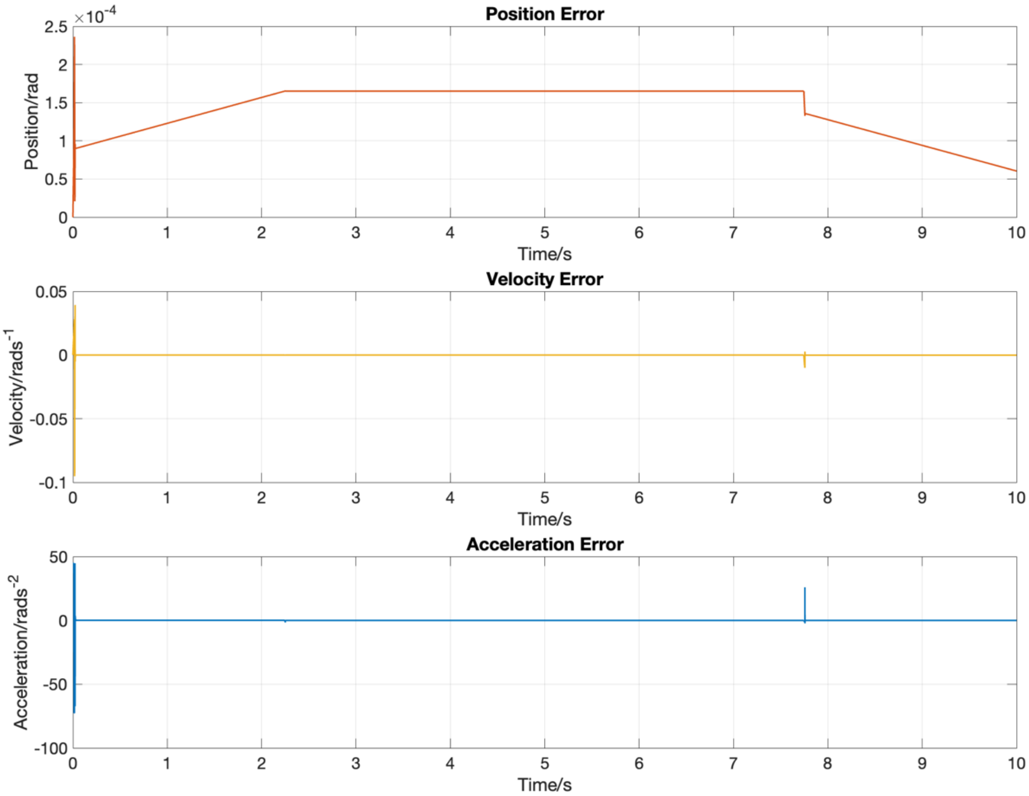

Figure 15 Load Motion Error Graphs

The graphs in Figure 15 show the error between the expected and actual values. The acceleration has large errors due to the reasons mentioned above while the error in the velocity and position are small enough to be considered insignificant.

The graphs in Figure 16 show the motor current and torque. The current and torque also suffer from the same problem as the acceleration. The proportional relationship between the current, torque and acceleration is seen in them having similarly shaped graphs.

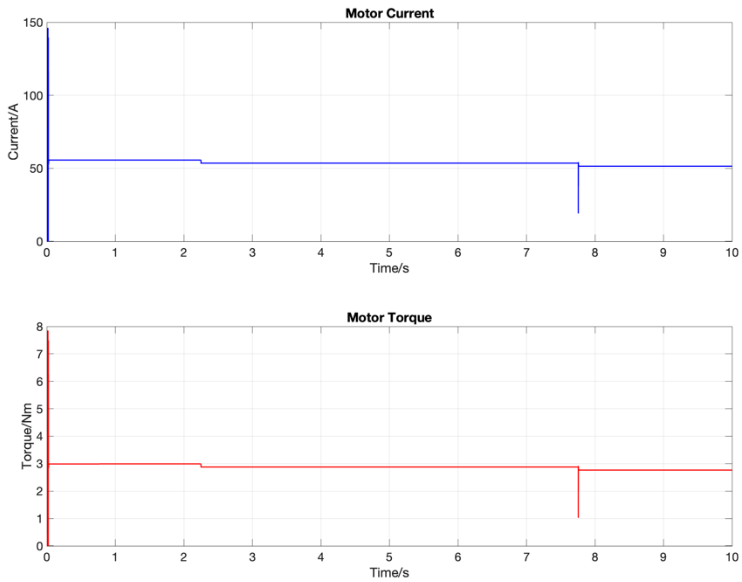

Figure 16 Motor Current and Torque Graphs

From the results obtained, specifically the error graphs, it is sufficient to conclude that the simulation model is verified and accurate.

3 Energy Parameterisation and Optimisation

3.1 Simple Energy Parameterisation

In simulation, the input power provided to the electric motor can be calculated using the following formula:

$$P = V * i$$

Where P is the input power (W), V is the input voltage (V) and i is the input current (A). The energy consumed, in Joules, by the electric motor can be calculated by integrating the input power during the simulation. This allows for energy measurements to be recorded in Simulink.

$$E = \int_{tp}^{tn} P \, dt$$

The energy is calculated by an integral block at each time step (denoted by tn and tp).

Outside of simulation, the power and energy equations can be expanded to be used for optimisation purposes. The energy measurements recorded in Simulink are independent of the motion profile and will be used to verify optimisation results.
It is possible to derive energy equations that are tailored to specific motion profiles. Doing so allows optimisation techniques to be applied to the energy equations as objective functions.

Using the simplest of the motion profiles, the trapezoidal profile, energy equations can be derived by integrating the power expression over the three main time periods of the motion (acceleration, constant velocity and deceleration periods) as seen in Figure 17.

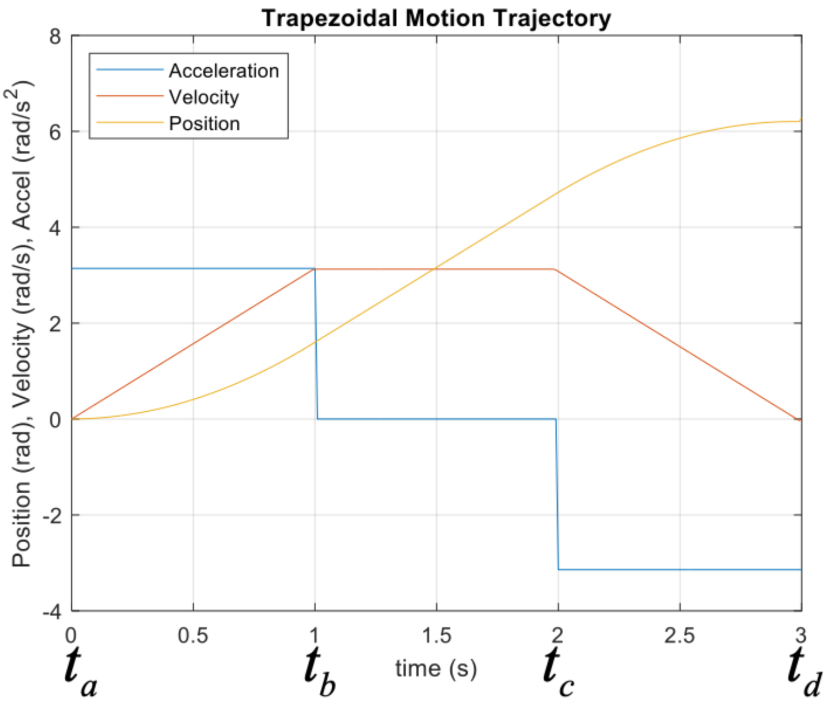

Figure 17 Trapezoidal Motion Profile Graphs

The power equation can be rewritten using Equation 2 and Equation 3:

$$P = Vi = Li\frac{di}{dt} + i^2R + iK_e\dot{\theta}$$

$$P = \frac{T_m L}{K_e{}^2}\frac{d(T_m)}{dt} + \frac{T_m{}^2 R}{K_e{}^2} + T_m\dot{\theta}$$

The T_m term can be expanded as seen in Equation 1,

$$P = C\dot{\theta}\dddot{\theta} + W\dddot{\theta} + S + P\dot{\theta} + M\ddot{\theta} + H\dot{\theta}\ddot{\theta} + G\dot{\theta}^2 + E\ddot{\theta}^2$$

However, the acceleration for a trapezoidal motion profile is a constant value as seen in Figure 9. At the acceleration and deceleration phases, the jerk is 0. Jerk is present when the acceleration changes, but this case is not considered as the jerk reaches infinity for the trapezoidal motion profile; $\dddot{\theta} \rightarrow \pm\infty$.

When analysing the energy, the integral of jerk at points of interest will be zero. Therefore,

$$P = S + P\dot{\theta} + M\ddot{\theta} + H\dot{\theta}\ddot{\theta} + G\dot{\theta}^2 + E\ddot{\theta}^2 \tag{5}$$

Equation 5 Trapezoidal Power Equation

Where S, P, M, H, G, E are coefficients derived from multiple combinations of the physical conveyor and motor parameters.

The energy is given the sum of integrals of power across the different time frames of the motion,

$$E = \int_{t_a}^{T} P\,dt$$

$$E = \int_{t_a}^{t_b} P\,dt + \int_{t_b}^{t_c} P\,dt + \int_{t_c}^{T} P\,dt$$

This evaluates to

$$E = X_1 T + X_2 l + X_3 \left(\frac{4l^2 \left(T - \frac{2}{3}t_1 - \frac{2}{3}t_3\right)}{(2T - t_1 - t_3)^2} \right) + X_4 \left(\frac{4l^2 (t_1 + t_3)}{(2T - t_1 - t_3)^2 t_1 t_3} \right) \tag{6}$$

Equation 6 Trapezoidal Energy Equation

Where l is the length, T is the total time taken (t_m).

In terms of the slew rate k, the energy equation can be rewritten as

$$E = X_1 T + X_2 l + \left(\frac{2l}{(T + kT)} \right)^2 \left[\frac{X_3}{3}(T + 2kT) + X_4 \cdot \frac{4}{(T - kT)} \right] \tag{7}$$

Equation 7 Trapezoidal Energy Equation in terms of k

Where:

$$X_1 = \frac{D^2 R}{{K_e}^2} \qquad X_2 = \frac{2BDR}{{K_e}^2} + D \qquad X_3 = \frac{B^2 R}{{K_e}^2} + B \qquad X_1 = \frac{A^2 R + ABL}{{K_e}^2}$$

The coefficients A, B, D are derived in Equation 1 and R, L, K_e represent the resistance, inductance and torque constant of the motor as seen in Equation 2.

3.2 Simple Energy Optimisation

To verify the energy model and quantify energy savings, a suitable parameter should be chosen to optimise the energy against. The slew rate k is suitable as it is related to the total time taken, the acceleration time and the deceleration time. Using Equation 7 and the initial example scenario, the value of k can be varied to prove that the energy consumption is indeed optimisable.

(Initial scenario: *moving a 25kg load through 30 radians in 10 seconds with k = 0.55*)

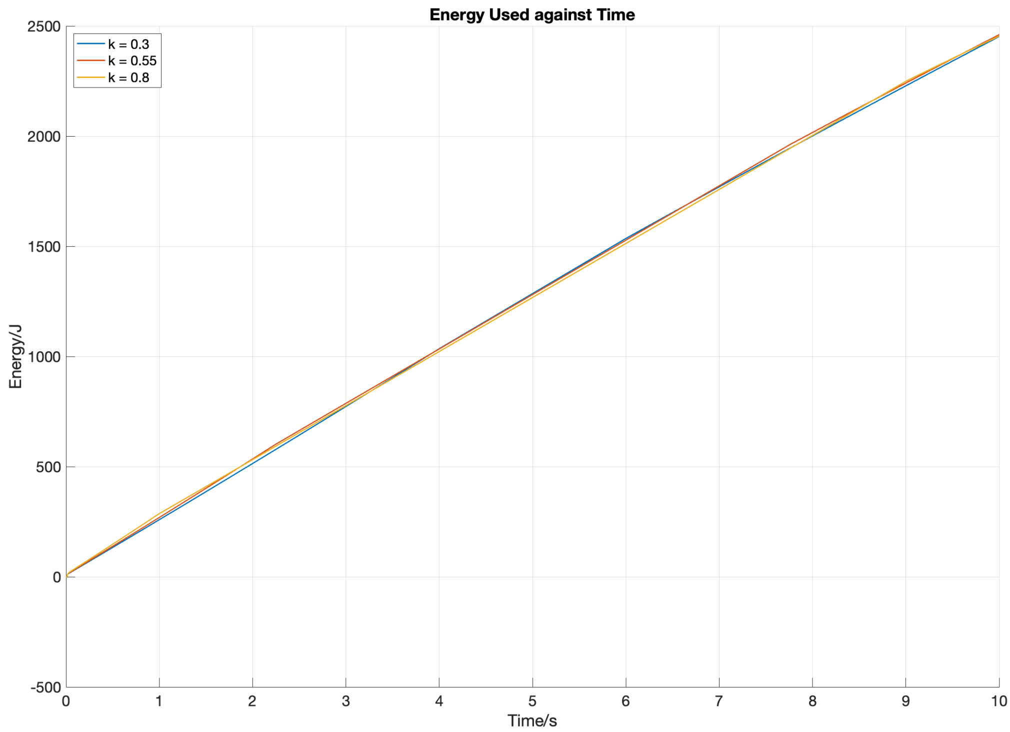

Figure 18 Simulated Energy Usage against Time with Different Slew Rates

The graph in Figure 18 shows the energy consumed by the motor with different slew rates being set for the motion profile.

The results are shown in Table 3:

k	**Energy / J**
0.3	2453.21
0.55	2460.29
0.8	2453.91

Table 3 Energy Usage for Different Slew Rates

Using the full range of the slew rate k, a graph of calculated energy against slew rate can be obtained as seen in Figure 19.

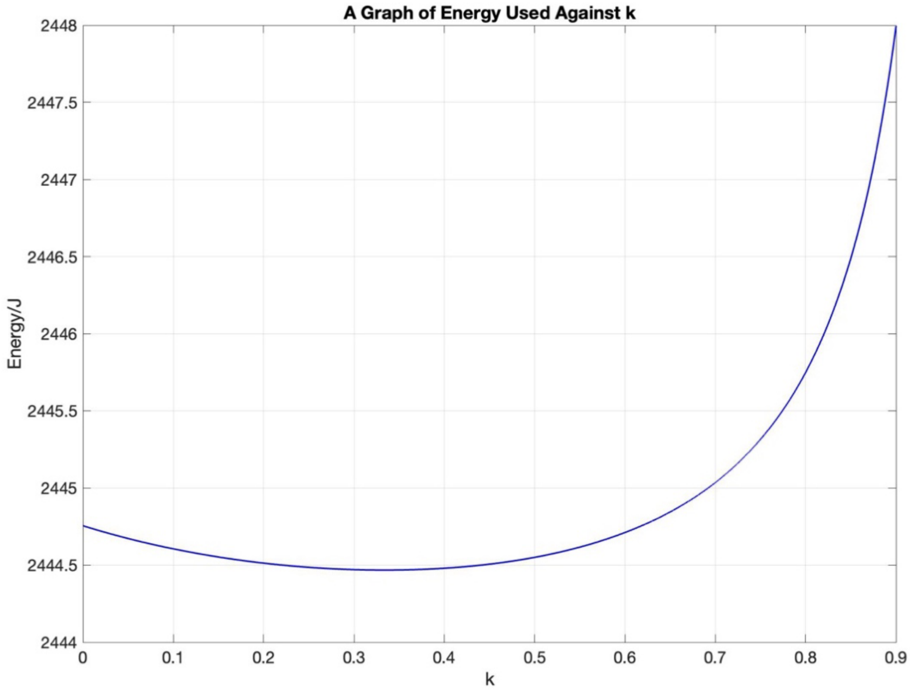

Figure 19 Energy against Slew Rate

From the graph in Figure 19, the minimum value of the curve in this scenario occurs around $k = 0.34$

To further verify this observation, mathematically:

$$\frac{dE}{dk} = \frac{8l^2T^2}{(T+Tk)^3}\left(\frac{2X_4(3k-1)}{(T-Tk)^2} - \frac{X_3k}{3}\right)$$

The turning points occur where $\frac{dE}{dk} = 0$,

$$\frac{8l^2T^2}{(T+Tk)^3}\left(\frac{2X_4(3k-1)}{(T-Tk)^2} - \frac{X_3k}{3}\right) = 0$$

$$(X_3T^2)k^3 - (2X_3T^2)k^2 + (X_3T^2 - 18X_4)k + 6X_4 = 0$$

The roots of the cubic equation (in k) provide optimal k values for the given trajectory.
For this example, the MATLAB command $polyvals$ produces the following solution:

$$\mathrm{k_{opt}} = [15.4181 \quad -13.7521 \quad 0.3340]$$
$$but, \qquad 0 \leq k < 1$$
$$therefore, \qquad k = 0.3340$$

The optimal k value, $\mathrm{k_{opt}}$, in this case is 0.3339; the other two values are eliminated as k must be greater than or equal to 0 and less than 1 (the *fraction* of the total time for which the velocity is constant).
$\mathrm{k_{opt}}$ can be used in the motion profile equations for time periods shown in Figure 10; the resulting durations are the optimal time periods ($t_{a_{opt}}, t_{b_{opt}}, t_{c_{opt}}, t_{d_{opt}}$).
However, it should be noted that $t_{d_{opt}} = t_m = T$ which is the total time taken for the motion (10s in this scenario).

By substituting $\mathrm{k_{opt}}$ into Equation 7, the optimal energy consumption can be obtained.

$$E_{opt} = 2444.57\,J$$

Using this $\mathrm{k_{opt}}$ value in the simulation, the energy consumption measured is

$$E_{opt_{sim}} = 2452.59\,J$$

The optimal energy value from the simulation is slightly higher than the optimal energy value from the calculations. This is a constant (but unpredictable) error for all simulation runs and it arises from the use of the PID controller; such errors will occur in real-life usage too. The error does not invalidate the solution.
From the simulated energy graph in **Error! Reference source not found.**, the lowest energy consumption occurs when

$$k = 0.3, \qquad E = 2453.21\,J$$

However,

$$k_{opt} = 0.3339 \quad E_{opt_{sim}} = 2452.59\,J$$

Using $\mathrm{k_{opt}}$, there is a decrease of 0.03% in energy usage.

Though this is a very small improvement, industrial machines such as conveyor belts run repetitively for long hours. Small energy savings accumulate over time into a significant amount.

The energy improvements will also vary greatly depending on the load and gear ratios. However, further improvements can still be made. The technique used enforces that $t_1 = t_3$, changing this restriction might improve the energy usage.

This method paves the way for implementing more efficient and suitable motion profiles; an issue with the trapezoidal motion profile is that at certain points infinite jerk is observed. This method also shows that the problem can be improved to account for other optimisation parameters such as gear ratios and efficiency.

3.3 Multivariable Optimisation Using a Genetic Algorithm

3.3.1 Multivariable Problems

The results obtained by simple optimisation using a single variable (slew rate) proved that the system is indeed optimisable. To further increase complexity and realism, multiple variables can be included in the optimisation task.

One disadvantage of using only the slew rate is that the system assumes the acceleration and deceleration of the load are the same. Therefore, the acceleration and deceleration times are also the same.

Such assumptions are not always possible in real-life scenarios, and the optimal energy solution may reside outside of these assumptions. These assumptions just helped simplify the system to prove that energy optimisation was feasible.

There are four variables which form part of the objective function:

- The total time taken to complete the trajectory, t_m
- The acceleration time, t_1
- The deceleration time, t_3
- The maximum velocity, V_m

The length of the trajectory must always be provided for the optimisation problem.

The maximum acceleration and deceleration can either be provided before optimisation begins or can be calculated after the optimisation finds the optimal acceleration and deceleration times.
The slew rate, k, can also be given beforehand or calculated after the optimal time values have been obtained.

Optimisation is set up in one of many configurations. A configuration being a combination of the objective variables, where some predefined values are used as constraints to the problem. For example, a configuration might be given as:

$$\textit{Find}:\ t_m,\ \ t_1,\ t_3,\ V_m$$

$$\textit{Given:}\ a_1,\ \ a_3,\ \ k$$

$$\textit{Where:}\ len = 9\pi\ rad\ and\ t_1 < 25s$$

In this configuration, the optimisation task is to find the optimal values of the four main variables given the maximum acceleration, a_1, the maximum deceleration, a_3, and the slew rate k. The given quantities are used to form constraints for the problem. Where constraints are mathematical equations involving limits, equalities and inequalities (linear and non-linear). The length, len, is a prerequisite for any problem.
The example configuration given is a rather easy problem to optimise, considering the number of constraints given.

3.3.2 Multivariable Objective Functions

Multivariable objective functions can be derived from the motion profiles by hand, or by using the MATLAB Symbolic Math Toolbox (Mathworks, 2023). The original energy and power expressions in Equation 5 and Equation 6 were derived by hand, but differentiation and integration become more complicated when involving multivariable equations (requires partial derivatives and integrals) and more complex motion profile equations. However, obtaining the original trapezoidal equations helped to verify the feasibility of the problem.

Using the MATLAB Symbolic Math Toolbox, the trapezoidal energy equation in terms of all motion profile time frames, the velocity, the acceleration, the deceleration, the total time taken and every physical parameter of the conveyor system is given as:

$$
\begin{gathered}
E = (6\,\mathrm{D}^2\,R\,\mathrm{tm} - 6\,\mathrm{D}^2\,R\,\mathrm{ta} - 3\,A_1\,\mathrm{D}\,\mathrm{Ke}^2\,{t_1}^2 - 3\,A_2\,\mathrm{D}\,\mathrm{Ke}^2\,{t_3}^2 - 6\,A_1\,\mathrm{D}\,\mathrm{Ke}^2\,\mathrm{ta}^2 + 6\,A^2\,{A_1}^2\,R\,t_1 + 6\,A^2\,{A_2}^2\,R\,t_3 \\
-12\,A^2\,{A_1}^2\,R\,\mathrm{ta} + 3\,A\,{A_1}^2\,\mathrm{Ke}^2\,{t_1}^2 + 3\,A\,{A_2}^2\,\mathrm{Ke}^2\,{t_3}^2 - 4\,{A_1}^2\,B\,\mathrm{Ke}^2\,{t_1}^3 + 2\,{A_2}^2\,B\,\mathrm{Ke}^2\,{t_3}^3 + 2\,{A_1}^2\,B\,\mathrm{Ke}^2\,\mathrm{ta}^3 \\
+3\,{A_1}^2\,B^2\,L\,{t_1}^2 + 3\,{A_2}^2\,B^2\,L\,{t_3}^2 - 4\,{A_1}^2\,B^2\,R\,{t_1}^3 + 2\,{A_2}^2\,B^2\,R\,{t_3}^3 + 2\,{A_1}^2\,B^2\,R\,\mathrm{ta}^3 - 12\,{A_1}^2\,B\,\mathrm{Ke}^2\,t_1\,\mathrm{ta}^2 + 12\,{A_1}^2\,B\,\mathrm{Ke}^2\,{t_1}^2\,\mathrm{ta} \\
+6\,{A_1}^2\,B\,\mathrm{Ke}^2\,{t_1}^2\,\mathrm{tm} + 6\,{A_1}^2\,B\,\mathrm{Ke}^2\,\mathrm{ta}^2\,\mathrm{tm} - 12\,{A_1}^2\,B^2\,R\,t_1\,\mathrm{ta}^2 + 12\,{A_1}^2\,B^2\,R\,{t_1}^2\,\mathrm{ta} + 6\,{A_1}^2\,B^2\,R\,{t_1}^2\,\mathrm{tm} + 6\,{A_1}^2\,B^2\,R\,\mathrm{ta}^2\,\mathrm{tm} \\
+6\,A\,{A_1}^2\,B\,L\,t_1 + 6\,A\,{A_2}^2\,B\,L\,t_3 - 12\,A\,{A_1}^2\,B\,L\,\mathrm{ta} - 6\,A_1\,B\,\mathrm{D}\,R\,{t_1}^2 - 6\,A_2\,B\,\mathrm{D}\,R\,{t_3}^2 - 12\,A_1\,B\,\mathrm{D}\,R\,\mathrm{ta}^2 + 6\,A_1\,\mathrm{D}\,\mathrm{Ke}^2\,t_1\,\mathrm{ta} \\
+6\,A_1\,\mathrm{D}\,\mathrm{Ke}^2\,t_1\,\mathrm{tm} - 6\,A_1\,\mathrm{D}\,\mathrm{Ke}^2\,\mathrm{ta}\,\mathrm{tm} + 6\,A\,{A_1}^2\,B\,R\,{t_1}^2 + 6\,A\,{A_2}^2\,B\,R\,{t_3}^2 - 6\,A\,{A_1}^2\,\mathrm{Ke}^2\,t_1\,\mathrm{ta} - 6\,{A_1}^2\,B^2\,L\,t_1\,\mathrm{ta} + 6\,A_1\,B\,\mathrm{D}\,L\,t_1 - 6\,A_2\,B\,\mathrm{D}\,L\,t_3 \\
-12\,A_1\,B\,\mathrm{D}\,L\,\mathrm{ta} + 12\,A\,A_1\,\mathrm{D}\,R\,t_1 - 12\,A\,A_2\,\mathrm{D}\,R\,t_3 - 24\,A\,A_1\,\mathrm{D}\,R\,\mathrm{ta} - 6\,A_1\,A_2\,B\,\mathrm{Ke}^2\,t_1\,{t_3}^2 + 6\,A_1\,A_2\,B\,\mathrm{Ke}^2\,{t_3}^2\,\mathrm{ta} - 6\,A_1\,A_2\,B^2\,R\,t_1\,{t_3}^2 \\
+6\,A_1\,A_2\,B^2\,R\,{t_3}^2\,\mathrm{ta} - 12\,{A_1}^2\,B\,\mathrm{Ke}^2\,t_1\,\mathrm{ta}\,\mathrm{tm} - 12\,{A_1}^2\,B^2\,R\,t_1\,\mathrm{ta}\,\mathrm{tm} + 12\,A_1\,B\,\mathrm{D}\,R\,t_1\,\mathrm{ta} + 12\,A_1\,B\,\mathrm{D}\,R\,t_1\,\mathrm{tm} - 12\,A_1\,B\,\mathrm{D}\,R\,\mathrm{ta}\,\mathrm{tm} - 6\,A\,A_1\,A_2\,\mathrm{Ke}^2\,t_1\,t_3 \\
+6\,A\,A_1\,A_2\,\mathrm{Ke}^2\,t_3\,\mathrm{ta} - 6\,A_1\,A_2\,B^2\,L\,t_1\,t_3 + 6\,A_1\,A_2\,B^2\,L\,t_3\,\mathrm{ta} - 12\,A\,{A_1}^2\,B\,R\,t_1\,\mathrm{ta} - 12\,A\,A_1\,A_2\,B\,R\,t_1\,t_3 + 12\,A\,A_1\,A_2\,B\,R\,t_3\,\mathrm{ta})/\mathrm{Ke}^2\,6
\end{gathered}
$$

Figure 20 Multivariable Trapezoidal Energy Equation

The expanded energy equation is mathematically equivalent to the derived equation seen in Equation 6.

Energy equations become more complex for advanced motion profiles like the cosine and cubic profiles.

The energy function can be written in the objective function format for MATLAB:

$$funE = @(x)(E);$$

In the energy equation E, the total time taken t_m, the acceleration time t_1, the deceleration time t_3 and the maximum velocity V_m are replaced by $x(1), x(2), x(3), x(4)$ respectively. This format informs the optimisation function that the optimisable variables are elements of the array x.

With a problem configured and ready to be optimised, the next step is to choose a suitable optimisation technique.

A suitable optimisation technique for multivariable problems is optimisation using a genetic algorithm. (Katoch et al, 2021)

3.3.3 Genetic Algorithm Theory

A genetic algorithm is a population-based optimisation technique which follows the principle of natural selection in evolution (survival of the fittest). In this process, the algorithm evolves to find the best solution. (Katoch et al, 2021)

The algorithm goes through iterations, otherwise known as generations, where a new set of solutions are produced. Ideally, the solution produced in one generation will be better than the solution produced in the previous generation. It is this evolutionary process which gives rise to the best solution.

Genetic algorithms are used to solve multivariable problems which involve finding the global minimum or maximum.
The solution at each generation is an assignment of values to each of the variables. The problem will also present a set of constraints and boundaries for the variables in the form of equalities, inequalities and limits.
At each generation, the algorithm attempts to find solutions which satisfies the constraints and simultaneously finds the global minimum or maximum (depends on the problem).
The solution improves across consecutive generations, as the newer solutions tend to move closer to the optimal values within the constraints.

Genetic algorithms have many properties and elements involved in its operation. Some of the main properties include initial population, fitness evaluation, selection, crossover and mutation.

Initialisation
The genetic algorithm starts by initialising some random population. The creation of the random population is commonly done by using a gaussian distribution with the constraints if possible. The operator may also provide a starting point to the algorithm. Providing a starting point may be useful in the case where some information about the final solution is known, allowing for the algorithm to find it faster.

Selection and Fitness Values
At each generation, the fittest members are chosen from the population to reproduce children for the next generation.
In order to find the fittest members, a fitness function is used to assign fitness values to the population. The choice of fitness function depends on the problem as a whole, but the underlying principle is that the fitness of a member is related to its closeness to the optimum and constraint satisfaction. For example, it can be assumed that a member with a numerically smaller solution is fitter than one with a larger solution (for a minimisation problem).
This is a simplification of the fitness function; they can be more complex.
Selection can be based on techniques such as rank selection, tournament selection and roulette selection. For example, rank selection uses the fitness values to rank the members, increasing the chances of higher ranked members to be chosen. (Mirjalili, 2019)
The selection stage chooses individuals who will become the parents of the next generation.

A common reproduction strategy is crossover.

Crossover

Crossover is a reproduction technique which works by swapping genes of parents to produce children. Genes refer to individual values of variables within a solution. Therefore, a set of genes of an individual comprises its solution.

In crossover, the genes of the parents are split and exchanged according to some specific rule. This might be a half split, where half of the genes are split and exchanged. It may even be for every odd or even numbered gene, or a stochastic assignment might be used.

The result of crossover in a generation is a new generation of children with a combination of genes from their parents.

Crossover happens at a rate known as the crossover rate. This controls how often crossover should happen in conjunction with parents surviving to the next generation. (Mirjalili, 2019)

Mutation

Mutation is a diversity technique which ensures that the population explores a large set of solutions. During mutation, a random gene from a random individual is changed to a random value. In other words, a random variable from a solution is changed to a random value.

This ensures that the population becomes more diverse, avoiding being trapped in a local minimum.

Mutation can be done using different methods such as linear and gaussian mutation.

Finalisation

At this point, the algorithm would have produced many generations to the extent where the solution no longer improves. The fitness value of the fittest individuals across subsequent generations does not change much, and the constraints are satisfied.

This is usually a good stopping criterion; however, it depends on the operator and the problem at hand.

Assuming fitness value and constraint satisfaction are the criteria, the algorithm halts and returns the values of the fittest individual as the final solution.

The algorithm can be ran multiple times to ensure that the global optimum has been found.

3.3.4 Hybrid Genetic Algorithm

Genetic algorithms are good at finding the region of the global minimum. They usually find a general region, or a set of valleys/peaks (local optima) where the global optimum is likely to reside.

However, due to the mechanics of the genetic algorithm (involving population size), it may take many generations to converge at the exact location of the global optimum. (Katoch et al, 2021)

One solution to this problem is to use a hybrid genetic algorithm. The hybrid function is usually a local optimisation method such as gradient descent.

The hybrid function can be set to activate after the genetic algorithm stalls for several generations. That is, where there is no significant change in the value of the best fitness function across a defined number of generations.

At this stage, it can be assumed that the algorithm is converging towards the optimum, and the local optimisation method can quickly find its location.

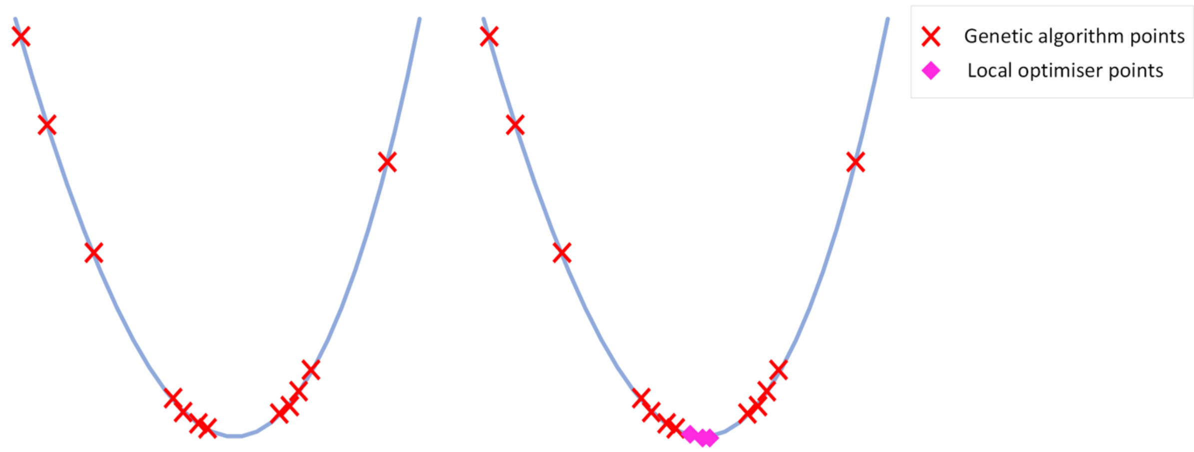

Figure 21 Simple Illustration Showing the Hybrid Genetic Algorithm

In Figure 21, the local optimiser quickly converges on the solution after the genetic algorithm terminates. This is a simplified illustration of how the hybrid genetic algorithm works; the search space becomes more complex as the number of variables increases.

For certain problems, the local optimiser may not find a suitable optimal solution within the constraints. In such cases, the solution found by the genetic algorithm is the optimal solution.

3.3.5 MATLAB Implementation

To begin the optimisation process, the following information must be provided to the optimiser:

- An objective function – The characteristic energy of the selected motion profile (either trapezoidal, cosine or cubic).
- An optimisation configuration – This describes the problem in terms of which variables to be optimised and which values have already been defined.
- Constraints, limits and boundaries – These are obtained from the chosen configuration. In the case where a hard limit is not provided for a variable, the default value of *50* units is used.
- Optimisation options – These are MATLAB specific options which determine how the genetic algorithm is deployed (maximum number of generations, selection method, mutation method etc.,).

The output of the optimiser are the optimal values of energy used, acceleration time, deceleration time and max velocity.

ga Function

The genetic algorithm is implemented using MATLAB's ga function from the Global Optimisation Toolbox (Mathworks, 2023). The function syntax is as follows

$$x = ga(fun, nvars, A, b, Aeq, beq, lb, ub, nonlcon, options)$$

Whereby

- fun is the objective function and $nvars$ is the number of variables to be optimised
- (A, b) and (Aeq, beq) represent linear inequalities and linear equalities respectively
- lb and ub represent the lower and upper bounds
- $nonlcon$ defines non-linear constraints
- $options$ contain the optimisation options in the $optimoptions$ format

The function returns the values of the optimal solution in an array, x.

ga options

The $options$ argument includes many tuneable parameters for the algorithm. Selecting the right combination of parameters is usually a difficult but crucial task, usually involving trial and error. The relevant parameters to the developed system, and their values, are listed below in Table 4:

Parameter	Value	Description
SelectionFcn	$selectionstochunif$	Chooses parents for the next generation. $selectionstochunif$ selects parents by moving along a line where distance between points represent scaled fitness values of the parents.
MutationFcn	$mutationadaptfeasible$	Chooses how mutation occurs (random changes). $mutationadaptfeasible$ simply applies random changes as long as the results satisfy the constraints.
MaxStallGenerations	40	The number of generations for which the algorithm can run without seeing significant changes in the fitness function value. 10 times the number of variables works well for smaller problems.
MaxGenerations	500	The number of generations for which the algorithm can run. Should be set to at least 100 times the number of variables.
ConstraintTolerance	10^{-6}	The amount of error acceptable for a solution to satisfy the constraints. A tolerance significantly less than 1 is sufficient.
FunctionTolerance	10^{-14}	The maximum amount of change in the best fitness function which is classified as a significant change (used during stall generations).
OutputFcn	$gabestplotf1$	Output function which executes as the algorithm runs. $gabestplotf1$ is a customised version of $gabestplotf$, for compatibility with the final application. The function plots a real-time graph of best and mean fitness scores against the number of generations.

Table 4 ga Function Optimisation Options

Hybrid Function: fmincon

The MATLAB function, $fmincon$, is used in conjunction with the ga function for hybrid optimisation. $fmincon$ is part of MATLAB's Optimisation Toolbox (Mathworks, 2023). After the genetic algorithm terminates, $fmincon$, resumes from the last point and locates the local optimum, which will be the global optimum in this case.

A simplified explanation of how $fmincon$ works is that it iteratively generates and solves smaller sequences of optimisation problems, within the constrained search space, to find the local optimum. This is the *interior-point* method, and it is one of the algorithms which can be chosen in the $fmincon$ options. (Mathworks, 2023)

$fmincon$ also has similar options which can be tuned. In this case, *interior-point* algorithm is the default algorithm, the constraint tolerance is set to 10^{-7} and the optimality tolerance (similar to ga function tolerance) is set to 10^{-8}.

The hybrid genetic algorithm is declared below in Figure 22.

```
...
hybridopts = optimoptions(
'fmincon', 'Display', 'none', 'Algorithm', 'interior-point',
'OutputFcn', @optimplotfval1, EnableFeasibilityMode=true,
ConstraintTolerance=1e-7, OptimalityTolerance=1e-8);

options = optimoptions(
'ga', 'OutputFcn', @(options,state,flag)gaplotbestf1(options,state,flag),
SelectionFcn="selectionstochunif", MutationFcn="mutationadaptfeasible",
MaxStallGenerations=35, MaxGenerations=500,
ConstraintTolerance=1e-6, FunctionTolerance=1e-14);

[OPTXS,~,~,OUTPUT] = ga(funE, 4,[],[],[],[],lb,ub, nonlinfcn,options);
...
[OPTXS,FVAL,~,OUTPUT] = fmincon(funE,OPTXS,[],[],[],[],lb,ub,nonlinfcn,hybridopts);
...
```

Figure 22 Code Snippet of Hybrid Genetic Algorithm Instantiation

$funE$ represents the objective function, $OPTXS$ stores the values of the optimal solution after each function runs, $FVAL$ stores the final energy value returned by $fmincon$.

The $OPTXS$ returned by the genetic algorithm is the same $OPTXS$ that $fmincon$ starts from. $OUTPUT$ holds values which are used for logging data out for the user.

4 Testing, Results and Analysis

4.1 MATLAB Application Packaging

To provide a finished product, the findings of this project have been compiled into a single MATLAB application. The functionality of the application is listed below:

- Input parameters for the conveyor, motor and motion profile
- Input settings for optimisation
- Run optimisation within the application, showing the results and logs separate of the main MATLAB window
- Run Simulink simulations within the application, showing graphs and results outside of Simulink
- Save and load configurations for later usage and to make analysis quick and easy

The application is a cohesive unit which allows the operator to design systems without having to work with numerous scripts and simulations.
However, it should be noted that the application developed is only a prototype, though it also passes as a minimum viable product. It has not been developed to industry standards.

4.1.1 Application Overview

The application was designed and programmed using the MATLAB App Designer.
All configurable settings are stored as variables in the base MATLAB workspace; all calls to external paths are also made from the working MATLAB path.

Any processing (such as calculations and optimisations) is started from within the application but done externally in the main MATLAB process. Separating the intensive processes from the application itself helps to prevent stalling and crashing when working with large data (future-proof). Simulations are ran using the sim command, evaluated in the base workspace.

Conveyor, motor and motion profile configurations can be saved as MAT files which can be loaded again for later use.

The intended workflow for using the application is as follows:

1. Load configuration files or enter values for the conveyor, motor and motion profile settings.
2. Run the Calculator/Optimiser. The optimiser is only activated when the ‘Optimise Params’ setting is enabled (in the motion profile panel).
3. Inspect the output log.
4. Run the Simulation.
5. Analyse the graphs.

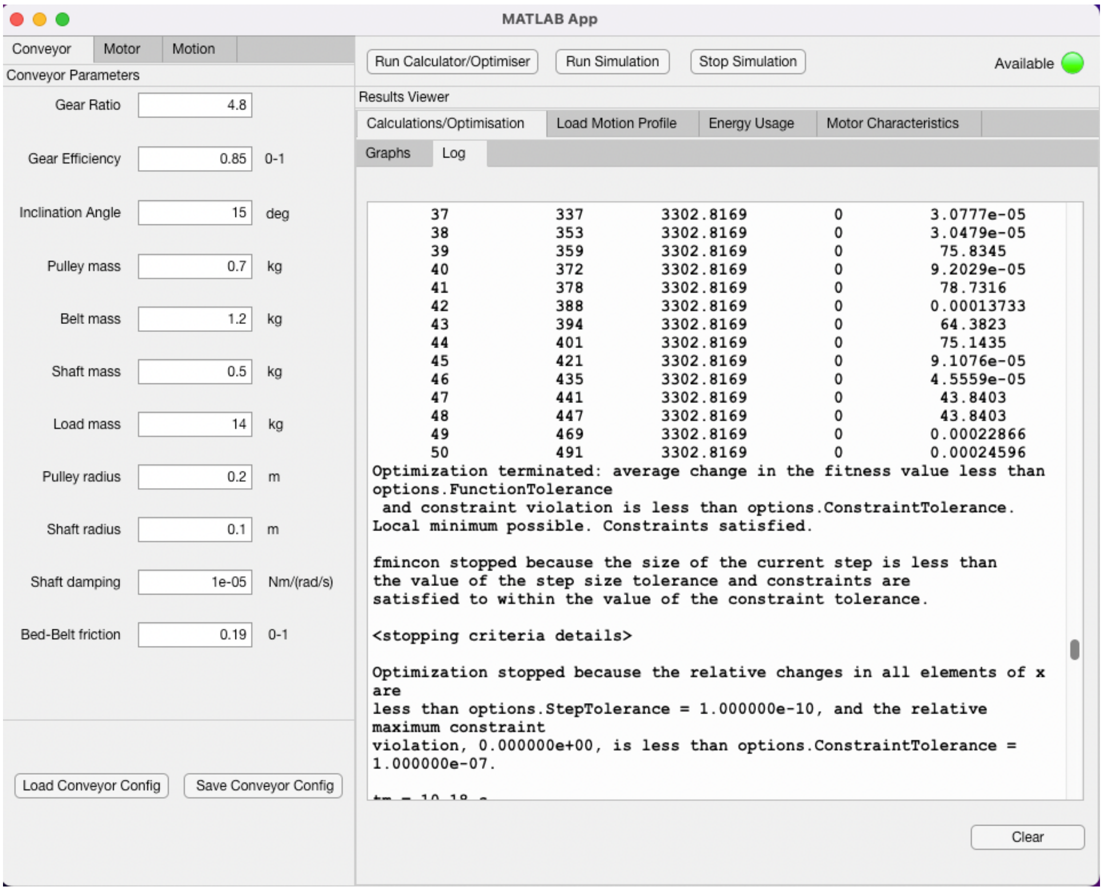

Figure 23 Snapshot of Prototype Application

4.2 Product Test Case

To verify the full product and the feasibility of the research carried out, an example scenario using real-life data from factories or manufacturers should be used.
However, the results of extensive searching online showed that manufacturers do not freely provide data about specific energy consumption of their conveyor belt systems in terms of how they are controlled. Some mining companies provide annual reports based on material flowrate; this is not suitable for the developed system under test.

Therefore, a fake scenario was created and used to verify the full product.

4.2.1 Briefing and Specifications

An Electricals and Electronics Manufacturer, E&E, for Heavy Goods Vehicles has a production line which includes a conveyor belt system.
The specifications of the current system are as follows:

Parameter	**Value**
Motor Used	Maxon RE 65
Motor Mode	OVERLOAD
Maximum Velocity	7.5 rad/s
Maximum Allowable Cycle Time	24 s
Minimum Allowable Cycle Time	12 s
Current Cycle Time	20 s
Gear Ratio	1:4.8
Gearbox Efficiency	85%
Bed-Belt Friction	0.19
Combined Load per Cycle	14 kg
Angle of Inclination	15° [Elevated]
Track Length	15 m
Pulley Radii	0.2 m
Shaft Radii	0.1 m
Belt Mass	1.2 kg
Pulley Mass	0.7 kg

Shaft Mass	0.5 kg
Shaft Damping	1e-05 Nm/(rad/s)
Type of Load	-

Table 5 E&E Conveyor Specification

As a Control Systems engineer, the objective is to minimise the energy consumption while meeting the requirements of the system. All changes are to be made in the programming of the system; the hardware is fixed. Assume the motor can operate in overload conditions.

4.2.2 Scenario Breakdown

From the briefing and the specifications in Table 5, the following observations can be made:

- The current system operates with a triangular motion profile. A triangular motion profile is just a modified trapezoidal profile with no constant velocity phase ($k = 0$). So, in this case, the load is accelerated to full speed and then immediately begins to decelerate. This rudimentary approach is easy to implement on a physical motor controller.
- The trajectory length in radians is given by dividing the track length in meters by the pulley radius. ($length = 0.2 * 14 = 75\ rad$)
- The cycle time range is $12s \leq t \leq 24s$
  This means the time taken for a product to be moved from the entry point to the end point should be within this range. The pick-and-place robot can be adjusted to operate within this time frame.
- The motor can operate comfortably in overload conditions. This may be due to the low operation speed.

With all the necessary parameters and a basic understanding of the current system, the Optimiser Application can be used to quantify the energy consumption.

4.2.3 Test and Results Analysis

Within the application, the conveyor and motor parameters are set according to the specifications given.

The relevant motion profile settings (boxed in red) for both the current and optimised systems are set as shown below in Figure 24:

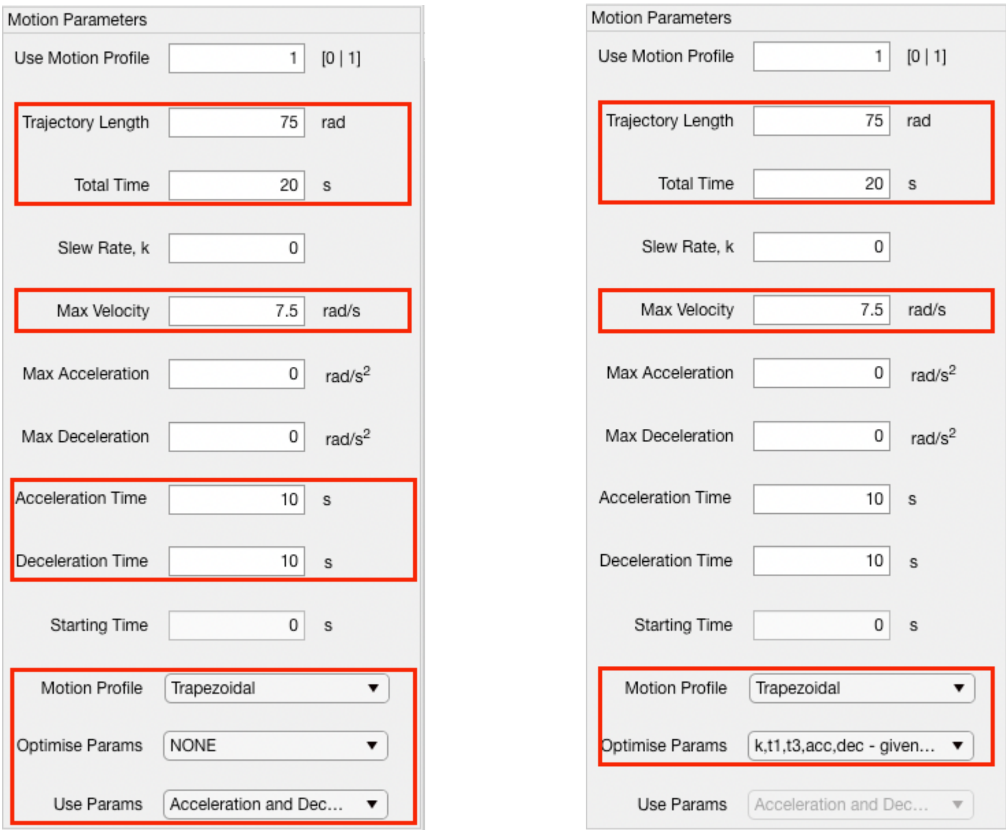

Figure 24 Motion Profile Settings for Current System (Left) and Optimised System (Right)

The results obtained are shown below in Table 6:

Parameter	**Current System**	**Optimised System**
Total time, $\boldsymbol{t_m}$	20.00 s	18.09 s
Acceleration time, $\boldsymbol{t_1}$	10.00 s	2.00 s
Deceleration time, $\boldsymbol{t_3}$	10.00 s	1.99 s
Max Velocity, $\boldsymbol{V_m}$	7.50 rad/s	4.66 rad/s
Slew Rate, $\boldsymbol{k}$	0.00	0.78
Energy Used, $\boldsymbol{E}$	6196.982 J	5629.613 J

Table 6 Results for The Current System and The Optimised System

As expected, the optimised system produces better results; a 9.2% decrease in energy consumption.

However, the scenario is not indicative of real-life conveyor operation. Furthermore, the current system was given a very simple and probably unrealistic motion profile. Despite this bias, the results prove the feasibility of the Optimiser Application and the research behind it.

5 Project Management

5.1 Brief Overview

An Iterative-Agile approach was used throughout the duration of the project since the workflow usually involved performing optimisations and backtracking to troubleshoot the original models. The fast iterative workflow was suitable, given that it was an individual project.

During the main development phase of the project, meeting where held regularly with the project supervisor to receive feedback. This emulates the interaction between the project managers and the stakeholders.

A Kanban-style priority list was used for progress tracking and task organisation. This usually involved having multiple lists where each was concerned with a different subsystem of the project. A simple note-taking application was used to implement this instead of established tools such as Trello, because it suited the individual's needs.

GitHub was used to manage and keep track of the MATLAB and Simulink code base. Updates occurred irregularly based on how much new (and seemingly important) changes were made. Ideally, updates should be done periodically.

5.2 Project Timeline

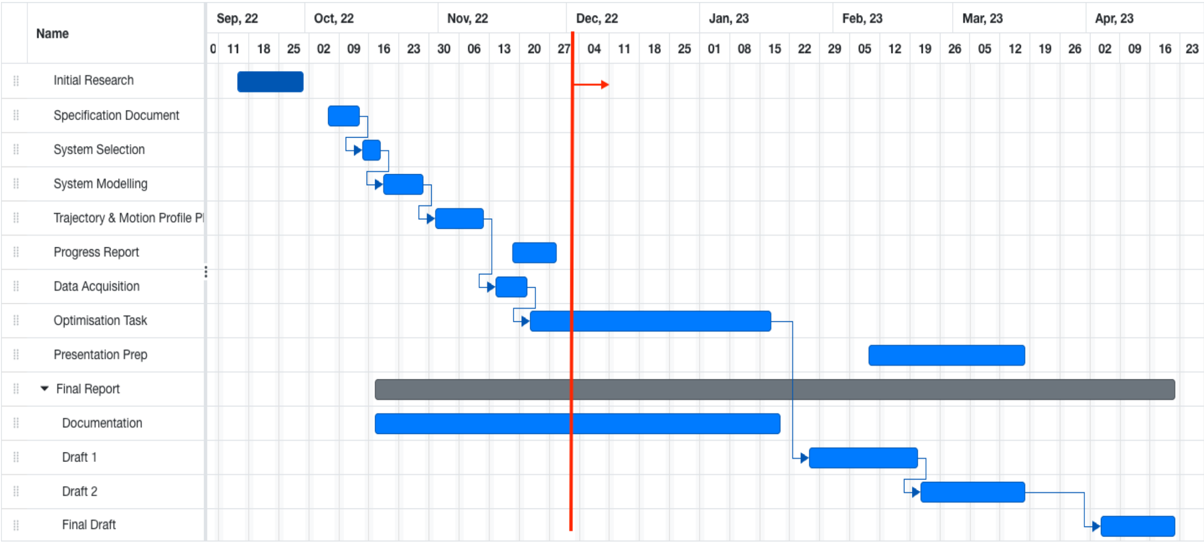

Figure 25 Initial Project Timeline

The project followed the initial timeline until midway through the optimisation task. The task was delayed by 3 weeks and an additional 2 weeks was needed to complete the work.

The inclusion of the application development (which was not originally planned) added an additional 2 weeks to the project.

In total, the project lagged by 5-6 weeks, proving to cause significant issues with time management during the final report stage.

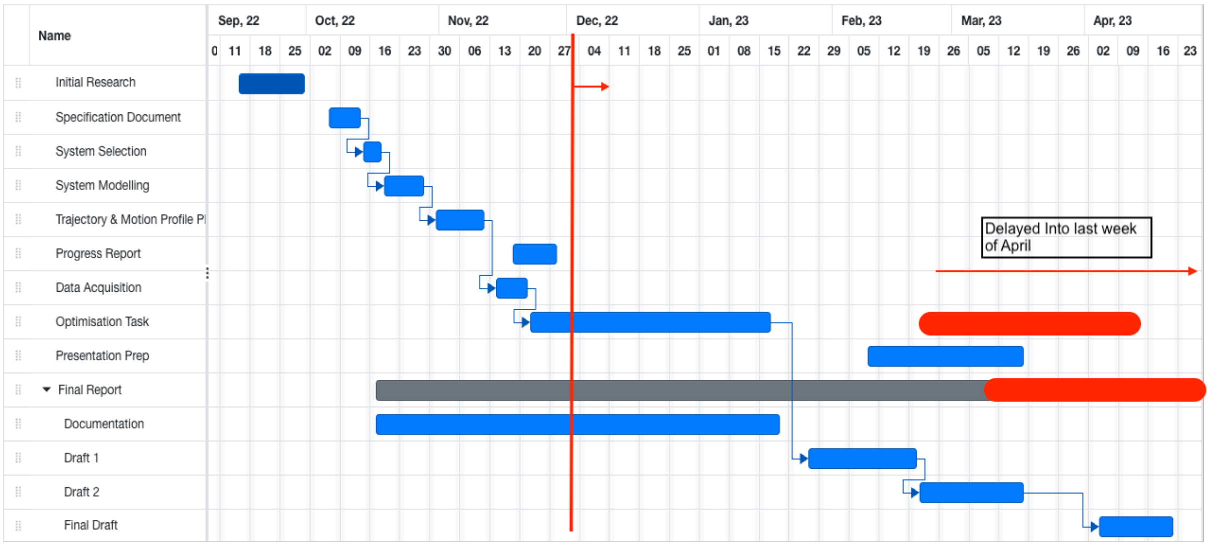

Figure 26 Final Project Timeline

One disadvantage of using the Agile approach in this case was the lack of strict timing constraints. Given the flexibility that comes with an individual project, there is an increased tendency to abandon all form of structure.

5.3 Project Tools

MATLAB R2021b was used for programming. Simulations were done in Simulink. Licenses for the software are provided freely by the University of Warwick.

All development was done on an M1 MacBook. Compatibility was successfully tested with Windows in the early stages of the project but is not guaranteed.

There are no legal, social or ethical issues pertaining to the project or any of the tools used.

6 Conclusions and Future Work

6.1 Additional Work

To complement the developed system, the following features were researched and/or implemented:

- Inclusion of gear ratio as an optimisable parameter. Considering how important gearboxes are in electromotive systems, the possibility of including them in the optimisation was also tested. In the same manner that the slew rate was tested, the gear ratio optimisation was tested; results proved the feasibility.
- Maximum current and torque ratings of motors. This option was also added to the system to further emulate real-life scenarios. Motors usually have a region where they can operate comfortably. Operating outside this region, in overload, can cause damage. The option allows users to toggle the operating limits.

6.2 Conclusions

From the results obtained throughout the project, it is acceptable to conclude that the project has contributed a substantial amount to this field of study.
The main outcomes of the project are as follows:

- Confirmation of the findings from previous work in the field
- Expansion of previous work in terms of the number and combination of optimisable variables
- Exploration of hybrid genetic algorithms for this use case
- Insight into the setup of the conveyor, motor and motion profile subsystems
- Provision of a prototype application capable of being used to create motion configurations for conveyor belt systems

Given these outcomes, there are a few downfalls that the project suffers from:

- The genetic algorithm may show inconsistencies between runs. The topic of global optimisation is a very complex one, and there is no perfect one-for-all solution. As such, the genetic algorithm may not perform too well in some cases due to its random nature. The best way to overcome this issue is to run the algorithm over several iterations to increase the chances of finding the optimal solution. It may also help to adjust the setup options. Despite this, the genetic algorithm will almost always outperform a singular local optimiser.

- False positives and confirmation bias. As the system was not tested with real-life data due to limitations on available data, it is difficult to claim that a perfect product was developed. It can be argued that the test case used worked to the benefit of justifying the methods employed. However, the results have laid the groundwork to expand upon.
- Application useability. As the developed application was intended for prototype use, the user interface and user experience may not live up to professional standards. For example, the user is expected to read the logs extensively in order to analyse the system. Ideally, important information should be easily accessible.

6.3 Future Work

To further develop the system and applications, the following improvements have been added to the work queue:

- Implementation of PWM (Pulse Width Modulation) voltage control for motors. PWM is a technique used to control motors by pulsing the input voltage from low to high across certain periods. At this stage, the motor is controlled by directly varying the input voltage. PWM is a more realistic method; including it will increase the system's industry potential.
- Allowing for the conveyor configuration to be changed, in terms of the layout. For example, changing the position and number of shafts, pulleys and gearboxes. This could be implemented by using an editable block diagram to allow users to drag-and-drop components around.
- Request real data from companies to verify the tool.
- Attain sponsorship from companies to test the tool in a real industrial setting.
- Improve the graphical user interface of the application.
- Provide options within the application to change the parameters of the PID controller and the hybrid genetic algorithm options. This would go a long way to increase the system's industry potential.
- Include an option to automate the number of runs for the hybrid genetic algorithm.

7 References

Atkinson, R.D. (2019) *Robotics and the Future of Production and Work. Information Technology and Innovation Foundation*. Available at: https://itif.org/publications/2019/10/15/robotics-and-future-production-and-work/ (Accessed: October 8, 2022).

Carabin, Giovanni & Scalera, Lorenzo. (2020). *On the Trajectory Planning for Energy Efficiency in Industrial Robotic Systems. Robotics*. 9. 89. 10.3390/robotics9040089.

Carabin, G., Vidoni, R., Wehrle, E. (2019). *Energy Saving in Mechatronic Systems Through Optimal Point-to-Point Trajectory Generation via Standard Primitives*. In: Carbone, G., Gasparetto, A. (eds) Advances in Italian Mechanism Science. IFToMM ITALY 2018. Mechanisms and Machine Science, vol 68. Springer, Cham. https://doi.org/10.1007/978-3-030-03320-0_3

Department for Business, Energy & Industrial Strategy (2022) *Quarterly Energy Prices UK April to June 2022*, p. 1. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1107499/quarterly_energy_prices_uk_september_2022.pdf (Accessed: October 7, 2022).
Howell, B. (2022) Top 7 most polluting industries in 2022, The Eco Experts. The Eco Experts. Available at: https://www.theecoexperts.co.uk/blog/top-7-most-polluting-industries (Accessed: October 8, 2022).

F. Assad, E. Rushforth, M. Ahmad, B. Ahmad and R. Harrison, (2018) "An Approach of Optimising S-curve Trajectory for a Better Energy Consumption," *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, 2018, pp. 98-103, doi: 10.1109/COASE.2018.8560587.

International Federation of Robotics (2021) *Executive summary world robotics 2021 Industrial Robots*. Available at:

https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2021.pdf (Accessed: October 6, 2022).

Katoch, S., Chauhan, S.S. & Kumar, V. (2021) *A review on genetic algorithm: past, present, and future*. Multimed Tools Appl 80, 8091–8126 Available at: https://doi.org/10.1007/s11042-020-10139-6 (Accessed: April 12, 2023)

Mathworks (2023). *fmincon*. Available at: https://uk.mathworks.com/help/optim/ug/fmincon.html (Accessed: January 22, 2023)

Mathworks (2023). *Symbolic Math Toolbox*. Available at: https://uk.mathworks.com/products/symbolic.html (Accessed: March 12, 2023)

Mathworks (2023). *Global Optimization Toolbox*. Available at: https://uk.mathworks.com/products/global-optimization.html (Accessed: January 22, 2023)

Mathworks (2023). *Optimization Toolbox*. Available at: https://uk.mathworks.com/products/optimization.html (Accessed: January 22, 2023)

Maxon (2023). *Motor Category* Available at: https://www.maxongroup.com/maxon/view/category/motor (Accessed: November 17, 2022)

Mirjalili, S. (2019). *Genetic Algorithm. In: Evolutionary Algorithms and Neural Networks. Studies in Computational Intelligence*, vol 780. Springer, Cham. https://doi.org/10.1007/978-3-319-93025-1_4 (Accessed: April 12, 2023)

Titan Conveyors (n.d) *Model 108 Slider Bed*. Available at: https://www.titanconveyors.com/products/slider-bed-belt/model-108-slider-bed (Accessed: March 13, 2023)

U.S. Energy Information Administration (2022) *September 2022 Monthly Energy Review*, pp. 38–39. Available at: https://www.eia.gov/totalenergy/data/monthly/pdf/mer.pdf (Accessed: October 7, 2022).

Waters, L. (2018) *Energy Consumption in the UK July 2018*, p. 6. Available at: https://webarchive.nationalarchives.gov.uk/ukgwa/20190511011505mp_/https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/729317/Energy_Consumption_in_the_UK__ECUK__2018.pdf (Accessed: October 7, 2022).