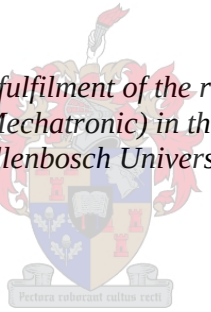
ALTERNATIVE METHODS OF MATERIAL HANDLING WITHIN A RECONFIGURABLE MANUFACTURING STATION

by
Matthew Marc Deacon



Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Mechatronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. Anton Basson

March 2017

DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third-party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2017

ABSTRACT

Alternative Methods of Material Handling Within a Reconfigurable Manufacturing Station

M.M. Deacon
Department of Mechanical and Mechatronic Engineering
Stellenbosch University
Private Bag X1, 7602 Matieland, South Africa
Thesis: MEng (Mechatronic)

March 2017

This thesis contributes to evaluating methods of material handling within a reconfigurable manufacturing station, as alternative to a six degree of freedom articulated robot arm. This research follows the design process of formulating the design requirements, considering different concepts and evaluating them, designing a selected concept in detail, validating the concept using test data and then applying the concept to a broader application. A few material handling methods are briefly considered before focusing on the use of a Cartesian robot. Different configurations of a Cartesian robot were considered.

As part of the design analysis, a model was developed which allows for the input of various station parameters and provides an estimate of the station's throughput and cost. This estimation model was implemented in MathCAD and split into two parts: a throughput estimate and a cost estimate. The inputs into the model are the process module configuration and the target kinematics. The model includes load and force calculations for each axis and component selection, as an input to the cost estimate.

A control system was developed, based on the PROSA architecture and implemented in C#. The design and implementation of this control system is discussed in this thesis.

To be able to validate the research results, a case study is used as an example implementation of the material handling method. However, the design is not limited to the case study, but rather provides a model for any process station with similar transport requirements.

The model was validated using a test setup in the Automation Laboratory that uses Festo components. The model therefore only provides for Festo components at this stage, but can easily be expanded upon if other manufacturers are to be considered.

After the model was validated, it was applied to the case study, including drive selection, to provide an estimate throughput and cost. These estimates are then compared to previous research that used a six degree of freedom articulated arm robot for a similar case. Other applications, different from the case study, of the model are also discussed.

UITTREKSEL

Alternatiewe Metodes vir Materiaalhantering Binne ‘n Herkonfigureerbare Vervaardigingstelsel

M.M. Deacon
Departement van Meganiese en Megatroniese Ingenieurswese
Universiteit Stellenbosch
Private Sak X1, 7602 Matieland, Suid-Afrika
Tesis: MIng (Megatronies)

Maart 2017

Hierdie tesis dra by tot die evaluering van metodes vir materiaalhantering binne ‘n herkonfigureerbare vervaardigingstasie, deur alternatiewe tot ‘n ses vryheidsgraad geartikuleerde robot arm te oorweeg. Hierdie navorsing volg die ontwerpproses van formulering van ontwerpvereistes, oorweging van verskillende konsepte, ontwerp van ‘n gekose konsep in detail, validasie van die konsep deur gebruik te maak van toetsdata en dan die konsep toe te pas tot ‘n breër toepassing. ‘n Paar materiaalhanteringsmetodes word vlugtig oorweeg voor daar gefokus word op die gebruik van ‘n Cartesiese robot. Verskillende konfigurasies van ‘n Cartesiese robot word oorweeg.

As deel van die ontwerpsanalise is ‘n model ontwikkel wat toelaat vir insette van verskeie stasie-parameters, en voorsien gevolglik ‘n beraming van die stasies se deurset en koste. Hierdie beramingsmodel is in MathCAD geïmplementeer en opgedeel in twee dele: ‘n deursetskatting en kosteskatting. Die insette tot die model is die proses-module-konfigurasie en die teiken kinematika. Die kosteskatting sluit in die las-en kragberekeninge vir elke as en komponent keuse as inset vir die kosteskatting.

‘n Beheerstelsel is ontwikkel, gebaseer op die PROSA argitektuur en geïmplementeer in C#. Die ontwerp en implementering van hierdie beheerstelsel was bespreek in hierdie tesis.

Om instaat te wees om die navorsingsresultate te valideer is ‘n gevallestudie gebruik as ‘n voorbeeld implementering van die materiaalhanteringsmetode. Die ontwerp is egter nie beperk tot die gevallestudie nie, maar kan voorsiening maak vir ‘n model vir enige prosesstasie met soortgelyke vervoervereistes.

Die model is gevalideer deur gebruik te maak van ‘n toetsopstelling in die Outomatisasie Laboratorium, deur gebruik te maak van Festo komponente. Die model voorsien daarom net vir Festo komponente op die stadium, maar kan maklik uitgebrei word as ander vervaardigers oorweeg moet word.

Na die model gevalideer is, is dit toegepas op ‘die gevallestudie; met inbegrip van die keuse van aktueerders, om ‘n geskatte deurset en koste te bepaal. Hierdie skattingswaardes is dan vergelyk met vorige navorsing wat ‘n ses vryheidsgraad geartikuleerde arm robot gebruik as ‘n soortgelyke geval. Toepassings van die model, vir gevalle wat van die gevallestudie verskil, word ook bespreek.

ACKNOWLEDGEMENTS

I would like to thank everyone that had input into my thesis or supported me through the process. I would like to specially thank the following people individually for their notable support:

- Prof. A.H. Basson, for his constant guidance, technical expertise and for providing a basic framework for the C# programming. Even though I was not the easiest student, your faith in my work kept me motivated.
- Reynaldo Rodriguez, for the many hours of assistance in the Automation Laboratory and for always being willing to help, even if it meant putting aside your own research for a time.
- Anton van den Bergh, and the workshop staff, for your input into the mechanical design work and for the hours spent manufacturing the components.
- My family, for the constant words of encouragement and for giving me the opportunity to study engineering.
- PJ Havenga, for your assistance in the laboratory, a second opinion when I was stuck and enjoyable banter through the late nights.

TABLE OF CONTENTS

1. Introduction 1
1.1. Background 1
1.2. Objectives 2
1.3. Motivation 3
1.4. Thesis Outline 3
2. Literature Review 5
2.1. Reconfigurable Manufacturing Systems 5
2.2. Traditional Control Architectures 5
2.3. Holonic Control 6
2.3.1. PROSA 6
2.3.2. ADACOR 7
2.4. Material Handling 7
2.4.1. Conveyors 7
2.4.2. AGVs 8
2.4.3. Robots 8
3. Design Requirements 10
3.1. Case Study 10
3.2. Functional Requirements 12
3.3. Performance Measures 13
4. Conceptual Design 14
4.1. Elementary Concepts 14
4.1.1. Conveyors 14
4.1.2. AGVs 14
4.1.3. Robots 15
4.2. Concept Refinement 16
4.3. Selected Concept 18
5. Detailed Design 19
5.1. Throughput Model 19
5.1.1. Throughput Estimate 19
5.1.2. Cost Estimate 21
5.2. Controller Design 24

5.2.1. Control Architecture 24
5.2.2. Supervisor (AHolonSupervisor) 26
5.2.3. External Communications Holon (MHolonECH) 26
5.2.4. Order (MHolonOrderMove) 27
5.2.5. Resource Holons 30
5.2.6. Cell Controller Emulator 35
5.3. Hardware 35
5.3.1. Sub-Assemblies 37
5.3.2. Other Design Features 40
5.3.3. Signal Interfaces 40
5.3.4. Robot Controller 42
6. Model Validation 43
6.1. Strategy 43
6.2. Configuration 43
6.3. Analytical Results 44
6.4. Experimental Results 45
6.5. Interpretation 46
7. Model Application 47
7.1. Case Study Configuration 47
7.2. Comparison to 6 DOF Robot 48
7.3. Other Applications 48
8. Conclusion 49
9. References 50
Appendix A - Model A-1
A-1: Throughput Estimate A-1
A-2: Cost Estimate and Force Calculations A-4
Appendix B – C# Implementation B-1
B-1: Holon Messages B-1
B-2: Robot Controller B-3
Appendix C – Hardware C-1
C-1: DAQ Pin Assignments C-1
C-2: Controller Position Tables C-2

C-3: Hardware Subassemblies C-2
C-4: Energy Chain Calculations.....C-7
C-4-1: Y-drive Energy Chain.....C-7
C-4-2: X-drive Energy Chain.....C-8
C-4-3: Z-drive Energy Chain C-9
Appendix D – Laboratory Results D-1

LIST OF FIGURES

Figure 1: Functional view of a process station 2

Figure 2: Traditional Control Architectures – (a) centralised, (b) heterarchical and (c) hierarchical. 5

Figure 3: Basic building blocks of an HMS (adapted from Van Brussel *et al* (1998)). 6

Figure 4: Conveyor 7

Figure 5: AGV 8

Figure 6: Cartesian robot (Vaughn, 2013) 8

Figure 7: 6 DOF articulated arm robot (KUKA, 2016) 9

Figure 8: SCARA robot (Omron Adept Technologies, Inc, 2016)....9

Figure 9: Exploded view of Q-frame circuit breaker (Hoffman, 2012)....10

Figure 10: Example cell layout (Hoffman, 2012)....11

Figure 11: Example electrical test station layout (Hoffman, 2012)....12

Figure 12: Cartesian robot primary axis 16

Figure 13: Cartesian robot module widths....17

Figure 14: Cartesian robot pallet system 18

Figure 15: Coordinate system and key dimensions 19

Figure 16: Velocity profile 20

Figure 17: Free body vector diagram example 22

Figure 18: Flowchart legend....24

Figure 19: Holon Interactions....25

Figure 20: Base autonomous holon loop 26

Figure 21 (a): Order holon state flow digram....28

Figure 22: RWT state flow diagram 31

Figure 23: Robot states 33

Figure 24: Robot move operation flow diagram....34

Figure 25: Cell Emulator GUI UPDATE 35

Figure 26: Experimental setup (a) CAD....36

Figure 27: End Effector 38
Figure 28: Support bracket 39
Figure 29: RWT and pallet setup 40
Figure 30: Electronics setup 41
Figure 31: End effector C-3
Figure 32: Mounted swivel module C-3
Figure 33: Mounted y-drive C-4
Figure 34: Mounted x-drives C-4
Figure 35: Y-drive mounting bracket C-5
Figure 36: Mounted z-drives C-5
Figure 37: Laboratory RWT and pallet table C-6

LIST OF TABLES

Table 1: Throughput estimate input variables 20
Table 2: Cost estimate input variables 22
Table 3: Drive and motor options 23
Table 4: Fixed costs 24
Table 5: Holon states 27
Table 6: Pallet holon states 30
Table 7: RWT holon states 31
Table 8: Robot holon states 32
Table 9: Cell controller messages 35
Table 10: Test scenarios 43
Table 11: Design variables 44
Table 12: Throughput estimate – validation case 44
Table 13: Cost estimate – experimental setup 44
Table 14: Average move times – laboratory setup 45
Table 15: Throughput – laboratory setup 46
Table 16: Throughput – comparison of model and laboratory results 46
Table 17: Throughput estimate – preferred case 47
Table 18: Cost estimate – preferred case, scenario 2 47
Table 19: 6 DOF robot throughput and costing 48
Table 20: Throughput estimate MathCAD nomenclature A-1
Table 21: Throughput estimate input variables A-4
Table 22: Holon inbox – supervisor staff holon B-1
Table 23: Holon inbox – move order holon B-1
Table 24: Holon inbox – extrnal communications holon B-2
Table 25: Holon inbox – pallet resource holon B-2
Table 26: Holon inbox – RWT resource holon B-3
Table 27: Holon inbox – robot resource holon B-3

Table 28: DAQ pin assignments C-1

Table 29: Position table – Y-drive C-2

Table 30: Position table – Z-drive C-2

Table 31: Laboratory test results - scenario 1 ..... D-2

Table 32: Laboratory test results - scenario 2 and 3 ..... D-3

NOMENCLATURE AND ABBREVIATIONS

Symbol	Description	Typical Units
a_x	Acceleration of drive in x-direction	m/s^2
a_y	Acceleration of drive in y-direction	m/s^2
a_z	Acceleration of drive in z-direction	m/s^2
$breakers_{pallet}$	Number of products per pallet	-
DAQ	Digital Acquisition Device	-
d_{mod_x}	Distance between modules in the x-direction	m
d_{mod_z}	Distance between modules in the z-direction	m
DOF	Degree Of Freedom	
gripper	Number of products held by gripper	-
J	Mass moment of inertia	$kg.m^2$
MADRG	Mechatronics, Automation and Design Research Group	
$RackSize_x$	Number of process modules in the x-direction	-
$RackSize_z$	Number of process modules in the z-direction	-
RFP	Request For Proposal	-
RMS	Reconfigurable Manufacturing Systems	-
RWT	Ramp Wave Tester	-
Stroke	The motion length of the linear drive	m
t_{pallet_switch}	Time taken for pallet to be switched by the conveyor	s
$t_{pickplace}$	Time taken for pickup or place operations	s
$t_{process}$	Product process time	s
v_x	Maximum velocity of axis in x-direction	m/s
v_y	Maximum velocity of axis in y-direction	m/s
v_z	Maximum velocity of axis in z-direction	m/s

1. INTRODUCTION

1.1. BACKGROUND

The global manufacturing market is under constant competition for cheaper and more efficient manufacturing methods. The use of automation and robotics has long been a solution for repetitive tasks that require a high level of accuracy.

Bi, et al. (2008) summarise the four requirements of a proficient manufacturing system as having:

a) Short lead-time. Product lead-time is the time between the initiation of a task and its completion.
b) More variants. By having a greater variety of products allows the company to accommodate more customers and therefore they can have a larger market share.
c) Low and fluctuating volumes. The required volumes of products are decreasing as there are shorter product life cycles and customer demand varies over time.
d) Low price. One of the primary features for a customer is the price and therefore it is important for manufacturers to reduce their production costs as much as possible.

As a result of the speed of technological advancements, the lifetime of a product line has been reduced and the need for manufacturing plants to handling more product varieties has increased. This type of manufacturing does not fit into the traditional methods of automation, where a manufacturing line is designed to handle high volumes of a single product type. This has led to the need for a new approach to manufacturing.

A potential solution to achieve these new requirements for effective manufacturing systems is the implementation of reconfigurable manufacturing systems (RMSs). These systems can adapt for product and station layout changes, without, for example, the need for the controller software to be rewritten and therefore there is less downtime between changes.

CBI-Electric Low Voltage is a South African company that produces a wide range of circuit breakers and other small electrical products that each have differing dimensions and manufacturing procedures. The company currently utilises a large amount of manual labour, but are looking to automate some of their processes through the use of robotics, as manual labour has become increasingly unreliable.

The University of Stellenbosch's Department of Mechanical and Mechatronic Engineering has a research group, Mechatronics Automation Design (MADRG), which has projects running with CBI-Electric and that is focused on the research of RMSs. The research into reconfigurability began with Sequeira (2008) who focused on the design of a fixture based reconfigurable automated spot welding system that could perform different spot welding routines based on different pallet layouts. This

research was further expanded on by Adams (2010) who researched the control of a reconfigurable assembly system (RAS). Kruger (2013) did a more in-depth study on the control of a feeder system for a RAS. Hoffman (2012) developed an electrical test station for an RMS cell. In his design, a six degree-of-freedom (DOF) articulated arm robot was used as a method for material transport within a reconfigurable manufacturing station.

Within most manufacturing stations, there is some sort of material handling that needs to be performed. There have been many advances within material handling methods and this has led to a large range of different material handling options when designing a manufacturing station. These options range from using conveyors, to automated guided vehicles (AGVs), to various types of robots and it is often difficult deciding which system to use.

The research presented here aims to develop a model of a particular material handling method that will allow different setups to be compared and evaluated. The material handling method that is considered here is a Cartesian robot.

1.2. OBJECTIVES

This research is aimed at evaluating alternative material handling approaches in a station within a RMS, particularly alternatives to a 6 DOF articulated arm robot (Hoffman, 2012), whilst using CBI Electric's electrical test station as a case study. The case study will provide certain boundaries on the final implementation, but the overall model should be applicable to a variety of possible case studies. The system size limitations that were imposed are that the product is roughly the size of a circuit breaker (100x100x10 mm) and the range of motion is in the order of one meter. A functional view of a process station is given in Figure 1. This shows the movement of products and information in and out of the station and the scope of the required transport method.

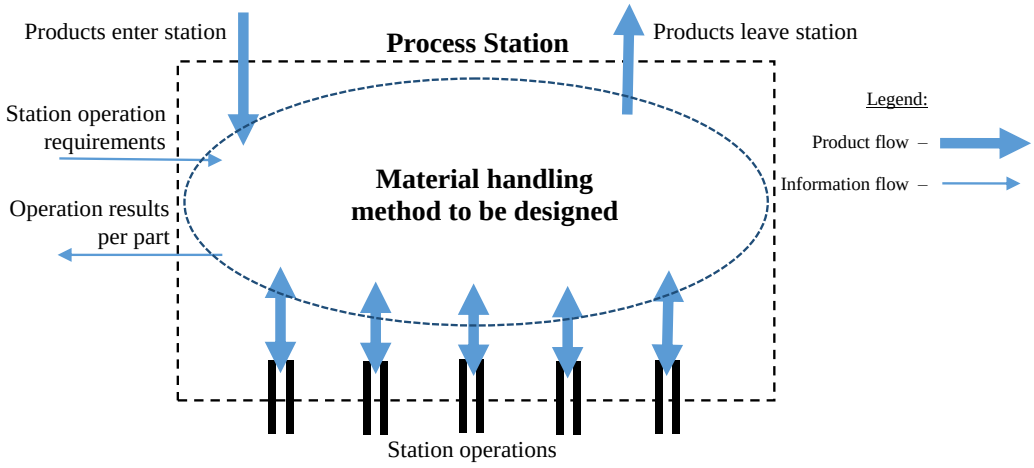


Figure 1: Functional view of a process station

The material transport method will be evaluated using a model that will estimate the station throughput and initial cost.

The model will be validated as a modification to the electrical test station developed by Hoffman (2012). The following has been excluded from the scope:

- How the material is transported into and out of the station.
- What the station's operations are and the duration of these operations, other than it is assumed to vary within a single product variety.

1.3. MOTIVATION

There is a need for new methods of material handling within a manufacturing station that have a high level of traceability, accuracy, speed and reconfigurability. The current solutions of either human handling or the use of a 6 DOF robot have been looked at and the results do not meet the requirements. For instance, Hoffman (2012) could not reach the required throughput without having too high an initial cost.

In Hoffman's (2012) station, the 6 DOF robot constituted a large cost element and became the single item that limited the throughput. Once the station grew to fully utilise the robot, the cell could only be scaled by adding more stations, each with their own 6 DOF robot.

Articulated arm (6 DOF) robots have many advantages, e.g. high dexterity. However, a typical test station, such as the one considered here, may use fewer than the six degrees of freedom. This thesis considers whether alternative approaches can compete with 6 DOF robot in these aspects.

1.4. THESIS OUTLINE

This thesis begins with a study of the literature on reconfigurable manufacturing, followed by traditional control architecture and the alternative holonic control. There is then a brief discussion on current material handling methods.

A set of design requirements are then formulated in Chapter 3, based on, but not limited to, a specific case study. This section includes briefly describing the case study, listing the functional requirements of the station and providing the performance measures of the model.

In Chapter 4, a concept is then developed based on the basic elementary material handling methods described in the literature review. Using the design requirements, a single concept, a Cartesian robot, is selected and further refined.

In Chapter 5, this concept is implemented in both a mathematical model and a laboratory setup. The mathematical model is split into two estimates: throughput and cost. The laboratory setup involved developing a controller, a test setup and integrating the two.

The model is then validated using the test setup in Chapter 6 followed by applying the validated model to other applications in Chapter 7.

Finally, a conclusion of the work completed is given followed by a list of possible further research ideas.

2. LITERATURE REVIEW

2.1. RECONFIGURABLE MANUFACTURING SYSTEMS

A reconfigurable manufacturing system is defined as a system with the ability to change and be rearranged in a cost-effective way (Setchi & Lagos, 2004). Koren *et al.* (1999) lists six core characteristics of a RMS system as:

- Modularity – operational functions are to be compartmentalised so that they can be shared between different operations.
- Integrability – modules are created with interfaces that allow quick and precise integration in a system.
- Customisation – the system can be customised to a range of product variants.
- Convertibility – the system can easily be adapted for a new set of system requirements.
- Diagnosability – any defects or faults in the system should be easily locatable and traceable.
- Scalability – additional capacity should easily be able to be added.

The attractiveness of RMSs is that they provide solutions to the current need for high variety, low volume manufacturing. Due to the characteristics mentioned above, a reconfigurable manufacturing plant can simply be implemented and adapted to these type of manufacturing situations.

2.2. TRADITIONAL CONTROL ARCHITECTURES

Meng *et al.* (2006) mention three common control architectures used in manufacturing. These are shown in Figure 2.

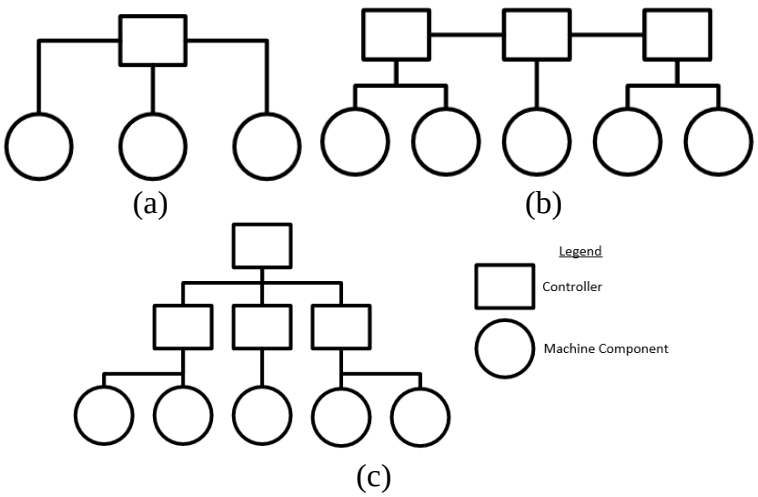


Figure 2: Traditional Control Architectures – (a) centralised, (b) heterarchical and (c) hierarchical.

A centralised control architecture is based on using one main controller that controls all the machine components in the cell. This is most commonly implemented in simple conventional control systems with only one controller.

A heterarchical control architecture is when there are multiple independent controllers that each have their own subsystems. There can be basic communication between each controller to allow for more complicated systems.

A hierarchical control architecture has different levels of control, where the main controller distributes control to the controllers beneath it, which then control specific machine components. Instructions are therefore passed down the tree and feedback passed back up.

2.3. HOLONIC CONTROL

Holonic control is a highly distributed control paradigm that allows for constant adaptation and a high level of flexibility (Van Brussel *et al* (1998)). It is based on the use of independent holons that each have different roles. This allows the system to adapt to frequent changes and disturbances. Holonic control can be seen as a mixture between hierarchical and heterarchical control.

2.3.1. PROSA

PROSA (**P**roduct-**R**esource-**O**rder-**S**taff **A**rchitecture) is defined in detail by Van Brussel *et al* (1998). The architecture consists of three basic holon types:

- Resource holon – represents a physical part, which includes a production resource and an information processing part.
- Product holon – acts as an information server that contains the product and process information required for manufacture.
- Order holon – represents a task and manages the logistical information related to the product being manufactured.

These basic holons interact through an exchange of knowledge as shown in Figure 3.

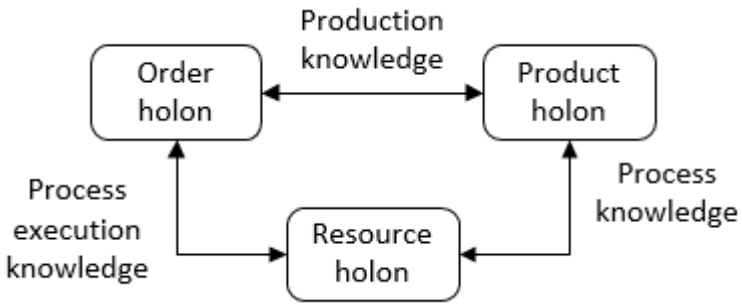


Figure 3: Basic building blocks of an HMS (adapted from Van Brussel *et al* (1998)).

There are also special staff holons that assist the above basic holons with certain tasks. These tasks include communication, calculations and other tasks that are used by a variety of holons.

As mentioned above, PROSA includes aspects of both hierarchical and heterarchical control. The structure of the system is decoupled from the control algorithm; adding integrability, modularity and allowing for more advanced control algorithms.

2.3.2. ADACOR

ADACOR (**ADA**ptive holonic **CO**ntrol a**R**chitecture for distributed manufacturing systems) is described by Leitao and Restivo (2008) and the key points of the architecture have been summarised below. The purpose of ADACOR is an adaptive control that dynamically balances a more centralised structure with a decentralised one.

As with PROSA, ADACOR defines four holon types: the product holons (PH), operational holons (OH), task holons (TH) and supervisor holons (SH). The first three holons have similar characteristics to the resource, product and order holons in PROSA. The supervisor holon is given an overview of the entire system and passes instructions down to the holons beneath it – in a hierarchical manner.

In ADACOR each holon contains a Logical Control Device (LCD) that is responsible for the inter-holon communication, decision making and interfacing with the holons' physical resource.

2.4. MATERIAL HANDLING

This section describes some of the different material handling concepts that are currently used in manufacturing system. Not much literature in the form of scholarly articles were found for material handling, therefore most of this literature was found in textbooks or online.

2.4.1. CONVEYORS

Conveyors are typically used to transport material in a single direction by the use of a belt or rolling mechanism. A simple example of a conveyor can be seen in Figure 4.

Figure 4: Conveyor

2.4.2. AGVs

An automatic guided vehicle (AGV) is an unmanned mobile robot that assists in material handling. An example of an AGV is shown in Figure 5.

Figure 5: AGV

2.4.3. Robots

A robot is defined as a machine that can perform a series of complicated tasks automatically. Three general types of industrial robots are described in this section: a Cartesian robot, a SCARA robot and a 6 degree of freedom (DOF) articulated arm robot. Industrial robots are grouped according to number of axes or DOFs, structure type, size of work envelope, payload capability, and speed (RobotWorx, 2016).

2.4.3.1. Cartesian

A Cartesian, or Gantry, robot (Figure 6) has three DOFs and moves in straight lines that are coincident with a Cartesian coordinate system (Nof, 1999). However, beyond the prismatic joints, a wrist can be attached to the end effector for reorientation.

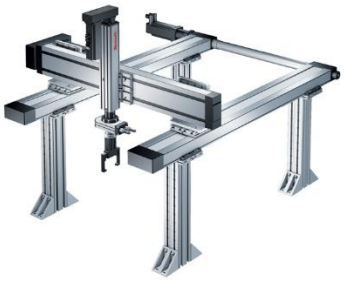


Figure 6: Cartesian robot (Vaughn, 2013)

2.4.3.2. 6 DOF Articulated Arm

This robot is designed to mimic a human arm and allows for any orientation of the end effector. Figure 7

Figure 7: 6 DOF articulated arm robot (KUKA, 2016)

2.4.3.3. SCARA

A selective-compliance-articulated robot arm (SCARA) is a 4-axis robot based on a two-link arm as shown in Figure 8. A SCARA robot is similar to a 6 DOF articulated arm robot, but has fewer axes.

Figure 8: SCARA robot (Omron Adept Technologies, Inc, 2016)

3. DESIGN REQUIREMENTS

This thesis considers scenarios where a discrete product needs to be moved from a single source to multiple destinations, called process modules, where some operation is to be performed. After the operation is completed, the product needs to be returned to a single final destination.

The requirements for this design were selected based on a particular case study, but a model is developed in the thesis that allows for a more general application as well. The case study is described in the next section, followed by functional requirements and performance measures based on the case study.

3.1. CASE STUDY

The reason for using a case study was to narrow down the goal for the material handling concept that is to be developed, so that it can be compared to the 6 degree of freedom (DOF) case. The chosen case study is that of the electrical test station from Hoffman (2012).

Hoffman (2012) researched the use of a 6 DOF articulated robotic arm in the reconfigurable manufacturing station. The station in particular was an electrical test station where the product, circuit breakers, have to be moved from a conveyor pallet to a ramp wave tester (RWT) and back to the conveyor pallet. An example of the circuit breakers that are to be processed can be seen in Figure 9. In this application, the circuit breakers are not riveted and therefore care has to be taken in holding them securely while transporting. Figure 10 shows an example of a possible cell layout. The electrical test station that this research is investigating is highlighted in the black box and shown in more detail in Figure 11.

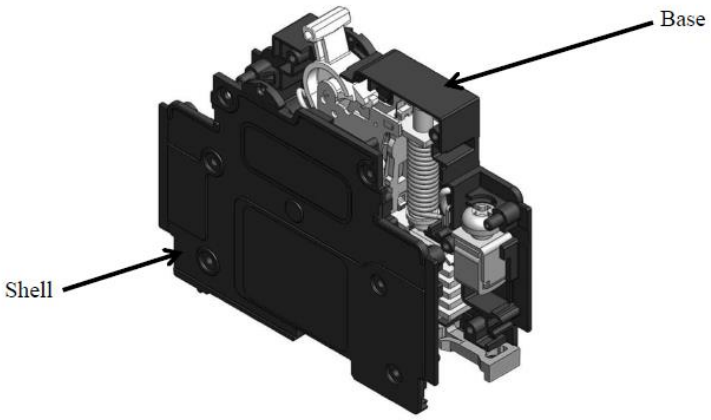


Figure 9: Exploded view of Q-frame circuit breaker (Hoffman, 2012)

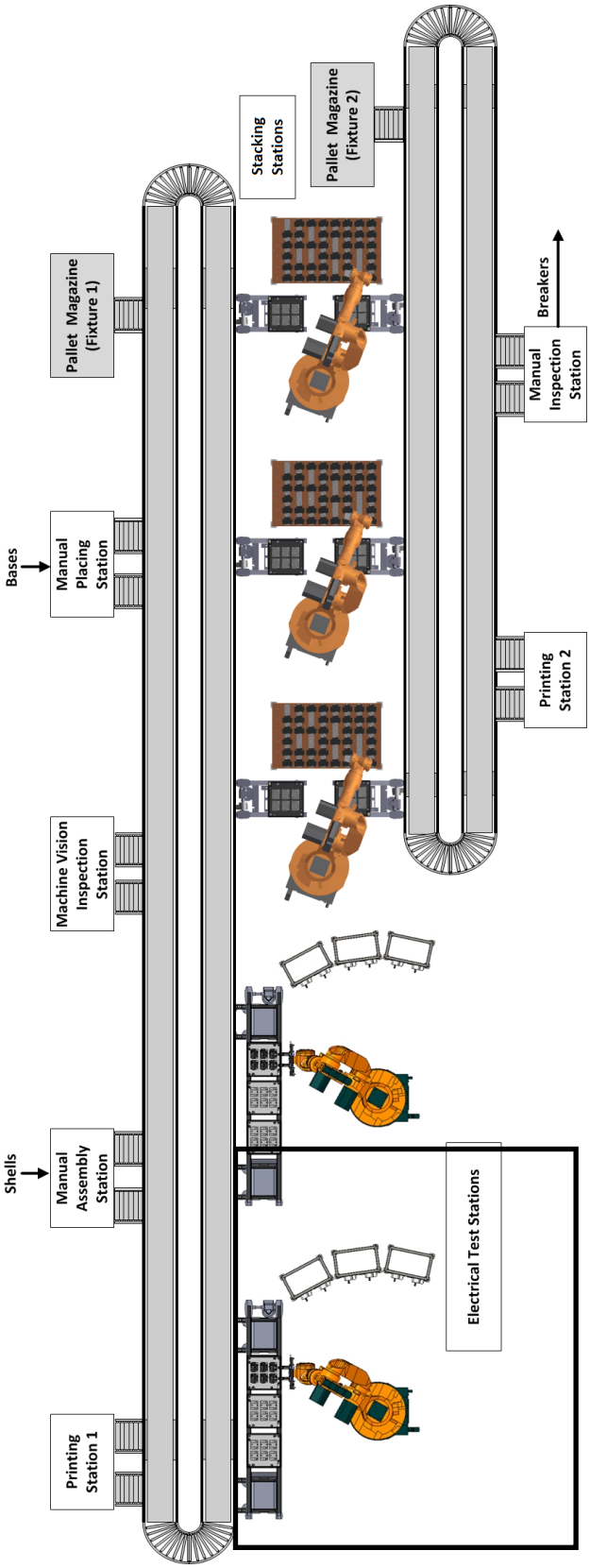


Figure 10: Example cell layout (Hoffman, 2012)

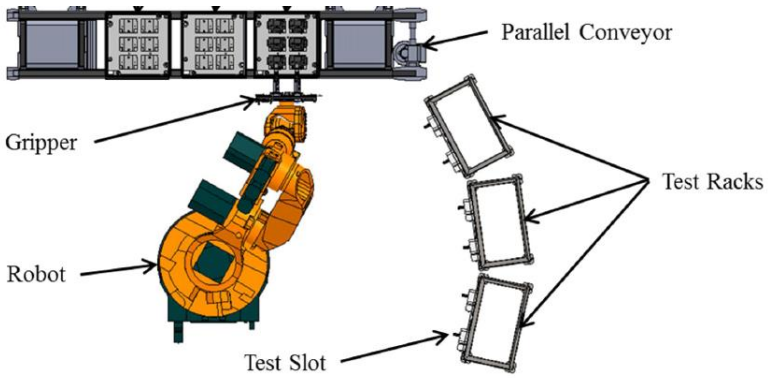


Figure 11: Example electrical test station layout (Hoffman, 2012)

3.2. FUNCTIONAL REQUIREMENTS

Figure 1 gives a schematic view of the station's functions. This is expanded upon to give a list of functional requirements in this section.

Without specifying the exact process that the products have to undergo, the following design requirements were formulated for the material handling in the station. These requirements are based on a list of characteristics of an automated material handling process that is given by Crowson (2006).

- There is a beginning and end point
  - The product will enter the station from a conveyor and leave the station by returning to the conveyor. There will, however, be multiple process module locations.
- The product's configuration or orientation may change
  - When the products enter the station, they are in a horizontal position in the pallet fixtures on the conveyor and may have to be rotated into a vertical position to be inserted into the process module.
  - The products may also not be properly bound and so needs to be securely gripped.
- Process times can differ between product varieties and amongst different instances of the same product
  - The products are considered to have an average processing time, but the station should allow for a variation in process times between a single product type.
- Safety of workers
  - During the stations operation, there will be no nearby workers.
- Coordination with support functions
  - The station relies on the cell controller to coordinate with the conveyor to supply pallets. The control involved in this coordination is beyond the scope of this research.
  - The station controller requires feedback from the process modules on the completion of a product's processing.

- Part tracking
  - The station needs to keep track of the products as they are moved around and processed so that it can inform the cell controller which product instances are leaving the station and the results of the process on the product.

For the research presented here, the circuit breakers used by Hoffman (2012) determine the typical size of the products being transported, i.e. approximately 75x75x100 mm. Also, the size of the processing modules used by Hoffman (2012) have a minimum centre-to-centre distance of 230x230 mm, which will be assumed to be the typical dimensions for the process rack. The overall range of motion of the station will also be limited to under 1 meter. These dimension ranges limit the scope of the study to allow the results to be comparable to Hoffman's (2012).

As shown in Figure 1, products enter and leave the station. In this research, it will be assumed that the products arrive at the station in a pallet configuration on a cell conveyor. This conveyor system is excluded from the scope of this research. Once the pallet has entered the station the products need to be moved to the process modules. After the process is completed, the products have to be moved back to a pallet and back on to the cell conveyor.

Figure 1 also shows that the information enters and leaves the station. Since the focus of this research is on the product transport, the information flow out of the station will not be considered beyond generating the information.

3.3. Performance Measures

This model will evaluate the station based on two performance measures. These are listed below:

- Throughput
  - The number of products that can be processed by the station per hour.
- Cost
  - The initial cost of the transport system within the station.
  - In the model the initial and operating costs of the process modules are not considered, but only the transport system.

4. CONCEPTUAL DESIGN

This section compares different material handling concepts that are currently used in manufacturing systems and evaluates them. These concepts are then used to conceptualise a design that fulfils the requirements set out in Section 3. A final concept is then shown and a short evaluation of it is given.

4.1. ELEMENTARY CONCEPTS

This section explains current material handling solutions and how they are used in manufacturing systems. This is a continuation from the literature review Section 2.4.

4.1.1. CONVEYORS

The main advantage of a using a conveyor system is that conveyors are relatively cheap over long distances. Conveyors also allow for parallel operation as they can carry multiple products at a time. A conveyor is also scalable by extending the conveyor or adding a new line.

The disadvantages of a conveyor system for this application is that the movement distances in the station are relatively small. The product also needs to be moved in the vertical direction which would require a lift conveyor and these lifts are a possible bottleneck. There is also a need for the products to be reoriented and possibly pressed into the process modules which would require another form of actuation.

4.1.2. AGVS

The concept of using an automatic guided vehicle (AGV) has two possible configurations for this application. These are explained briefly below

The first is an AGV that has a 4 DOF robotic mechanism built on it that could retrieve the circuit breaker from the conveyor and then insert it into an open test slot. This concept requires each AGV to have a large amount of expensive moving parts and therefore this would make the cost very high. However, since it would be easy to add an extra AGV, the solution is highly scalable and because each AGV handles one part it also has good parallelism and reconfigurability.

The second configuration is that the AGV is used purely for transportation. This would mean there would need to be a mechanism for loading the AGV from the conveyor and one for offloading the product into the process module. This requires a complicated mechanism at the process module that would be very similar to that of the Cartesian robot mentioned in Section 2.4.3.1.

Both configurations would be attractive if there was a large distance between the offload point of the conveyor and the process modules. However, this is not the case and the cost of an AGV for short distance travel would be more than for using a conveyor or a robot. The main application for AGVs in current manufacturing

systems involves moving high mass payloads between cell stations, while the products considered here weigh only a few grams. Therefore, AGVs are not considered to be an attractive solution for the present requirements.

4.1.3. ROBOTS

In this section the three general types of industrial robots there were mentioned in Section 2.4.3 are evaluated for their application in the process station.

4.1.3.1. Cartesian

An advantage of a Cartesian robot is that the system is scalable by extending an axis. They are also modular as only the required axes are supplied, so if only two degrees of freedom are needed the cost can be reduced. Further, the design of Cartesian robots has become significantly easier through the use of vendor design tools and the cost of Cartesian robots has decreased by 25% over the last 5 years (Vaughn, 2013).

However, a Cartesian robot, like all robots, can only move one end effector at a time and this can create a bottleneck. Each Cartesian robot will also have a fixed range of motion which reduces the reconfigurability of the system. If the supplied vendor design tools do not meet requirements, a custom design and manufacture may be required.

4.1.3.2. 6 DOF Articulated Arm

A 6 DOF articulated arm robot was the solution that was analysed by Hoffman (2012). This robot is designed to mimic a human arm and allows for any orientation of the end effector. In Hoffman's (2012) research he focused on using a large 6 DOF articulated arm robot as this allowed for space for a human operator to man the station if the robot was down. This, however, reduced the throughput of his system as the resulting large robot moved slower than a small one would and the distances moved were also increased (Hoffman, 2012).

The advantage of using a 6 DOF articulated arm robot is the freedom of picking and placing the products from any location and orientation and moving them to any location and orientation within the robot's reach. However, these robots have relatively small payloads for the size of the robots and that limits the gripper design (Hoffman, 2012).

4.1.3.3. SCARA

SCARA robots are typically very fast and accurate, but are mostly used for pick and place operations from above. This limits their use in this research as the products need to be reoriented between the pallet and process modules.

4.2. CONCEPT REFINEMENT

The concept that was selected in this research is a mixture of the above basic technologies, with the focus being a Cartesian robot. A Cartesian robot was chosen as it fits into a small space and when designed correctly can be relatively quick.

There are a few configurations of the Cartesian robot that were considered and refined to a final concept for detailed investigation. The first configuration aspect is how to modularise the Cartesian robot. This includes determining the size and orientation of the robot axes. The next is the how pallets from the conveyor will be accommodated in the station.

The choice of the Cartesian robot modules being either horizontal or vertical is illustrated in Figure 12. If the Cartesian robot was split in horizontal modules (Figure 12(b)) there would be the need for a lifting sub-system to bring the circuit breaker to the correct vertical position. The other option (Figure 12(a)) is where vertical modules are used. This vertical setup allows for the Cartesian robot to remove and place the parts directly from the conveyor pallets. Since using vertical modules removed the need for extra subsystems, this was the chosen configuration.

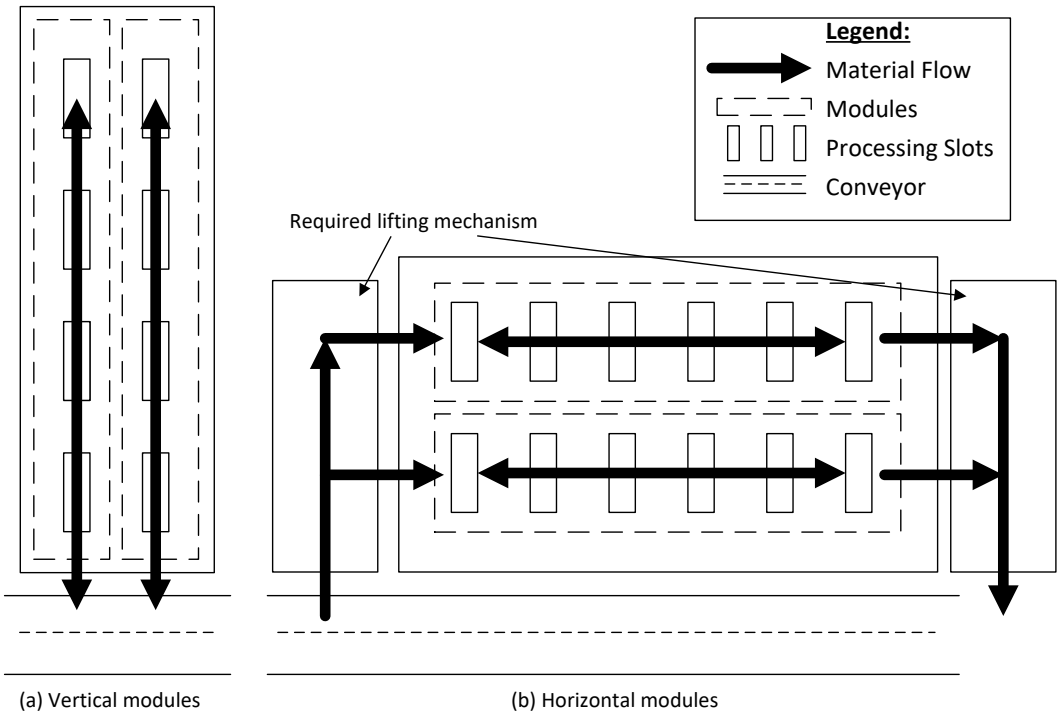


Figure 12: Cartesian robot primary axis

The configuration choice is the number of the process modules in the horizontal direction. The modules can be either single width (Figure 13(a)) or in multiple widths (Figure 13(b)). The advantage of a single process module width is that it eliminates the need for one of the degrees of freedom. The next option is a two process module width setup, which would require only two possible positions and

can be achieved using a simpler actuator, under bang-bang control, such as a pneumatic drive. The final option is using a setup with more than two process modules and this will therefore require a more expensive linear drive capable of position control.

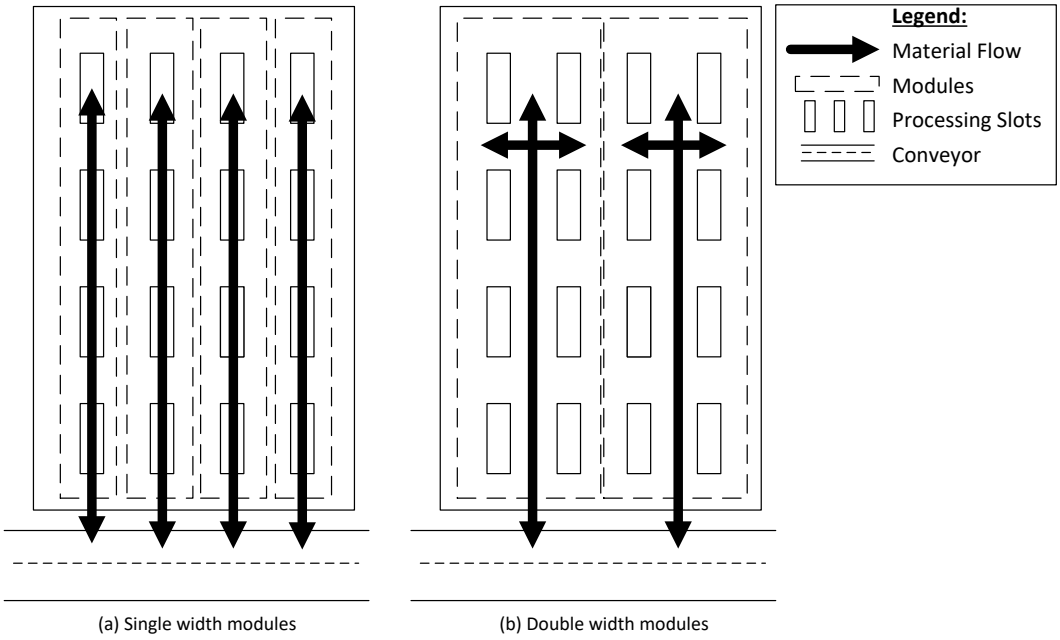


Figure 13: Cartesian robot module widths

The next configuration choice is how to handle pallets in the station and the offload and return of products to the pallets. The control of pallet movement is outside the scope of this research, but the method used to store them in the station is important. The first option is using an auxiliary conveyor (Figure 14(a)) where multiple pallets will be stored in the station. These pallets will be unloaded on a first in and loaded on as a first out basis. This setup allows for a buffer of pallets in the station, but requires a large amount of extra conveyor and multiple horizontal robot positions. The next option is to use transverse conveyors to move the pallets off the main conveyor and store them on the transverse conveyor in the station.

This choice is directly linked to the width of the process modules mentioned previously, since if the Cartesian modules are single width, there will be only one transverse conveyor per robot module and therefore only one pallet can be active in the system at a time (Figure 14(b)). However, when the module width is two or more, there can be a number of active pallets (Figure 14(c)) per robot, as there can be multiple transverse conveyors. The problem with only having one active pallet in the system is that all the products have to be removed from the pallet, before a processed product can be returned to the pallet. This is because the products are removed from the pallet fixtures on a last in, first out basis.

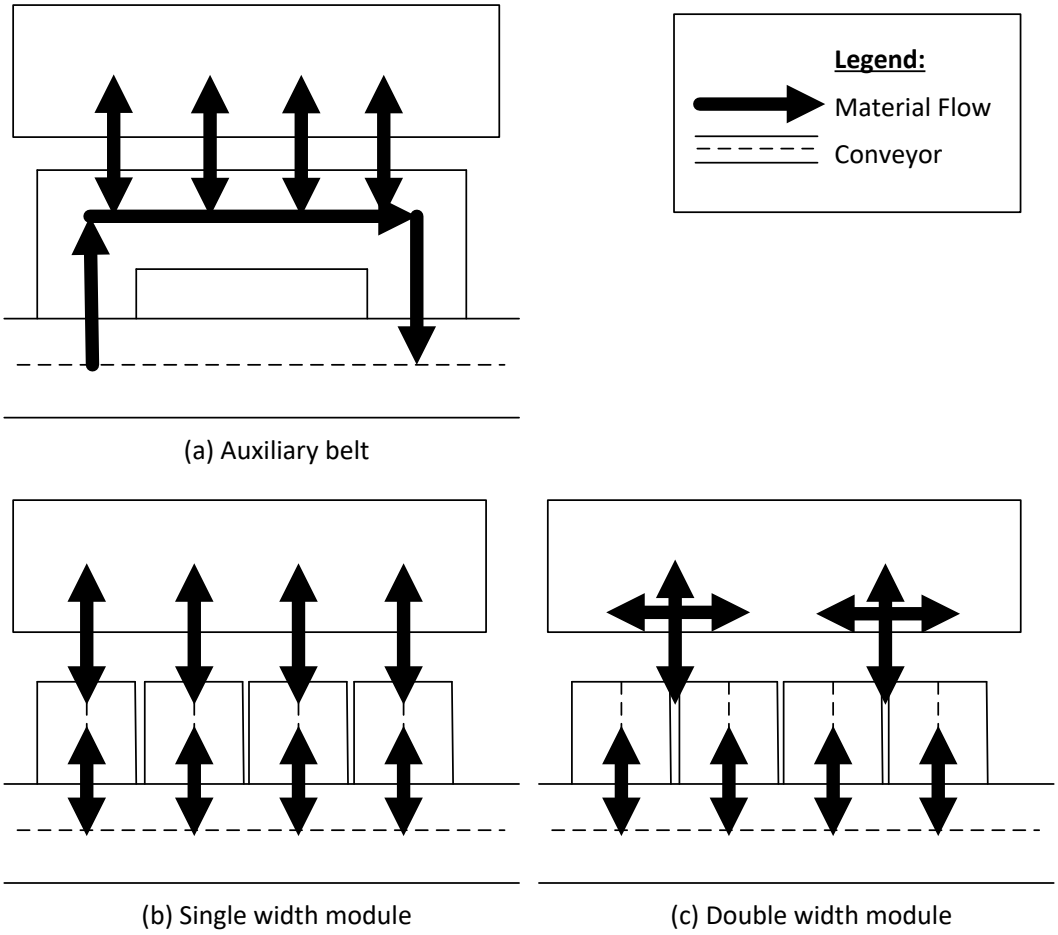


Figure 14: Cartesian robot pallet system

4.3. SELECTED CONCEPT

The concept that was selected for further evaluation utilises transverse conveyors to remove pallets from the main conveyor (Figure 14(b) and (c)) and a Cartesian robot that can simulate both single and double widths of process modules of different vertical heights. It is assumed that the pallet positions are aligned with the process modules so that the x-axis does not need to be adjusted when moving a product from a pallet to a process module on the same side.

The Cartesian robot has an end effector with two grippers that can pick up horizontally placed products from a pallet and rotate them using a swivel module to place them in the vertical process modules. This end effector is mounted on a y-axis to move laterally from the fixtures on the pallets to the process modules. The y-axis is then mounted on the horizontal x-axis. This x-axis is then mounted on a vertical z-axis.

This concept is illustrated and explained in more detail in the next chapter.

5. DETAILED DESIGN

The design is presented three sections: the throughput model, the controller and the experimental hardware.

5.1. THROUGHPUT MODEL

The throughput model comprises of a throughput estimate based on the process station size and target kinematics and a cost estimate based on a drive and motor selection done.

5.1.1. THROUGHPUT ESTIMATE

The throughput estimate was programmed in MathCAD (refer to Appendix A-1 for the detail) which calculates the throughput rate of the station. The inputs for the algorithm were split into design variables, which the user can change depending on the station size is to be simulated, and design independent variables, which are fixed by external factors. These variables are listed in Table 1. The coordinate system and some key dimensions around the process rack are illustrated in Figure 15.

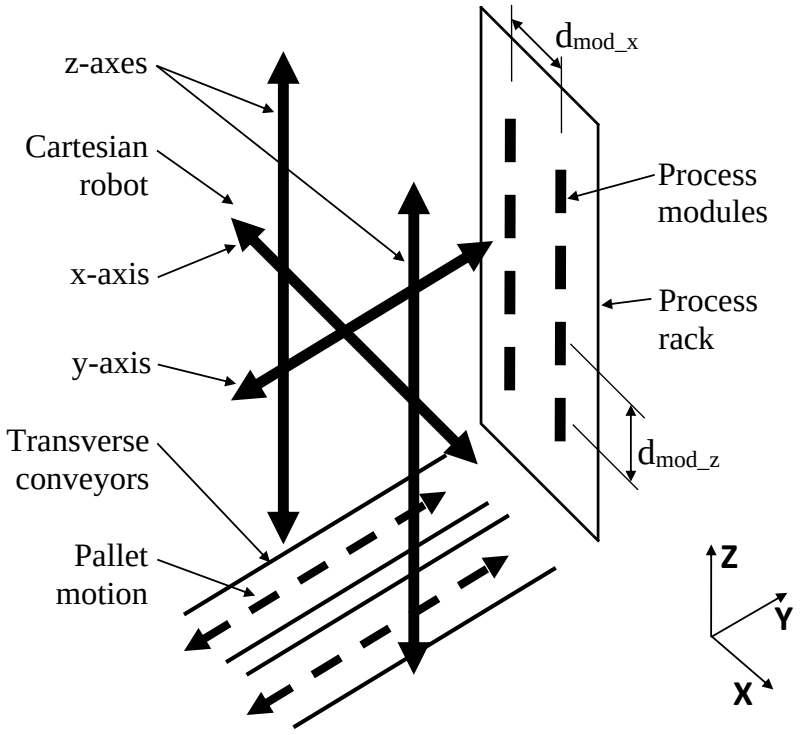


Figure 15: Coordinate system and key dimensions

Table 1: Throughput estimate input variables

Design Variables	**Description**
$RackSize_z$	Number of process modules in the z-direction
$RackSize_x$	Number of process modules in the x-direction
gripper	Number of products held by gripper
a_x	Acceleration of axis in x-direction
a_y	Acceleration of axis in y-direction
a_z	Acceleration of axis in z-direction
v_x	Maximum velocity of drive in x-direction
v_y	Maximum velocity of drive in y-direction
v_z	Maximum velocity of drive in z-direction
$t_{pickplace}$	Estimate time taken for pickup and place operations
Design Independent Variables	**Description**
t_{pallet_switch}	Time taken for pallet to be exchanged by the conveyor
d_{mod_z}	Distance between process modules in the z-direction
d_{mod_x}	Distance between process modules in the x-direction
$products_{pallet}$	Number of products per pallet
$t_{process}$	Product processing time

The estimate is based on the time it would take to move the product over the average distance of all possible moves. This average distance, called d_{ave_z} for the z-direction, is calculated using Equation 1.

$$d_{ave\_z} := d_{mod\_z} \cdot \frac{RackSize_z}{2} \tag{1}$$

It is assumed that all drive motions follow the velocity-time profile shown in Figure 16.

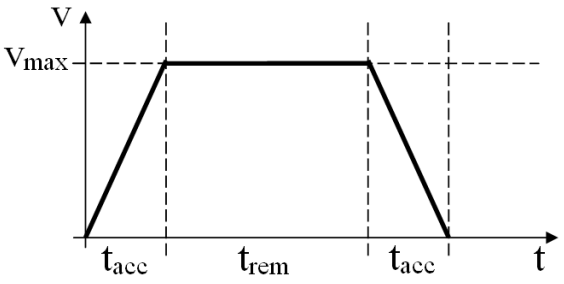


Figure 16: Velocity profile

The total time of the motion was split in three parts: the time to reach maximum velocity (t_{acc_z}, Equation 2), the time to cover the remaining distance at maximum velocity (t_{rem_z}, Equation 3) and an estimate time for the pick and place operations ($t_{pickplace}$). The reason the pick and place operations were split up was as a result of the operations being very complicated and taking a significant length of time to complete. This pick and place time was taken as an average from experimental results and is case specific.

$$t_{acc\_z} := \frac{v_z}{a_z} \tag{2}$$

$$t_{rem\_z} := \frac{d_{ave\_z} - 2 \cdot 0.5 \cdot a_z \cdot t_{acc\_z}^{\ 2}}{v_z} \tag{3}$$

The total move time was then calculated using Equation 4, which assumes the drive takes the same length of time to decelerate as it did to accelerate.

$$t_{cycle\_move\_z} := 2 \cdot t_{acc\_z} + t_{rem\_z} + 2 \cdot t_{pickplace} \tag{4}$$

There are two possible scenarios that limited the throughput: either the robot was too slow to fill all the process modules in the process rack or there were too few process modules and therefore the processing time was the limiting factor.

The first scenario is when there are an excess number of process slots and the limitation on the system is the time it takes to move the breakers. In this scenario, the throughput is the number of moves the system can do per second multiplied with the number of products moved per move, which is calculated using Equation 5.

$$throughput_{gripper} := \frac{gripper_{num\_teeth}}{t_{cycle\_move}} \tag{5}$$

The other scenario is when all the process slots are full and the process time is the limiting factor. In this scenario, the throughput is calculated as the rack size divided by the process time, Equation 6.

$$throughput_{racksize} := \frac{(RackSize_z \cdot RackSize_x)}{t_{process}} \tag{6}$$

5.1.2. Cost Estimate

The drive selection and cost estimate was also programmed in MathCAD (refer to Appendix A-2 for details). The cost estimate has the inputs listed in Table 2.

Table 2: Cost estimate input variables

Kinematic Variables	**Description**
a_x	Acceleration of drive in x-direction
a_y	Acceleration of drive in y-direction
a_z	Acceleration of drive in z-direction
α_{DSM}	Rotational acceleration of swivel module
Station Dimensions	**Description**
$RackSize_z$	Number of process modules in the z-direction
$RackSize_x$	Number of process modules in the x-direction
gripper	Number of products held by gripper
d_{mod_z}	Distance between modules in the z-direction
d_{mod_z}	Distance between modules in the z-direction
d_{mod_x}	Distance between modules in the x-direction
$Stroke_y$	Distance to move in y-direction

The cost estimate is calculated by doing a set of drive selections. These selections are based on choosing drives that can handle the system loads that are found through a force analysis of the system. The drive selection is here limited to the Festo catalogue as these were the drives available in the automation laboratory, but the model can be expanded upon to include other manufacturers.

This force analysis was done using first principles: the centre of gravity of each moving part was found using CAD models and then, using Newton's second law, the reaction forces were calculated as shown in Figure 17. Both static and dynamic equivalent forces were considered.

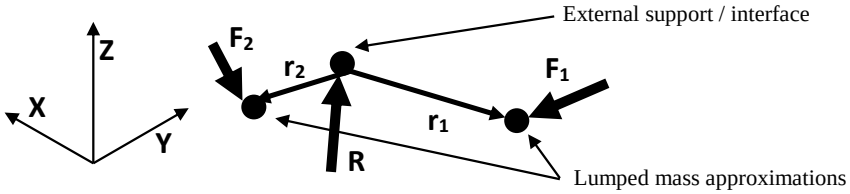


Figure 17: Free body vector diagram example

Drive selection is started by selecting a drive size and type. The model then calculates the drive safety factors. The same is done when choosing a motor to power the drive. To simplify the linear drive and motor selection process in this research, it was limited to the Festo catalogue, more specifically the models listed in Table 3. These choices can, in future, be adapted in the MathCAD file to allow for other product ranges.

The drive options include toothed belt axes drives with roller guides, DGE, and pneumatic linear drives, DGC. The pneumatic linear drive was included for axes that only have two positions, as it is substantially cheaper than a belt drive and

motor combination. The drive cost estimates are based on obtaining two quotes for different drive lengths and interpolating or extrapolating for the required length.

Table 3: Drive and motor options

Drives	**Motors**
DGE-25-...-ZR-RF	EMMS-ST-57-S/M
DGE-40-...-ZR-RF	EMMS-ST-87-S/M
DGC-25-...-G-PPV-A	EMMS-AS-55-S/M
DGC-40-...-G-PPV-A	EMMS-AS-70-S/M

The next selection is between using a stepper or servomotor and the size of the motor. The first step in this selection is the calculation of the mass moment of inertia of the drive and its load, J_L, using Equation 9, which is for a DGE-25-...-ZR-RF drive (FESTO, 2016)).

$$J_L := \left(1.75 + 0.188 \cdot \frac{Stroke}{m} + 2.052 \cdot \frac{Load}{kg}\right) kg \cdot cm^2 \tag{9}$$

Where
Stroke : The working length of the drive [m]
Load : The total moving load on the drive [kg]

The inertia ratio is calculated from this inertia and the inertia of the motor, J_M, using Equation 10.

$$InertiaRatio := \frac{J_L}{J_M} \tag{10}$$

When selecting a motor, the aim is to match the inertias so that the inertia ratio is less than 10. It is also important to ensure an acceptable safety factor of the maximum motor torque vs required torque and the required rotational speeds are achieved. If the inertia ratio is too high, the use of a gearbox, of gear ratio GR, should be considered. If a gearbox is used, the inertia ratio would be calculated using Equation 11.

$$InertiaRatio_{Gearbox} := \frac{J_L}{J_M \cdot GR^2} \tag{11}$$

When a gearbox is used, the effect of the gearbox on the output torque (multiplied by GR0 and rotational speed (divided by GR) must also be taken into account.

In addition to the cost estimate obtained from the model, there are some costs that are constant for all the cases. These fixed hardware costs are listed in Table 4. The drive mount fixtures costs are split into the purchase cost of the materials and the manufacture cost in the workshop. The end effector used was designed by Hoffman (2012) and is discussed in more detail in Section 5.1.3. The estimated purchase cost of this end effector is given in Table 4. The other components listed are based on purchasing the components new (RS Components, 2016).

Table 4: Fixed costs

Cost	Amount [R]
Drive mounts - materials	3927
Drive mounts - manufacture	25000
End effector	20000
2 x DAQ - NI USB-6501	3921
24 V power supply	304
48 V power supply	2998
Relays	500
Total:	56650

The costs that were excluded from the model were the design cost of the project and the cost of the process rack. However, a common practice is to take the design cost as a percentage of the system cost, normally at 100%.

5.2. Controller Design

The control architecture for this station is based on PROSA by Van Brussel *et al* (1998), with a few deviations.

The control architecture is programmed in C#, which was chosen as it is often used by the research group. The language's inheritance property makes it highly favourable in holonic control architectures as base holons can be created for each holon type. There are also standard C# libraries for most signal interfaces, as explained in Section 5.3.3.

In the next section the overall control architecture is described, followed by sections describing each holon type in more detail. The C# class name associated with each holon is given in the section heading in brackets. The legend for the flowcharts is given in Figure 18.

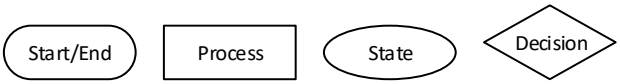


Figure 18: Flowchart legend

5.2.1. Control Architecture

The overall interactions between the holons can be seen in Figure 19 where the boundary between the station and the cell controller is also shown. In this research the cell controller is emulated using a windows form (Section 5.2.6) and interacts with the station through an external communications staff holon. There is another staff holon, named supervisor, that facilitates the start-up of the holons and the beginning and completion of order holons. This supervisor holon was included to allow for possible system optimisation, which is an adaptation from ADACOR (Leitao & Restivo, 2006).

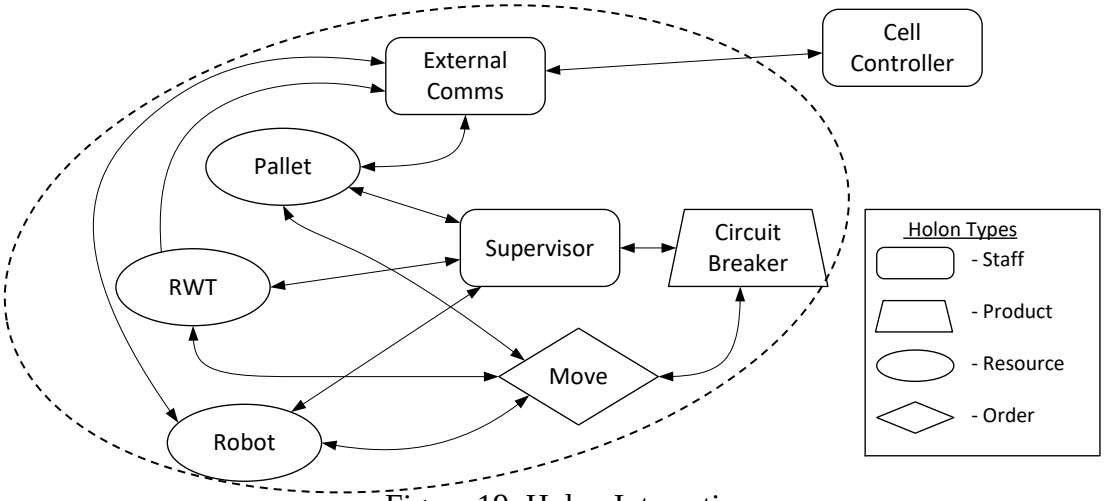


Figure 19: Holon Interactions

As is customary in the PROSA (Van Brussel *et al*, 1998) the order holon ("Move") performs a list of operations which it obtains from the relevant product holon ("Circuit Breaker"). Each of these operations requires one or more processes to be completed by one or more resource holons. The order holon is described in more detail in Section 5.2.4.

The resource holons in this system are the pallet holons, RWT holons (representing the process modules) and robot holon. Each resource holon receives a request for proposal (RFP) from the order holon and, if the resource can fulfil the required request, responds with a proposal. The order holon then decides which proposal is the best. The order holon then accepts the best proposal and the resource holon performs the required process. This is repeated until the order holon has finished all the required operations for the product.

Each holon in the architecture inherits its basic behaviours from a base holon class. The behaviour of this holon is shown in Figure 20. This shows that each holon executes a start-up procedure, followed by a loop where it processes any messages in its inbox, executes its behaviours, handles anything on its agenda and performs any low-level control. If the holarchy is to be shut down, it executes a shutdown procedure and ends.

A description of all the messages sent between holons is provided in Appendix B.

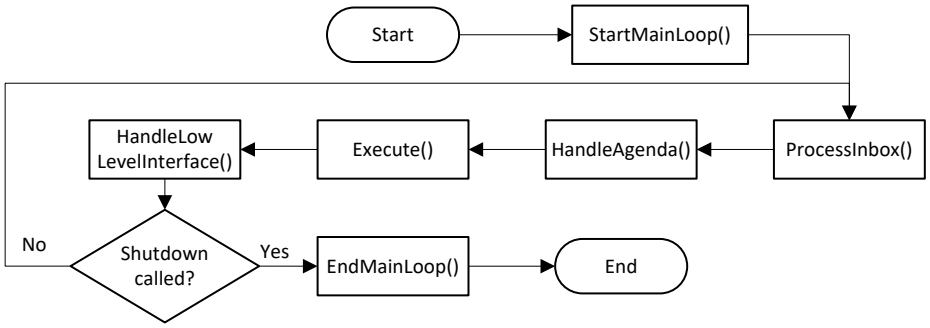


Figure 20: Base autonomous holon loop

5.2.2. SUPERVISOR (AHOLONSUPERVISOR)

The supervisor staff holon handles the starting of order and resource holons. At start-up, the resource holons are created based on the size of the system: the number of pallets and the size of the process rack.

The supervisor follows the behaviour illustrated in Figure 20. When the supervisor gets a message from the cell controller to add an order, it creates a new order holon and then sends that holon a message of what product the order holon must process. This order holon is described in more detail in Section 5.2.4. When the order holon requires a list of all current resource holons, the supervisor holon sends it a resource list. The supervisor then handles the completion and failure messages of the order holons.

This staff holon also manages the information flow associated with the movement of products around the system: when the robot holon has performed a move, it sends the move information to the supervisor which in turns sends messages to the relevant resource holons to update their product tables.

This idea of a supervisor holon is based on the ADACOR architecture, where the supervisor holon is tasked with optimising the system. This has not been implemented in this architecture, but the functionality can be added at a later stage. This can be done as the supervisor can keep track of the state of all the resource holons in the system.

5.2.3. EXTERNAL COMMUNICATIONS HOLON (MHOLONECH)

The external communications holon handles all communication between the cell controller and the station controller. This holon would normally include parsing the messages from the cell controller (in Hoffman's (2012) case these messages were XML strings), but this was not implemented here since the communication with the cell controller was out of the scope of the research. An overview of the messages the holon handles are listed below:

- Initialising communication with the cell controller
- Starting new orders and feedback on the order end status

- Registering of the resource holons so that the status of each resource holon can be displayed in the cell controller GUI as described in Section 5.2.6
- The movement of pallets in and out of the station

5.2.4. ORDER (MHOLONORDERMOVE)

This is the holon that handles the execution of a queue of operations. The holon states and its behaviour in each state are described in Table 5. The flow between the different states are shown in Figure 21.

Table 5: Holon states

State	**Description**
Dormant	Order holon is dormant.
WaitingForProductData	Sent a request for operation queue to product holon and awaiting response.
WaitingForDevList	Sent a request for a list of resource holons to supervisor holon and awaiting response.
WaitingForProposals	Sent an RFP to all resource holons. When proposals are received, the best proposals are selected.
WaitingForOperation ToComplete	Sends an accept to the best proposals and waits for the resource holon to send a reply when its processes are complete.

The order holon begins in a dormant state that awaits a request to process a product from the supervisor staff holon. Once a product request has been received, the task holon requests a list of operations the product requires to be performed from the product holon. Once a list of operations has been received the order holon then requests a list of the resource holons from the supervisor holon.

Next the order holon sends a request for proposal (RFP), that contains the information of operation that needs to be completed, to all the resources holons. The resource holons reply with proposals if they are able to perform the operations. Once all the resource holon proposal bids are received by the order holon, the best proposals are selected. This is done by looping through the bids and using the method *IsBetterThan*, contained in the bid class definition, to compare the current bid to the best bid and replacing the best bid if the current bid is superior. This *IsBetterThan* method selects the pallet with the most available products or fixtures as the best and the lowest process slot available, with preference given to a process slot in the same x-position as the pallet if there are multiple process slots at the same height.

Some operations require multiple resources and the order holon sends an accept to all required resource holons. The order holon then waits for the resource holons to finish their processes and when a complete is received from the resource, the order holon moves on the next operation until there are no more operations left in the operation queue.

The above procedure is mostly the standard behaviour for a PROSA (Van Brussel, et al., 1998) order holon. In this application, the operations that the order holon handles are:

- Find - A find operation does not contain an actual process a resource holon performs, but is a tool for selecting an available pallet and process module and booking the specific products or destinations.
- Move - A move operation contains the information for a robot to move a product from a pallet to a process module or from a process module back to a pallet. It contains the accepted bids for the process module and the pallet of the move, as well as the direction of the move.
- Process - A process operation contains the information needed for the processing of a product in the process module.

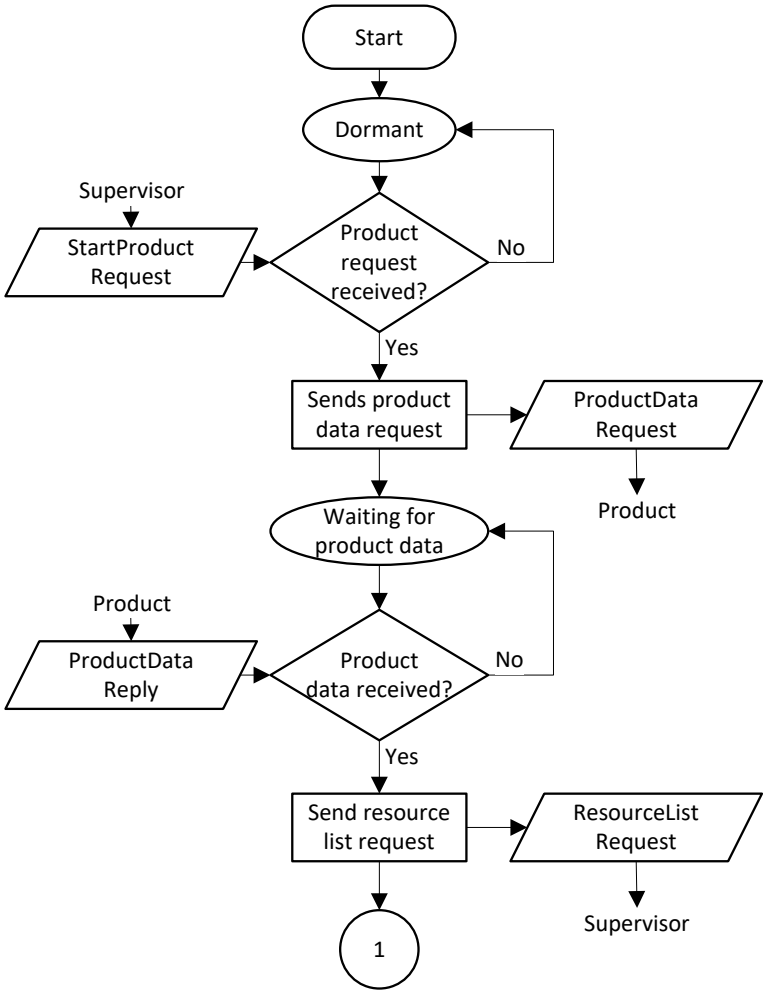


Figure 21 (a): Order holon state flow digram

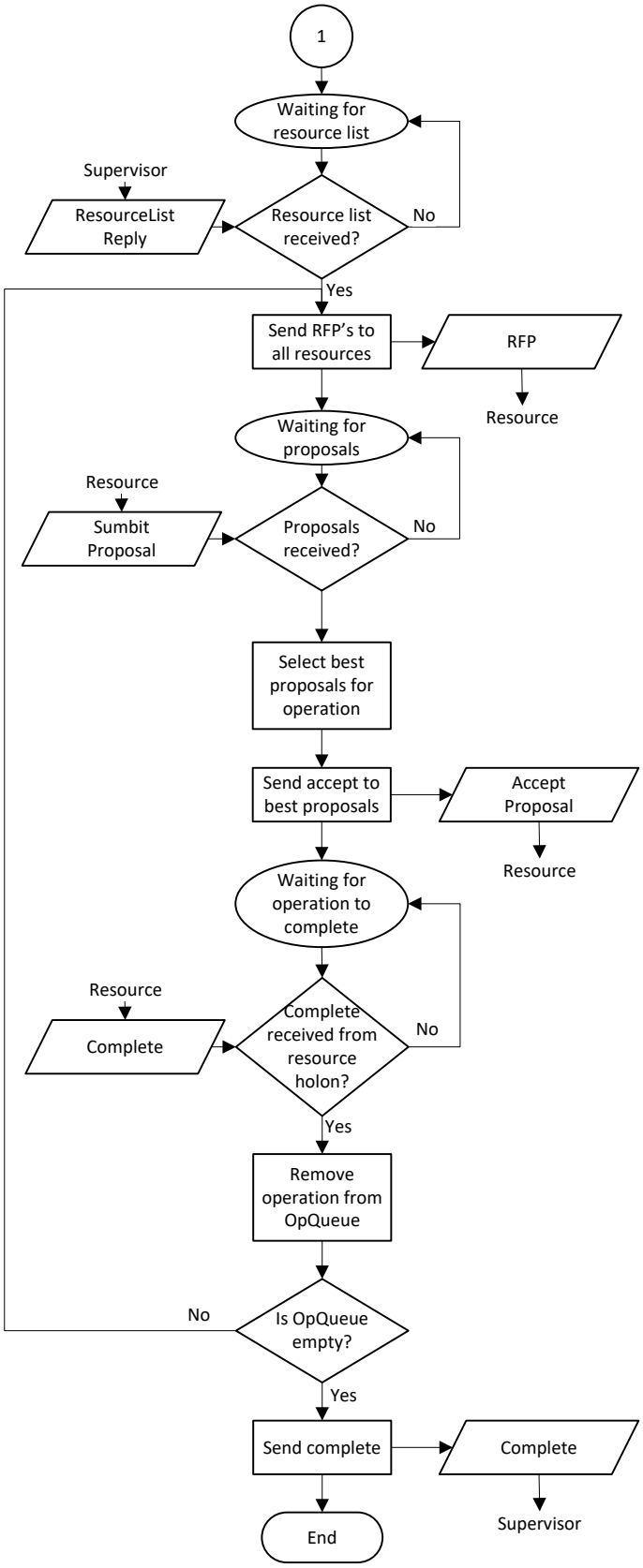


Figure 21 (b): Order holon state flow digram cont.

5.2.5. RESOURCE HOLONS

The resource holons are based on the autonomous holon class (Figure 20), but contains extra functionality, contained in the resource holon base class. Each resource holon has a generic inbox for processing RFPs and proposal accepts. When an RFP is received the holon checks whether it can fulfil the required operation and, if it can, sends a proposal back. If the proposal is accepted by the order holon, the resource holon performs the required operation.

5.2.5.1. Pallet (MHolonResourcePallet)

This resource holon models the conveyor pallets. The holon has three states that are described in Table 6. The pallet is assumed to have multiple fixtures that each contain a product. This is modelled in the holon using an integer table that contains the state of the products on the pallet, as well as each product's unique identification number. The holon also contains a pointer that keeps track of the active fixture position. This pointer is included as the pallet can only have products accessed from one side and therefore operates on a last-in, first-out basis.

Table 6: Pallet holon states

State	**Description**
None	No pallet on transverse conveyor.
In	Pallet is accepting products.
Out	Pallet has unprocessed products.

The product table is updated when a pallet is added or removed by the conveyor, which is done after receiving a message from the cell controller. The product table is also updated when the robot removes an unprocessed product from or places a processed product onto the pallet.

When an RFP is received, the resource holon checks whether it can either provide an unprocessed product or if it has an available empty fixture. It then responds with a proposal containing a bid that includes the current pallet fixtures table, the pallet position, the active fixture pointer and the number of available products or fixture positions.

If the pallet's proposal is accepted, the selected products are flagged as booked. The holon then waits for the supervisor holon to confirm the products have been moved from the pallet and then updates the fixture table to reflect the current state of the pallet.

5.2.5.2. RWT (MHolonResourceRWT)

This operational holon models the behaviour of the process modules, which in this case study are ramp wave testers (RWTs) and are referred to as such in this section. The states of the holon are described in Table 7 and the holon's operation is illustrated in Figure 22. Each RWT holon is designed to represent two physical RWT slots to comply with the end effector that moves two products at a time.

When an RFP is received, the holon checks if it is empty and submits a bid containing the RWT position. If the proposal is accepted the RWT slots are booked. The RWT then waits for a second accept message, with a test operation bid, which is sent to the RWT when a product is added, and testing starts.

In the laboratory implementation, the testing is simulated using a timeout that is activated by the second accept message. When the timeout has fired, a pass (95% chance) or fail is assigned to each breaker and a complete message is sent to the order holon.

Table 7: RWT holon states

State	**Description**
Empty	Process slots are empty
Booked	Process slots are booked for a move operation.
Processing	Process slots are currently processing products.
Finished	Process slots have processed products that need to be moved.

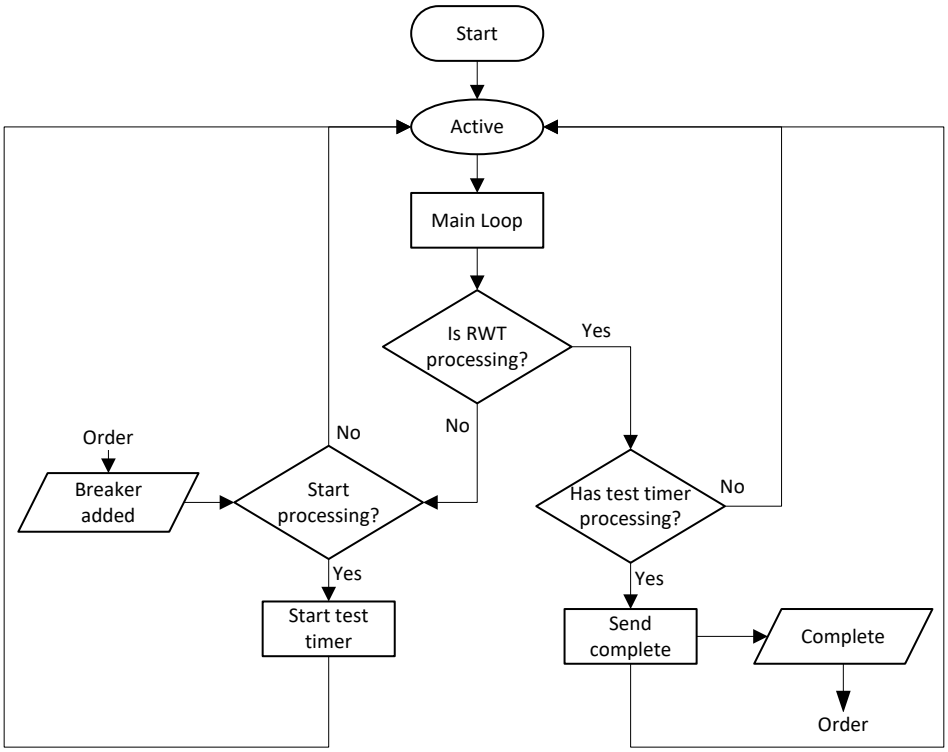


Figure 22: RWT state flow diagram

5.2.5.3. Robot (MHolonResourceRobot)

This resource holon handles the control of the Cartesian robot and gripper. This holon has three states that are described in Table 8. To prevent the robot from trying to move the drives when they are not properly initialised, the robot needs to receive

an “online” message from the cell controller to move it from the dormant to available state.

Whilst the robot is online, it accepts move operations from order holons. These move operations are placed in an operation queue and when the robot is available, it selects the move operation with the nearest starting position to the robot’s current position and performs this move. This behaviour is illustrated in Figure 23.

Table 8: Robot holon states

Name	**Description**
Dormant	Robot holon is running, but is not interfaced with hardware.
Available	Robot is connected to hardware and awaiting a move operation.
Moving	Robot is busy performing a move operation. Move operation steps shown in Figure 24.

The interaction between the robot holon and the physical hardware is done through a signal interface which is explained in more detail in Section 5.3.3. The status of the robot is read from the signal interface and stored in a Boolean table. The information stored in this table can be found in Appendix B-2.

The robot handles two different move operations. One places products from a pallet fixture into a process module and the other returns them to a pallet. The different steps that are performed in each move are shown in Figure 24. Not shown in the flow chart is an intermediary step between each pair of operations where the robot holon waits for confirmation from the robot controller that the operation has been completed.

The control of the robot can be broken down into two components: the gripper control and the Cartesian robot control. The gripper control is through changing the actuators valves control on the signal interface – see Section 5.3.3. The Cartesian moves are performed by calling the function, “Cartesian Move”, which accepts the parameters of an X, Y and Z position. These parameters refer to locations on a position table that is hardcoded onto the drive controllers. The processes involved in controlling the drives is explained further in Section 5.3.3.

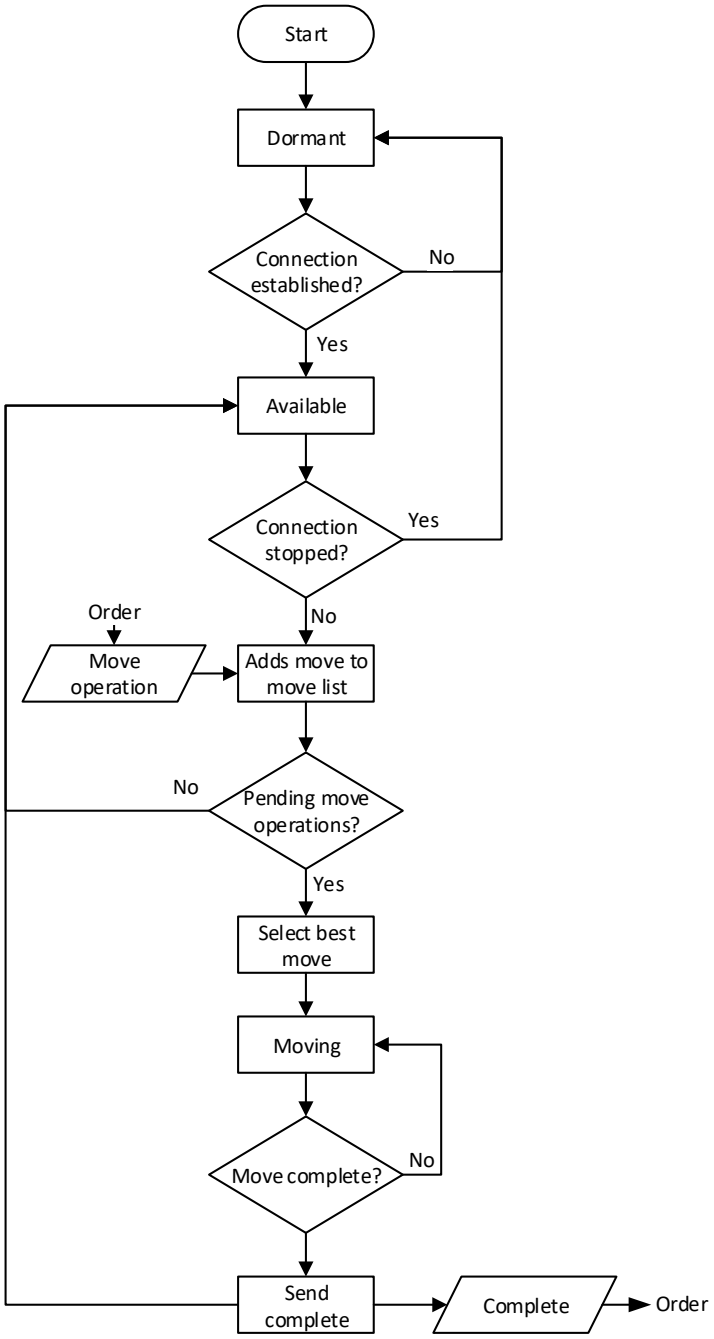


Figure 23: Robot states

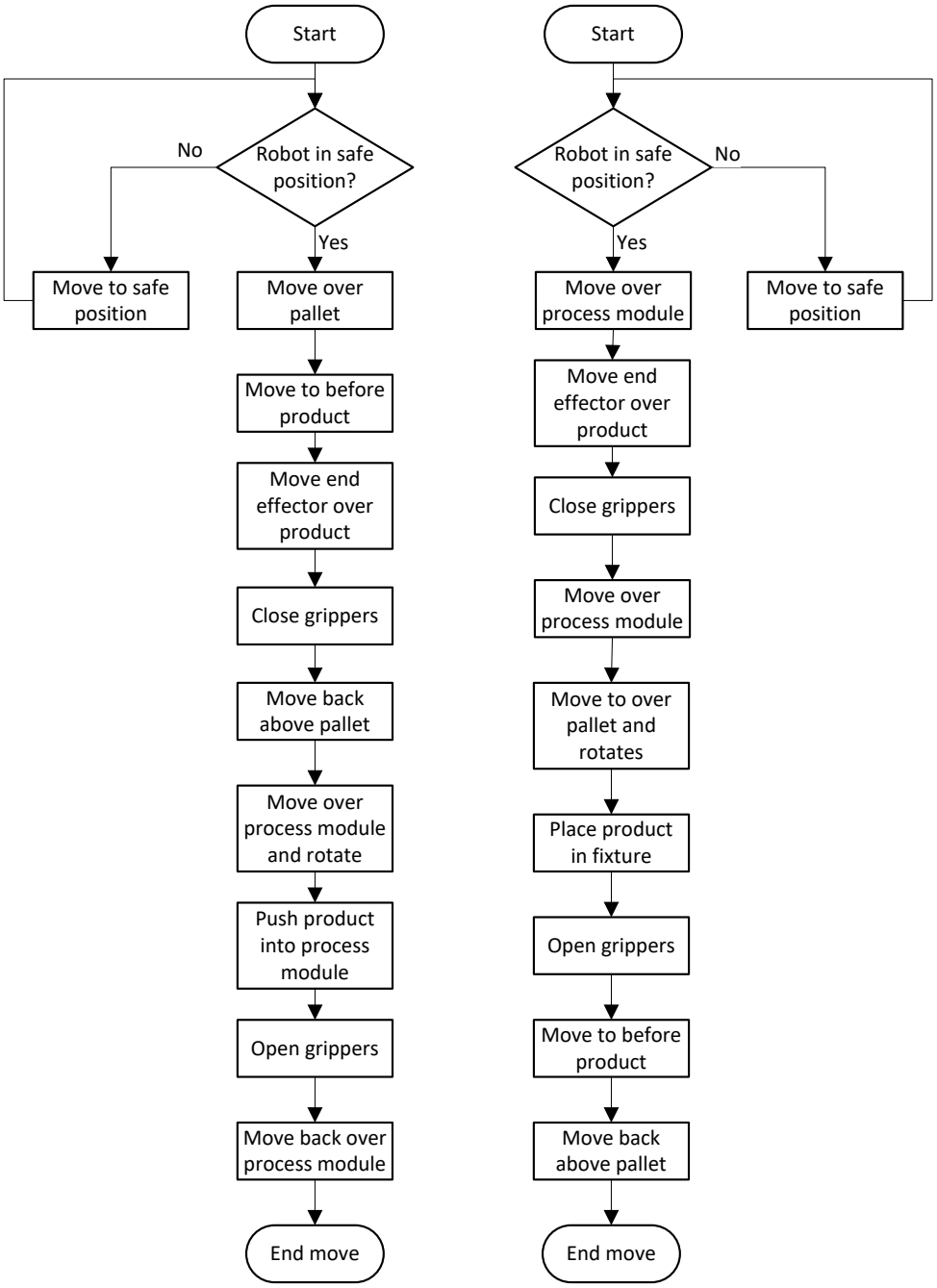


Figure 24: Robot move operation flow diagram

5.2.6. CELL CONTROLLER EMULATOR

This windows form simulates the information that would be sent to and from the cell controller and can be seen in Figure 25. It offers visual feedback of the status of the resource holons in the station. The messages the cell controller sends to the external communications holon are listed in Table 9.

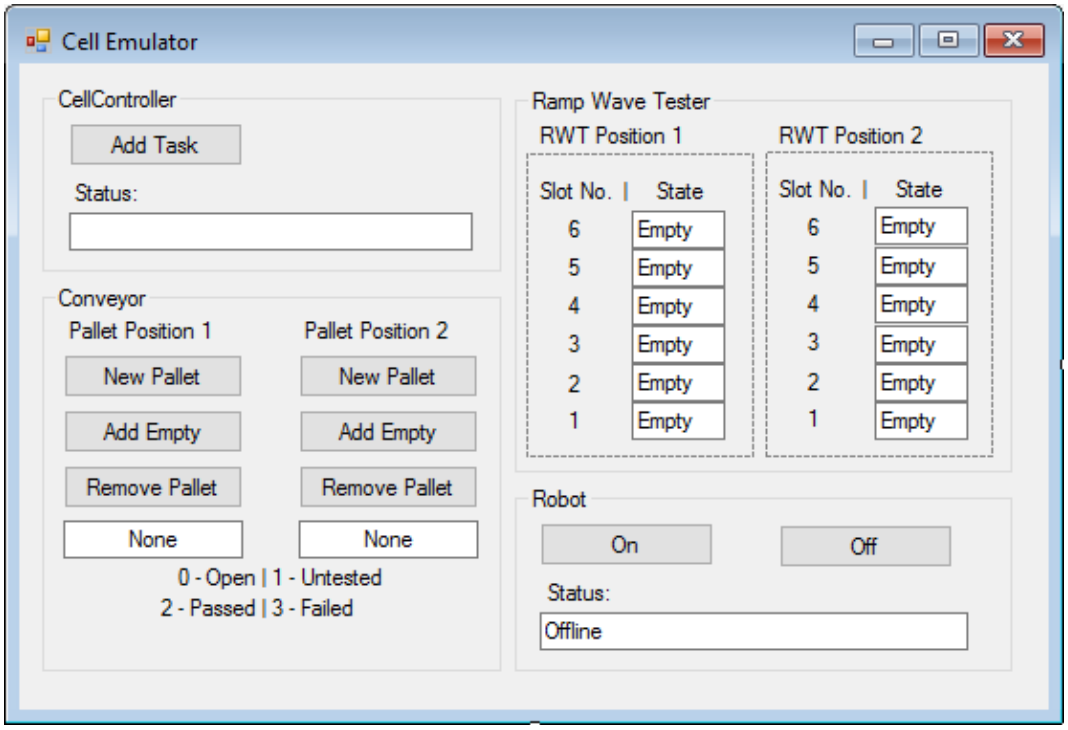


Figure 25: Cell Emulator GUI UPDATE

Table 9: Cell controller messages

Message Name	**Description**
AddTask	Sends message to start a new task.
InitialiseComms	Registers form with external communications holon.
PalletAdded/Removed	Simulates the addition and removal of pallets from the pallet table.
RobotOn/Off	Disables or enables the robot holon.

5.3. HARDWARE

The test setup was designed around FESTO (2016) drives that were available in the Automation Laboratory. There are five main sub-assemblies for the robot, i.e. the end effector, mounted swivel module, mounted x-drives, mounted y-drive and mounted z-drives. The 3D CAD models for each of these subassemblies can be found in Appendix C-3.

The degrees of freedom are numbered as the largest axis (Z) first to the smallest (swivel module) last. A mock-up of the ramp wave tester rack was also designed to represent the process modules. A limitation to the system was the lack of integrating with the conveyor system and therefore two pallet stations were designed to simulate two transverse conveyor end positions. The complete setup can be seen in Figure 26 with (a) showing the CAD model, (b) showing the laboratory setup and (c) showing a more detailed view of the laboratory end effector, process modules and pallet.

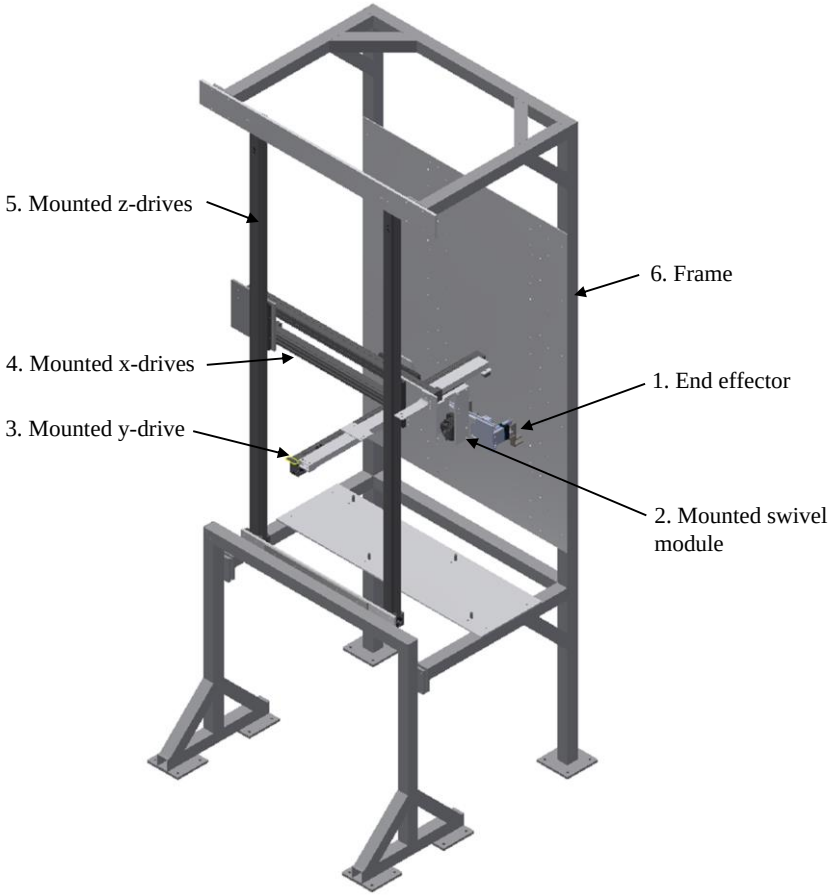
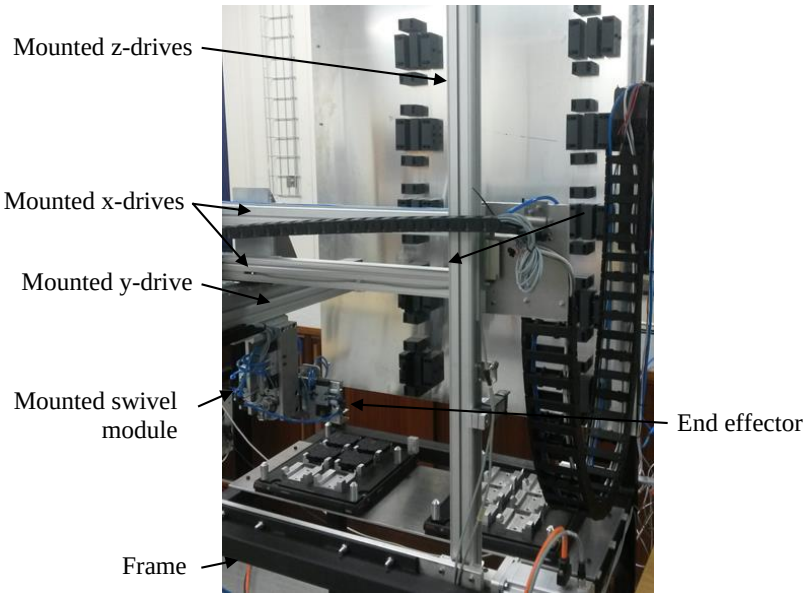
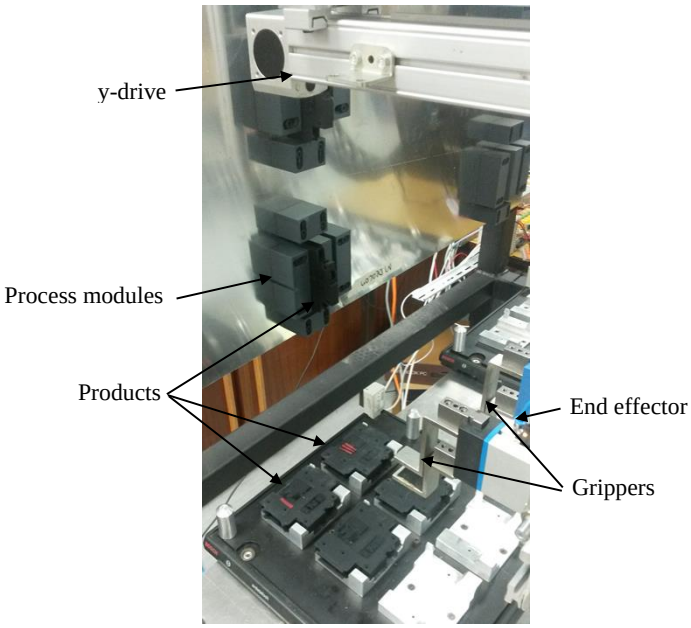


Figure 26: Experimental setup (a) CAD



(b)



(c)

Figure 26: (b) Laboratory setup, (c) End effector, pallet and process modules

5.3.1. SUB-ASSEMBLIES

Each subassembly and its function is briefly described in the following subsections. The force calculations for each sub assembly are shown in detail in Appendix A-2. All the parts were manufactured with dowel pin holes to ensure accurate positioning throughout the system.

5.3.1.1. End Effector

The end effector (Figure 27) was designed by Hoffman (2012). The end effector allows for two circuit breakers to be carried at a time, with individual control over each gripper. Its grippers can also be brought together so that the products can be picked up from the pallet where they are closely packed and pressed into the more widely spaced process modules.

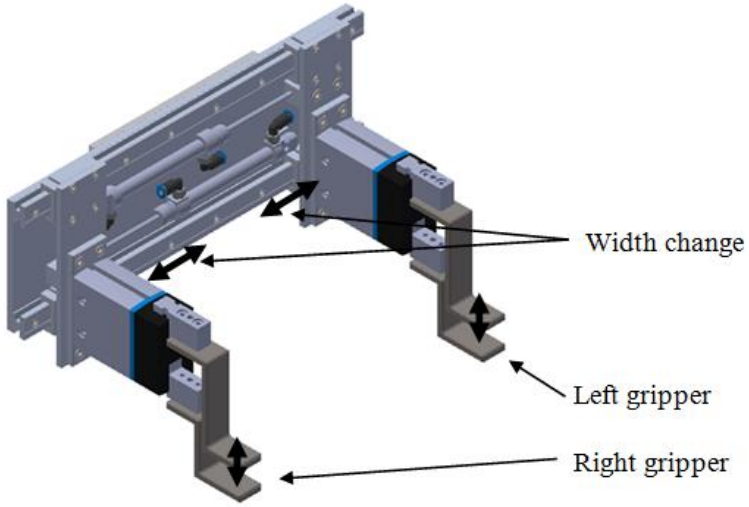


Figure 27: End Effector

5.3.1.2. Swivel Module

The fourth degree of freedom of the robot is the rotation of the products from their horizontal positions on the pallets to the vertical process modules. A pneumatic swivel module was chosen for this DOF as it is cheaper than using a drive and only two positions were needed. The swivel module was mounted to two side plates that were held together at the top by a plate that connects to the y-drive.

5.3.1.3. Mounted Y-Drive

For the third degree of freedom (the in and out motion towards the process modules) a Festo linear drive (DGE-ZR-RF (FESTO, 2016)) is used. The drive was mounted upside down to allow the swivel module to hang beneath it. An aluminium tube was used to support the drive. The drive is powered by a stepper motor (EMMS-AS-55 (FESTO, 2016)) and controller (CMMS-AS (FESTO, 2016)). This controller was operated in the digital position set mode. The positions that were programmed into the controller are listed in C-2: Controller Position Tables

Table 29 in Appendix C-2.

5.3.1.4. Mounted X-Drives

For the second degree of freedom (side-to-side) a pneumatic linear drive from Festo (DGC-G (FESTO, 2016)) was used. A pneumatic drive was used since the positioning is limited to only two x-positions. A support drive (FDG (FESTO,

2016)) was added to help reduce the bending moment around the x-axis caused when the weight of the gripper when the t-drive was fully extended. The slides of these two drives were connected with a bracket shown in Figure 28 and the y-drive was mounted below it.

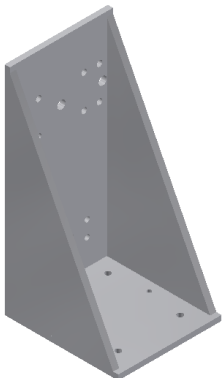


Figure 28: Support bracket

5.3.1.5. Mounted Z-Drives

The first degree of freedom (up-and-down) is also provided by two Festo DGE-ZR-RF (FESTO, 2016) drives mechanically coupled by a connecting shaft. These drives were powered by a stepper motor (EMMS-ST-87-M (FESTO, 2016)) and controller (CMMS-ST (FESTO, 2016)). The operation of these drives is the same as the y-drives and the position table can be found in Table 30 in Appendix C-2. These drives were connected on each end by an aluminium plate. Instead of mounting these drives along a frame, it was decided to use the drives themselves as structural members, and only mounting the bottom connecting plate to the frame.

5.3.1.6. Frame with Pallet Table and RWT Rack

The frame was designed out of mild steel tubing (50x50x2) and is split into two sections. The first section is the section the robot is mounted to and the second is the section that contains the pallet table and process modules. The connections between these two sections are adjustable to account for any warping of the frame during welding that could lead to the robot not being aligned with the process rack.

The pallet table was designed based on the distance between two transverse conveyors. This was done as the transverse conveyors in the laboratory were not long enough for use in this concept and the integration of the conveyor control was beyond the scope of this thesis.

The RWT slots were designed by Hoffman (2012). They allow for multiple different sized circuit breakers to be tested and use a spring-loaded mechanism to hold the products in place while the process is being simulated. The size of the process rack was set to be six modules high and two modules wide.

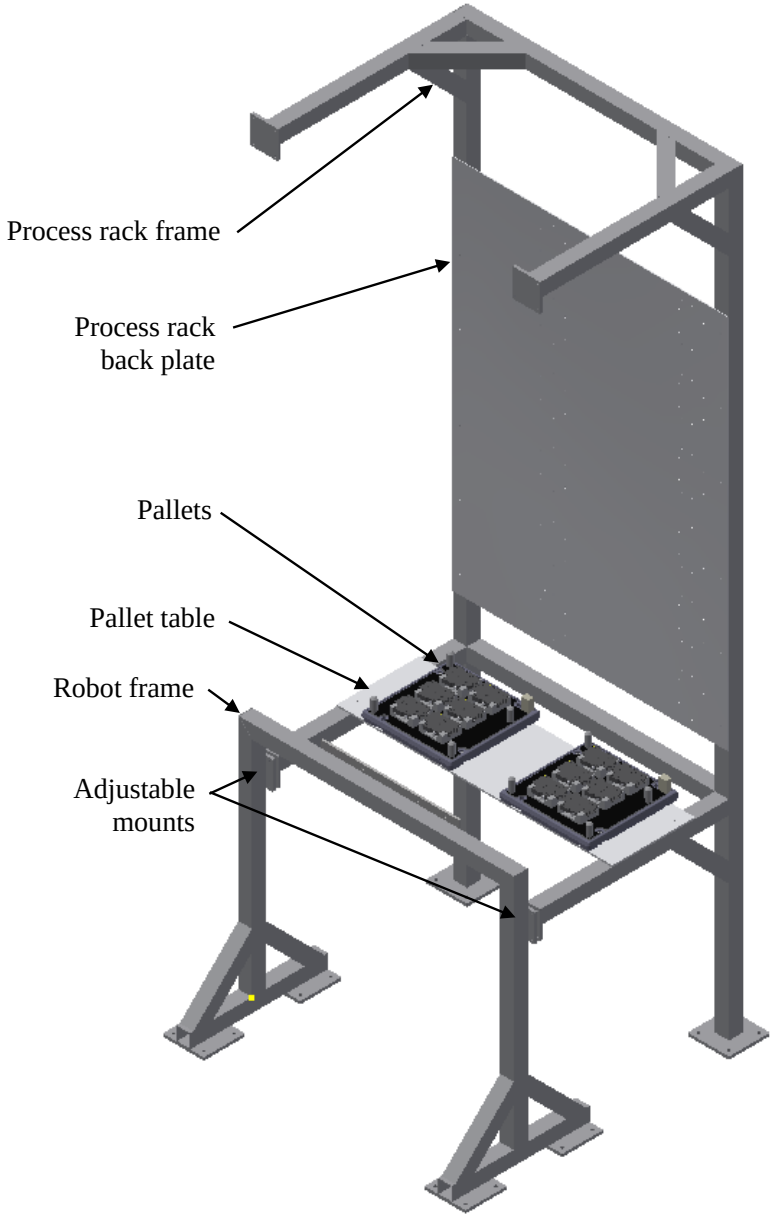


Figure 29: RWT and pallet setup

5.3.2. Other Design Features

All pneumatics and wiring were placed in energy chains on the three moving axes. The calculations for these energy chains can be found in Appendix C-4.

5.3.3. Signal Interfaces

A digital control interface between the holonic architecture running on the computer and the hardware was needed. The two choices considered were a programmable logic controller (PLC) and a digital acquisition device (DAQ). The option of a PLC would involve writing a program for the PLC to translate the signals from the computer and set the PLC's digital outputs. The DAQ, however, requires no extra

program as it connects directly to the computer and is accessed using prewritten C# libraries. There are also a large number of digital I/Os required, which were too many for the available PLCs in the Automation Laboratory. Therefore, a DAQ was selected for the signal interface.

The electronics setup is shown in Figure 30. The biggest problem with using a DAQ was that it is USB bus powered and therefore limited to 5 V. Therefore, relay boards were required to step up the outputs to 24V and voltage dividers (R_1 = 38 kΩ and R_2 = 10 kΩ) were used to bring the inputs signals down to 5 V. The relay boards and voltage divider used can also be seen in Figure 30. An emergency stop was also included to stop the drives in the chance of collision; this was wired to a "Stop" pin on the motor controllers.

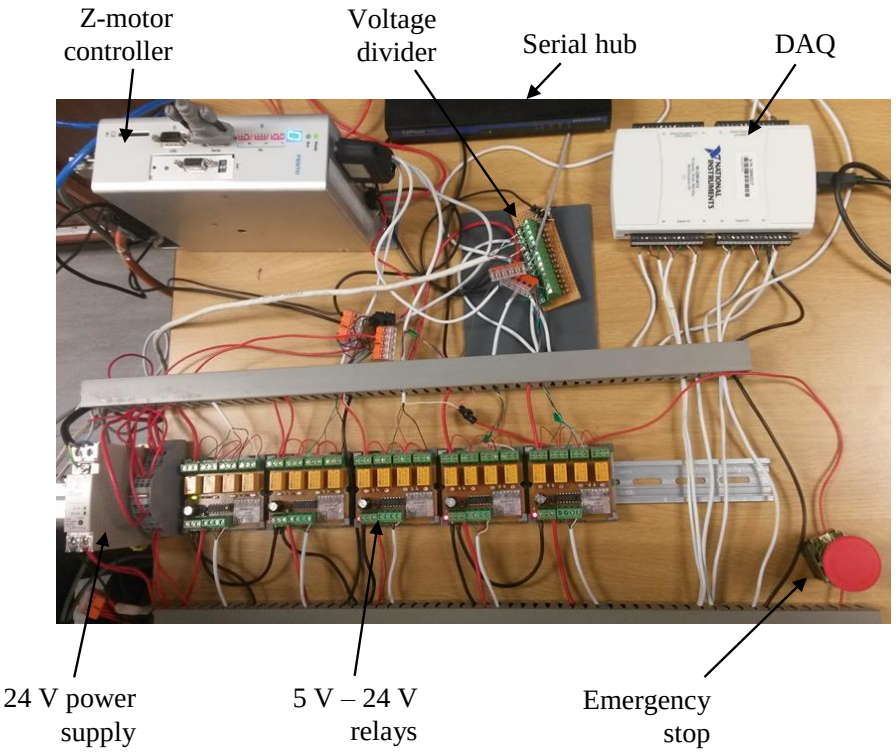


Figure 30: Electronics setup

The exact DAQ used was a National Instruments USB-6212 which also has the ability to handle analogue signals and is therefore over designed for the present application. A much cheaper DAQ, NI USB-6501, could have been used instead. The DAQ has a total of 32 digital I/Os which are mapped as shown in Table 28. The letter I or O after the port designation refers to the line being either input or output.

The DAQ was integrated into the holonic architecture in the robot holon. When the holon starts up data streams are created. These streams are written to and read from during each cycle of the holon, in the holons low level control section.

5.3.4. ROBOT CONTROLLER

A simple secondary program was written to initialise and shutdown the robot. The graphical user interface (GUI) for this program can be seen in FIGURE in Appendix C-2, along with operating instructions for the GUI. The purpose of this program is to ensure the robot is in a safe state before the holonic architecture attempts to use it.

6. MODEL VALIDATION

Before the model can be accepted, it needs to be validated. This is done by running a set of tests on an experimental setup based on the E-test station case study. These experimental results are then compared to the results of the model with the same parameters. This chapter presents the test strategy, a brief description of the experimental configuration and an interpretation of the results.

6.1. STRATEGY

To prove that the model is accurate, a number of test scenarios were considered. These scenarios were selected to give a range of comparable throughput rates that are each limited by a different factor. The three scenarios that will be considered are given in Table 10.

Table 10: Test scenarios

Design Variable	**Scenario 1 [s]**	**Scenario 2 [s]**	**Scenario 3 [s]**
$RackSize_x$	1	2	2
$RackSize_z$	6	4	6
Number of pallets	1	2	2

For each test scenario, a repeat point was selected. This repeat point is chosen as a point in the test cycle where the robot has reached the same point it was at previously, i.e. a full pallet of untested breakers. Each test scenario is run 5 times and an average of each robot move was calculated. Using these averages a total cycle time was calculated. The number of breakers that are moved in this cycle is divided by the length of the cycle to get a throughput rate that can be compared to the model.

A set of common, fixed design variables (given in Section 6.3) of the system were selected, based on the case study described in Section 3.1. There are also case dependant variables (given in Section 6.3) which were configured to the laboratory setup.

6.2. CONFIGURATION

The Automation Laboratory has a number of Festo linear drives (both belt driven and pneumatic) that were used in this research. The available size of these drives is limited to the 25 mm models. Refer to Figure 26 for images of the laboratory setup that was used.

Since this model was limited to the drives available in the lab, some of the drives do not meet the required design margins. This was considered acceptable in this application, because the tests are only running in short cycles and not continuous operation.

6.3. ANALYTICAL RESULTS

This section gives the results of the numerical model based on the experimental setup. The design variables used are listed in Table 11. For a full example calculation for both the throughput estimate and cost estimate refer to Appendix A.

Table 11: Design variables

Design Variables	**Value**
t_{pallet_switch} [s]	6
$t_{process}$ [s]	30
$t_{pickplace}$ [s]	4.5
d_{mod_x} [mm]	500
d_{mod_z} [mm]	230
$products_{pallet}$	6
$Gripper_{num\_teeth}$	2
a_x, a_y, a_z [m/s^2]	1.25
v_x, v_y, v_z [m/s^2]	0.5

The results of the throughput estimate for this validation case are given in Table 12. The throughput was rounded to the near whole product.

Table 12: Throughput estimate – validation case

	Scenario 1	**Scenario 2**	**Scenario 3**
Throughput	278	326	301

The drives and motors that were used from the Automation Laboratory are listed in Table 13 along with their limiting design margins and component cost.

Table 13: Cost estimate – experimental setup

Component	**Selection**	**Limiting design margin**	**Cost**
Swivel module	DSM-25	F_Z : 3.39	4075
X-drive	DGC-25-500-G	F_Z : 1.26	6539
X-support drive	FDG-25-500-ZR-RF	F_X : 0.78	12010
Y-drive	DGE-25-500-ZR-RF	F_Z : 2.30	16287
Y-motor	EMMS-AS-55	T : 12.78	32310
Z-drive	2 xDGE-25-1400-ZR-RF	Load : 1.24	29387
Z-motor	EMMS-ST-87	T : 1.20	18735
		Subtotal	119343

The overall cost of the laboratory setup is not of any significance, but is included for interest sake. This cost, including the fixed costs mentioned in Section 5.1.2, is R 157386.

It should be noted that the inertia ratios on the motors in this setup are too high. These inertia ratios should be reduced through the use of gearboxes; however, these were not available in the Automation Laboratory and therefore not considered.

The design margin on the load on the Z-drives was also below 1 in this setup. This was because the entire moved load was considered to be carried by a single drive as a worst-case scenario, however the load was split between two drives and so this was considered acceptable for the validation testing.

6.4. Experimental Results

This section contains the results of the experimental tests. The C# controller logged the time that each step was executed and the differences between each step were calculated and this data can be found in Table 31 in Appendix D. The average time for each robot move is given in Table 14 which is a summary of the data from Table 31. Using these move times, a cycle time, as described in Section 6.1, was calculated. Using this cycle time, along with the number of products moved in the cycle, the resultant throughput rates are calculated (Table 15). The accuracy of the cycle times is limited to the C# controller's ability to log each step.

Since the robot's speed was limited by the drives used, it was not able to utilise more than the 2 x 4 process module setup (scenario 2) and therefore when extra process modules were added (scenario 3) the throughput was not changed. As a result, scenario 2 and 3 are combined in Table 14.

Table 14: Average move times – laboratory setup

Move	**Scenario 1 [s]**	**Scenario 2 & 3 [s]**
Above pallet	1,384	2,06
Before breaker	1,098	1,65
Over breaker	0,737	1,51
Above pallet	1,545	1,82
Process module	2,849	1,93
Place	1,874	2,49
Moving to safe space	1,171	2,14
Front of process mod	4,014	1,76
Over product	2,302	1,29
Front of process mod	1,972	1,08
Above pallet	2,651	1,01
Above fixture	1,458	1,78
Drop in fixture	1,013	1,06
Above pallet	2,263	1,19
Total	**26.33**	**22.77**

Table 15: Throughput – laboratory setup

Design Variables	**Scenario 1**	**Scenario 2 & 3**
Products processed	6	6
Average cycle time	85.00	68.30
Throughput	254	316

6.5. INTERPRETATION

A comparison between the throughput rates of the model for the validation case and of the laboratory setup is given in Table 16.

Table 16: Throughput – comparison of model and laboratory results

Scenario	**Model throughput**	**Laboratory throughput**	**Difference [%]**
Scenario 1	278	254	-7,75
Scenario 2	326	316	-3.16
Scenario 3	301	316	4.75

These results show a close relation between the model throughput results and the laboratory setup for the electronic test station validation case.

It should also be noted that the throughput was largely limited by pickup and place operations, since the time to pick up or place a product was significantly larger than the main movements. This was a result of the complicated operation that was required to get the gripper under the product to hold the product together while moving.

7. MODEL APPLICATION

This section begins with giving the model results of the electrical test station case study, where the drives were selected using the model and acceptable design margins. There is then a comparison of these results with the results from Hoffman (2012) and a discussion on alternative applications where this model can be used.

7.1. CASE STUDY CONFIGURATION

The fixed design variables used for this model are listed in Table 11. The inputs and results of the throughput estimate for the preferred case are given in Table 17.

Table 17: Throughput estimate – preferred case

Design Variables	**Scenario 1**	**Scenario 2**	**Scenario 3**
a_x, a_y, a_z [m/s^2]	5		
v_x, v_y, v_z [m/s^2]	1		
$t_{pickplace}$ [s]	3.5		
$RackSize_x$	1	2	2
$RackSize_z$	6	4	6
Number of pallets	1	2	2
$t_{pickplace}$	6	6	6
Throughput	522	636	611

Since scenario two has the highest throughput, this scenario was used for the cost estimate. This table contains the total cost for each component as outputted from the model. A performance measure of base cost per throughput is calculated. This value can then be compared the model results to other material handling methods. The time $t_{pickplace}$ was shortened by 1 s to account for the faster drives.

Table 18: Cost estimate – preferred case, scenario 2

Component	**Selection**	**Limiting design margin**	**Cost [R]**
Swivel module	DSM-25	F_Z : 2.25	4075
X-drive	DGC-25-500-G	F_Z : 1.13	6539
X-support	FDG-40-500-ZR-RF	F_X : 1.23	16328
Y-drive	DGE-25-500-ZR-RF	F_X : 1.72	15967
Y-motor	EMMS-AS-55-S	T : 12.78	32310
Z-drive	2x DGE-25-1400-ZR-RF	Load : 0.97	44814
Z-motor	EMMS-ST-87-M	T : 1.08	19143
		Subtotal	**139175**
		Fixed cost	56650
		Total	**195825**
		Cost per part per hour	**307,9**

7.2. Comparison to 6 DOF Robot

The throughput that Hoffman (2012) achieved is given in Table 19. The main cost of his setup is the KUKA-KR16 6 DOF articulated arm robot and the other costs are the gripper and a Beckhoff PLC. Using these values a cost per hour per part is calculated. The costs in the Table 19 are obtained as the current retail price of the components.

Table 19: 6 DOF robot throughput and costing

Throughput	**960**
Cost of robot	250000
Gripper cost	20000
Beckhoff PLC with extra modules	34000
Total cost	304000
Cost per part per hour	**316,7**

The above shows that according to the model the Cartesian model provides a lower throughput than the 6 DOF robot, but is slightly cheaper than the 6-DOF robot.

7.3. Other Applications

A major limitation to the test cases is the pickup and place times being much longer than the overall move times between pallets and process modules. As a result, there is not much change in average throughput when adapting the process module size. If the pickup and place operations of a process station were quicker, the model would vary more and therefore be of more assistance in sizing a process station.

Another factor to look at is the throughput increase involved in the addition of the x-axis (double process module widths). If the x-axis was removed, not only is the overall cost is reduced, but so is the loading on the z-axis. If the X-axis was removed, the cost per throughput for the case study configuration rises to 331,3. However, this does not take into account the fact that the z-drives may be smaller cause of the reduced load.

One other application could be DNA testing, where a single source material needs to be deposited in multiple test tubes and left for a set amount of time. This model could assist in sizing the number of test tubes in the station to optimise the throughput.

8. CONCLUSION

This thesis considered alternative material handling methods to that of a 6 DOF articulated arm robot in a reconfigurable manufacturing process station. The concept of using a Cartesian robot was selected for detailed evaluation. This evaluation involved creating a model that estimates a throughput rate and cost for the robot that meets the process station's specifications.

As a case study, an electrical test station (Hoffman, 2012) was considered as the process station. The case study was used as a basis of the station's requirements, but the model can be applied to similar process stations.

To prove the model, a laboratory test station was designed and built, including the control of the station. The station was designed using Festo linear drives, motors and other actuators that were available in the Automation Laboratory. The controller for the station was implemented in C# and was based on the PROSA holonic architecture (Van Brussel, et al., 1998). This architecture allowed for the easy reconfiguring of the process module rack size and has the ability to adapt to multiple product variants. This test station did not meet the required design margins, but was still used to verify the model at reduced speeds. The differences in throughput between the laboratory test case and the model were under 8% over three different test scenarios.

When compared to the 6 DOF articulated robot arm (Hoffman, 2012) it was shown that the maximum throughput of a single Cartesian robot using the selection of Festo products was less than the 6 DOF robot. However, if the cost per throughput was compared, the Cartesian robot is more efficient.

This thesis demonstrates that the throughput and cost model developed here is suitable for evaluating alternative Cartesian robot configurations in process stations similar to the case study. The controller developed here is also suitable for alternative robot configurations due to its holonic architecture.

Reviewing the work presented in this thesis, the following recommendations can be made for further research:

- Different manufacturers, other than Festo, could be considered. These manufacturers could perhaps have cheaper or more suitable products for the application. The inclusion of gearboxes into the model should also be added.
- The supervisor staff holon that was included could be improved upon by adding optimisation. A possible optimisation that could be added is to the pairing of products from the pallet with a process module that would reduce the overall move time. It can also assist the robot in choosing which moves to execute first. Both of these choices were handled very simply in the order and robot holons, respectively, in this thesis.
- Further research could be conducted into the station's reconfigurability and evaluated against the characteristics detailed in Section 2.1, especially the architecture's diagnosability and error handling.

9. REFERENCES

Automation, E., 2016. *LTV - Load Transfer Vehicle AGV.* [Online]
Available at: http://www.egemin-automation.com/en/automation/material-handling-automation_ha-solutions_agv-systems_agv-types/load-transfer-agv
[Accessed 20 Novemeber 2016].

Bi, Z. M., Lang, S. Y. T., Shen, W. & Wang, L., 2007. Reconfigurable Manufacturing Systems: The State of the Art. *International Journal of Production Research,* Volume 46:4, pp. 967-992.

Crowson, R., 2006. *Assembly Processes: finishing, packaging, and automation.* 1st ed. New York, NY: Taylor & Francis Group.

FESTO, 2016. *FESTO Product Catalog.* [Online]
Available at: https://www.festo.com/cat/en-za_za/products
[Accessed 2016].

Hoffman, A., 2012. *IEC 61131-3-Based Control of a Reconfigurable Manufacturing Subsystem,* Stellenbosch University: Mechanical and Mechatronic Engineering Thesis.

Koren, Y. et al., 1992. Reconfigurable Manufacturing Systems. *Annals of the CIRP,* 48(2), pp. 527-540.

Kruger, K., 2013. *Control of the Feeder for a Reconfigurable Assembly System,* Stellenbosch University: MSc Eng. Thesis. Department of Mechanical and Mechatronic Engineering.

KUKA, 2016. *KR 16-2.* [Online]
Available at: http://www.kuka-robotics.com/en/products/industrial_robots/low/kr16_2/start.htm
[Accessed 20 Nov 2016].

Leitao, P. & Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry,* Issue 57, pp. 121-130.

Nof, S. Y., 1999. *Handbook of Industrial Robotics.* 2nd ed. New York, NY: Wiley & Sons, Inc.

Omron Adept Technologies, Inc, 2016. *Adept Cobra SCARA Robots - Cobra s800.* [Online]
Available at: http://www.adept.com/products/robots/scara/cobra-s800/downloads
[Accessed 20 November 2016].

RobotWorx, 2016. *What is an industrial robot?.* [Online]
Available at: https://www.robots.com/faq/show/what-is-an-industrial-robot
[Accessed 20 November 2016].

Sequeira, M., 2008. *Conceptual Design of a Fixture-Based Reconfigurable Spot Welder, MEng Thesis,* Stellenbosch University: MSc Eng. Thesis. Department of Mechanical and Mechatronic Engineering.

Setchi, R. M. & Lagos, N., 2004. Reconfigurability and Reconfigurable Manufacturing Systems - State-of-the-art Review. *IEEE International Conference of Industrial Informatics: Collaborative automation – one key for intelligent industrial environments.*

Van Brussel, H., Wyns, J., Valckenaers, P. & Bongaerts, L. &. P. P., 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry,* Volume 37, pp. 255-274.

Vaughn, R., 2013. *The Difference between Cartesian, Six-Axis, and SCARA Robots.* [Online]
Available at: http://machinedesign.com/motion-control/difference-between-cartesian-six-axis-and-scara-robots
[Accessed 20 November 2016].

APPENDIX A - MODEL

This appendix includes all the MathCAD code. The first section contains the throughput estimate, the second contains the force calculations and cost estimate and the third section contains other smaller calculations.

A-1: THROUGHPUT ESTIMATE

The purpose of the throughput estimate was described in more detail in Section 5.1.1. The code is split into two sections: the input section and the calculations section. In the input section, the design and design independent variables listed in Table 1 are entered. The calculation section then uses those input values and calculates a throughput for the system using the formulas described in Section 5.1.1. The code for the validation case, scenario 2 is given on the next page.

The nomenclature for the different variables used in this section is listed in Table 20.

Table 20: Throughput estimate MathCAD nomenclature

Variable	**Unit**	**Description**
a_x	m/s^2	Acceleration of drive in x-direction
a_y	m/s^2	Acceleration of drive in y-direction
a_z	m/s^2	Acceleration of drive in z-direction
d_{acc}	m	Distance covered while accelerating
d_{ave}	m	Average distance needed to move in specific distance
d_{ave_y}		Distance for y-direction
d_{mod_x}	m	Distance between modules in the x-direction
d_{mod_z}	m	Distance between modules in the z-direction
d_{rem}	m	Remaining distance to be covered at design velocity
gripper	-	Number of products held by gripper
no_{moves_pallet}	-	Number of move operations that can be performed from one pallet
$no_{moves\_process}$	-	Number of move operations that can be performed during one process cycle
$products_{pallet}$	-	Number of products per pallet
$RackSize_x$	-	Number of process modules in the x-direction
$RackSize_z$	-	Number of process modules in the z-direction
t_{acc}	s	Time accelerating before design velocity is reached in specified direction
t_{cycle_move}	s	Time to move in specific direction
$throughput_{hour}$		Products processed per hour
$throughput_{sec}$	/s	Products processed per second
t_{pallet_switch}	s	Time taken for pallet to be switched by the conveyor
$t_{pickplace}$	s	Time taken for pickup or place operations
$t_{process}$	s	Product process time
t_{rem}	s	Time to cover remaining distance at design velocity
v_x	m/s	Maximum velocity of drive in x-direction
v_y	m/s	Maximum velocity of drive in y-direction
v_z	m/s	Maximum velocity of drive in z-direction

Input

Design variables

$RackSize_z := 4$ $RackSize_x := 2$ $gripper := 2$

$a_x := 1.25 \frac{m}{s^2}$ $a_y := 1.25 \frac{m}{s^2}$ $a_z := 1.25 \frac{m}{s^2}$

$v_x := 250 \frac{mm}{s}$ $v_y := 500 \frac{mm}{s}$ $v_z := 250 \frac{mm}{s}$

Design independent variables

$t_{pallet\_switch} := 6\ s$ $t_{pickplace} := 4.5\ s$ $t_{process} := 30\ s$

$d_{mod\_x} := 500\ mm$ $d_{ave\_y} := 170\ mm$ $d_{mod\_z} := 230\ mm$

$products_{pallet} := 6$

Calculations

$d_{ave\_z} := d_{mod\_z} \cdot \frac{RackSize_z}{2}$ Average move distance

$t_{acc\_z} := \frac{v_z}{a_z}$ Time to reach max velocity

$d_{acc\_z} := 0.5 \cdot a_z \cdot t_{acc\_z}^{\ 2}$ Distance to reach max velocity

$d_{rem\_z} := d_{ave\_z} - 2 \cdot d_{acc\_z}$ Remaining distance

$t_{rem\_z} := \frac{d_{rem\_z}}{v_z}$ Time to cover remaining distance

$t_{cycle\_move\_z} := 2 \cdot t_{acc\_z} + t_{rem\_z} + 2 \cdot t_{pickplace} = 11.04\ s$ Time to move in one direction

$t_{acc\_y} := \frac{v_y}{a_y}$

$d_{acc\_y} := 0.5 \cdot a_y \cdot t_{acc\_y}^{\ 2}$

$d_{rem\_y} := d_{ave\_y} - 2 \cdot d_{acc\_y}$

$t_{rem\_y} := \frac{d_{rem\_y}}{v_y}$

$t_{cycle\_move\_y} := 2 \cdot t_{acc\_y} + t_{rem\_y} + 2 \cdot t_{pickplace} = 9.74\ s$

$$t_{cycle\_move} := \begin{cases} 2 \cdot t_{cycle\_move\_y} & \text{if } t_{cycle\_move\_z} < t_{cycle\_move\_y} \\ 2 \cdot t_{cycle\_move\_z} & \text{else} \end{cases} + \begin{cases} \frac{gripper \cdot t_{pallet\_switch}}{products_{pallet}} & \text{if } (RackSize_x = 1) \\ 0\ s & \text{else} \end{cases} = 22.08\ s$$

$$no_{moves\_process} := \frac{t_{process}}{t_{cycle\_move}} = 1.359$$

$$no_{moves\_pallet} := \frac{products_{pallet}}{gripper} = 3$$

$$throughput_{sec} := \begin{cases} \dfrac{gripper}{t_{cycle\_move}} & \text{if } \left(no_{moves\_process} \cdot gripper\right) < RackSize_z \cdot RackSize_x \\ \dfrac{\left(RackSize_z \cdot RackSize_x\right)}{t_{process}} & \text{else} \end{cases} = 0.091 \ \frac{1}{s}$$

$$throughput_{hour} := throughput_{sec} \cdot \left(60 \ min\right) = 326.087$$

A-2: COST ESTIMATE AND FORCE CALCULATIONS

The purpose of the cost estimate was described in more detail in Section 5.1.2. The cost estimate starts with an input page where the target kinematics are entered and the system size is defined. This is followed by selection data of the different bought components for multiple sizes. This data was obtained from the relevant catalogs and can be expanded upon if more products need to be considered.

The rest of the calculations are performed per subassembly. Each subassembly has a free body vector diagram that shows the locations of each parts center of gravity and a distance vector from the subassembly interface with the next subassembly and the center of gravity. The design distances are inputted followed by the masses of each part in the subassembly. This data is put into position and force vectors. Using these vectors the reaction forces at the subassembly interface are calculated. If a bought component selection is required at the interface this is done by choosing a component size and checking if the various safety factors are met.

The costs that are included here were obtained from the Festo website.

Table 21: Throughput estimate input variables

Variable	**Unit**	**Description**
a_x	m/s^2	Acceleration of drive in x-direction
a_y	m/s^2	Acceleration of drive in y-direction
a_z	m/s^2	Acceleration of drive in z-direction
d_{mod_x}	m	Distance between modules in the x-direction
d_{mod_z}	m	Distance between modules in the z-direction
DrivingT	N.m	Maximum driving torque of drive
d_y		Distance for y-direction
$d_{z\_toPallet}$	m	Distance between pallet and lowest process module in the z-direction
F	N	Force
F_v		Combined loading safety factor
gripper	-	Number of products held by gripper
I	$kg.m^2$	Moment of inertia
J	$kg.cm^2$	Effective enertia
Load	kg	Maximum load weight of drive
M	N.m	Moment
m	kg	Mass of part
Price	R	Cost of part
$products_{pallet}$	-	Number of products per pallet
R	N	Reaction force
r	m	Position vector
$RackSize_x$	-	Number of process modules in the x-direction
$RackSize_z$	-	Number of process modules in the z-direction
SF	-	Safety factor
$Stroke_x$	m	Stroke of drive in x-direction
$Stroke_y$	m	Stroke of drive in y-direction
$Stroke_z$	m	Stroke of drive in z-direction
T	N.m	Torque

Inputs:

Target kinematics:

$a_x := 1.25 \ \frac{m}{s^2}$ $a_y := 5 \ \frac{m}{s^2}$ $a_z := 5 \ \frac{m}{s^2}$

$\alpha_{DSM} := \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} \frac{rad}{s^2}$

Station setup variables:

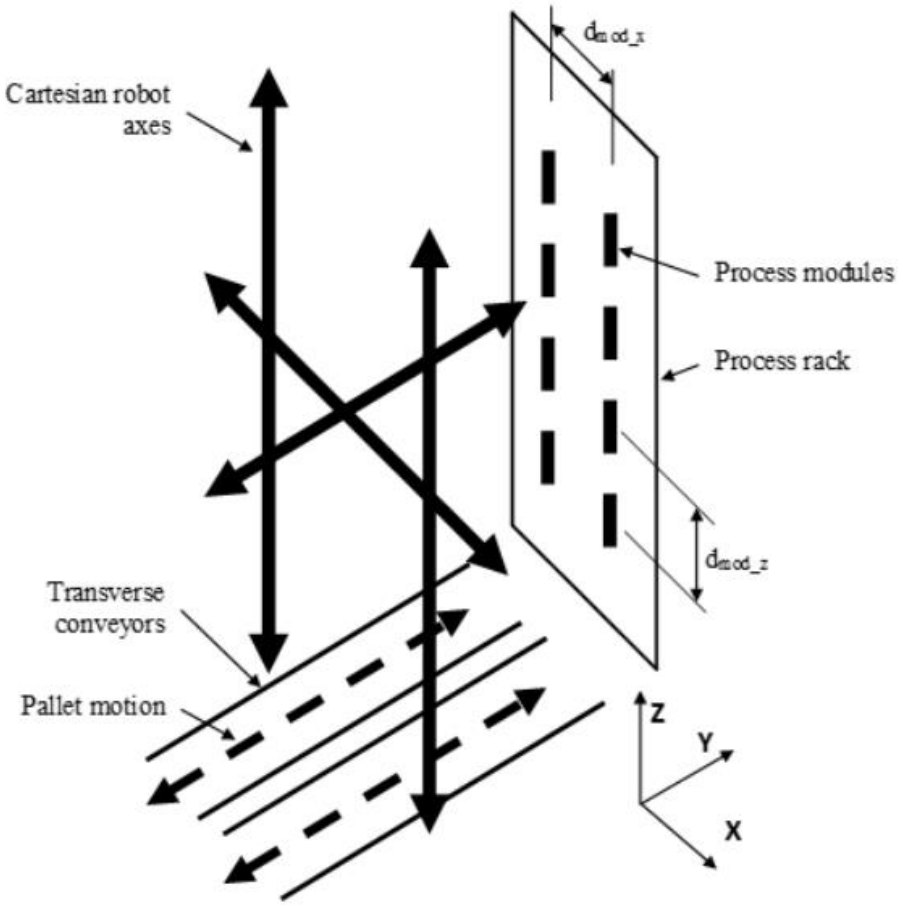
$RackSize_x := 2$ $d_{mod\_z} := 230 \ mm$ $d_{z\_toPallet} := 220 \ mm$

$RackSize_z := 6$ $d_{mod\_x} := 500 \ mm$ $gripper := 2$

$Stroke_x := (RackSize_x - 1) \cdot d_{mod\_x} = 0.5 \ m$

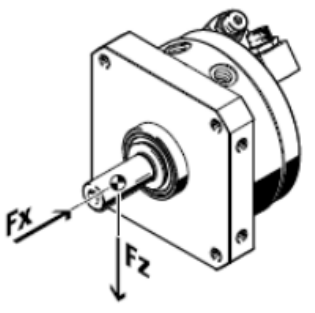
$Stroke_z := (RackSize_z - gripper) \cdot d_{mod\_z} + d_{z\_toPallet} = 1.14 \ m$

$Stroke_y := 400 \ mm$

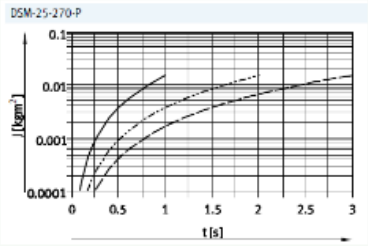


Festo datasheet info and prices

Specifications for FESTO DSM-...-270-FW-P-A-B from datasheet



Mass moment of inertia J as a functipn of swivel time t
With elastic cushioning components (P)



$$F_{X\_DSM}(Size) := \begin{cases} 30\ N & \text{if } Size = 16 \\ 50\ N & \text{else if } Size = 25 \end{cases}$$

$$F_{Z\_DSM}(Size) := \begin{cases} 75\ N & \text{if } Size = 16 \\ 120\ N & \text{else if } Size = 25 \end{cases}$$

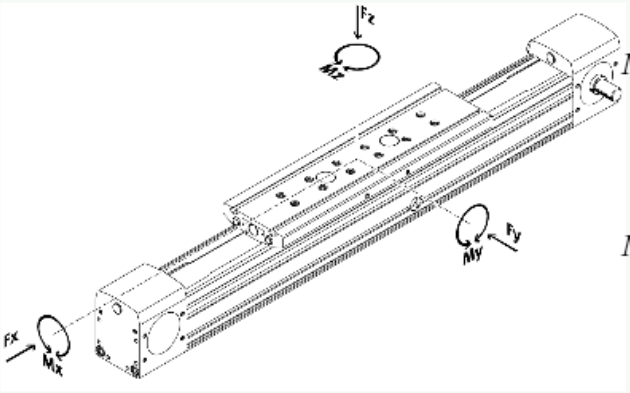
$$T_{DSM}(Size) := \begin{cases} 2.5\ N \cdot m & \text{if } Size = 16 \\ 5\ N \cdot m & \text{else if } Size = 25 \end{cases}$$

$$m_{DSM}(Size) := \begin{cases} 0.51\ kg & \text{if } Size = 16 \\ 0.725\ kg & \text{else if } Size = 25 \end{cases}$$

$$Price_{DSM}(Size) := \begin{cases} 3150 & \text{if } Size = 16 \\ 3475 & \text{else if } Size = 25 \end{cases} + 600$$

Includes estimate cost of cabling and sensors

Specifications for FESTO DGE-...-ZR-RF-GK from datasheet



$$M_{X\_DGE}(Size) := \begin{cases} 7\ (N \cdot m) & \text{if } Size = 25 \\ 18\ (N \cdot m) & \text{if } Size = 40 \end{cases}$$

$$M_{Y\_DGE}(Size) := \begin{cases} 15\ (N \cdot m) & \text{if } Size = 25 \\ 60\ (N \cdot m) & \text{if } Size = 40 \end{cases}$$

$$F_{X\_DGE}(Size) := \begin{cases} 260\ N & \text{if } Size = 25 \\ 610\ N & \text{if } Size = 40 \end{cases}$$

$$M_{Z\_DGE}(Size) := \begin{cases} 15\ (N \cdot m) & \text{if } Size = 25 \\ 90\ (N \cdot m) & \text{if } Size = 40 \end{cases}$$

$$F_{Y\_DGE}(Size) := \begin{cases} 150\ N & \text{if } Size = 25 \\ 300\ N & \text{if } Size = 40 \end{cases}$$

$$Load_{DGE}(Size) := \begin{cases} 15\ kg & \text{if } Size = 25 \\ 30\ kg & \text{if } Size = 40 \end{cases}$$

$$F_{Z\_DGE}(Size) := \begin{cases} 150\ N & \text{if } Size = 25 \\ 300\ N & \text{if } Size = 40 \end{cases}$$

$$DrivingT_{DGE}(Size) := \begin{cases} 3.7\ (N \cdot m) & \text{if } Size = 25 \\ 12.1\ (N \cdot m) & \text{if } Size = 40 \end{cases}$$

$$m_{DGE}(Size, Stroke) := \begin{cases} (2.61 + 3 \cdot Stroke \div m)\ kg & \text{if } Size = 25 \\ (7.75 + 6.1 \cdot Stroke \div m)\ kg & \text{if } Size = 40 \end{cases}$$

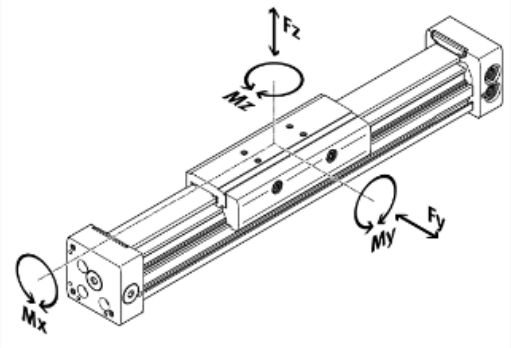
$$d_{DGE}(Size) := \begin{cases} 28.65\ mm & \text{if } Size = 25 \\ 39.79\ mm & \text{if } Size = 40 \end{cases}$$

$$J_{A\_DGE}(Size, Stroke, Load) := \begin{cases} (1.75 + 0.188 \cdot Stroke \div m + 2.052 \cdot Load \div kg)\ kg \cdot cm^2 & \text{if } Size = 25 \\ (9.89 + 0.933 \cdot Stroke \div m + 3.958 \cdot Load \div kg)\ kg \cdot cm^2 & \text{if } Size = 40 \end{cases}$$

$$Price_{DGE}(Size, Stroke) := \begin{cases} 13190 + 3.193 \cdot Stroke \div mm & \text{if } Size = 25 \\ 17808 + 4.557 \cdot Stroke \div mm & \text{if } Size = 40 \end{cases} + 1500$$

Includes estimate cost of cabling and 2 sensors, excludes motor and controller

Specifications for FESTO DGC-G... from datasheet



$$F_{X\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 180\ N \\ \text{else if } Size = 40 \\ \quad 610\ N \end{cases}$$

$$M_{X\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 4\ (N \cdot m) \\ \text{else if } Size = 40 \\ \quad 12\ (N \cdot m) \end{cases}$$

$$F_{Y\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 180\ N \\ \text{else if } Size = 40 \\ \quad 610\ N \end{cases}$$

$$M_{Y\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 20\ (N \cdot m) \\ \text{else if } Size = 40 \\ \quad 60\ (N \cdot m) \end{cases}$$

$$F_{Z\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 540\ N \\ \text{else if } Size = 40 \\ \quad 610\ N \end{cases}$$

$$M_{Z\_DGC}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 5\ (N \cdot m) \\ \text{else if } Size = 40 \\ \quad 25\ (N \cdot m) \end{cases}$$

$$m_{DGC}(Size, Stroke) := \begin{cases} \text{if } Size = 25 \\ \quad (1.004 + 5.4 \cdot Stroke \div m)\ kg \\ \text{else if } Size = 40 \\ \quad (7.0 + 6.1 \cdot Stroke \div m)\ kg \end{cases}$$

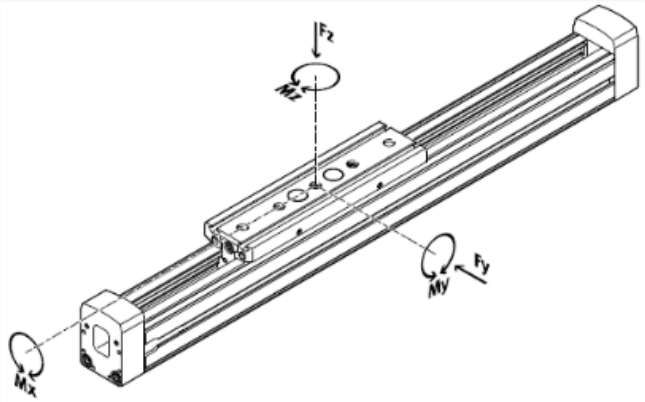
$$Price_{DGC}(Size, Stroke) := \begin{cases} \text{if } Size = 25 \\ \quad 3960 + 1.157 \cdot Stroke \div mm \\ \text{else if } Size = 40 \\ \quad 5508.7 + 1.497 \cdot Stroke \div mm \end{cases} + 2000$$

$$F_{DGC\_6Bar}(Size) := \begin{cases} \text{if } Size = 25 \\ \quad 295\ N \\ \text{else if } Size = 40 \\ \quad 483\ N \end{cases}$$

$$F_{v\_X}(F, F_{max}, M, M_{max}) := \left|\frac{F_1}{F_{max_1}}\right| + \left|\frac{F_2}{F_{max_2}}\right| + \left|\frac{M_0}{M_{max_0}}\right| + \left|\frac{M_1}{M_{max_1}}\right| + \left|\frac{M_2}{M_{max_2}}\right|$$

Includes estimate cost of cabling and 2 sensors and solenoid valve

Specifications for FESTO FDG-ZR-RF-... from datasheet



$$M_{X\_FDG}(Size) := \begin{cases} 7\ (N \cdot m) & \text{if } Size = 25 \\ 18\ (N \cdot m) & \text{else if } Size = 40 \end{cases}$$

$$M_{Y\_FDG}(Size) := \begin{cases} 15\ (N \cdot m) & \text{if } Size = 25 \\ 60\ (N \cdot m) & \text{else if } Size = 40 \end{cases}$$

$$F_{X\_FDG}(Size) := \begin{cases} 260\ N & \text{if } Size = 25 \\ 610\ N & \text{else if } Size = 40 \end{cases}$$

$$M_{Z\_FDG}(Size) := \begin{cases} 15\ (N \cdot m) & \text{if } Size = 25 \\ 90\ (N \cdot m) & \text{else if } Size = 40 \end{cases}$$

$$F_{Y\_FDG}(Size) := \begin{cases} 150\ N & \text{if } Size = 25 \\ 300\ N & \text{else if } Size = 40 \end{cases}$$

$$Load_{FDG}(Size) := \begin{cases} 15\ kg & \text{if } Size = 25 \\ 30\ kg & \text{else if } Size = 40 \end{cases}$$

$$F_{Z\_FDG}(Size) := \begin{cases} 150\ N & \text{if } Size = 25 \\ 300\ N & \text{else if } Size = 40 \end{cases}$$

$$DrivingT_{FDG}(Size) := \begin{cases} 3.7\ (N \cdot m) & \text{if } Size = 25 \\ 12.1\ (N \cdot m) & \text{else if } Size = 40 \end{cases}$$

$$m_{FDG}(Size, Stroke) := \begin{cases} (2 + 0.29 \cdot Stroke \div m)\ kg & \text{if } Size = 25 \\ (6.1 + 0.59 \cdot Stroke \div m)\ kg & \text{else if } Size = 40 \end{cases}$$

$$Price_{FDG}(Size, Stroke) := \begin{cases} 9026.7 + 2.967 \cdot Stroke \div mm & \text{if } Size = 25 \\ 12879.7 + 3.897 \cdot Stroke \div mm & \text{else if } Size = 40 \end{cases} + 1500$$

$$F_{v\_Y}(F, F_{max}, M, M_{max}) := \left|\frac{F_0}{F_{max_0}}\right| + \left|\frac{F_2}{F_{max_2}}\right| + \left|\frac{M_0}{M_{max_0}}\right| + \left|\frac{M_1}{M_{max_1}}\right| + \left|\frac{M_2}{M_{max_2}}\right|$$

$$F_{v\_Z}(F, F_{max}, M, M_{max}) := \left|\frac{F_1}{F_{max_1}}\right| + \left|\frac{F_2}{F_{max_2}}\right| + \left|\frac{M_0}{M_{max_0}}\right| + \left|\frac{M_1}{M_{max_1}}\right| + \left|\frac{M_2}{M_{max_2}}\right|$$

Specifications for FESTO EMMS-ST from datasheet
All drives without break, with encoder

$$T_{EMMS\_ST}(Size, Length) := \begin{cases} \text{if } Size = 57 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 0.8\ N \cdot m \\ \text{else if } Length = 2 \\ \quad 1.4\ N \cdot m \end{cases} \\ \text{else if } Size = 87 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 2.5\ N \cdot m \\ \text{else if } Length = 2 \\ \quad 5.9\ N \cdot m \end{cases} \end{cases}$$

$$J_{EMMS\_ST}(Size, Length) := \begin{cases} \text{if } Size = 57 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 0.29\ kg \cdot cm^2 \\ \text{else if } Length = 2 \\ \quad 0.48\ kg \cdot cm^2 \end{cases} \\ \text{else if } Size = 87 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 1\ kg \cdot cm^2 \\ \text{else if } Length = 2 \\ \quad 1.9\ kg \cdot cm^2 \end{cases} \end{cases}$$

$$Price_{CMMS\_ST} := 10507$$

$$m_{EMMS\_ST}(Size, Length) := \begin{cases} \text{if } Size = 57 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 0.97\ kg \\ \text{else if } Length = 2 \\ \quad 1.2\ kg \end{cases} \\ \text{else if } Size = 87 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 2.1\ kg \\ \text{else if } Length = 2 \\ \quad 3.2\ kg \end{cases} \end{cases}$$

$$Price_{EMMS\_ST}(Size, Length) := \begin{cases} \text{if } Size = 57 \\ \quad \text{if } Length = 1 \\ \quad\quad 3477 \\ \quad \text{else if } Length = 2 \\ \quad\quad 3716 \\ \text{else if } Size = 87 \\ \quad \text{if } Length = 1 \\ \quad\quad 5052 \\ \quad \text{else if } Length = 2 \\ \quad\quad 5224 \end{cases} + Price_{CMMS\_ST} + 1594 + 1410$$

$$Price_{EMMS\_ST\_Break} := 3511$$

Specifications for FESTO EMMS-AS-LS from datasheet
All drives without break, with encoder

$$T_{EMMS\_AS}(Size, Length) := \begin{cases} \text{if } Size = 55 \\ \quad \text{if } Length = 1 \\ \quad\quad 1.65\ N \cdot m \\ \quad \text{else if } Length = 2 \\ \quad\quad 2.7\ N \cdot m \\ \text{else if } Size = 70 \\ \quad \text{if } Length = 1 \\ \quad\quad 5\ N \cdot m \\ \quad \text{else if } Length = 2 \\ \quad\quad 8.3\ N \cdot m \end{cases}$$

$$m_{EMMS\_AS}(Size, Length) := \begin{cases} \text{if } Size = 55 \\ \quad \text{if } Length = 1 \\ \quad\quad 1.3\ kg \\ \quad \text{else if } Length = 2 \\ \quad\quad 1.6\ kg \\ \text{else if } Size = 70 \\ \quad \text{if } Length = 1 \\ \quad\quad 2.1\ kg \\ \quad \text{else if } Length = 2 \\ \quad\quad 2.7\ kg \end{cases}$$

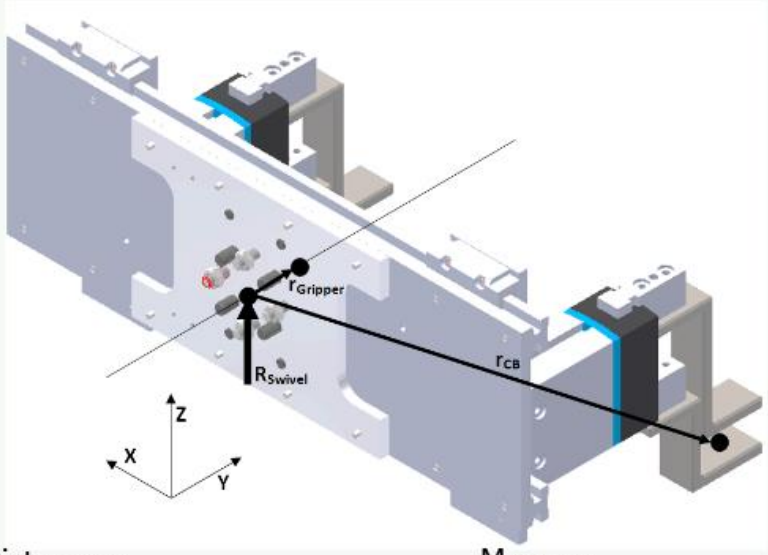
$$Price_{CMMP\_AS} := 14800$$

$$Price_{EMMS\_AS}(Size, Length) := \begin{cases} \text{if } Size = 55 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 12851 \\ \text{else if } Length = 2 \\ \quad 14506 \end{cases} \\ \text{else if } Size = 70 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 14881 \\ \text{else if } Length = 2 \\ \quad 16450 \end{cases} \end{cases} + Price_{CMMP\_AS} + 1594 + 1410$$

$$Price_{EMMS\_AS\_Break} := 3511$$

$$J_{EMMS\_AS}(Size, Length) := \begin{cases} \text{if } Size = 55 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 0.13\ kg \cdot cm^2 \\ \text{else if } Length = 2 \\ \quad 0.223\ kg \cdot cm^2 \end{cases} \\ \text{else if } Size = 70 \\ \quad \begin{cases} \text{if } Length = 1 \\ \quad 0.379\ kg \cdot cm^2 \\ \text{else if } Length = 2 \\ \quad 0.611\ kg \cdot cm^2 \end{cases} \end{cases}$$

Gripper



Distances:

$d_{X\_Gripper} := 0\ mm$
$d_{Y\_Gripper} := 42\ mm$
$d_{Z\_Gripper} := 0\ mm$

$d_{X\_CB} := 108\ mm$
$d_{Y\_CB} := 178\ mm$
$d_{Z\_CB} := 0\ mm$

Masses:

$m_{Gripper} := (2.85 + 0.15)\ kg$

$m_{CB} := 0.1\ kg$

$m_1 := m_{Gripper} + 2 \cdot m_{CB}$

Position vectors:

$$r_{Gripper} := \begin{bmatrix} d_{X\_Gripper} \\ d_{Y\_Gripper} \\ d_{Z\_Gripper} \end{bmatrix}$$

$$r_{CB} := \begin{bmatrix} d_{X\_CB} \\ d_{Y\_CB} \\ d_{Z\_CB} \end{bmatrix}$$

Forces:

$$F_{Gripper} := -m_{Gripper} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

$$F_{CB} := -m_{CB} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

Rotational Inertia:

$$I_{YY\_CB} := \frac{0.4\ kg \cdot \left((0.02\ m)^2 + (0.1\ m)^2\right)}{12}$$

$$I_{YY\_Gripper} := \frac{2.2\ kg \cdot \left((0.3\ m)^2 + (0.1\ m)^2\right)}{12} = 0.0183\ kg \cdot m^2$$

$$I_{YY\_GripperCB} := I_{YY\_Gripper} + 2 \cdot \left(I_{YY\_CB} + 0.4\ kg \cdot {d_{X\_CB}}^2\right) = 0.0284\ kg \cdot m^2$$

Calculations:

$$R_{Swivel} := -\left(F_{Gripper} + 2 \cdot F_{CB}\right)$$

$$M_{Swivel} := r_{Gripper} \times F_{Gripper} + 2 \cdot r_{CB} \times F_{CB} + I_{YY\_GripperCB} \cdot \alpha_{DSM}$$

Results:

$$R_{Swivel} = \begin{bmatrix} 4 \\ 16 \\ 47.3813 \end{bmatrix} N \qquad M_{Swivel} = \begin{bmatrix} -2.1092 \\ 0.3198 \\ 0.094 \end{bmatrix} N \cdot m$$

Swivel Module Selection:

$$DSM_{Size} := 25$$

$$F_{X\_DSM\_Selected} := F_{X\_DSM}\left(DSM_{Size}\right) = 50\ N$$

$$F_{Z\_DSM\_Selected} := F_{Z\_DSM}\left(DSM_{Size}\right) = 120\ N$$

$$T_{DSM\_Selected} := T_{DSM}\left(DSM_{Size}\right) = 5\ m \cdot N$$

$$m_{DSM\_Selected} := m_{DSM}\left(DSM_{Size}\right) = 0.725\ kg$$

$$Cost_{DSM\_Selected} := Price_{DSM}\left(DSM_{Size}\right) = 4075$$

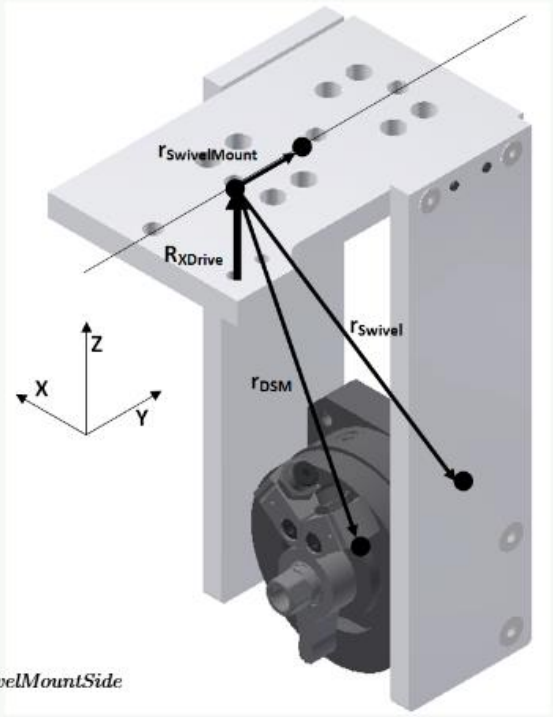
Safety factors:

$$SF_{X\_Swivel} := F_{X\_DSM\_Selected} \div R_{Swivel_0} = 12.5$$

$$SF_{Z\_Swivel} := F_{Z\_DSM\_Selected} \div R_{Swivel_2} = 2.5326$$

$$SF_{T\_Swivel} := T_{DSM\_Selected} \div M_{Swivel_1} = 15.6336$$

Y Drive



Distances:

$d_{X\_DSM} := 0\ mm$
$d_{Y\_DSM} := 69\ mm$
$d_{Z\_DSM} := 188\ mm$

$d_{X\_SwivelMount} := 0\ mm$
$d_{Y\_SwivelMount} := 50\ mm$
$d_{Z\_SwivelMount} := -73\ mm$

$d_{X\_Swivel} := 0\ mm$
$d_{Y\_Swivel} := 109\ mm$
$d_{Z\_Swivel} := -188\ mm$

Masses:

$m_{SwivelMountSide} := 0.318\ kg$
$m_{SwivelMountTop} := 0.352\ kg$
$m_{SwivelMount} := m_{SwivelMountTop} + 2 \cdot m_{SwivelMountSide}$
$m_2 := m_1 + m_{SwivelMount} + m_{DSM\_Selected}$

Position vectors:

$$r_{DSM} := \begin{bmatrix} d_{X\_DSM} \\ d_{Y\_DSM} \\ d_{Z\_DSM} \end{bmatrix}$$

$$r_{SwivelMount} := \begin{bmatrix} d_{X\_SwivelMount} \\ d_{Y\_SwivelMount} \\ d_{Z\_SwivelMount} \end{bmatrix}$$

$$r_{Swivel} := \begin{bmatrix} d_{X\_Swivel} \\ d_{Y\_Swivel} \\ d_{Z\_Swivel} \end{bmatrix}$$

Forces:

$$F_{DSM} := -m_{DSM\_Selected} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

$$F_{SwivelMount} := -m_{SwivelMount} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

Calculations:

$$R_{DriveY} := -\left(F_{DSM} + 2 \cdot F_{SwivelMount} - R_{Swivel}\right)$$

$$M_{DriveY} := -\left(-M_{Swivel} + r_{Swivel} \times \left(-R_{Swivel}\right) + r_{SwivelMount} \times F_{SwivelMount} + r_{DSM} \times F_{DSM}\right)$$

Results:

$$R_{DriveY} = \begin{bmatrix} 7.3763 \\ 29.505 \\ 87.374 \end{bmatrix} N$$

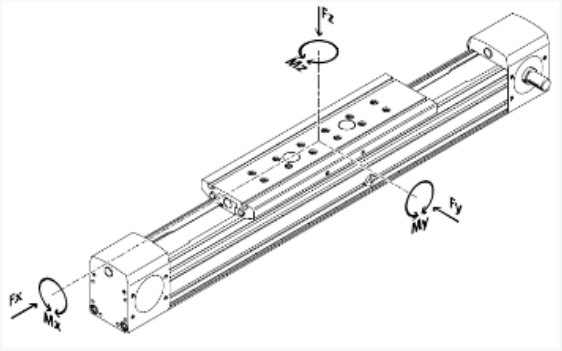
$$M_{DriveY} = \begin{bmatrix} 7.2147 \\ -0.352 \\ -0.4663 \end{bmatrix} N \cdot m$$

Drive selection:

$$DriveY_{Size} := 25$$

$$F_{DriveY\_Max} := \begin{bmatrix} F_{Y\_DGE}(DriveY_{Size}) \\ F_{X\_DGE}(DriveY_{Size}) \\ F_{Z\_DGE}(DriveY_{Size}) \end{bmatrix}$$

$$M_{DriveY\_Max} := \begin{bmatrix} M_{Y\_DGE}(DriveY_{Size}) \\ M_{X\_DGE}(DriveY_{Size}) \\ M_{Z\_DGE}(DriveY_{Size}) \end{bmatrix}$$



$$m_{DriveY} := m_{DGE}(DriveY_{Size}, Stroke_y) \qquad Load_{DriveY\_Max} := Load_{DGE}(DriveY_{Size})$$

$$DrivingT_{DriveY\_Max} := DrivingT_{DGE}(DriveY_{Size})$$

$$Cost_{DriveY} := Price_{DGE}(DriveY_{Size}, Stroke_y) = 15967.2$$

$$EffectiveD_{DriveY} := d_{DGE}(DriveY_{Size})$$

$$J_Y := J_{A\_DGE}(DriveY_{Size}, Stroke_y, m_2) = 11.9067\ kg \cdot cm^2$$

Safety factors for Y-Drive in robot coordinate system:

$$SF_{F\_DriveY} := F_{DriveY\_Max} \div R_{DriveY} = \begin{bmatrix} 20.3355 \\ 8.8121 \\ 1.7168 \end{bmatrix}$$

$$SF_{M\_DriveY} := M_{DriveY\_Max} \div M_{DriveY} = \begin{bmatrix} 2.0791 \\ -19.8888 \\ -32.1694 \end{bmatrix}$$

$$F_{v\_Y}(R_{DriveY}, F_{DriveY\_Max}, M_{DriveY}, M_{DriveY\_Max}) = 1.194$$

$$SF_{DrivingT\_DriveY} := DrivingT_{DriveY\_Max} \div \left(R_{DriveY_0} \cdot EffectiveD_{DriveY}\right) = 17.5082$$

$$SF_{Load} := Load_{DriveY\_Max} \div m_2 = 3.0531$$

Y-Motor selection:

$$T_{MotorY} := T_{EMMS\_AS}(55, 2) = 2.7\ N \cdot m$$

$$J_{MotorY} := J_{EMMS\_AS}(55, 2)$$

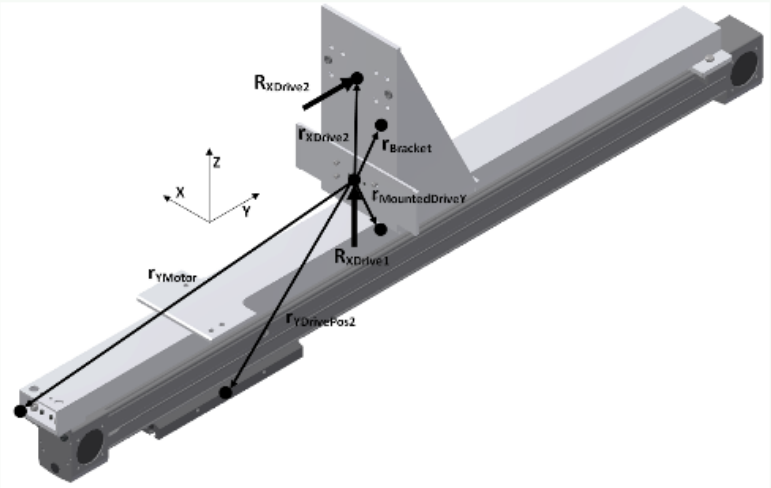
$$J_{RatioY} := J_Y \div J_{MotorY} = 53.3932$$ Not ideal - what is available in lab

$$m_{MotorY} := m_{EMMS\_AS}(55, 1) = 1.3\ kg$$

$$SF_{T\_MotorY} := T_{MotorY} \div \left(R_{DriveY_0} \cdot EffectiveD_{DriveY}\right) = 12.7763$$

$$Cost_{MotorY} := Price_{EMMS\_AS}(55, 2) = 32310$$

X Drives



Masses:

$m_{Bracket} := 0.807\ kg$

$m_{MountY} := 1.083\ kg$

$m_{MountedDriveY} := m_{MountY} + m_{DriveY}$

$m_3 := m_2 + m_{MotorY} + m_{MountedDriveY} + m_{Bracket} = 11.913\ kg$

Distances:

$d_{X\_YMotor} := 78\ mm$

$d_{Y\_YMotor} := -375\ mm$

$d_{Z\_YMotor} := -108\ mm$

$d_{X\_Bracket} := 0\ mm$

$d_{Y\_Bracket} := 30\ mm$

$d_{Z\_Bracket} := 34\ mm$

$d_{X\_MountedDriveY} := 0\ mm$

$d_{Y\_MountedDriveY} := 28\ mm$

$d_{Z\_MountedDriveY} := -93\ mm$

$d_{X\_YDrivePos1} := 0\ mm$

$d_{Y\_YDrivePos1} := -92\ mm$

$d_{Z\_YDrivePos1} := -142\ mm$

$d_{X\_YDrivePos2} := 0\ mm$

$d_{Y\_YDrivePos2} := 308\ mm$

$d_{Z\_YDrivePos2} := -142\ mm$

$d_{X\_XDrive2} := 0\ mm$

$d_{Y\_XDrive2} := 0\ mm$

$d_{Z\_XDrive2} := 116\ mm$

Position vectors:

$$r_{YMotor} := \begin{bmatrix} d_{X\_YMotor} \\ d_{Y\_YMotor} \\ d_{Z\_YMotor} \end{bmatrix}$$

$$r_{Bracket} := \begin{bmatrix} d_{X\_Bracket} \\ d_{Y\_Bracket} \\ d_{Z\_Bracket} \end{bmatrix}$$

$$r_{MountedDriveY} := \begin{bmatrix} d_{X\_MountedDriveY} \\ d_{Y\_MountedDriveY} \\ d_{Z\_MountedDriveY} \end{bmatrix}$$

$$r_{YDrivePos1} := \begin{bmatrix} d_{X\_YDrivePos1} \\ d_{Y\_YDrivePos1} \\ d_{Z\_YDrivePos1} \end{bmatrix}$$

$$r_{YDrivePos2} := \begin{bmatrix} d_{X\_YDrivePos2} \\ d_{Y\_YDrivePos2} \\ d_{Z\_YDrivePos2} \end{bmatrix}$$

$$r_{XDrive2} := \begin{bmatrix} d_{X\_XDrive2} \\ d_{Y\_XDrive2} \\ d_{Z\_XDrive2} \end{bmatrix}$$

Forces:

$$F_{MotorY} := -m_{MotorY} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix} \qquad F_{MountedDriveY} := -m_{MountedDriveY} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

$$F_{Bracket} := -m_{Bracket} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

Assumptions:

Full load carried by bottom drive, whilst Myy is handled by support drive.

Calculations:

$$R_{DriveX1} := -\left(F_{MotorY} + F_{MountedDriveY} + F_{Bracket} - R_{DriveY}\right)$$

$$M_{DriveX1} := -\left(-M_{DriveY} + r_{YDrivePos2} \times \left(-R_{DriveY}\right) + r_{YMotor} \times F_{MotorY}\right)$$

$$M_{DriveX1} := M_{DriveX1} - \left(r_{MountedDriveY} \times F_{MountedDriveY} + r_{Bracket} \times F_{Bracket}\right)$$

$$R_{DriveX2} := \begin{bmatrix} 0\ \boldsymbol{N} \\ M_{DriveX1_0} \div d_{Z\_XDrive2} \\ 0\ \boldsymbol{N} \end{bmatrix}$$

$$M_{DriveX2} := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{N \cdot m}$$

$$M_{DriveX1_0} := M_{DriveX1_0} - R_{DriveX2_1} \cdot d_{Z\_XDrive2} + 0.1\ \boldsymbol{N \cdot m}$$

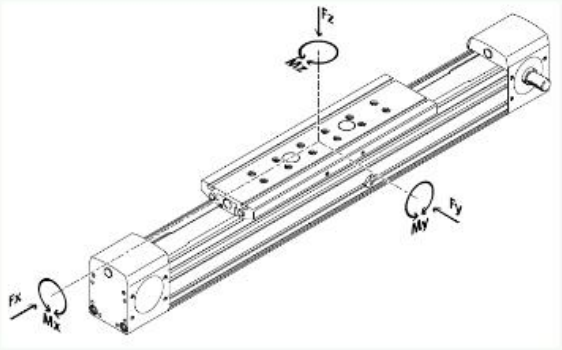
$$R_{DriveX1} := R_{DriveX1} - R_{DriveX2}$$

Results:

$$R_{DriveX1} = \begin{bmatrix} 16.1263 \\ -248.6366 \\ 191.0206 \end{bmatrix} \boldsymbol{N} \qquad R_{DriveX2} = \begin{bmatrix} 0 \\ 313.1416 \\ 0 \end{bmatrix} \boldsymbol{N}$$

$$M_{DriveX1} = \begin{bmatrix} 0.1 \\ -3.6108 \\ -1.8233 \end{bmatrix} \boldsymbol{N \cdot m} \qquad M_{DriveX2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{N \cdot m}$$

X Drive selection:



$$DriveX_{Size} := 25 \qquad SupportX_{Size} := 40$$

$$F_{DriveX\_Max} := \begin{bmatrix} F_{X\_DGC}(DriveX_{Size}) \\ F_{Z\_DGC}(DriveX_{Size}) \\ F_{Y\_DGC}(DriveX_{Size}) \end{bmatrix}$$

$$M_{DriveX\_Max} := \begin{bmatrix} M_{X\_DGC}(DriveX_{Size}) \\ M_{Z\_DGC}(DriveX_{Size}) \\ M_{Y\_DGC}(DriveX_{Size}) \end{bmatrix}$$

$$F_{PneumaticsX} := F_{DGC\_6Bar}(DriveX_{Size})$$

$$m_{DriveX} := m_{DGE}(DriveX_{Size}, Stroke_x)$$

$$m_{SupportX} := m_{FDG}(SupportX_{Size}, Stroke_x)$$

$$m_{DrivesX} := m_{DriveX} + m_{SupportX}$$

$$Cost_{DriveX} := Price_{DGC}(DriveX_{Size}, Stroke_x)$$

$$Cost_{SupportX} := Price_{DGC}(DriveX_{Size}, Stroke_x)$$

$$F_{SupportX\_Max} := \begin{bmatrix} F_{X\_FDG}(SupportX_{Size}) \\ F_{Z\_FDG}(SupportX_{Size}) \\ F_{Y\_FDG}(SupportX_{Size}) \end{bmatrix}$$

Safety factors for Y-Drive in robot coordinate system:

$$SF_{F\_DriveX} := F_{DriveX\_Max} \div R_{DriveX1} = \begin{bmatrix} 11.1619 \\ -2.1718 \\ 0.9423 \end{bmatrix}$$

$$SF_{M\_DriveX} := M_{DriveX\_Max} \div M_{DriveX1} = \begin{bmatrix} 40 \\ -1.3847 \\ -10.9691 \end{bmatrix}$$

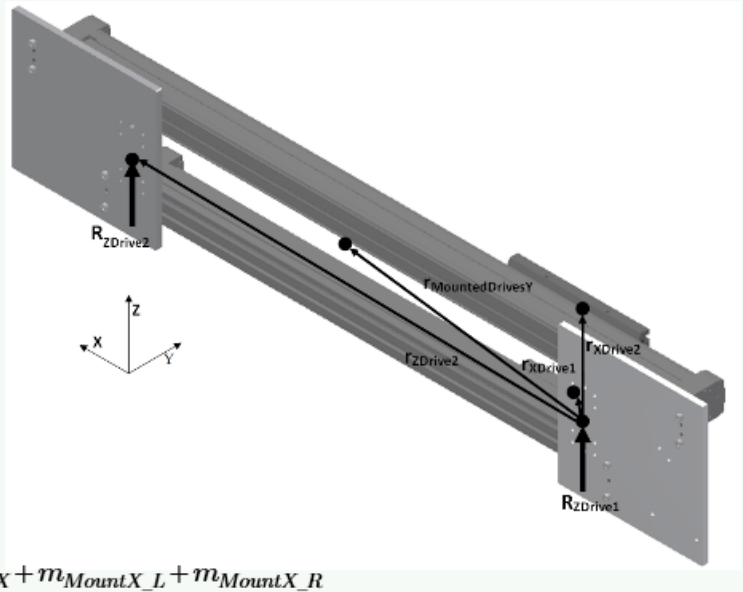
$$SF_{F\_SupportX} := F_{SupportX\_Max_1} \div R_{DriveX2_1} = 0.958$$

$$F_{v\_X}(R_{DriveX1}, F_{DriveX\_Max}, M_{DriveX1}, M_{DriveX\_Max}) = 2.36$$

$$a := F_{PneumaticsX} \div m_3 = 24.7629 \ \frac{m}{s^2}$$

This acceleration is limited to required acceleration by the valve.

Z Drives



Masses:

$m_{MountX\_L} := 0.892\ kg$

$m_{MountX\_R} := 0.895\ kg$

$m_{MountedDrivesX} := m_{DrivesX} + m_{MountX\_L} + m_{MountX\_R}$

$m_4 := m_3 + m_{MountedDrivesX} = 24.205\ kg$

Distances:

$d_{X\_XDrive1} := 78\ mm$

$d_{Y\_XDrive1} := 77\ mm$

$d_{Z\_XDrive1} := 3\ mm$

$d_{X\_XDrive2} := 78\ mm$

$d_{Y\_XDrive2} := 77\ mm$

$d_{Z\_XDrive2} := 119\ mm$

$d_{X\_MountedDriveY} := 330\ mm$

$d_{Y\_MountedDriveY} := 31\ mm$

$d_{Z\_MountedDriveY} := 3\ mm$

$d_{X\_ZDrive2} := d_{mod\_x} + 155\ mm$

$d_{Y\_ZDrive2} := 0\ mm$

$d_{Z\_ZDrive2} := 0\ mm$

Position vectors:

$$r_{DriveX1} := \begin{bmatrix} d_{X\_XDrive1} \\ d_{Y\_XDrive1} \\ d_{Z\_XDrive1} \end{bmatrix}$$

$$r_{DriveX2} := \begin{bmatrix} d_{X\_XDrive2} \\ d_{Y\_XDrive2} \\ d_{Z\_XDrive2} \end{bmatrix}$$

$$r_{MountedDrivesX} := \begin{bmatrix} d_{X\_MountedDriveY} \\ d_{Y\_MountedDriveY} \\ d_{Z\_MountedDriveY} \end{bmatrix}$$

$$r_{DriveZ2} := \begin{bmatrix} d_{X\_ZDrive2} \\ d_{Y\_ZDrive2} \\ d_{Z\_ZDrive2} \end{bmatrix}$$

Forces:

$$F_{MountedDrivesX} := -m_{MountedDrivesX} \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix}$$

Assumptions:

No moment around Y axis from mounted Y drives (supported by second Z drive)

$$ratio := \frac{d_{X\_XDrive1}}{d_{X\_ZDrive2}} = 0.1191$$

Calculations:

$$R_{DriveZTot} := -\left(F_{MountedDrivesX} - R_{DriveX1} - R_{DriveX2}\right)$$

$$M_{DriveZ1} := -M_{DriveX1} - M_{DriveX2} - \left(r_{DriveX1} \times \left(-R_{DriveX1}\right) + r_{DriveX2} \times \left(-R_{DriveX2}\right)\right)$$

$$M_{DriveZ1} := M_{DriveZ1} - F_{MountedDrivesX} \times r_{MountedDrivesX}$$

$$R_{DriveZ2} := \begin{bmatrix} R_{DriveZTot_0} \div 2 \\ R_{DriveZTot_1} \div 2 \\ R_{DriveZTot_2} \cdot ratio + M_{DriveZ1_1} \div d_{X\_ZDrive2} \end{bmatrix}$$

$$M_{DriveZ1_1} := 0.1\ \boldsymbol{N \cdot m}$$

$$M_{DriveZ2} := \begin{bmatrix} M_{DriveZ1_0} \div 2 \\ M_{DriveZ1_1} \div 2 \\ M_{DriveZ1_2} \div 2 \end{bmatrix}$$

$$R_{DriveZ1} := R_{DriveZTot} - R_{DriveZ2}$$

$$M_{DriveZ1} := M_{DriveZ1} - M_{DriveZ2}$$

Results:

$$R_{DriveZ1} = \begin{bmatrix} 15.7456 \\ 62.9825 \\ 254.1378 \end{bmatrix} \boldsymbol{N}$$

$$M_{DriveZ1} = \begin{bmatrix} -13.6835 \\ 0.05 \\ -7.0963 \end{bmatrix} \boldsymbol{N \cdot m}$$

$$R_{DriveZ2} = \begin{bmatrix} 15.7456 \\ 62.9825 \\ 118.8862 \end{bmatrix} \boldsymbol{N}$$

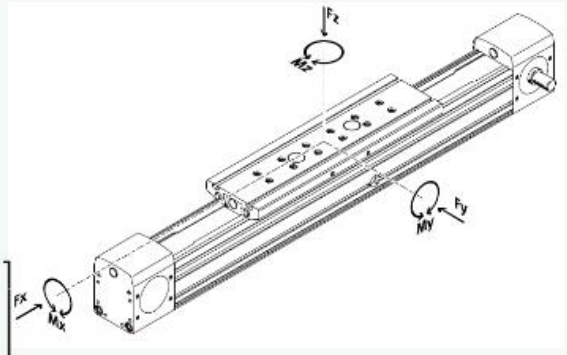
$$M_{DriveZ2} = \begin{bmatrix} -13.6835 \\ 0.05 \\ -7.0963 \end{bmatrix} \boldsymbol{N \cdot m}$$

Z Drive selection:

$$DriveZ_{Size} := 40$$

$$F_{DriveZ\_Max} := \begin{bmatrix} F_{Y\_DGE}(DriveZ_{Size}) \\ F_{Z\_DGE}(DriveZ_{Size}) \\ F_{X\_DGE}(DriveZ_{Size}) \end{bmatrix}$$

$$M_{DriveZ\_Max} := \begin{bmatrix} M_{Y\_DGE}(DriveZ_{Size}) \\ M_{Z\_DGE}(DriveZ_{Size}) \\ M_{X\_DGE}(DriveZ_{Size}) \end{bmatrix}$$



$$m_{DriveZ} := m_{DGE}(DriveZ_{Size}, Stroke_z) \qquad Load_{DriveZ\_Max} := Load_{DGE}(DriveZ_{Size})$$

$$DrivingT_{DriveZ\_Max} := DrivingT_{DGE}(DriveZ_{Size})$$

$$Cost_{DriveZ} := 2 \cdot Price_{DGE}(DriveZ_{Size}, Stroke_z) = 49005.96$$

$$EffectiveD_{DriveZ} := d_{DGE}(DriveZ_{Size})$$

$$Stroke_z = 1.14\ m$$

$$J_Z := J_{A\_DGE}(DriveZ_{Size}, Stroke_z, m_4)$$

Safety factors for Z-Drive in robot coordinate system:

$$SF_{F\_DriveZ1} := F_{DriveZ\_Max} \div R_{DriveZ1} = \begin{bmatrix} 19.0529 \\ 4.7632 \\ 2.4003 \end{bmatrix}$$

$$SF_{M\_DriveZ1} := M_{DriveZ\_Max} \div M_{DriveZ1} = \begin{bmatrix} -4.3848 \\ 1800 \\ -2.5365 \end{bmatrix}$$

$$SF_{F\_DriveZ2} := F_{DriveZ\_Max} \div R_{DriveZ2} = \begin{bmatrix} 19.0529 \\ 4.7632 \\ 5.131 \end{bmatrix}$$

$$SF_{M\_DriveZ2} := M_{DriveZ\_Max} \div M_{DriveZ2} = \begin{bmatrix} -4.3848 \\ 1800 \\ -2.5365 \end{bmatrix}$$

$$SF_{DrivingT\_DriveZ1} := DrivingT_{DriveZ\_Max} \div \left(R_{DriveY_2} \cdot EffectiveD_{DriveY}\right) = 4.8337$$

$$SF_{LoadZ1} := Load_{DriveZ\_Max} \div m_4 = 1.2394$$

$$F_{v\_Z}(R_{DriveZ1}, F_{DriveZ\_Max}, M_{DriveZ1}, M_{DriveZ\_Max}) = 1.2494$$

Safety factors on drives low. This is a result of using undersized drives for this section.

Z-Motor selection:

$T_{MotorZ} := T_{EMMS\_ST}(87, 2) = 5.9\ N \cdot m$

$J_{MotorZ} := J_{EMMS\_ST}(87, 2)$

$J_{RatioZ} := J_Z \div J_{MotorZ} = 56.1879$ Not ideal - what is available in lab

$m_{MotorZ} := m_{EMMS\_ST}(87, 2) = 3.2\ kg$

$SF_{T\_MotorZ} := T_{MotorZ} \div \left(R_{DriveZ1_2} \cdot EffectiveD_{DriveZ}\right) = 0.5835$

$Cost_{MotorZ} := Price_{EMMS\_ST}(87, 2) = 18735$

Costs:

$Cost_{DSM\_Selected} = 4075$

$Cost_{DriveX} = 6538.5$

$Cost_{SupportX} = 6538.5$

$Cost_{DriveY} = 15967.2$

$Cost_{MotorY} = 32310$

$Cost_{DriveZ} = 49005.96$

$Cost_{MotorZ} = 18735$

$Cost_{Total} := Cost_{DSM\_Selected} + Cost_{DriveX} + Cost_{SupportX} + Cost_{DriveY} + Cost_{MotorY}$

$Cost_{Total} := Cost_{Total} + Cost_{DriveZ} + Cost_{MotorZ} = 133170.16$

APPENDIX B – C# IMPLEMENTATION

This appendix contains all the extra information related to the C# implementation of the station controller. The first section contains details on the messages that are passed between holons. The messages are sorted by the message receiver and in the implementation, are given prefixes of the sender and receiver.

B-1: HOLON MESSAGES

Table 22: Holon inbox – supervisor staff holon

Message Name	**Sender Holon**	**Description**
AddTask	ECH	Supervisor initiates a new task with the specified product ID.
Complete	Task	Task holon saying it completed successfully.
Failed	Task	Task holon saying it failed.
DeviceListRequest	Task	Task holon asking for a list of device (operational) holons.
Failed	Product	Product holon response for failing to get product data.

Table 23: Holon inbox – move order holon

Message Name	**Sender Holon**	**Description**
StartProductRequest	Supervisor	Begins request for operation list for product.
ProductDataReply	Product	List of operations from product holon.
SubmitProposal	Operational	Proposal from operational holons.
Failed	Operational	Operational holon failed to perform operation.
Complete	Operational	Operational holon completed operation.
DeviceListReply	Supervisor	List of operational holons in holonic system.

Table 24: Holon inbox – extrnal communications holon

Message Name	Sender Holon	Description
InitialiseComms	Cell Controller	Registers the cell controller.
AddTask	Cell Controller	Relays a new task must be started to the supervisor.
ProductDataRequest	Cell Controller	Requests product data from product holon.
PalletAdded	Cell Controller	Relays a new pallet was added to a pallet holon.
PalletRemoved	Cell Controller	Relays a pallet was removed from a pallet holon.
PalletFull	Operational	Relays a message telling the cell controller a pallet needs to be removed.
RobotOn	Cell Controller	Relays the robot is online.
RobotOff	Cell Controller	Relays the robot is offline.
RegisterPallet	Operational	Registers a pallet holon for communication.
RegisterRobot	Operational	Registers a robot holon for communication.
UpdateTaskStatus	Supervisor	Relays a status message to the cell emulator.
UpdateRWTStatus	Operational	Relays a status message to the cell emulator.
UpdateRobotStatus	Operational	Relays a status message to the cell emulator.
UpdatePalletStatus	Operational	Relays a status message to the cell emulator.

Table 25: Holon inbox – pallet resource holon

Message Name	Sender Holon	Description
RFP	Task	Checks if the pallet can fulfil the task holons request and sends a proposal if it can.
AcceptProposal	Task	Books the accepted proposals breakers or fixture position.
PalletAdded	ECH	Sets the fixtures array to either pallet full of untested breakers or an empty pallet.
PalletRemoved	ECH	Clears fixtures array.
PalletBreakerChanges	Supervisor	Changes fixtures array to reflect the pallet state after a robot move operation.

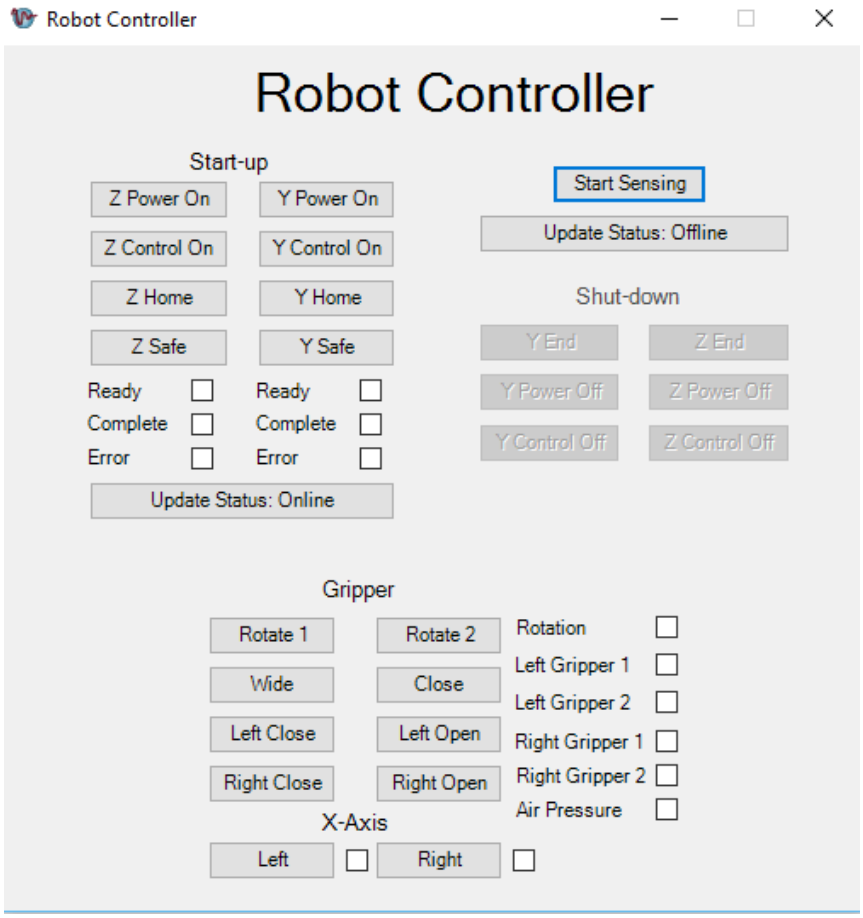
Table 26: Holon inbox – RWT resource holon

Message Name	Sender Holon	Description
RFP	Task	Checks if the RWT is available and can test the specific breaker.
AcceptProposal	Task	If in state Empty: Books the accepted proposals RWT slot. If in state Booked: Starts testing procedure.
BreakerRemoved	Task	Sets the fixtures array to either pallet full of untested breakers or an empty pallet.

Table 27: Holon inbox – robot resource holon

Message Name	Sender Holon	Description
RFP	Task	Checks if the pallet can fulfil the task holons request and sends a proposal if it can.
AcceptProposal	Task	Books the accepted proposals breakers or fixture position.
RobotConnectionEstablished	ECH	Sets the fixtures array to either pallet full of untested breakers or an empty pallet.
RobotConnectionEnded	ECH	Clears fixtures array.

B-2: ROBOT CONTROLLER



B-3

APPENDIX C – HARDWARE

This appendix contains the extra information related to the hardware implementation. The first section is related the signal interface used. The second contains the safety report and operating instructions for the laboratory setup. The third contains images of the hardware subassemblies. The fourth contains the calculations for the energy chains.

C-1: DAQ PIN ASSIGNMENTS

The pin assignments of the DAQ are given in Table 28. The notation given on the port name is P[port number].[line number] [I/O] where I stands for an input line and O stands for an output line. Each linear drive has a number of record bits (RB) that are used to select a position from the position tables (Section C-2) that were programmed into the Festo motor controllers using the Festo Configuration Tool software. Each drive also has two enable lines (Epower and Econtrol) and a start positioning (StartPos) line. The enable lines can only be changed using the robot controller described in Section 5.3.4. Each drive also has three output lines: ready for when the drive can be moved, motion complete for when the drive finishes a movement and an error line.

Table 28: DAQ pin assignments

Port 0	**Function**	**Port 1**	**Function**	**Port 2**	**Function**
P0.0 O	X-Move	P1.0 O	Rotate	P2.0 I	X-Right
P0.1 O	Z-RB1	P1.1 O	Widen	P2.1 I	X-Left
P0.2 O	Z-RB2	P1.2 O	Left grip	P2.2 I	Y-Ready
P0.3 O	Z-RB3	P1.3 O	Right grip	P2.3 I	Y-Motion Complete
P0.4 O	Z-Epower	P1.4 I	Left grip1	P2.4 I	Y-Error
P0.5 O	Z-Econtrol	P1.5 I	Left grip2	P2.5 I	Z-Ready
P0.6 O	Z-StartPos	P1.6 I	Right grip1	P2.6 I	Z-Motion Complete
P0.7 O	Y-RB1	P1.7 I	Right grip2	P2.7 I	Z-Error
P0.8 O	Y-RB2				
P0.9 O	Y-RB3				
P0.10 O	Y-RB4				
P0.11 O	Y-Epower				
P0.12 O	Y-Econtrol				
P0.13 O	Y-StartPos				
P0.14 O	Pressure				
P0.15 O	Rotation				

C-2: CONTROLLER POSITION TABLES

Table 29: Position table – Y-drive

Position number	**Position [mm[**	**Description**
0	Homing	Initiates homing sequence.
1	0	Zero position and pre-fixture 1 on pallet.
2	30	Fixture 1 on pallet.
3	90	Pre-fixture 2 on pallet.
4	120	Fixture 2 on pallet.
5	180	Pre-fixture 3 on pallet.
6	210	Fixture 3 on pallet.
7	290	Pre-RWT
8	340	RWT

Table 30: Position table – Z-drive

Position number	**Position [mm]**	**Description**
0	Homing	Initiates homing sequence.
1	2	Zero position and fixture height
2	10	Just above fixture for breaker releasing
3	150	Safe position above pallet
4	210	RWT slot 1
5	667	RWT slot 2
6	1125	RWT slot 3
7	-3	Release position

C-3: HARDWARE SUBASSEMBLIES

This section contains inventor images of each individual subassembly that was designed and manufactured.

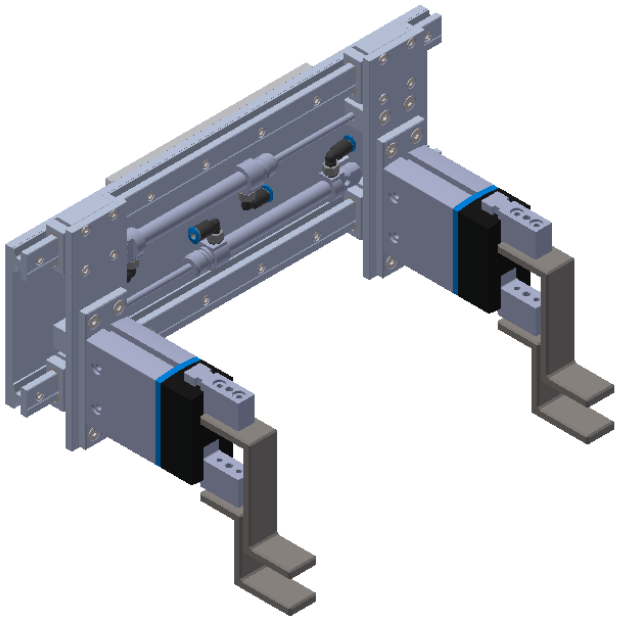


Figure 31: End effector

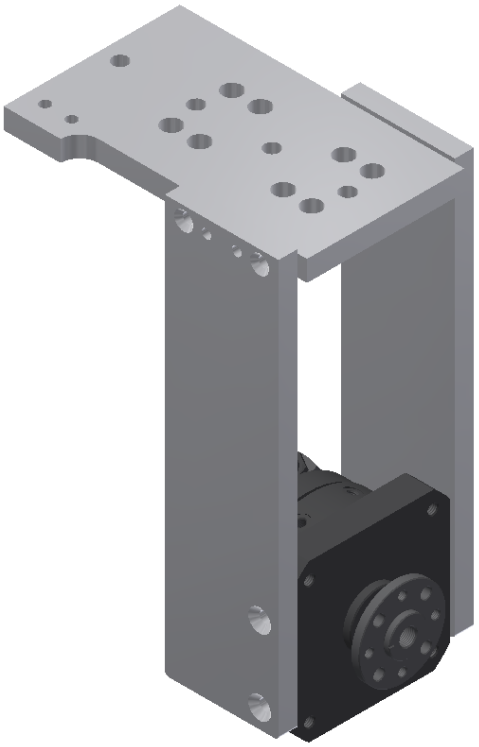


Figure 32: Mounted swivel module

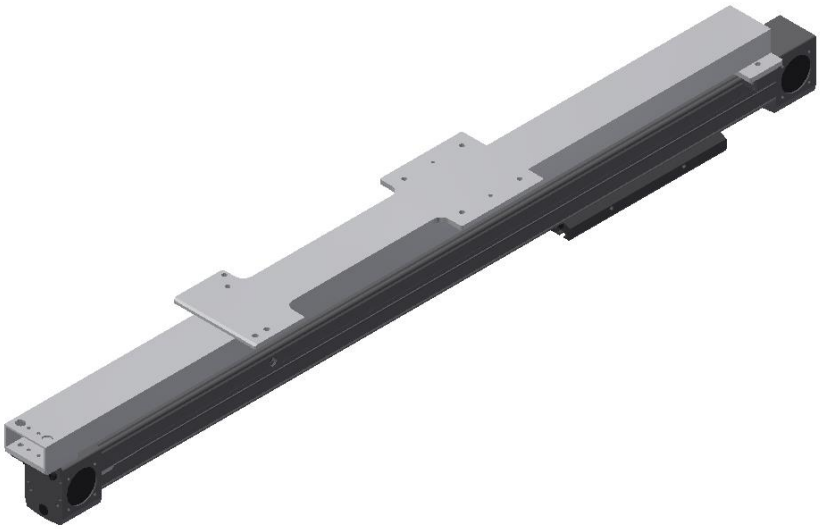


Figure 33: Mounted y-drive

Figure 34: Mounted x-drives

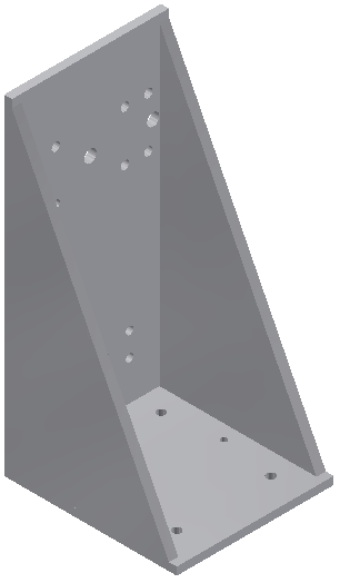


Figure 35: Y-drive mounting bracket

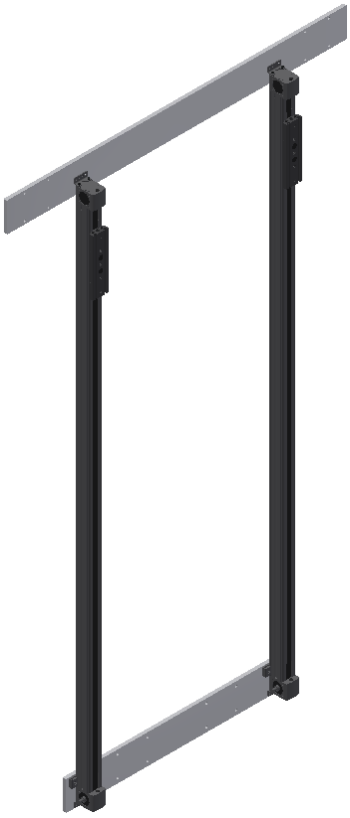


Figure 36: Mounted z-drives

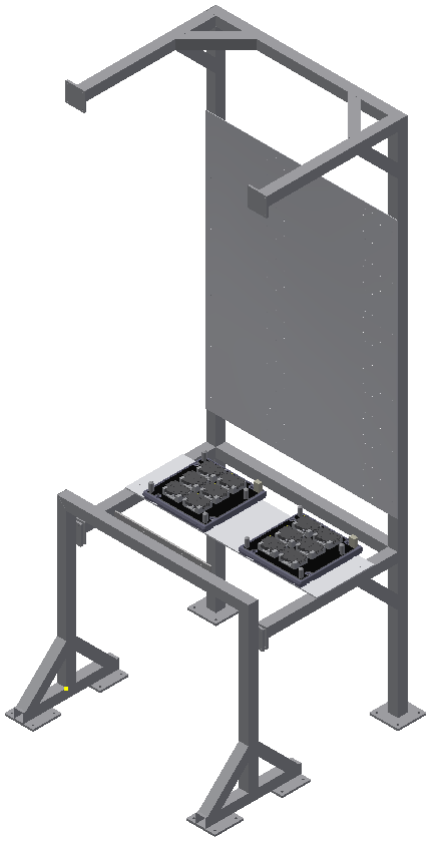


Figure 37: Laboratory RWT and pallet table

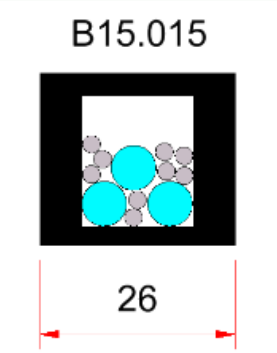
C-4: ENERGY CHAIN CALCULATIONS

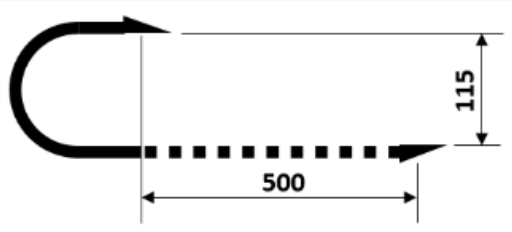
The energy chain calculations were done using manually, taking the distance of the chain and dividing by the chain link pitch, and also compared to the results given by the software provided by IGUS. In each calculation, a list of the cables that are to be contained in the chain is given as well as an image illustrating how they will fit in the chain. An image illustrating the moving length of the chain and bend radius is also given. The calculations to find the amount of links is then done.

C-4-1: Y-DRIVE ENERGY CHAIN

Pnuematics:
2 x 6 mm
Control:
7 x 3 mm
2 x 3 mm
Bend Radius:
< 60 mm

Chosen chain:
B15.015.035.0





$$R_{ellipse}(a,b,h) := \pi \cdot (a+b) \cdot \left(1 + \frac{3 \cdot h}{10 + \sqrt{4 - 3 \cdot h}}\right) \div 2$$

$$h_{ellipse}(a,b) := \frac{(a-b)^2}{(a+b)^2}$$

$pitch_{B15} := 30.5\ mm$ $\quad R\_b_1 := 48\ mm$

$pitchEnd_{B15} := 38\ mm$

$dM_1 := 500\ mm$

$R_1 := \frac{116}{2}\ mm = 58\ mm$

$D := 105\ mm$ $\quad h_1 := h_{ellipse}(R_1, R\_b_1)$

$d\_tot_1 := dM_1 + R_{ellipse}(R_1, R\_b_1, h_1) + 2 \cdot (D - R\_b_1) - 2 \cdot pitchEnd_{B15} = 0.7049\ m$

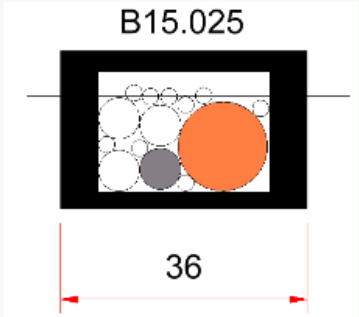
$links_1 := d\_tot_1 \div pitch_{B15} = 23.1107$

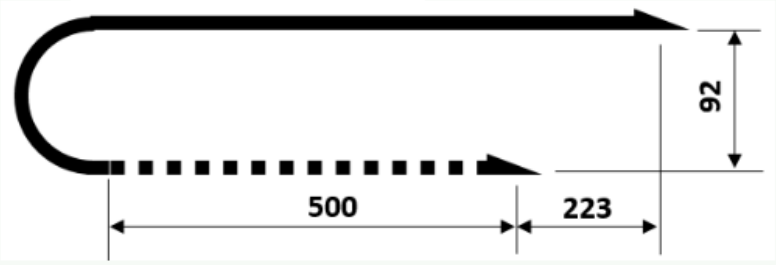
C-4-2: X-DRIVE ENERGY CHAIN

Pnuematics:
2 x 6 mm
Control:
7 x 2.4 mm
3 x 2.4 mm
Motor (AS):
1 x 13 mm power
*CHECK BEND
1 x 6 mm control

Bend Radius:
< 75 mm

Chosen chain:
B15.025.075.0





$R\_b_2 := 38\ mm$

$dM_2 := 500\ mm$

$R_2 := \frac{200}{2}\ mm = 100\ mm$

$dF_2 := 223\ mm$

$D_2 := 132\ mm$ $\qquad h_2 := h_{ellipse}(R_2, R\_b_2)$

$d\_tot_2 := dM_2 + R_{ellipse}(R_2, R\_b_2, h_2) + 2 \cdot (D_2 - R\_b_2) + dF_2 - 2 \cdot pitchEnd_{B15} = 1.0629\ m$

$links_2 := d\_tot_2 \div pitch_{B15} = 34.8477$

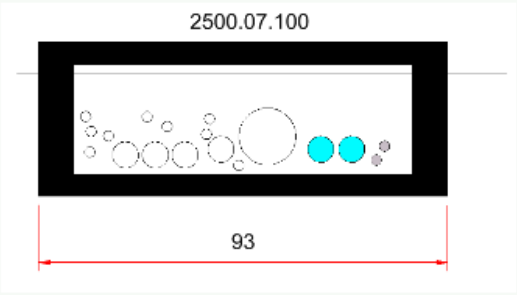
$links_2 := links_2 - 22 = 12.8477$

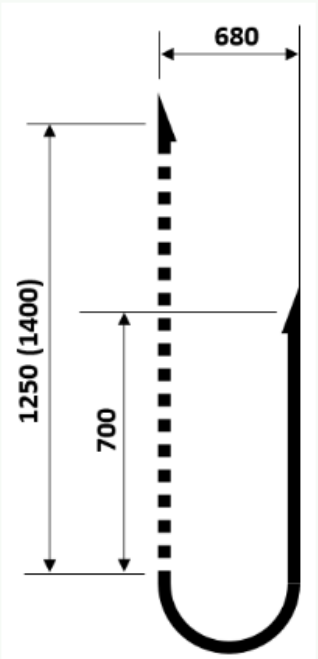
C-4-3: Z-DRIVE ENERGY CHAIN

X direction:
Pnuematics:
3 x 6 mm
Control:
7 x 2.4 mm
2 x 2.4 mm
Motor (AS):
1 x 13 mm power
1 x 6 mm control
Pnuematic Drive
2 x 6 mm
2 x 2.4 mm

Bend Radius:
< 200 mm

Chosen chain:
240.07.100 & 2500.07.100





$R\_b_3 := 100\ mm$

$dM_3 := 700\ mm$

$R_3 := \frac{650}{2}\ mm = 325\ mm$

$dF_3 := 0\ mm$

$pitch_{B15} := 46\ mm$

$pitchEnd_{B15} := 38\ mm$

$D_3 := 157\ mm$

$h_3 := h_{ellipse}(R_2, R\_b_2)$

$d\_tot_3 := dM_3 + R_{ellipse}(R_3, R\_b_3, h_3) + 2 \cdot (D_3 - R\_b_3) + dF_3 = 1.5157\ m$

$links_3 := d\_tot_3 \div pitch_{B15} = 32.9505$

APPENDIX D – LABORATORY RESULTS

This appendix contains the full laboratory test results. The data was written at each move step in the C# controller. The description in the first column follows the move steps from Figure 24 in Section 5.2.5.3 which describes the robot holons operation. The data that is listed in Table 31 and Table 32 is then averaged to provide the data in Table 14. The missing data entries are as a result of omitting data with a very high deviation from the normal time as a result of the DAQ performing slowly.

Table 31: Laboratory test results - scenario 1

Move description	Move ID	Run 1 [s]	Run 2 [s]	Run 3 [s]	Run 4 [s]	Run 5 [s]	Average [s]
Above pallet	20	0,18	0,17		0,16	0,13	0,16
Before breaker	30	0,98	1,02	0,91	1,09	0,91	0,98
Over breaker	40	0,70	0,72	0,59	0,63	0,64	0,66
Above pallet	60	1,39	1,40	1,38	1,40	1,37	1,39
Process module	70	1,42	1,42	1,45	1,44	1,49	1,44
Place	80			1,02	0,98		1,00
Moving to safe sapce	91			0,84			0,84
Above pallet	20	1,09	1,15	1,35	1,30	1,32	1,24
Before breaker	30	0,98	0,99	1,04	1,00	1,01	1,01
Over breaker	40	0,72	0,78	0,72	0,66	0,71	0,72
Above pallet	60	1,48	1,48	1,42	1,42	1,46	1,45
Process module	70	2,42	2,41	2,46	2,34	2,42	2,41
Place	80	1,19	1,12	1,12	1,14	1,35	1,18
Moving to safe sapce	91	0,87	1,08	0,91			0,95
Above pallet	20	2,56	3,03	2,53	2,63	3,03	2,75
Before product	30	1,03	1,76	1,05	1,12	1,58	1,31
Over product	40	0,80		0,88	0,83		0,83
Above pallet	60		2,22	1,59	1,46	1,91	1,80
Process module	70	4,34	5,12	4,34	5,00	4,67	4,70
Place	80	2,96	3,47	3,03	4,19	3,54	3,44
Moving to safe sapce	91	1,67	2,00	1,50			1,72
Front of process mod	20	4,16	4,99	4,18	5,17	5,01	4,70
Over product	30	2,01		2,14		2,65	2,27
Front of process mod	50	1,73	2,47	1,68		2,05	1,98
Above pallet	60	0,92	1,57	0,40	1,93	1,32	1,23
Above fixture	70	1,30	1,94	1,30		2,03	1,65
Drop in fixture	90	0,95	1,34	0,92		1,56	1,19
Above pallet	0	1,54		1,47			1,50
Front of process mod	20	2,46	2,63	2,49	3,48	3,14	2,84
Over product	30	2,02	2,35	2,06	3,00	2,68	2,42
Front of process mod	50	1,77	1,92	1,72			1,80
Above pallet	60	2,32	2,40	2,21	3,29	3,09	2,66
Above fixture	70		1,34	1,24		1,66	1,41
Drop in fixture	90	0,69	0,78	0,82	0,93	1,00	0,84
Above pallet	0	3,30	3,02	4,07	2,76	4,15	3,46

Table 32: Laboratory test results - scenario 2 and 3

Move description	**Move ID**	**Run 1 [s]**	**Run 2 [s]**	**Run 3 [s]**	**Run 4 [s]**	**Average [s]**
Above pallet	20	2,09	2,10	2,09	2,10	2,09
Before product	30	1,01	1,00	1,01	1,00	1,01
Over product	40	2,03	1,59	1,71	1,99	1,83
Above pallet	60	1,86	1,98	1,92	1,91	1,92
Process module	70	2,26	2,27	2,29	2,37	2,30
Place	80	3,29	3,04	1,08	3,33	2,68
Moving to safe space	0	1,64	1,63	1,18	1,81	1,57
Front of process mod	20	2,05	2,29	2,34	2,36	2,26
Over product	30	0,91	1,85	1,84	2,12	1,68
Front of process mod	50	1,46	1,83	1,53	1,60	1,60
Above pallet	60	0,65	0,64	0,63	0,76	0,67
Above fixture	70	2,60	2,61	2,71	2,56	2,62
Drop in fixture	90	1,71	1,83	1,67	1,72	1,73
Above pallet	0	1,60	1,57	1,51	1,61	1,57
Front of process mod	10	1,03	0,91	1,02	0,93	0,97
Over product	20	0,91	0,91	0,88	2,22	1,23
Front of process mod	30	7,41	2,80	8,26		6,16
Above pallet	50	1,83	1,80	1,83	0,12	1,40
Above fixture	60	2,25	2,29	2,38	0,14	1,77
Drop in fixture	70	1,26	2,84	1,24	1,01	1,59
Above pallet	90	0,82	1,40	0,81	0,49	0,88
Above pallet	20	0,87	2,36	1,19	0,92	1,33
Before product	30	2,07	1,51	1,53	2,30	1,85
Over product	40	1,46	1,00	1,08	0,17	0,93
Above pallet	60	1,84	1,62	1,64	1,26	1,59
Process module	70	2,28	2,62	2,28	1,01	2,05
Place	80	3,34	2,90	1,45	0,51	2,05
Moving to safe space	0	2,29	2,05	1,72	1,10	1,79
Front of process mod	20	2,69	2,91	2,89	3,58	3,02
Over product	30	2,30	2,17	2,27	1,98	2,18
Front of process mod	50	2,07	2,14	2,10	0,18	1,62
Above pallet	60	2,50	2,60	2,49	1,90	2,37
Above fixture	70	2,82	3,86	3,22	1,00	2,73
Drop in fixture	90	1,88	1,76	1,65	0,54	1,46
Above pallet	0	2,54	2,20	2,20	1,11	2,01