Capacity Analysis and Throughput Increase of an Automated Robotic Fastening System

by

Matthew John Cragun

B.S., Brigham Young University (2008)
M.E., Old Dominion University (2010)

Submitted to the Department of Mechanical Engineering and
MIT Sloan School of Management
in partial fulfillment of the requirements for the degrees of
Master of Science in Mechanical Engineering
and
Master of Business Administration
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2017

Author .. Signature redacted ....
Department of Mechanical Engineering and
MIT Sloan School of Management
May 12, 2017

Certified by. Signature redacted ....
Tahid R. Zaman
Assistant Professor, Sloan School of Management
Thesis Supervisor

Certified by.. Signature redacted ....
Warren W. Hoburg
Assistant Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Certified by.................. Signature redacted ......
Daniel E. Whitney
Senior Research Scientist Emeritus, MIT Leaders for Global Operations
Thesis Reader

Accepted by.................... Signature redacted ....
Maura Herson
Director of MIT Sloan MBA Program
MIT Sloan School of Management

Accepted by...... Signature redacted ....
Rohan Abeyaratne
Chairman of the Committee on Graduate Students
Department of Mechanical Engineering

Capacity Analysis and Throughput Increase of an Automated Robotic Fastening System

by

Matthew John Cragun

Submitted to the Department of Mechanical Engineering and
MIT Sloan School of Management
on May 12, 2017, in partial fulfillment of the
requirements for the degrees of
Master of Science in Mechanical Engineering
and
Master of Business Administration

Abstract

Large Structure Manufacturing (LSM) is in the midst of implementing a new robotic fastening system that has been designed with flexibility as a key feature. This thesis describes work done build a model of the new system in order to estimate the capacity of the system. It then presents several recommendations to increase the throughput of the system. Additionally, lessons learned from this work are presented so they can be applied to future automation projects.

Thesis Supervisor: Tauhid R. Zaman
Title: Assistant Professor, Sloan School of Management

Thesis Supervisor: Warren W. Hoburg
Title: Assistant Professor, Department of Aeronautics and Astronautics

Thesis Reader: Daniel E. Whitney
Title: Senior Research Scientist Emeritus, MIT Leaders for Global Operations

Acknowledgments

In deep gratitude to my supportive wife Nicole and our wonderful children who are constant reminders to me of what really matters.

Contents

List of Figures

List of Tables

List of Terms

AFS Automated Fastening System

LSM Large Structure Manufacturing

DES Discrete Events Simulation

RMFT Required Mean Fastener Time

RSI Robotic Systems, Inc.

PFS Precision Fastening Systems

OLP Offline Program

MRP Mobile Robot Platform

MCP Master Control Panel

SOW Statement of Work

AGV Automatic Guided Vehicle

PLC Programmable Logic Controller

Chapter 1

Introduction

1.1 Motivation

In this thesis, a Discrete Events Simulation (DES) is used to estimate the capacity of a new flexible manufacturing system. Along with a capacity estimate, several recommendations for system improvement are presented that arose as a result of the model and the model-building process. From a broader perspective, this thesis will also demonstrate that in building a new production system, especially in a complex organization, a model is imperative and that the model needs to be improved as the system develops in order to capture more subtle opportunities for improvement. Finally, lessons learned will be presented as they apply to future automation projects.

1.1.1 Background information

Large Structure Manufacturing (LSM) is a company in a highly competitive industry producing large, high-value, metal structures. They have one major competitor and several smaller, new market entrants. There are currently several issues facing the factory including: cost, quality, schedule, and worker safety. In order to make improvements on all fronts, LSM has undertaken a major initiative to modernize one of its production lines using new robotic technologies.[1]

This initiative is being implemented on one of their mature products, the Product

A. Product A is a component of a larger product and has been in production for a number of years; a staple for the company. However, soon, LSM plans the rollout of a major update to this product called Product B. It is the company's plan to implement this new robotic production system on Product A, a product it understands well, in order to prepare for the rollout of Product B.

1.1.2 A new automated factory

The new factory process being implemented is called Automated Fastening System (AFS); it is a new method for manufacturing the primary structure of Product A. The Product A is made of 12-15 different vendor-supplied panels which must be aligned and then fastened together using roughly 25,000 fasteners. The fasteners are placed along horizontal lap joints or vertical circumferential joints. Fasteners are naturally grouped by the structure of Product A into smaller work packages called bays. Currently the entire process is done by hand by highly skilled assembly workers. AFS is a greenfield production system comprised of robotic arms mounted on mobile platforms in a new purpose-built facility. Attached to each robot arm is a multi-function end effector that is able to drill holes and then insert the required fasteners.

LSM is implementing this initiative with the intent to reduce costs, improve quality, and reduce worker injury. Removing workers from the production line will reduce the variable costs of assembly. High quality is a major focus for LSM as its industry is highly regulated and the products must pass certification. LSM plans to improve quality and repeatability using robots as opposed to human workers. Lastly, AFS is designed to reduce worker injury rates. The repetitive process of drilling and fastening results in injuries for the assembly workers, especially when the work is performed in some of the less-ergonomic positions. The fastening process in particular is especially violent and leads to one of the highest injury rates in the entire factory.

1.1.3 Ramp-up to meet production targets

AFS has been in development for more than five years and is currently being implemented in a new building at the factory. To ramp up to full production, LSM is phasing in the new line by progressively scheduling more Product A's to be built in the new building using AFS instead of the current factory using the legacy system. However, the implementation is complex and behind schedule. In order for LSM to meet its full production rate with the new factory, the flowtime needs to decrease significantly.

One of the challenges in the ramp up is that the production system has been implemented piece-meal with different features coming online at different times. Because of this, the end state system capacity is not well understood. Beyond this, the effort to get the system up and running quickly resulted in many inefficiencies that are not well-tracked or accounted for.

1.2 Problem statement

This project was undertaken with two main objectives. First, use modeling to define the capacity of the AFS, and second, use the insights gained from the model to make recommendations for system improvements.

1.2.1 Capacity model of the automated system

The first goal in the project was to develop a model of AFS and use the model to predict the capacity of the system at its end state. LSM had a business case for AFS with a target flowtime for Product A, but was still unsure if the system being developed would, in the end, be able to meet the requirements. The model built for this project was designed to answer this question and then be a tool for LSM engineers to evaluate future decisions.

1.2.2 Make recommendations for flow reduction

The second focus of the project was to make recommendations based on the model for LSM to reduce the flowtime of the system. Because of the limited time frame of the research period, recommendations were to be implemented after the end of the project.

1.3 Project methodology

The methodology for the project had five main steps as follows:

1. Understand the product
2. Understand the production process
3. Model the production system
4. Build confidence in the model
5. Make recommendations based on data obtained from the model

1.4 Thesis document layout

The remainder of this document will first provide the reader with a more in-depth understanding of Product A, the manufacturing process, and the function of AFS. Then the document will turn toward the steps used in creating the model and gaining confidence in the results. Insights gained from the model will then be leveraged in a section containing recommendations for system improvements along with implementation suggestions. Finally conclusions and learnings will be discussed that can be applied toward future automation projects.

Chapter 2

Literature Search

Flexible automations systems are designed to allow the system to be changed and adapted as part of their implementation. There are different scopes and degrees of flexibility which can be designed into an automation system. AFS was designed with flexibility as a main requirement. One of the main reasons for flexibility cited was the ability to copy the system or move the system to another manufacturing site with minimal resources or time. This chapter will discuss a summary of other literature that has been published on flexible automation systems and then discuss how it relates to the task at hand with AFS.

2.1 What are the benefits of flexible automation?

In the history of manufacturing, automation has led to massive increases in efficiency. The primary reasons for automation are typically cost reduction, safety, and repeatability[2].

Typically, the process to implement a new automation system has been to automate tasks or groups of tasks in an assembly process and then to connect those groups with material handling processes in between[3]. Lines built this way, however, have the challenge of being rigid; any changes require changing the process. As technology has improved, flexibility has been introduced into the manufacturing process. A great example of flexibility in the manufacturing process is a CNC mill. The mill has the capability to perform many different operations and to produce a variety of parts with simple setup changes.

Abd classifies production systems into assembly lines and assembly cells[4]. A line is a traditional dedicated production system and a cell is a collection of flexible components. A similar comparison is made for robotic systems by Makino as shown in the table below[4][5] comparing Robotic Assembly Lines(RAL) and Robotic Assembly Cells(RAC):

Table 2.1 – A comparison of Robotic Assembly Lines(RAL) vs. Robotic Assembly Cells(RAC)

Characteristics	RAL	RAC
Process	Divided	Integrated
Task of each station	Simplified Specialized Standardized	Complex Versatile Not standardized
Cycle time	Short (3–60 s)	Long (3–20 min)
Flow of work piece	In-line One-way	Circulating Network
Number of robots	1–100	1–4
No. of assembly parts	1–6	1–50
Tool change	Single tool Multi finger	Switchable tool Automatic tool change
Production volume	High	Low

Michalos describes four types of flexibility[6] that can be added to a manufacturing system:

Routing flexibility: Changing the order in which parts are routed through an existing production process

Structural flexibility: The ability to change or physically alter the production system's structure

Resource flexibility: The ability to change the systems capacity for specific tasks using existing resources. E.g. reconfiguring a robot to weld instead of grind.

Expansion flexibility: The ability to add additional resources to an existing production system

As an example of flexibility, a typical large machine shop will have a collection of CNC machines giving it great resource flexibility, or the capability to produce a wide array of products. Jobs shops such as this, while having the capability to produce a wide array of products, are still limited in their structural flexibility because the machines themselves are not mobile. This is less of a problem for smaller parts, but as the parts become larger and material handling becomes a bigger factor, adapting a production line to accommodate different products becomes more important. Robotic systems are increasingly common that provide an even wider range of movement and task flexibility. This additional flexibility is needed to handle increasing demand for product customization[5]. Many times, robots are bolted to the floor on special foundations limiting the structural flexibility of the system. However, these systems can be augmented with mobile bases to that can be structurally reconfigured easily.

A flexible manufacturing system may not be right in all circumstances[7]. Factors such as complexity and time to market have a significant influence. Tseng finds that in industries where there is more variety or fewer firms in competition, the benefit of flexible systems is greater. Conversely, for products that are low mix or industries with a high number of competitors, the benefits of a flexible system are significantly reduced[8]. Additionally, the value of a flexible system can also take other factors into account such as market uncertainty[9] by allowing the system to be reconfigured to meet market demands quickly.

2.2 Typical challenges with flexible automation

One of the first challenges with any automation system is determining the right level of automation. When implementing on the basis of cost, the primary anticipated savings from automation is a decrease in direct labor. A system integrator mentions direct cost reduction on their website with a caveat:

> The important thing to remember is that eliminating direct labor results

> in a cost reduction only when the actual headcount is lower than what would be required to produce the same volume manually[2].

These benefits are typically paid for by high initial capital investment and technical risks of implementation. One consideration is that even if the amount of labor required is reduced, higher–wage technical labor is required to keep the machines running. This is important when considering the business case of the project. There are many factors that go into a cost reduction effort and predicting all factors ahead of time is difficult. Often reducing costs is possible but obtaining the minimum cost solution for a flexible system is very difficult[10].

Another major challenge in large systems is proper integration. In a recent survey of automation projects at a large manufacturing company, it was found that significant issues exist when a systems level approach is not taken, especially when using new, unproven technologies. In some cases this causes projects to overrun budgets by millions and schedules by months[11]. Wilson[12] states some of the challenges to implementing automation systems including:

- Technical risk using unproven technologies and processes
- Working with integrators vs. developing skill inhouse
- Employee views on automation (some are threatened by increased automation)

Flexible automation systems are also often difficult to implement due to organizational and strategic reasons. According to Inman:

> Other problems can result from a lack of technical literacy, management incompetence, and poor implementation of the FMS process. If the firm misidentifies its objectives and manufacturing mission, and does not maintain a manufacturing strategy that is consistent with the firm's overall strategy, problems are inevitable. It is crucial that a firm's technology acquisition decisions be consistent with its manufacturing strategy[7].

2.3 Where has flexible automation been employed successfully?

The automotive industry has been developing flexible manufacturing systems for several decades. Early automotive efforts brought in flexible machines such as CNC systems that can produce a variety of parts on demand[13]. More recent efforts have focused on using reconfigurable robotics for assembly[14]. One of the major initiatives has been to focus on equipment that can handle a variety of tasks and produce a variety of products. Its worth noting that while the products are different, they are all similar. Tesla's new auto factory can easily adapt to producing their different models, but it would have a difficult time being configured to produce boats or computer chips. Thus there is a limit to flexibility. Flexible systems are also being used for tasks such as warehouse management[15] and farming[16]. Over the coming decades the number of applications for flexible automation systems will continue to grow rapidly.

2.4 How has planning and simulation been implemented with flexible systems?

Simulation during the early stages of development provides an inexpensive opportunity to learn a significant amount about the process prior to implementation. Wilson suggests this as a meaningful way to reduce risk in a project[12]. He goes on to suggest two types of commonly used simulations:

> Two types of simulation may be appropriate. The first is kinematic simulation of the robot and associated equipment. The second is a discrete event simulation that provides a model of the operation of the facility, including any automation to check for production rates and resource requirements[12].

Discrete Events Simulations(DES) typically look at a process as a set of tasks to be executed, each with an associated requirement for resources. Kinematic simulations utilize 3D models of the equipment and parts being processed. Where DES models are excellent for capacity and resource planning, kinematic models are excellent for evaluating cycle times, interferences, and reducing commissioning time. Commercial software is available for both types of simulations.

Another challenge associated with a flexible system is planning and scheduling. As constraints on the order of task completion are relaxed, and as agents with different capabilities are introduced, scheduling becomes more challenging. Inherently, the realized capacity of a system is tied to the optimality of its schedule. When considering the different system flexibilities that can be included, the problems become extremely challenging. Imagine a system made of many robots, each with the capability to perform any of many tasks, in any order, where the robots are mobile and can be in any location in the factory at any time. The complexity of such problems makes them NP-hard and impossible to find an exact solution.

While these problems can be impossible to solve, there are efficient algorithms to find near optimal solutions in a practical amount of time. In most cases the types of flexibility that exist will be limited for straight-forward reasons. Much work has been done looking at both flexible job shop scheduling systems with machines that are fixed but can perform a variety of tasks[17], [18], [19]. Significant work has also been done to optimize schedules for mobile robots[20], [21], [22].

2.5 Summary

When considering AFS, one of the main reasons for flexibility cited was factory flexibility—the ability to copy the system or move the system to another manufacturing site with minimal resources or time. The primary benefit of flexibility that LSM hopes to achieve with AFS is packaging the complexity that used to be handled by large monumental tooling, complex factory systems, and skilled assembly workers and capture it all in robotics and software so that it is portable. Ideally the factory could be moved in a period of weeks or a new factory could be built by simply buying more robots and putting them in a building with a flat concrete floor, air, and power.

An observation about AFS is that they have traded flexibility at the system level for flexibility at the factory level. There are many concepts that could have been used for AFS' design, but the chosen one requires massive robots on massive platforms. The robot platforms are mobile, but to provide the stability needed, they are very large. This limits the number of robot pairs in the system to four, and each is limited to its respective quadrant of Product A. In the old manual system with assembly workers, there used to be a great deal of labor flexibility in the production process. If the build started to fall behind schedule, the solution was simply to add more workers. If AFS had used a different concept, for instance robots on a gantry or a structures bolted to the floor, more robot workers could have been added increasing system capacity and production flexibility but making the system more difficult to break down and transport or copy.

As pointed out in the literature, a flexible system may not be right in all circumstances. The cost may not be justified when there is low mix or when the time to market is slower. The Product A is a low mix product has been in production for almost 20 years, and it is anticipated that it replacement which is due in several years, will be in production for another 15 years. Once this system is up and running, it will only rarely need to be used to produce new products. LSM does have some other structural products that are much smaller where this system, once developed, could possibly be used, however, it is unclear whether it could do so without significant modifications and rework. It would certainly not be the equivalent of taking a spot welding robot and have it weld a truck instead of a sedan.

Another important point is that because there are so many variables and so few constraints in a flexible system, modeling and simulation are important tools to design the system and understand its capacity. There are potentially an infinite number of ways that the AFS robots could process a Product A. The ideal sequence used to process a Product A may not be obvious, especially if the work to be done and the critical path to do it are not completely understood. The next chapter will provide a more detailed description of the production process and of AFS that will be important to understand so that the modeling rationale is clear.

Chapter 3

Description of the system

The first step in improving a system is understanding the current process. The purpose of this section is to provide a deeper understanding of the system being studied including its configuration and constraints as context for the modeling work performed and the improvements suggested.

3.1 Description of the manufacturing process

The Product A has been in production for about 20 years. It is a large, closed, metal structure that has internal support ribs that run longitudinally and circumferentially and is similar to what is found in the mining, construction, aerospace, or shipbuilding industries. A generic diagram of the structure is shown in Figure 3-1. The Product A is comprised of several sub-sections that are butt-spliced together. Each sub-section is made of several panels. Panels are comprised of an outer piece plus attached support structures which come to LSM from tier-1 suppliers.

To manufacture Product A by the legacy manual process, the panels are attached to a large assembly jig that locates the panels relative to one another. Once the panels are loaded into the assembly jig, assembly workers proceed to drill holes along all the seams. Once the holes are in place, the joints are split apart, de-burred, and a sealant is applied. An important note is that the sealant is applied joint by joint as they are processed. Finally, the joints are fastened using either several types of fasteners.

The fastening process requires two assembly workers, one on the outside and one on the inside to respectively install the fastener and provide backing pressure. This a very physically demanding process and for some installation locations is difficult ergonomically.

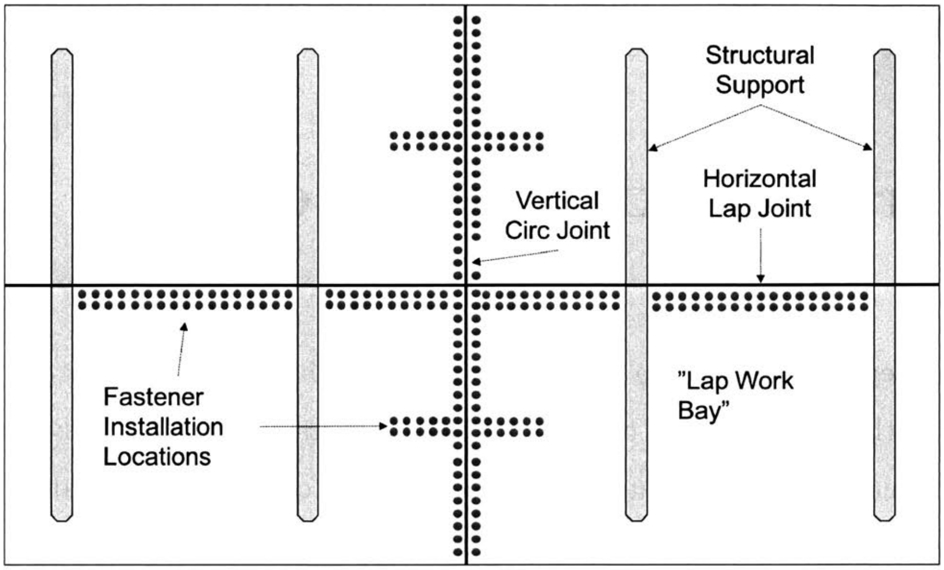


Figure 3-1 – A general diagram of Product A structure.

3.2 AFS concept

The Automated Fastening System concept started more than 5 years ago with initial research looking into the automation of the fastening process. Over time the fastening work progressed into a manufacturing solution that formed the basis of AFS. As AFS was developed, one of the company mandates that LSM gave to its design team was "No roots, No vines." In other words, the system was to be designed so that there was no monumental tooling and no fixed factory requirements other than a flat concrete floor with air/power stations. The intent was to allow a factory that was easily reconfigurable, easy to replicate, and easy to move. To expand capacity, new factories could easily be created, or if need arose, the entire factory could be moved in a matter of weeks to a more advantageous location.

The manufacturing line was designed with three work cells. The first cell is pre-automation where Product A panels are loaded and prepped. Cell two is the automation portion where AFS operates. This cell contains robots on the inside and outside of Product A that work together to install the fasteners. Then there is a final post-automation cell to finish the remaining manual tasks on Product A before moving it out to another part of the factory.

3.2.1 Pre-automation

One of the first outcomes of the AFS mandate was a change in the way that the panels were brought into the factory and prepared for manufacturing. Instead of being loaded into a large fixture that locates all the panels, the panels self-locate relative to one another using a determinant assembly method. Lego is one of the best examples of determinant assembly; it uses the built-in features of the brick to both locate and attach the next brick. This method required the panel design to be altered slightly at the supplier to include accurately placed, pre-drilled locating holes on all the parts. To assemble Product A panels, the first set of panels that form that length of the tube are loaded into a mobile, adjustable cradle system. The cradles are adjusted until the locating holes align. Then the panels are fixed together by inserting steel pins into the locating holes.

Once the base is laid, the sides of the structure are built-up. The first level of side panels are brought in using cranes and then assembly workers pin them to the base panels. After that, a structural cross-member that horizontally bisects the main structure is loaded and attached (this eventually becomes a floor for the inner robot). Once the floor structure is in place, another row of panels forming the upper sides are loaded and then finally the upper crown is put in place. All of these panels are joined to one another using removable steel pins.

Once all the panels are in place, assembly workers apply sealant and then begin drilling holes and tack fastening the panels together with temporary fasteners. Workers will place roughly 2 tack fasteners for every 50-60 fasteners ultimately required. These fasteners are undersized and later will be drilled out and replaced with full-size fasteners. The tack fasteners serve two main purposes: 1. They provide the structural rigidity Product A needs before the robots can begin work. 2. The tack fasteners are used by the robot vision system as a means of locating the robot relative to the to the structure. This means it is important that the holes are drilled and the tack fasteners are inserted with high precision. Once the tack fasteners are in place, the steel pins are removed.

At the completion of tack fastening, the unit is ready to be fully fastened by the robots. The application of the sealant prior to tack fastening is an important milestone in the build process of Product A. The sealant has a working life of several days and all fasteners must be installed prior to the sealant clock running out. To install fasteners after the seal has cured, the joint must be split apart, the sealant must be scraped out and reapplied, and then the fasteners manually installed.

3.2.2 Automation

System overview – A single automation cell is composed of a team of robots similar to the system shown in Figure 3-2. The robots are on mobile platforms and work in pairs with one robot inside Product A to provide backing pressure and one on the outside that does the drilling and fastening. The Product A has an inner structure along its horizontal centerline which effectively divides Product A into upper and lower sections. Furthermore, the robots are also assigned to one side or the other of Product A. Therefore, a team of robots will typically consist of eight total robots: an upper left-hand pair, and upper right-hand pair, a lower left-hand pair, and a lower right-hand pair. All the robots sit on mobile platforms. The platforms for each outer robot rest on the concrete floor and have a base height that allows them to reach either the upper or lower section they are assigned to. The inner robots from both sides are attached to a single platform. There is one platform for the upper section and one platform for the lower section. The lower robot platform is able to ride directly on the structure of Product A, while the upper platform rides the centerline structure.

Outer robots – AFS has four outer robots that have been assigned to either side of the structure as well as to an upper or lower position. The outer robots carry

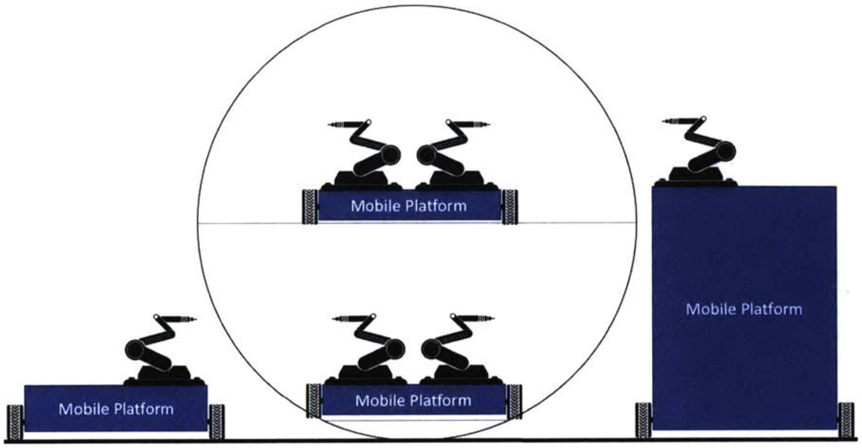


Figure 3-2 – A general diagram of the configuration of the robots and Product A.

the end effector that has the primary components for the fastening process, namely, the fastener feed system, the drilling modules, the fastener insertion modules, and the fastener hammering module. Each of the outer robots and all their necessary components sit on their own platform that can be moved with an automatic guided vehicle(AGV). Every outer robot has a corresponding inner robot.

Inner robots – The inner robots are each paired with an outer robot. As an example, an outer robot assigned to the upper left-hand side of the structure will be paired to an inner robot assigned to the upper left-hand side. The inner robots carry two end effectors used for providing backing pressure during fastener installation. One of the main constraints of the inner robots is that they are mounted in pairs on their mobile platforms. An upper set of robots consists of both a right and left-hand robot which are mounted to a single platform. Consequently the pair must be moved from location to location as single unit. As will be discussed later, this is a major system constraint.

Air/power coupling – The robots move around the factory on battery power. However, when performing manufacturing operations, they must be tethered to air and power stations. Various air and power stations have been placed around the factory to allow the robots to work flexibly. Coupling and uncoupling is automated. Once a robot system is coupled it is only able to move the length of the tether that is attached.

End effectors – The end effectors used by the outer robots are designed to perform multiple functions. The end effectors carry several tools that can be selected without making a major tool change. The tools include at least one of each of the following: a drill module, a fastener insertion module, and a hammering module that provides the impact necessary for the fastening process. Depending on the configuration of the module, there may be additional tools on board.

The inner robots use two different styles of end effectors. Both are used to provide backing pressure to the outer robot during its various operations. The first of the end-effectors is called the standard end effector(SEE). The SEE provides backing pressure during drilling and hammering and has two points of contact, one axis-aligned touch point for directly accessible fasteners and one off-axis touch point for fasteners in more difficult to reach areas such as under structural ribs.

The second inner robot effector called the heavy-duty end effector(HDEE) is a sturdier module that can provide off-axis backing pressure at higher loads and is used after the SEE as a secondary operation when needed. It is required because the load requirements to install the some fasteners are much higher than the SEE can support in some instances.

Fastener feed system – The AFS receives its fasteners from an automated fastener feed system that sends a fastener just prior to insertion through an air-powered sending system.

Fastener types – There are five main operation types (three of which include installing a fastener) that the robots can process that are listed below. The distribution of these processes for a typical build is shown in Figure 3-3.

1. **Type 1 (T1)** These are the basic fasteners on Product A represent the bulk of the installations and are installed using the SEE. After robotic installation, T1 fasteners require no additional processing or human work. A typical Product A uses Type 1 fasteners with a variety of diameters and lengths.

2. **Type 2 (T2)** These are a second type of fastener that are first installed using

the SEE but then require a secondary operation using the HDEE to apply a higher load. T2 fasteners do not require any human work after the automation process.

3. **Type 3 (T3)** These are a third type of fastener that are used in far fewer locations using only the SEE. All T3 fasteners require a secondary human operation after the automation completes.

4. **Drill-only (DO)** There are a variety of fastener locations where the robots only drill a hole but no fastener is inserted. The fasteners for these holes are installed after the automation work by an assembly worker during the post-automation period. Drill-only holes are typically done because a fastener head would get in the way of subsequent fasteners being installed or to allow for a quality inspection later.

5. **No Process (NP)** These are locations where the robots only do a fly-over. These locations are tracked in the robot programs, but no work is performed.

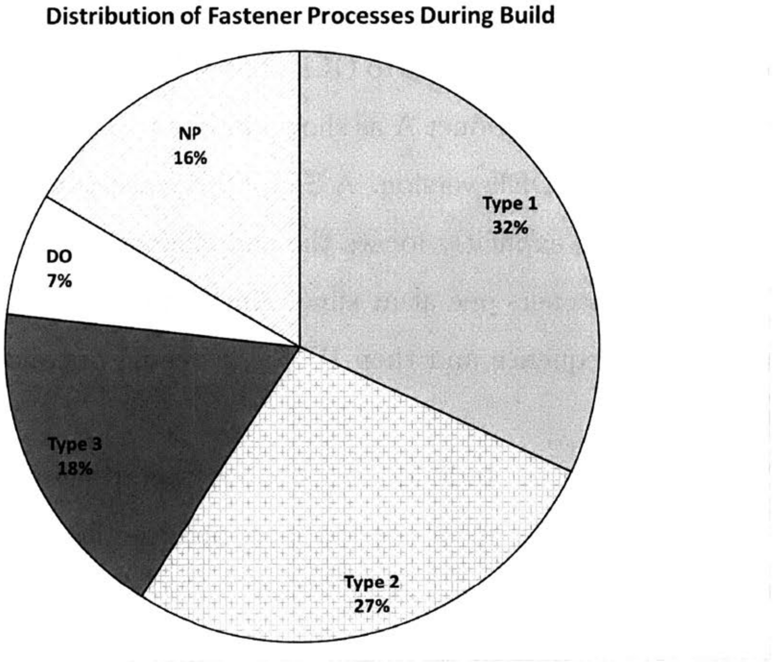


Figure 3-3 – The distribution of the fastener types installed in a typical build.

Master control panel – The entire system is designed to be driven by a master control panel (MCP) that controls the entire factory. The server holds the build plan for each product variant, dispatches the robots to their work locations, and then commands them to execute pre-defined programs.

The control sequence is important to understand. In the build of a Product A, there are programs and execution sequences that exist at a variety of depths.

1. MCP Command: The MCP issues commands to be executed at a high level. These typically consists of commands to:

- Move a robot platform to a new work location
- Couple or decouple the robotic platforms to power/air stations
- Perform laser metrology on the system to ensure coordinate system alignment
- Execute an NC program (OLP)

2. Offline NC Program (OLP): The Product A is divided up into many small work packages. Each work package is accessible from a single robot platform parking position and will usually include somewhere between 20-50 fastening operations. Operations are most often assigned to OLPs based on naturally occurring boundaries (bays) in the structure of Product A as shown in Figure 3-1. OLPs come in two types: a SEE version and an HDEE version. A SEE OLP consists of three main tasks: execute a vision macro to explicitly locate the end effector and the work location, then drill and insert all fasteners one at at time. An HDEE OLP follows a SEE OLP and consists of a vision sequence and then HDEE fastening operations in locations that require higher loads.

3. Operation Macros: Individual fastening operations are defined by specific macros. Macros are made up of the low-level machine language needed to actuate the robots and their end effectors and contain the specific parameters needed such as drill size, drill speed, hammering force, hammering time, etc. There are individual macros for every unique operation that can be performed in a build.

The workflow – At a high level, building a Product A follows the process shown in Figure 3-4.

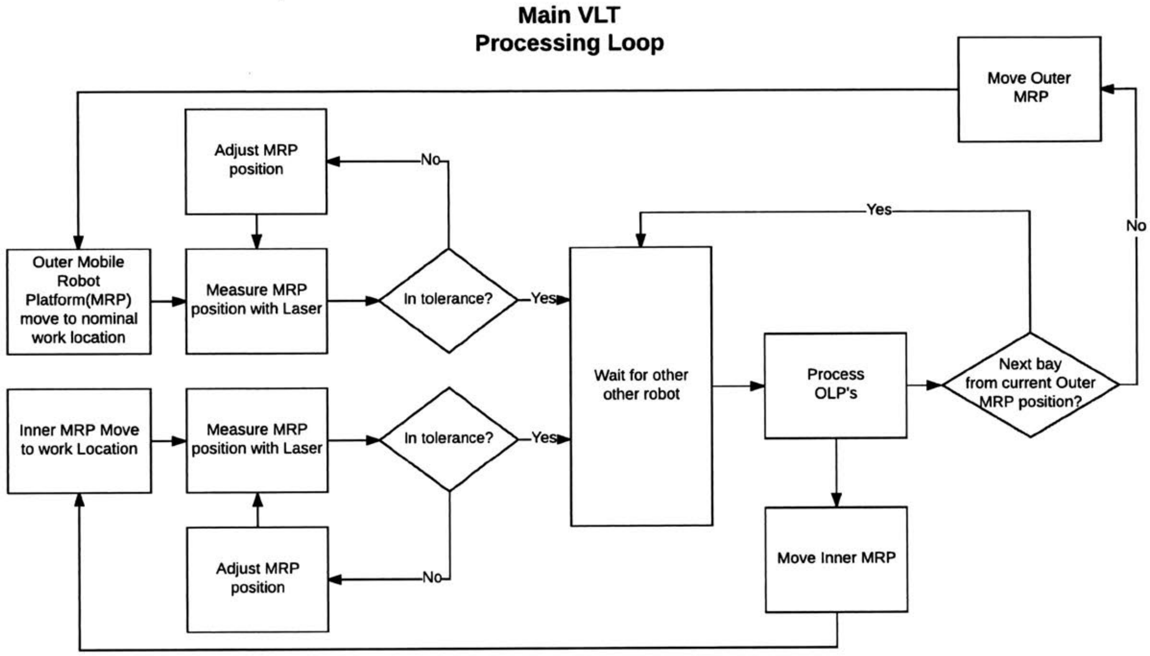


Figure 3-4 – A flowchart describing the main steps in processing a Product A.

The robots move from bay to bay around Product A, and when in position execute the OLPs for that bay. OLPs are separated into two categories: SEE and HD. The steps for each type of OLP are listed below:

SEE OLP

1. Vision: The robots each first execute a vision sequence to find the tack fasteners installed earlier by the mechanics. Using these fasteners, the machines are able to calculate their exact position relative to the work area and make small adjustments if needed.

2. On-Axis Operations: Does not require any secondary robot operation.

    2.1. Move effector to hole location

    2.2. Clamp the inner and the outer robots to the structure

    2.3. Drill the hole

    2.4. Insert the fastener
    2.5. Light hammer to ensure the fastener is inserted (interference fit)
    2.6. Final hammer
    2.7. Unclamp the robots from the structure

3. Off-Axis Load Operations: Requires HDEE to follow
    3.1. Move effector to hole location
    3.2. Clamp the structure by applying pressure from the inner and the outer robots to the structures surface
    3.3. Drill the hole
    3.4. Insert the fastener
    3.5. Light hammer to ensure the fastener is inserted (interference fit)
    3.6. Unclamp the robots from the structure

HDEE OLP

1. Switch inner robot end effectors from SEE to HDEE
2. Vision (Same process as SEE)
3. Paddle Operations: Use the HDEE which can withstand higher fastening loads on fasteners installed using the SEE that need it.
    3.1. Move effector to hole location
    3.2. Clamp the inner and the outer robots to the structure
    3.3. Final hammer
    3.4. Unclamp the robots from the structure

3.2.3 Post-automation

After the automation statement of work is completed, manual post-automation work needs to be done by assembly workers. This work includes secondary manual operations on T3 fasteners and installation of any fasteners that the robots were unable to reach. Currently it also includes drill-out of all tack fasteners and insertion of the proper sized fasteners. Additionally, there are some other manual clean-up and installation operations that need to be done before Product A can be moved. In the case that AFS does not accomplish its full statement of work during the automation schedule window, this is often the period when the production group will try and make up the lost time.

3.3 System Qualification

Qualification receives a great deal of attention by the AFS team. Product A's are costly to produce and have very high quality and performance requirements. In order for AFS to operate on a Product A, the system undergoes a series of tests to ensure it is functioning within specification.

The first set of tests is a fastener quality check that occurs in a mock production setting. To do this test, coupons are used. A test coupon is a stack of rectangular metal sheets with a similar stack-up as specific locations on Product A. Different stacks are used to represent different structural features on Product A. The coupons are held in a special rigid fixture while the robots run a test fastener installation pattern. The pattern is typically a simple rectangular grid over the surface of the coupon with roughly 50 fasteners. Coupon tests are run for every fastener type that the robots will install on Product A. After the coupons are processed, each of the fasteners in the coupon is inspected to ensure quality installation. Anytime there is a change in tools, fastening parameters, or machine firmware, the coupon qualification process is repeated.

The other main qualification process that takes place is OLP validation. When an OLP is created, it is first checked for clashing between the robot and Product

A structure in CAD. However, before being allowed to run at full rate on Product A, the first time it is run in production, it is first run as a fly–over process with movements only (no holes are drilled or fasteners inserted) at a reduced movement rate. Operators watch the robots carefully to ensure that they do not contact the structure of Product A. Once the fly–over has been completed, the OLP is re–run and the program is considered validated.

3.4 AFS requirements

3.4.1 Fastening requirements

Because Product A is a pressure vessel, it has extremely strict structural requirements. It is also required to have a premium exterior surface finish. The quality of the installed fasteners is extremely important. Parameters for the parts of the fastening process that actually touch the surface of Product A (i.e. drilling speeds, clamping forces, and hammering times) are tightly controlled by the technology development group. These requirements are kept in a program document and require significant validation to be changed.

3.4.2 Rate requirements

The business case for AFS was built around 25,000 fasteners being installed by four pairs of robots at an expected hole-to-hole rate. This expected rate is referred to in this document as the Required Mean Fastener Time (RMFT). The RMFT is used to nondimensionalize the actual data used in this project. It will be discussed later, but the business case does not account for several factors such as setup time, robot platform movement, and down time. This makes the RMFT an extremely optimistic goal. If the system were to install at this rate with the above assumptions it would take 85% of the allotted automation time. In reality, as will be discussed in much more detail later, the assumptions of the business case do not hold well. Some of the business case assumptions are:

- Robots will insert all 25,000 fasteners
- The factory will operate three shifts/day 5 days/week with only 2 operators required to run all automated equipment
- Non-value add time (setup, movement, scheduled and unscheduled downtime) allocated 15% of automation time
- All maintenance will happen on weekend
- Robots work is evenly distributed side to side and top to bottom

3.5 AFS system constraints

3.5.1 Temporal constraints

The first macro constraint is the production schedule. The automated portion of work has been allocated a fixed amount of time equal to the takt time of the factory. The next major constraint is that because Product A is a pressure vessel, a seal is applied in between the panels. The sealant has a limited working time which is a hard constraint for AFS. If the seal life is exceeded, the sealant on unfastened areas must be manually removed and then reapplied by assembly workers as the remaining fasteners are installed by hand. If the full-rate production target is met, the seal life is not an issue because it is greater than the time allocated to AFS operation. However, during ramp-up the seal life constraint has almost always been the limiting factor to the amount of work that is done. It is worth noting that the seal life constraint is a non-issue in the manual process because the sealant is applied as the fasteners are installed, whereas, in the automated process, the sealant is applied on all panels prior to beginning the automation sequence.

3.5.2 Spatial constraints

Each outer robot arm is mounted on a mobile platform; the platforms are extremely large. Logically, two robots cannot be in the same place at the same time. Each side of Product A has an upper robot and a lower robot, each on their own platform. Thus upper work and lower work may not be done at the same place at the same time. To maintain a buffer zone between the systems, the robots are required to maintain a minimum distance between themselves. Additionally, because the inner robots are on the same platform, effectively, the upper right-hand robot is tied to the upper left-hand robot and they essentially function as a single unit. The same is true for the lower set. This means that the uppers and lower must each move as a set.

Another constraint that exists in regard to positioning is that the outer robots are only allowed to reach two bays ahead and two bays behind a platform parking location (shown in Figure 6-2). The actual reach of the arm is greater than the ±2 bay requirement, however, quality became less reliable beyond this reach under certain circumstances. So as a conservative measure, a program requirement was implemented limiting the reach to ±2 bays.

An additional constraint on the robots is that when doing a circumferential joint (butt splice joining two sections), the two bays on either side of the centerline at the top and at the bottom of Product A only allow either the right hand robot or the left hand robot to be working at a time; both can not work simultaneously. This means that during this portion of the build, one robot is forced to be idle while the other works. As an an example, when the upper set of robots is doing a circumferential joint working on the upper-most bays at the top of the structure, only one end effector can be in the space at a time. So the right hand robot will go first and process two bays while the left hand robot waits idle. Then the left hand robot will process two bays while the right hand robot waits idle. At that point the two robots can continue to process in parallel. A similar situation occurs for the lower set of robots in the bottom-most bays.

Spatial constraints are also an issue for the inner robots. The structure on the inside of Product A is complex, and limits the access of the inner robots to the fasteners. This is one of the reasons the inner robots have to perform so many movements. In addition there is a program requirement that the inner robots are only allowed to perform work on the bay immediately in front of the platform. This is for similar reasons as the bay limitations on the outer robots, except it is more restrictive because of the additional structure the inner robots must work around.

3.6 Current state vs. future state

AFS has several known issues that are limitations to its current capability. These are important understand because in order to model the properly assumptions need to be made about the future state. The first of these, and the one with the greatest impact, is that the mobile platforms that the robots are mounted to must be moved manually by a human operator. The end state anticipates that these movements will be fully automatic. Because of the manual movement, the time required to move the platforms in the current state is much larger than expected in the end state. In general the platforms move more slowly and less accurately (often requiring several positioning attempts) when done manually.

Another significant issue that exists is the lack of a central command system—the MCP. This is also in development, but currently, all programs and commands delivered to the robots are loaded and sent through a handheld pendant. This means that currently new NC programs must be delivered to the pendants via a thumb drive. Although the process of loading a program via thumb drive is not a terribly long process, what tends to be a problem is the communication and coordination that takes extra time that will be eliminated when it is all automatic.

3.7 Summary

AFS is a complex system. The primary goal of automating the assembly process is to reduce direct labor, improve quality, and improve worker safety. At the same time, in its implementation, the system is designed to maximize the flexibility of the factory. As mentioned in the previous chapter, this forced several constraints on the system. The main limitations of AFS are that:

- The maximum number of robot pairs that can work on a Product A at any one time is 4

- The robots are only able to work in their assigned quadrant and the work is not balanced top to bottom or left to right meaning there are natural bottlenecks in the system
- The inner robots are joined together on a single platform which essentially forces the upper robots and lower robot to function as units
- Everything in the factory is mobile with no fixed reference points making positioning, especially during movements, more difficult
- There are several features, primarily automated movement, that are not fully developed yet that are key to the performance of the system

The next chapter will discuss some of the challenges of implementing AFS faced by the organization and the additional impact it has on performance. Following that, the model that was developed will be introduced.

Chapter 4

Organizational considerations

The implementation of AFS by LSM is a major automation project, that by its nature, is highly complex. One of the inherent challenges to the implementation of AFS is that it takes place in a "real–world" setting. This distinction is important because as opposed to a well-controlled laboratory, in a corporate setting, events, organizations, and timelines often do not work ideally and must be accounted for. Implementing a project at the scale of AFS requires a large team made of LSM engineers, assembly workers, subcontractors, suppliers and more. This section highlights some of the organizational challenges that are relevant to the implementation of a highly complex system that have an impact on the system"s performance. This section also shows that the modeling process can be used to better bring out information from various groups that might otherwise remain buried. Some of the content of this chapter is from personal observations and anecdotes that seem representative of the challenges LSM faces.

4.1 The AFS team

The groups implementing AFS are divided among two primary organizations: the R&D organization which is in charge of developing new technology and the production group which is charge of building Product A. The R&D organization has a relatively small head count compared to the overall number of people working on the AFS program and has an entirely different reporting structure all the way up to the CEO of LSM. The R&D group is primarily responsible to ensure that the equipment has the capabilities needed to be viable before implementation in a production setting.

The other main organization in the Product A department is the production organization. The production organization is in charge of operating the equipment and producing Product A's to the factory schedule. The production organization has many sub-organizations including separate groups that handle equipment design, robotic programming, manufacturing planning, system integration as well as the manufacturing staff.

One of the major challenges for me was working across the two main departments. Many times there seemed to be overlap between groups and often parallel initiatives (including initiatives to model the system) were taking place. Because AFS was being implemented in the production line at the same time that many of the features and capabilities were still being implemented, it was often unclear which group had true ownership over projects, tasks, and data.

4.2 Working with subcontractors

Subcontractors play a major role in the development and implementation of AFS. As a starting point, the integration and mechanical development of several of the systems are done by subcontractors. Subcontractors involved include Robotic Systems, Inc. (RSI) a robot manufacturer who is also functioning as overall system integrator. Also heavily involved is Precision Fastening Systems (PFS) who is responsible for development of the end-effector. There are also several other 2nd and 3rd–tier suppliers involved in this project supplying major system components such as tooling, fastener feed systems, programmable logic controllers (PLCs), etc.

In general, the relationship with RSI, the primary subcontractor, was strained. In particular, because AFS production rates were not improving as fast as desired, both LSM employees and RSI employees were feeling pressure to perform. Keeping the relationship and communication open was, at times, difficult. These statements are not a judgement, but simply an observation that although everyone had a stake in AFS' performance, as the pressure to perform grew stronger, the relationship between supplier and customer was also challenged.

One of the main issues that the heavy reliance on subcontractors presents is that much of the technical expertise around these systems exists outside of LSM. Many times issues that need to be resolved are outside the capability of the core LSM team and take longer to resolve due to tracking down the right subcontractor with the appropriate skills.

The involvement of so many subcontractors means that integration of the system is a major challenge. An example of this that will be discussed in more detail later is the tool change system. The NC programs that the robots execute are really a series of predefined macros. The various macros for different steps in the fastening process were written and maintained by LSM and several different subcontractors. In the example, because of the challenge of communicating across so many groups, tool changes were happening much more frequently than needed.

4.3 System improvement in a production environment

LSM made a decision to move AFS into production as soon as possible under the assumption that the fastest way to bring the system up to speed would be to build actual Product A"s with it. AFS underwent some early stage testing on a representative test structure and after the first structure passed qualification, the decision was made to move the system into a production setting. The goal was to start slow and ramp up the production rate over time as they progressed through the learning curve. However, this has presented several formidable challenges.

One of the fundamental challenges that the production environment creates for system development is time pressure. This often forces a conflict between short and long term goals. The short term goals are tied to meeting production targets on the current Product A being assembled. The long term goals are tied moving down the learning curve as fast as possible to achieve the long-term production rates. An example of this trade-off is when an end-effector jammed due to chip build-up. The decision was made to clean out the effector and continue production and do the root-cause analysis after production was finished. The problem in this case was that root-cause analysis was difficult to do after the fact and would have been easier to do in the moment. However, it also would have greatly slowed production of that Product A.

Another constraint the production process places on implementing new features is that when robots are used in production, they are not available for new feature development. New software and hardware must first be validated before being used on a production Product A, so upgrading software or performing tests on the equipment was slowed significantly. An example of this impact was that software to automatically collect data from the robots was delayed for several months. At the beginning of the research period, there was one functional set of robots that had been in production for four months and automatic data capture had not yet been implemented. All timing had to be done manually which greatly limited the access to and quality of performance data for the robots. This was eventually solved with a software update, however, the software update took longer to develop than it could have because there was no way to develop and test the software while the robots were in use on a production job.

There was also a staff issue. The staff needed for new feature development and qualification were required to be on–hand during production work. This was so that in case anything went wrong, there would be technical staff readily available to fix the equipment and minimize downtime during production. This limited their time spent to bring new system enablers online. In the end, the decision to do as much learning as possible in a production setting slowed the progress of system capability.

4.4 Organizational and political challenges

One of the major organizational challenges I experienced during this work (and is representative of many others in AFS) was the difficulty of getting information from silo'd organizations. The following are two personal anecdotes that are indicative of larger challenges in the integration of a complex system such as AFS.

During the creation of the simulation model (discussed in-depth later), one of the tasks that needed to be addressed was how drill bits were changed by the machines. The process logic of a tool change was assumed to be as follows:

1. A new hole is specified
2. The machine checks to see if it has the required drill bit loaded in one of the two end effector drill bit slots
3. If not, the machine returns to the tool carousel to retrieve the required tool
4. The machine must decide which of the two drill bits currently loaded in the end effector to remove and replace with the new drill bit

In my attempt to validate the logic used by the system to do tool changes, many people were consulted. The first stop was the NC programming department. The NC programming group programs the tool paths for the robots. When programming a specific hole, detailed instructions for the robot (such as which tool to use) are actually handled by a lower-level set of machine instructions. From there I went to the equipment engineering group, who said that the logic for tool changing was handled by the integrator RSI. RSI said that part of the code was written by one of their sub-contractors. In the end, I found someone who could describe the tool change process who worked for a second-tier subcontractor. One of the surprises that came out of this was that the tool assignments were hard-coded into the base level macro definitions. This was based on some simple assumptions and was done to speed up deployment of the code, but ultimately this led to many extra tool changes. Other cases similar to this were discovered because the work was divided in a way that it was difficult for any one person to have a complete vertical view of the system.

In other cases there were major challenges getting data from various groups. Interior political challenges meant that groups often would not share data or work with other groups. At times suspicion existed between groups that data was being withheld or manipulated to gain an advantage. In one such case, there was a discussion about whether or not performance data from the machines existed. In the discussion, the data was believed to exist, but several people said that it was retained by RSI who was not releasing it because it would reflect poorly on RSI. After approaching RSI directly, I was given full access to the database. The conflict was purely imaginary, but it had still prevented the team from collecting data it needed.

4.5 Summary

In AFS, LSM has undertaken an extremely complex project which fundamentally requires a large organization spanning many groups and teams. The purpose of this chapter is to show that in the process of implementing such a system, ensuring that groups communicate and share data has a material impact on the performance of the system being implemented. The process of modeling the system is a vertical role that required interfacing with almost all departments in the group and served as an important go-between. The two personal experiences described above (or others similar to it) were repeated many times while working across the AFS group. These observations are made for two reasons. First, to highlight the fact that organizational performance has a real impact on the integration and performance of the system. In these cases, decisions were being made about the implementation of AFS without access to all the data. The second point is that part of the value of a model or simulation is that it forces groups to integrate in ways that they might not otherwise. As in the case of the sub-optimal tool-change programming, such information might otherwise remain buried, limiting the performance of the system.

The next chapter will discuss the model specifically and the decisions that were made in building it. The decisions made during modeling are based on both the system, and its current state as was discussed in the previous chapter, but also some of the organizational considerations such as who would be using the model and what their role was is in the AFS group.

Chapter 5

Modeling the system

This chapter describes the model that was built of AFS to help determine the system capacity and provide insights into its performance. The modeling process required piecing information together from a variety of sources and working across the AFS group. The decisions on what on what type of model to build, how much detail to include, and some of the challenges in obtaining the needed data are discussed as well as the results and insights gained.

5.1 Previous system models

Prior to describing the current model, it is worth considering other models that have been built for AFS, why they were created, and the challenges that they have faced in implementation. These challenges have been both technical and organizational.

5.1.1 Back of the Napkin

The earliest models of AFS used during its conception and through development of the business case were rough estimates that made large assumptions. This is also where the Required Fastener Time (RMFT) comes from. The model consisted of simple math that said if there are X fasteners on a Product A, and there are 4 robot teams, and the cycle time for a single fastener is Y seconds (1 RMFT), then the total time to produce a Product A is given by:

$$ProductA_{Time} = \frac{X_{\text{fasteners}}}{N_{\text{robots}}} * Y_{\text{sec/fastener}} \tag{5.1}$$

These early calculations were done to build the business case and justify the creation of AFS. When the baseline numbers are applied, the build time for a single Product A is 86% of the allotted schedule time. Although the work time fits within the schedule window, it is based on several assumptions including:

- Robots will insert all 25,000 fasteners
- The factory will operate three shifts/day 5 days/week with only 2 operators (a major factor for the business case) required to run all automated equipment
- Non-value add time (setup, movement, scheduled and unscheduled downtime) allocated 15% of automation schedule time
- All maintenance will happen on weekend
- Robots' work is evenly distributed side to side and top to bottom

After the system was approved for production, other models were generated. However, the later models did not significantly address many of the initial assumptions.

5.1.2 The Fox Guards the Hen House

During the design phase for the system, a model was requested from the primary contractor on the project. The contractor built a discrete events simulation. The model included significant additional detail beyond the original model. However, because it was done early, there was never any validation done. The results showed that the time to build a Product A was equal to the time allotted. LSM engineers on the AFS project reported that the numbers were simply adjusted by RSI until the model showed the outcome LSM expected. (This was the common perception of many LSM engineers, but the origin of the underlying data could never be confirmed since the actual model or its creators could not be located. In general, the relationship between AFS and RSI was strained during this research period.) This model was used to fulfill a required step early in the design process and was never maintained. Only the output of the model was provided to LSM, never the model itself.

This model makes the following assumptions:

- Structure divided into groups of bays and work time was assigned at a panel level
- All times rough estimates
- All system technologies fully implemented
- No allocation for downtime or maintenance

5.1.3 Early ProModel

At the same time the contractor was asked to build a model of the system, the industrial engineering department started to build their own model. One of their engineers built a Discrete Events Simulation(DES) model in ProModel. The model included the extra detail of breaking Product A into its various sections and individual work statements for each of the robots. The movements of the robot platforms also better reflected reality. It was a step forward from the initial business case model. However, this model was abandoned because it disagreed with the contractor's model which was being built at the same time. The disagreement between the two models created political conflict in the department because its predictions for Product A flow time were much longer than the schedule time allotted for building Product A. Ultimately, the engineer on the project was asked to stop and moved onto another project.

5.1.4 Manufacturing Group Excel Model

At the start of the research period, the only model of the build process being used in the AFS group was an Excel model. This model was built and used by the manufacturing engineering group who were in charge of manufacturing planning for each

Product A build. This model captured the work at a bay to bay level and assumed a common time for each of the bays. It also dealt with robot platform movement and assumed an optimistic 85% utilization rate. It was a good tool to do a high-level analysis on the process. The details it missed were:

- It did not account for individual operations in each bay, it just assumed all bays were equal
- It did not account for machine-machine interactions (The robots often get in the way of one another and there was no way to model this practically in Excel)
- The input data was based on cycle times from testing using an early version of AFS

The Excel tool captured the main tasks that the equipment needed to perform a build. The model results showed that AFS didnt have the capacity to complete the full statement of work. In general, this model was often dismissed because people argued that it was too simplistic and didnt have enough detail or that the input data was based on old performance data and future versions of the system would be faster. Probably more critical to the acceptance of the model, the individual who maintained the tool was quite unpopular and people often sought reasons to discredit him because he had contrarian views and expressed them with little regard for tact.

5.1.5 The Academic Model

Two thirds of the way through the research period, a new model was introduced by the R&D group that had been in parallel development to the DES model developed as part of this thesis. The R&D department had been funding a model that the production group was not aware of (hence the parallel effort). The work was being done by a university professor with a team of students. The model captured all of the work-bays and machine locations, modeled robot-robot interactions, and accounted for the various constraints that the robots must respect. It was built with a simple user interface and allowed easy selection of work packages for the robots to perform.

The major advantage of this model was that the assigned work for the robots could be arbitrary and the tool would not only predict the time needed to finish the work, it would find the optimal schedule for the robots to accomplish the work. The optimization methods involved were quite advanced and many of the schedules produced were non-intuitive. This model agreed with the Excel model suggesting that, with the given inputs and assumptions, the optimized build plan would not meet the required flow time.

The main limitations of this model were:

- The data used in the tool was based on generalizations instead of explicit work requirements for each bay. All bays were just given a standard time when in reality the time for different bays could vary significantly depending on the work to be performed
- Cycle times were all rough estimates and often overly optimistic
- The software was license restricted and plans at the time did not allow for general use of the tool
- The tool was bespoke software and would likely require significant further outside investment
- The tool was controlled by the R&D group, however the primary need for the tool was in the production group. The two groups often had issues working together.

Overall, this tool was the most complete to date. Its primary weaknesses were poor inputs and access. It is likely that the ideal tool would be a combination of the tool built by the academic team plus many of the data inputs used as part of this thesis. Such a tool would have good estimates for process times, a means to accurately get the tasks to be done at each work-bay, and the capability to optimize the schedule.

5.2 Model Methodology

5.2.1 Need for a new model

At the start of this research period, the production organization requested help to determine the capacity of the system at full-rate production. As was just described, several models and tools existed. However, for technical reasons and for organizational and political reasons, none of the tools provided a result that was satisfactory. The tools that had been created had either been abandoned because of politics or were incomplete and did not have the confidence of the other groups in the department. Sometimes the different models were assumed to have been created in order to discredit others. These tools were especially inadequate when it came to selecting different statements of work or specifying different schedules for the robots to perform. In addition, they all had suffered from several issues:

1. They were not based on valid cycle time data. To this point, all performance data the models were built on had been based on early estimates or optimistic expectations.

2. The models were not based on the actual work the robots were assigned to do--instead high-level generalizations were used.

3. Politics had made most of the tools unavailable to production engineering, manufacturing engineering, and industrial engineering--the groups who had the primary need form them.

4. The tools that did exist were generally not trusted by groups that had no part in their development.

5. Validation of the tools had been a challenge because of a lack of good data from the production floor.

This thesis was an attempt to fix those issues. This was done by picking a level of detail for the model that would be useful for planning purposes, basing the model on real data, providing validation, and getting input from many stakeholders across the group so that the various groups would feel confident in its use.

5.2.2 What is important to learn (The four buckets)

When establishing the key outcomes of the model, it was clear the sequence was not well understood. The constraints were known, but the bottlenecks were not. It was decided that the time spent by the robots needed to be better understood to be able to find the critical path and understand where improvements could be made. Along those lines, the first step was to assign time spent by the robots in one of four categories:

1. **Waiting:** An individual robot is in position ready to start work. However, it is waiting for another robot.
2. **Working:** A robot is executing an OLP-this means it is moving the end effector to a new hole, performing localization with the vision system, drilling holes, inserting fasteners, or hammering fasteners. This also includes the robot arms moving from hole–to–hole.
3. **Moving:** A robot platform is moving from one work location to another. As will be discussed later, platform movements were modeled as ideal and automatic as opposed to the current manual movement method.
4. **Blocked:** A robot has finished its work but cannot proceed to its next work location because there is another robot in its path.

5.2.3 What to model

Finding the right level of detail in a model is part of the art of modeling. There are two opposing concepts that are both important: accuracy and simplicity. During the development of a system, there is a time and place for both types of models. Simple models tend to be fast and robust, but often lack the nuance and the detail to provide realistic answers beyond a rough estimate. Efforts such as back of the napkin calculations or simple Excel spreadsheets fall into this category. Additional granularity will make the model more accurate, however, it is also easy to include too much detail. Too much detail will often make the model break easily when trying to handle situations that are non-standard or the model will become so cumbersome that it is slow and requires highly specialized training to use.

On the low detail side, models had already been built such as the Excel model mentioned above. They provided general estimates, but were often met with a lack of confidence because they were considered 'too simple.' These models left out some key details such as the number and types of fasteners in each bay and the interactions between the robots. Some of the details, such as robot–robot interactions would lengthen the predicted time to complete the work. Others, such as the bay processing time would reduce overall work time since the bay processing time used was on average longer than many bays took to process.

At the other end of the spectrum was the option to create a simulation of the entire system operating down to each fastener installation with full kinematic simulation. There is software to do this, such as Siemens Tecnomatix Process Simulate, that can simulate all robot movements and procedures. This software features the ability to create a digital twin of the robots and drive the robots from the PLC system. In fact, there was a parallel effort to bring this software in by a different group (albeit for a different purpose). The challenge here is that the level of detail is so fine and the complexity of the inputs so high that it would take a dedicated team of engineers to model the system and operate the model. At some point, this is a step that should be done. The value of a model of this detail is that it would solve many other problems and allow the group to debug the system, validate new robotic programming prior to deployment, and evaluate 'what if' scenarios much more easily than their current ability which relies almost entirely on a physical test or production cell. This, of course, interrupts production.

For the current project requirements, neither detail level was a great fit. The solution for this problem was to find something in the middle. The decision was to use a discrete events simulation (DES) which models a series of tasks by a set of agents as discrete time events. It was decided to model each of the robots and drill down to the level of the OLPs they execute. Each OLP was treated as a time event along with each of the larger steps in the process. The OLPs themselves would be modeled as the sum of all the expected processes called out in their code.

5.2.4 Tool choice

There were several software packages considered for this job. The primary requirement, however, was that upon completion of the project, the tool could be transferred to another engineer who could pick it up and use it with as shallow of a learning curve as possible. This meant that the solution should:

- Be capable of modeling the appropriate level of detail
- Avoid writing the tool from scratch
- Allow access to multiple users
- Use software that the company already uses and supports

The software that best met these requirements was ProModel. ProModel has several limitations, but its availability and established user base at LSM won out over other options.

5.2.5 Model concept

The model was designed to use ProModel as the simulation engine, but the inputs and outputs were setup in Excel so that the simulation would be available users who were not familiar with ProModel. The required input was a build sequence for Product A plus parameters governing the process times. In general, the concept looks like the diagram shown in Figure 5-1.

The output is automatically exported to an Excel file that automatically builds plots of the process. Figure 5-2 is a screen shot from the Excel output showing the position of the inner robot platform during the build process as well as the number of

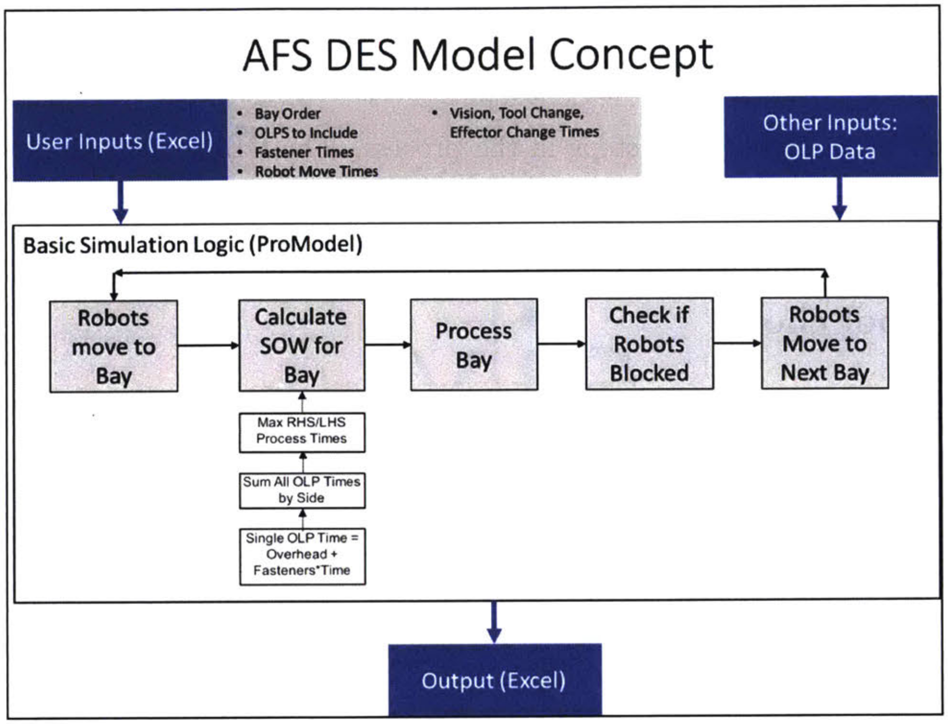
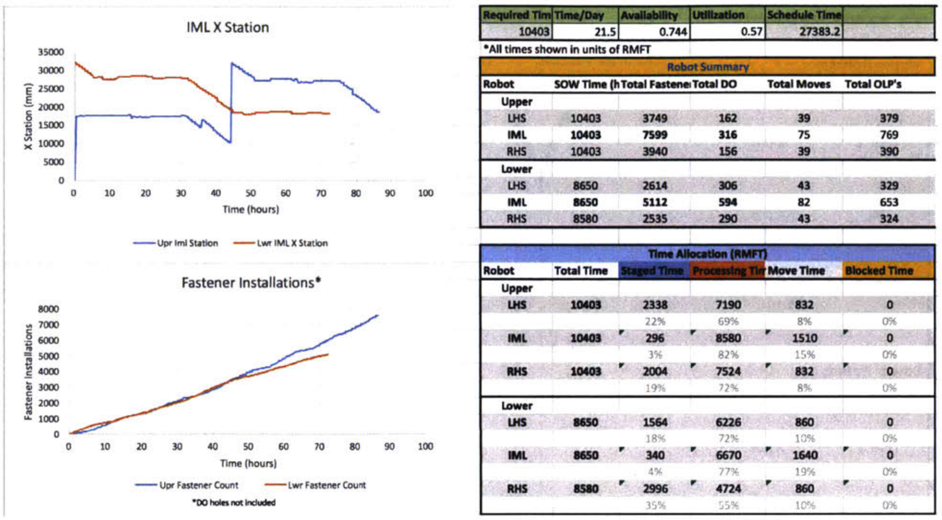


Figure 5-1 – An overview of the model with inputs, outputs, and general simulation logic

fasteners installed over time. The main output is the required time in machine hours to complete the input statement of work. Machine time is then expanded to schedule hours using historical averages for availability and utilization.

5.2.6 Model components

The components of the model included an entity for each robot. The physical locations of the robots were tracked to understand when they would block one another, but they were not modeled as location objects. Instead the locations represented the 4 states mentioned previously: Waiting, Processing, Moving, and Blocked. When a robot arrived at a particular location, logic was included to tell it how long to stay there. In the case of Waiting, it would stay in the Waiting location until the other set of robots arrived. In the case of moving, a move time parameter was used, and it would simply wait the specified time. In the case of Processing, the unit would wait for an amount of time that was calculated based on the work required by the OLPs in that physical location. In the case of Blocked, it would move to Blocked when the



Required Tim	Time/Day	Availability	Utilization	Schedule Time
10403	21.5	0.744	0.57	27383.2

*All times shown in units of RMFT

Robot Summary					
Robot	**SOW Time (h**	**Total Fastene**	**Total DO**	**Total Moves**	**Total OLP's**
Upper					
LHS	10403	3749	162	39	379
IML	10403	7599	316	75	769
RHS	10403	3940	156	39	390
Lower					
LHS	8650	2614	306	43	329
IML	8650	5112	594	82	653
RHS	8580	2535	290	43	324

Time Allocation (RMFT)					
Robot	**Total Time**	**Staged Time**	**Processing Tir**	**Move Time**	**Blocked Time**
Upper					
LHS	10403	2338	7190	832	0
		22%	69%	8%	0%
IML	10403	296	8580	1510	0
		3%	82%	15%	0%
RHS	10403	2004	7524	832	0
		19%	72%	8%	0%
Lower					
LHS	8650	1564	6226	860	0
		18%	72%	10%	0%
IML	8650	340	6670	1640	0
		4%	77%	19%	0%
RHS	8580	2996	4724	860	0
		35%	55%	10%	0%

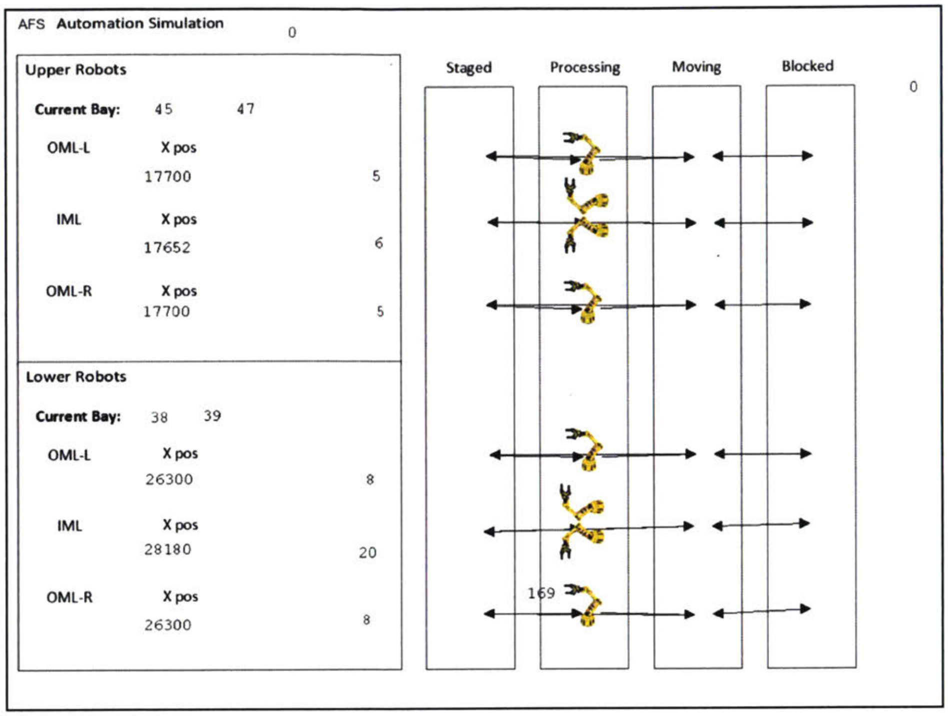
Figure 5-2 – A screenshot of the model's Excel output showing system times (in units of RMFT) as well as plots of position and fastener counts.

other set of robots was in the location it needed to go next. When the other set of robots' physical location changed, the blocked set would attempt another move. The model in ProModel with the various components is shown in Figure 5-3.

5.2.7 Data pipeline

To drive the model, a build plan is first established that includes the order in which each bay will be processed. This is a known limitation of the model because, as will be mentioned later, the ideal sequence may not be intuitive. This sequence is typically decided by the manufacturing engineering group. Each bay has a time associated with it that is the sum of the time to complete the OLPs at that location. The OLP times are the sum of the individual tasks–calibration, holes, fastener installations–that the OLP is programmed to execute. To compile this data, code was written to take the OLP programs and parse them. This step is important to accurately capture the actual work to be done in each bay.

Timing for the individual tasks was based on data from the machines on the floor. Times for the fasteners were based into groups of drill only, T1, T2, T3, and HD operations. These times could be further refined by the actual fastener part number to be installed, but that level of detail seemed too deep for the type of output needed.


AFS Automation Simulation 0

Upper Robots

Current Bay: 45 47

OML-L X pos 17700 5

IML X pos 17652 6

OML-R X pos 17700 5

Lower Robots

Current Bay: 38 39

OML-L X pos 26300 8

IML X pos 28180 20

OML-R X pos 26300 8

Figure 5-3 – A screenshot of the ProModel simulation layout.

Production data was decided upon instead of timing from validation coupons because there are some important time differences between the two. A single fastener time includes the movement time of the arm from the previous hole to the current hole. Movements during coupon testing were often not representative of movements during production; they were often much simpler and consequently faster.

For larger tasks, such as platform movements, predicted times were used instead of data taken on the floor. The reason for this is that the robots had been shown to move at the ideal speed in limited circumstances. In practice, times had been much slower and extremely variable. This was because up until that point automated platform movement was not available and all the platforms had to be moved manually. This resulted in moves that were slower because the machine speed was limited for safety and moves times that were more variable because the operators had to position the robots by lining them up with marks on the floor. Prior to full-rate production, new computerized dispatch capabilities are to come online. When that happens, the ideal moved times are expected. Because of this, ideal times are used to estimate the end-state capacity even though they are still assumed.

5.2.8 Inner robot platform constrains the process

One of the choices made in designing the model was how to handle the inner robot platoform. The inner robot is really two robots joined at the hip. When the platform is parked, both robots act independently with their outside counter-part. However, the robot platform moves as a single entity and moving is not an option until both robots have completed their work. Because of this, the inner robot platform is modeled as a single entity and in the model it tracks whether each side has completed its assigned task at a given work location. It is also important to note that although the inner robot platform is able to move to the next location regardless of where the other robots are positioned, to do any work at the new location, an outer robot is required, and if the outer robot is blocked, the inner robot platform is by extension as well. More generally, because the inner robots are mounted to one platform, the upper and lower robots essentially act as individual systems. This matters in terms of downtime as well. If any of the upper robots break, all of the upper robots are down until the issue is addressed; the same is true for the lower set.

5.3 Data sources

As with any simulation, a model is only as reliable as its inputs. One of the key aspects of building a DES model is understanding the length of time it takes a machine to execute its tasks at each stage of the process. The timings from individual steps came from several sources.

5.3.1 Rough estimates

Early on, many of the cycle times for individual tasks existed at the beginning only as rough estimates. By the end better data was available for most tasks. However, as mentioned earlier, at the completion of the project, rough estimates were still being used for robot platform movements.

5.3.2 Early test data

Early test data was available for certain tasks. The data available came from coupon testing when the robots were being qualified to install the various fastener types. Timing was done with stopwatches for the total time to finish a qualification coupon. The average time per fastener was simply the total time divided by the number of fasteners installed. This was good initial data, and better than a rough estimate, but there were some problems with it. Specifically, it assumed that installing a fastener in a coupon was representative of an installation process in a Product A which wasn't always the case. The main differences between test coupons and the production setting are:

- The hole pattern in a test coupon is much simpler, causing a non-representative mean travel time between fasteners
- The coupons are stacks of flat aluminum sheets compared to the more complex surface curvature of Product A, causing non-representative times for the machine head to normalize itself to the surface at each hole
- Tack fastener placement in the coupons have less variability in their location than tack fasteners on Product A increasing vision times
- The coupons are held in a rigid fixture that prevents the coupon from flexing under load as much as the wall of Product A
- The internal structure of Product A is not present meaning robot arm movements are simpler
- The floor of the test fixture is more rigid meaning vision may be faster due to improved platform positioning
- The lighting in the test fixture is better (influences the vision-based calibration system)

5.3.3 Production data

As machine based data became available, it became the foundation for the model. Figure 5-4 shows the cycle time for the different operations that comprise an OLP. Data is shown for both coupon and production data. The first thing to notice about the cycle time data shown in Figure 5-4 is that the only operations that are close to an RMFT value of 1.0 are T3 fasteners and drill-only operations. The T1 and T2 operations are 1.5-2.0 RMFT. This means that AFS has a fundamental problem with fastener cycle times. This will be discussed further in later sections.

Using production data also confirmed the prior expectation that data from early testing was optimistic. When comparing test data to production data, the times to install a T1 fastener or perform a drill-only operation are almost identical. Installing T3 fasteners in production is about 12% faster than qualification. Although the difference is significant, the number of T3 fasteners installed as an overall percentage of fasteners is low. However, installing a T2 fastener is 16% slower and the follow-on HD operation is 33% slower in production. In production, the first choice method for fastener installation is the on-axis method since it is a single operation. However, in cases where the on-axis method is not available, such as an installation under overhanging structure, the off-axis method is used–these types of installations also require more complex movements.

Also of note is the 40% time increase in the vision sequence that is performed at the start of every OLP. The primary reason for the difference here is the impact of variability in production versus the relatively controlled environment of a coupon. The way the vision sequence works is to move the end effector to the nominal coordinate location where it expects to find the tack fastener. If it is unable to locate the fastener immediately, it begins searching in a spiral pattern moving outward until the tack fastener is found. Thus if the fastener is not in the exact location it is expected, or more commonly, if the robot platform is poorly parked, then the variability of their relative positioning is increased and as a result the vision sequence takes longer.

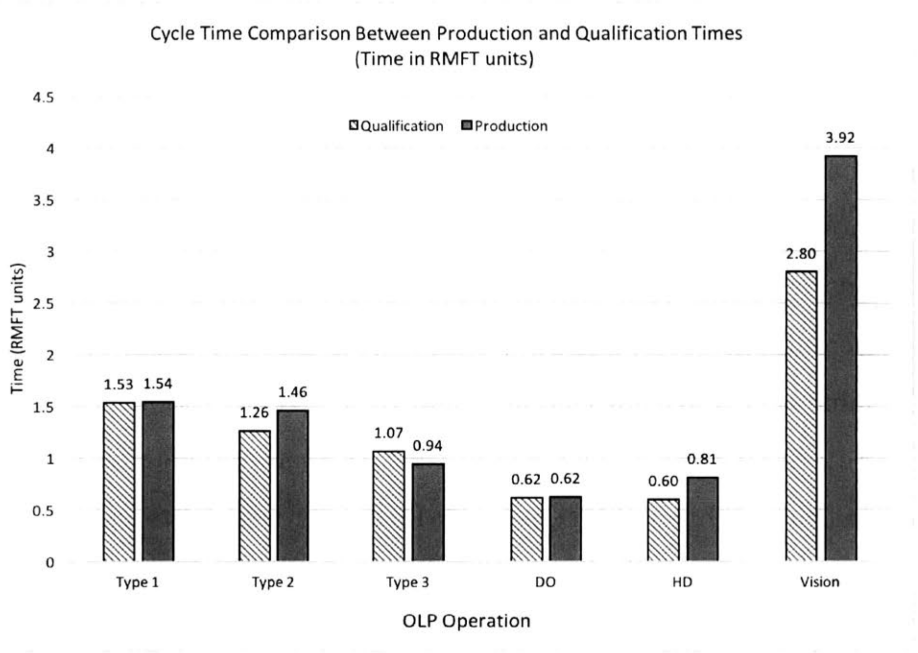


Figure 5-4 – A plot showing the average cycle time for various operations during test and actual production.

5.3.4 Other simulation tools

Another opportunity to improve the cycle time data in the DES model is to use other simulation tools to model individual parts of the build process. LSM was working toward implementation of some of these tools, but had not implemented any during the research period. The movements and fastener installations for each OLP's could be simulated as the OLP is created. The output from these tools for cycle times and bay processing times could feed the inputs to the DES model instead of using average fastener cycle times. However, these tools require a significant commitment by LSM to dedicate engineers and to have them receive the needed training.

5.4 Model validation

Validating the model was an objective of this project in order to build support for the model among the many stake holders in AFS. Early efforts at validation are shown in Table 5.1. The actuals show rough agreement with the predicted times considering the actual number of fasteners that were installed. However, there is significantly more work that can be done in this area.

Table 5.1 – Comparison of predicted fastner processing time with actuals for two similar production builds. Note that move time, setup time, and downtime are not included.

		Predicted Fastener Install Time (ProModel)			**Actual Fastener Install Time (Machine Data)**		
Line Number	**Machine**	**Time (RMFT)**	**Fasteners**	**Drill Only Count**	**Time (RFMT)**	**Fastener Count**	**Drill Only Count**
1	**RH Robot**	**3000**	**1252**	**620**	**2868**	**1097**	**793**
	LH Robot	**3000**	**1256**	**659**	**2304**	**840**	**1011**
2	**RH Robot**	**3000**	**1252**	**620**	**2820**	**1077**	**756**
	LH Robot	**3000**	**1256**	**659**	**2616**	**972**	**868**

Validation was still a challenge by the end of the research period because of limitations in the data from the production process. The biggest weakness currently is the amount and quality of the data from the machines. There are planned system updates for AFS that should enable tracking of machine movements, coupling times, time spent laser scanning, and matching specific fastener installation data to the the OLP that created it. This will allow much better system-level validation along with more granular insight to determine where the variance between actual and predicted performance originates.

5.5 Model results

The results of the model and the process itself of building the model provided several good insights into the system. The largest statement of work(SOW) that the robots are capable of doing during the automation sequence is 14,000 fasteners (the robots have access issues for the remaining fasteners). A baseline model was built with a statement of work for these 14,000 fasteners.

5.5.1 Fundamental system challenges

The baseline model represents significantly fewer fasteners than the business case called for. The RMFT value of 1 is the amount of time allotted, on average, to each of the 25,000 fasteners during a build according to the business case. One of the first things that is apparent from the time data collected and shown in Figure 5-4 is that the T1 and T2 operations take significantly more time than the allotted 1 RMFT. T3 fasteners take about 1 RMFT. Drill-only operations take less than 1 RMFT, however, these don't contribute much to the the business case from a cost, quality, or safety perspective because they still require an assembly worker to perform the fastening. As shown earlier in Figure 3-3, the bulk of the fasteners are T1 and T2. Simply weighting the averages by their distribution, the average installation time across Product A is 1.56 [1].

In other words, just looking at the fastening times and using the original back-of-the-napkin (equation 5.1) calculation, the business case does not hold. The time just to install the fasteners would take 35% more time than the production schedule allots (without even considering vision, downtime, setup time, etc). Looking at the feasible SOW case of 14,000 fasteners, as a first pass using equation 5.2, the time to complete just the fastening work is 76% of the time allotted before accounting for setup time, downtime, maintenance, etc.

$$ProductA_{Time} = \frac{X_{fasteners}}{N_{robots}} * Y_{sec/fastener} \tag{5.2}$$

Drilling down a bit further into the data, it is useful to look at what the robots are doing while they are executing OLPs and installing fasteners. The data shown in 5-5 shows the time spent by the outer robot across all OLPs it processed for a particular Product A build. It should be noted that the RMFT includes hole-to-hole movement of the robot arms, but it does not include vision. The 'Other' category is time that the outer robot spent waiting for the inner robot either to perform an arm movement or to finish a vision sequence. The chart does not include platform movement or any of the other system setup tasks.

---

[1] This assumes that all No-process fasteners are tack fasteners which will eventually be drilled out and have a T1 installed with the on-axis tool

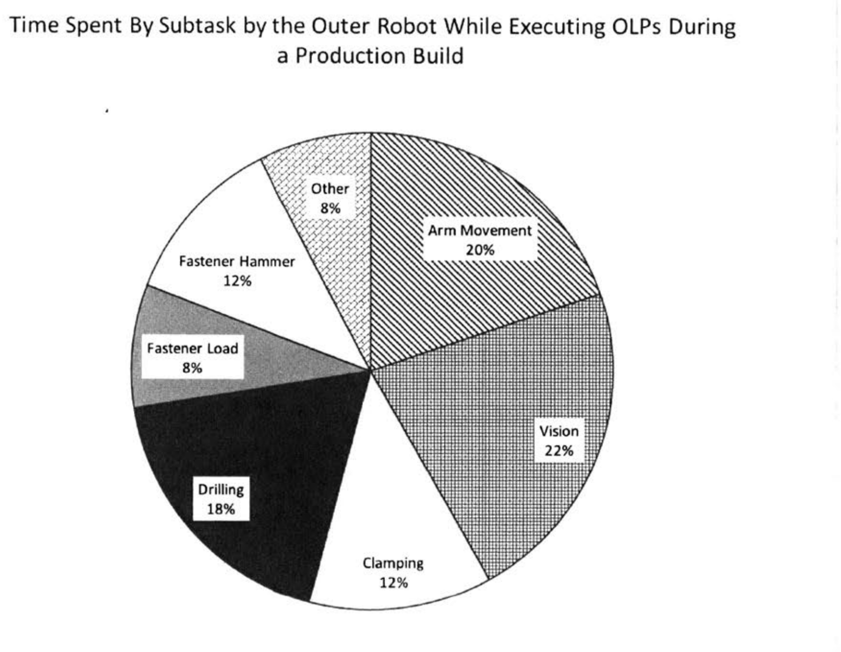


Figure 5-5 – A plot of how the time was spent by the outer robots while processing OLPs. Data taken from a production build.

5.5.2 General insights

The model offers some general insights into AFS concerning its capacity and how time is being spent by the machines. These general insights are useful because they put some general bounds around the problem.

1. **Fundamental Capacity**The machine time required (downtime not included) to complete the installation of 14,000 currently planned (and reachable) fasteners was 10440 RMFT. The allotted schedule time was 7,200 RMFT. Looking more closely, the lower robots' machine time is 8,650 RMFT while the uppers 10,440. This means that there is a fundamental capacity issue. Neither robot will finish its work in the allotted time. Looking more closely, the upper robots require 8,540 RMFT machine time just for OLP execution (Processing time). Even if robot platform is not considered, there is not enough time to complete the upper statement of work. Fundamental changes will need to be considered to the way OLPs are executed.

2. **Staged Bucket** One of the primary insights that will be mentioned in the next section on the critical path is that the critical path is determined by the inner robot platform and the path itself moves between the right hand side and left hand side. This is apparent when looking at the Staged time of the outer robots. The upper left hand robot spends 13% of the total build time waiting for the right hand side to finish its work and the right hand side robot spends 10% of its time waiting for the upper left hand side to finish. Because the upper robots operate as a unit, at a given bay, if one robot finishes before the other, it must simply wait until the other side finishes. This means that for 23% of the total build time, only one of the robots is operating. The rest of the Staged time spent by the outer robots is spent waiting for the inner robot to move.

3. **Moving Bucket**One of the greatest technical issues facing the group is the positioning of the robot platforms, and particularly the inner robot platform. Best-case predictions were that a single (ideal automated) move would take a time of 20 RMFT and that the move time is equal between inner and outer platforms. Assuming the move time is the same for both inner and outer robots, the real focus to reduce movement and setup times needs to be on the inner robot platform movement because it moves significantly more. The inner robot platform is required to move every work bay, but the outer platforms are only required to move at least once every four work bays. The inner robot spends almost twice as much time moving as the outer robots. Move time is currently 15% of total required machine–this puts a floor on the improvement that can be had.

4. **Inefficient OLPs** There were 28 OLPs in the system that each processed 5 fasteners or less. This is a problem because, the average fastener install rate for these bays get even worse because each OLP has overhead (vision, movement, etc) that takes up to an additional 4-6 RMFT. They also all require an additional platform move. Some of these OLPs should be considered for removal from the statement of work.

5. **System Slack**The time to complete the statement of work on the upper path was longer than the lowers by 1800 RMFT which means that there is slack in the lower robots' capacity. There was a task force within the AFS group that was looking at ways to add fasteners back into the work statement that were currently unavailable because of technical reasons (robot reach, end effector clearance, Product A internal structure problems, etc). The task force should prioritize adding fasteners to the lower statement of work because it won"t add to the overall build time of Product A.

5.5.3 The critical path

One of the primary outcomes of the model is the ability to identify the critical path and to reinforce the fact that the improvements that will yield the greatest benefit are the ones that reduce the time to complete the critical path. The critical path depends on the work the robots are asked to do. This model allows the critical path to be identified for a given work statement.

The Product A is essentially divided into four quadrants. The upper-right quadrant has the greatest statement of work-this is because the upper-right contains some extra structural features that require additional fasteners. For a long time, the upper-right was assumed to be the critical path. Something that became apparent though is that, because the inner robots are tied together, the critical path, while dominated by the upper-right, at times is constrained by the upper-left. The critical path jumps from side to side depending on which robot has the larger statement of work for a particular location. Depending on the work to be done and the schedule that is used, the constraining resources can also jump from the top to the bottom if the top robots are blocked in their movement by the bottom set.

For the 14,000 fastener baseline, the critical path contained the following items:

- 75 Inner Robot Platform Moves
- 390 OLPs
  - 234 SEE OLPs

  - 156 HDEE OLPs

- 3940 Fasteners

- 127 tool changes

- 4 power decoupling/coupling operations

This information is useful because it helps identify potential savings in the system. For instance, if an improvement enabled fastener processes to happen on average 1 second faster, it would roughly result in a system improvement of 3940 seconds, or just better than an hour savings. Similarly, if the vision sequence, which is performed at the start of every OLP, were reduced by 10 seconds each, it would result in just over an hour savings. These tasks are directly in the critical path and this knowlege can help triage system improvements to AFS by their potential impact on the critical path.

5.5.4 System sensitivity

Another reason this model is useful is to identify system sensitivities. As an example, one of the primary factors the model considers is robot-robot interactions. Because the robots can't be in the same place at the same time, if one robot is in a work zone, the other robot must stay out of that zone plus maintain a buffer. It is very easy for the robots to be scheduled to work in a location where another robot is already working and consequently force it to wait. This is a temporal-spatial problem that needs to be considered carefully to have an optimal schedule. Figure 5-6 shows the overall production time for a set of simulations where the average move time varies. If a simple linear estimate is used to predict runtime, at 60 RMFT the prediction if off by just a few percent, but then the error increases quickly.

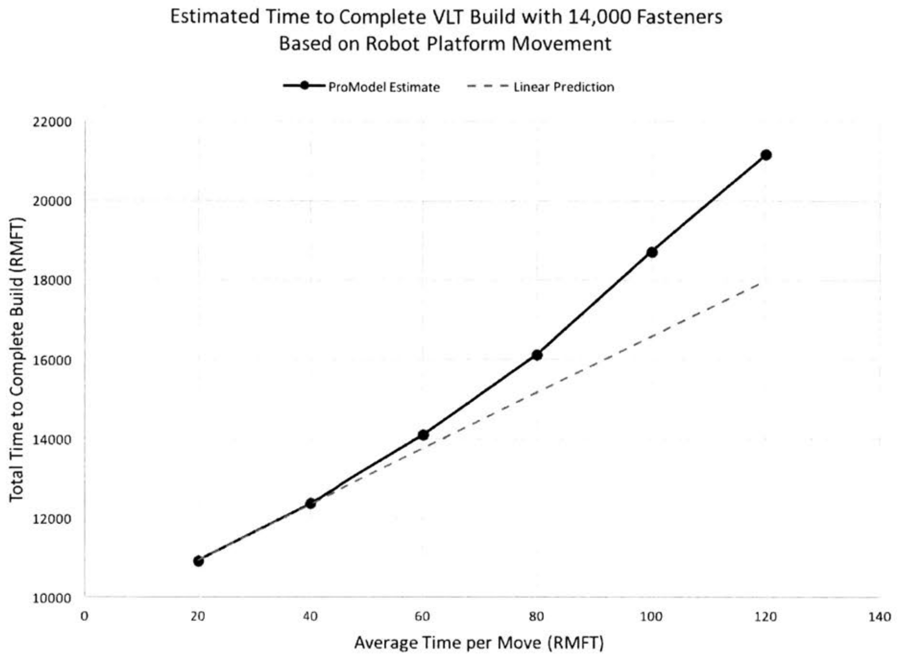


Figure 5-6 – A plot of the estimated build time (for a Product A with 14,000 fasteners) as average robot platform movement time increases.

5.6 Model next steps

There are two main things that would improve the usefulness of the model. The first is improving the inputs to the system. Ideally as more features of the system come online, more accurate data will be available. In particular when the MCP is controlling the system, data logging should be much more accurate and reported for all major and minor tasks.

The second feature that would make the tool more useful is the ability of the tool to optimize the work schedule based on the work to be performed for a specific build. Currently, the build sequence has to be determined ahead of time. Because of the complex nature of the system, it would be even better if the user only had to choose the work for the robots to do and then an efficient algorithm could be written to determine the best sequence in which to perform the work.

5.7 Summary

The model that was built strives to meet the needs of the AFS group and to provide the right level of detail to enable decision making and gain insight into the performance of the system. The model shows, consistent with several of the previous models, that AFS—as it is currently being developed—is incapable of meeting its flowtime requirement. Improvements will need to be made to the system in order for it to meet its required flowtime. The general insights discussed above provide some ideas of where to look for flow improvements.

Identifying the critical path is an important outcome of the model because it allows decisions to be made based on whether or not they will actually increase system throughput. Only reductions to the critical path result in reduced flowtime. In the next section the recommendations presented all have an impact on reducing processes from the critical path.

Another reason the model is important is because of insights that arose during the process of gathering information and building the model. As was discovered in the literature review, automation projects that were successful required a systems-level approach to their development. The model building process was truly a cross-functional, systems-level exercise that required input from across the AFS department. It was the first time that anyone at LSM had pulled together all of the information and data to understand the process from a high-level build plan all the way down to what happened when each fastener was installed. It also was the impetus to finally make data available from the machines to the group. Prior to this work there was no machine-based cycle-time information for any of the processes.

All of this information lead to many possibilities to improve AFS. In the next chapter a few selected recommendations that came out of the model and the impact they will have on the system will be discussed. Following that will be a discussion on lessons-learned for LSM as the company looks toward future automation projects.

Chapter 6

Recommendations to improve the system

This chapter presents several recommendations that can improve the performance of AFS.

6.1 Summary of opportunities

When considering ways to improve the system, there are a few points to take into account. Obviously with enough money, time, and will-power, anything in the system can be changed. However, there are things about the production process that are easier to change than others. In general, changes are one of two types, physical and procedural. Physical changes refer to changing either Product A itself or to modification of the equipment that is used for production (the robots, platforms, sensors, etc.) Currently there is no appetite for major structural changes to the equipment or Product A. Concepts such as spilting the internal robot platform or adding extra actuators to the inner robots to do fine positioning do not get organizational traction. On the other hand, procedural changes refer to either the programming, operational parameters, or sequence used to operate the system. For this particular study, all of the changes proposed are procedural in nature and dont require a modification to the equipment. Both types of changes have cost and schedule implications.

Regarding procedural changes, they can be further grouped as either open or requirement-controlled. A good example of this is the fastener installation processes. Referring back to Figure 5-5 for fastener installation timing, the parameters for several of the events during installation—drilling, fastener loading, hammering, and clamping—are tightly controlled in AFS program requirement documentation. For instance, the drill rpm and pecking speed are specified and cannot be changed without reason and significant data to back it up. When parameter changes are made to the requirements, all of the machines need to be re-qualified. These are difficult changes to make. On the other hand, changing something open, such as the arm-movement speed, only requires agreement from the various engineering groups and adjustment of the parameters in the system. For Product A build represented by the pie chart in Figure 5-5, 50% of the time spent 'processing' Product A was open for change including vision, arm-movement, and anything in the 'other' category. The recommendations that follow include both open and controlled changes.

6.2 Recommendation 1: Reduce tool changes

6.2.1 Problem background

One of the events that was included in the model was drill bit changes. Overall there are 10 unique drill bits used to complete a Product A build. Figure 6-1 shows a more detailed diagram of an outer robot with its drill bit configuration. The outer robot end effector has an onboard capacity for six tools, of which two are allowed to be drill bits. During production, if the end effector does not have the required drill bit on board, it must retrieve the new drill bit from the tool carousel. Changing a drill bit takes 2 RMFT. For a complete build, the model shows a single robot would perform over 120 of tool changes equating to about 1% of the total Product A build time. These tools changes were sub-optimal and represented an opportunity for savings.

The tool changes occur whenever the end effector doesnt have the needed drill bit onboard. The current system has a sub-optimal tool change policy. In an effort to

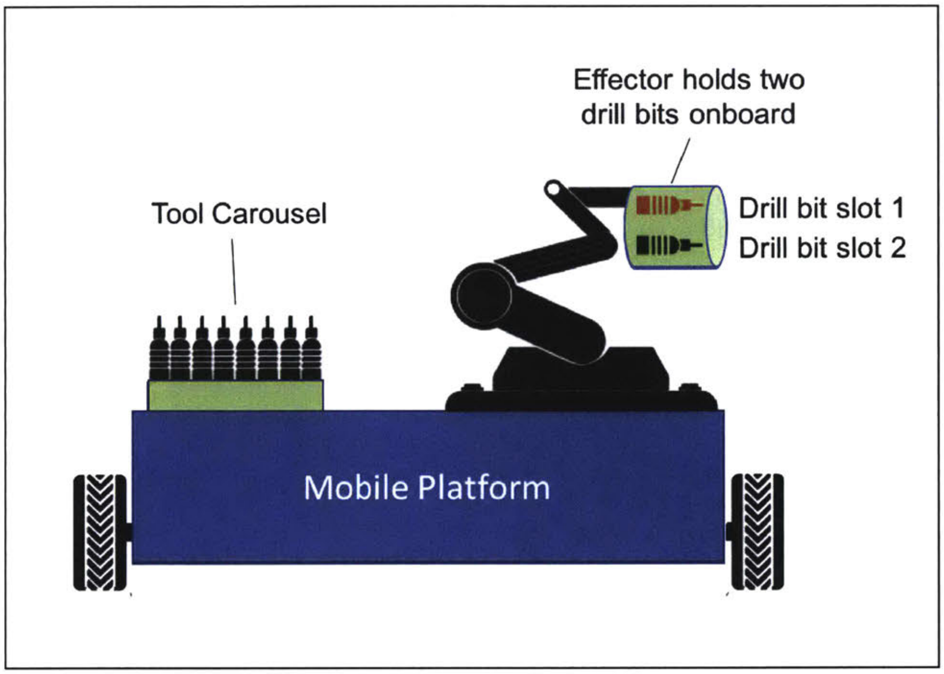


Figure 6-1 – A more detailed view of the tool change system is shown. The end effector can hold tool drill bits while all other drill bits are stored on a tool carousel on the robot platform. AFS currently has the drill bit in slot 1 hard-coded (fixed) while the drill bit in slot 2 is free to be changed with the tools in the carousel. This is sub-optimal and an opportunity for improvement.

get the AFS up and running quickly, the programmers in charge of the tool change routine made the decision that the easiest way to deal with tool slot assignments was to have the most frequently used drill bit continuously loaded in tool slot one on the end effector. Tool slot two is used for all other drill-bits. In practice, there are many sequences in the build when only drill bits 2-10 are required and so the robot performs tool changes effectively using only a single tool-holder. If both tool slots are available, the number of tool changes in these sequences is greatly reduced.

In determining the fastener order of a Product A build, the tool-change constraint is far subordinate to other constraints such as movement and structural interferences. In general, the ordering is determined first by robot platform position and then it is ordered to ensure that the machine head has enough clearance to operate. The local ordering of the fasteners needs to be arranged properly so that the fasteners can be installed with respect to the structure of the airplane. Because of this, the fastener order is determined first and then the tool-change sequence can be set.

6.2.2 Anticipated benefit

By using two tool holders across all for all 10 bits, on the baseline Product A build sequence with 14,000 fasteners, the new tool change sequence yields an overall reduction from 123 tool changes to 27 for the worst robot, saving around 96 RMFT. In the ideal workflow, the fastener ordering would be made, and then the tool change optimization would run to ensure that the tools are changed out at the correct time and put in the ideal slots.

6.2.3 Optimization strategy

In setting the optimal tool change policy, the problem becomes more complex as the number of tools or tool slots increases. For instance, with 1 tool slot, there is no optimization to be done. Once the fastener order is fixed, the optimal drill bit loading policy is simply the order of the fasteners. However, if the number of slots is increased from one to two, when a new tool is required, the machine must decide in which slot to load the new drill bit. If one of the two loaded drill bits is going to be used again immediately following the current hole, it would be ideal to change out the tool that would not be used. If the number of tool slots is further increased to 3, the problem expands even further.

For two tools slots, the logic of the optimal policy is fairly straight-forward, and could still be worked out by hand. However, because it is to be implemented as part of an automated process, it needs to be encoded in an algorithm. The other advantage is that if the number of available tool slots is increased, the algorithm easily adapts. A mixed integer program is encoded below. For this project, it was solved using Julia, but it could be programmed in any number of solvers. A particular robot has N holes to drill, using M different drill bits which can be loaded into one of L Tool holders. Additionally, as an input, at each hole i, the drill to be used d, must be specified. The problem formulation is encoded below:

$$i = \text{Hole index } \{1..\text{N}\}$$

$$j = \text{Drill bit index } \{1..\text{M}\}$$

$$k = \text{Effector tool holder index } \{1..\text{L}\}$$

$$d = \text{Required drill bit to drill each hole } \{1..\text{M}\}$$

Decision Variables:

$$x_{ijk} = \{0,1\}$$

$$y_i = \{0,1\}$$

Objective:

$$minimize \sum_{i=1}^{N} y_i \tag{6.1}$$

Subject To:

$$\sum_{k=1}^{L} x_{id_ik} = 1 \quad \forall \quad i \tag{6.2}$$

$$\sum_{j=1}^{M} x_{ijk} = 1 \quad \forall \quad i, k \tag{6.3}$$

$$x_{ijk} - x_{(i-1)jk} \leq y_i \quad \forall \quad i, j, k \tag{6.4}$$

$$-x_{ijk} + x_{(i-1)jk} \leq y_i \quad \forall \quad i, j, k \tag{6.5}$$

In this case, the goal is to minimize the number of allowed tool changes y. Constraints 6.2 and 6.3 ensure that the tool requirement for a particular hole is satisfied and that only one tool is assigned per holder. Equations 6.4 and 6.5 force the number of tool changes to respect the decision to allow tool changes or not for a specific hole.

The end effector has two slots for drill bits on board. The current system configuration has one of the drill bit slots permanently assigned to one of the 10 drill bits. The other drill bits are cycled in and out of the remaining drill bit slot resulting in 123 tool changes that occur in the critical path. By removing the hard assignment of the 1st drill bit to the 1st slot and optimizing the tool changes, the number of tool changes in the critical path can be reduced to 27.

6.2.4 Path to implementation

The largest barrier to implementation is the additional programming that will be required to make this step automatic. The current NC programming files that are written specify a drilling/insertion macro to be executed for each hole. The macros are a collection of low-level robotic commands to be executed, including the information about which tool slot should be used to hold the drill bit. The macros are simply called, and do not accept input arguments (such as a desired tool slot). Using the current system, in order to implement alternative tool loadings, other than those currently programmed, a duplicate set of macros will need to be made for each tool holder. For two available tool slots, two complete sets of macros will be required. The addition of a third available tool slot would require a third set of macros to be added. The macros are maintained by one of the subcontractors and dont have any interface other than execution.

Making this implementation slightly more complex is the matter that currently the macros are specified when writing the NC program. However, when the NC programs are written, the tool loading sequence is unknown-it isnt determined until after all NC programs are written. This means that the tool loading sequence will have to be decided **after** the OLPs (which contain the tool-slot specific macros) are written. If the suggested implementation is made, it would proceed as follows:

1. Create robot paths and create OLPs
2. Write out OLP files
3. Parse all OLPs to determine final hole order from start to finish
4. Run optimization to determine tool loading sequence
5. Automatically modify OLP files to reflect tool-holder specific macro
6. Send modified OLPs to robot file server for execution

The other option for implementation is for the subcontractor to modify the structure of the macro files so that they are able to accept arguments. This would allow the creation of a single set of macros with the ability to specify the tool holder to use when the macro is called.

6.3 Recommendation 2: Vision sequence reduction

6.3.1 Problem Background

One of the steps in the drilling and fastening process is the localization of the robot using vision. At the start of every OLP the robot executes a vision sequence. The vision sequence establishes an explicit relationship between the position of the robot and the fastener locations to be installed. Currently vision tasks make up almost 25% of the total time spent processing Product A. In every location where an OLP will be executed, there are two special fasteners installed by mechanics called tack fasteners. The fasteners are located by the robot using a camera mounted on the end effectors and a machine vision algorithm. Once the fasteners are located, the hole pattern is adjusted from its nominal position and orientation to match actual locations of the installed tack fasteners and account for any minor misalignment.

In general, this process is required because:

1. There is no absolute reference point
2. The robots are on a mobile platform that is not physically constrained relative to Product A latterally or longitudinally
3. The as-built condition of Product A varies from the nominal design and exact localization needs to be used to ensure the holes are inserted properly

The current inefficiency with this process is that it is repeated unnecessarily. When the robots are processing a work bay, they first work the bay with the SEE end effector. This drills all holes and insert alls fasteners for that bay. Then the end effector is changed to the HDEE and all T2 fasteners are processed a second time using the HDEE. Changing the end effector is an automatic process where the inner robot arm puts the old end effector in a specially designed rack, then moves to the next end effector and couples to it. At the beginning of each OLP before any fastening operations using either end effector, the vision sequence is repeated.

The vision sequence during the OLP for the HDEE was included as standard practice early on to simplify development but it is redundant with the following assumptions:

1. The tool center points exist at known offset positions
2. When an end effector is loaded, the tolerances of the coupling system are very small (similar industry-standard couplers are documented to 0.0006 mm X,Y,Z positional repeatability[23].)
3. The robot is repeatable to a very high relative tolerance (similar type products are documented to 0.06-0.07 mm positional repeatability[24][25])
4. The structure/surfaces of Product A is not altered/deformed during the primary drilling and fastening operations
5. The platform that the inner robots are attached to does not shift or move during the primary drilling and fastening operations.

If any of these assumptions are not true, it would signal much larger flaws in the design of the system with respect to its accuracy and repeatability. However, under these assumptions, the HDEE can be explicitly located relative to the work surface using the locating information from the first vision sequence and a simple tool offset.

6.3.2 Anticipated Benefit

The primary benefit from implementing this change will be a time savings of 300–600 RMFT. There are almost 150 secondary vision sequences in the critical path.

The time required to perform a vision sequence varies depending on how accurately the inner robot platform is positioned and how closely Product A"s structure is to nominal. The greater the variation in either of these, the greater the vision sequence time. At the low end, it is 2 RMFT and at the high end it is 4 RMFT. At an average of 3 RMFT, the overall savings is very large.

There is a second important reason to remove the secondary vision sequence. The original plan for AFS calls for the tack fasteners to be drilled out and have a production fastener inserted automatically by the robots. Under the current implementation, this is not possible. Currently over 1500 tack fasteners have to be removed and their production fasteners installed by hand during the post-automation sequence. If it is to be automated, the tack fasteners will need to be removed and replaced while the SEE end effector is attached. These gives two options:

1. OLP #1 with the SEE end effector drills and replaces the tack fasteners. The end effector swap is made to the HDEE. OLP #2 is executed with no vision sequence because there are no longer any tack fasteners to reference.
2. OLP #1 with SEE end effector does regular work. Swap end effectors. OLP #2 is executed with HDEE. Swap back to SEE end effector. Execute third OLP with vision sequence and replace tack fasteners.

6.3.3 Path to implementation

The first barrier to implementation is a programming issue with the vision process. Currently each OLP that is executed is a stand-alone program and there is no interface between programs. There is no way to store or pass the results of a vision sequence. The changes required would include storing the location information from the primary vision sequence and then passing it as an input to the HD OLP vision process with a predetermined tool offset. These programs are developed and maintained by RSI.

A reasonable implementation step for this change would be to leave the HD vision sequence in place, but instead of moving the end effector to the nominal location of the tack fasteners, moving it to the known locations from the SEE vision sequence. This should decrease the time for the execution of the HD vision sequence because it should be able to locate the tack fasteners immediately instead of having to compensate for the variation both times. Then, over time, as confidence increases, the entire vision sequence for the HD OLPs can be removed.

6.4 Recommendation 3: Ahead/behind OLP processing

6.4.1 Problem Background

The biggest bottleneck of the entire AFS system is the movement of the inner robots between bays. Figure 6-2 shows the relative positions of the inner and outer robots when processing bays. A system requirement states that the inner-robots may process only the bay immediately in front of the robot platform as shown in Figure 6-2. This requirement was defined hastily in an effort to get the machines working sooner. The reach of the robots is greater than a single bay, however, the inner robots must support extreme loads during the bucking process and as the arms are extended further to reach more distant fasteners, the moments exerted on the arm become much greater. The platform that the robots are mounted to is used to counter these forces and moments. During early testing, it was observed that fastener quality suffered when the arm was over-extended because of a loss of rigidity of the arm and the base. There was not enough time however to explore the issue further and so the reach of the robots was simply limited conservatively to the single bay immediately in front of the platform.

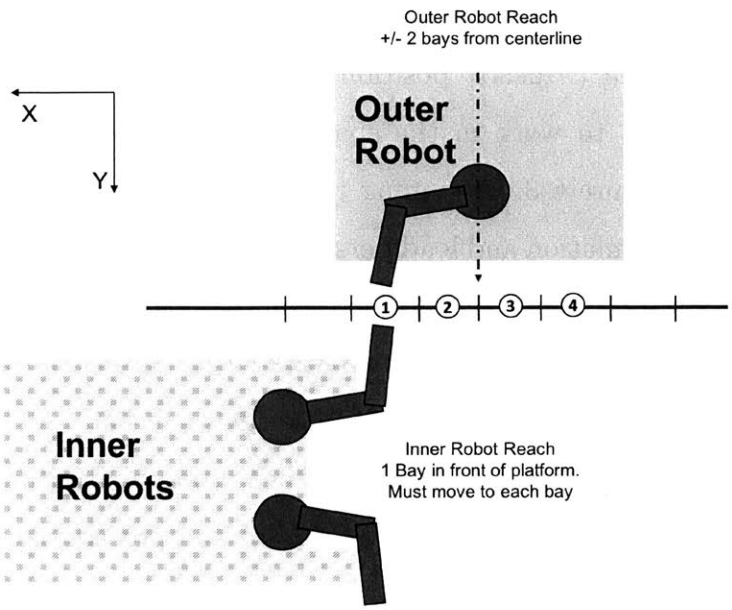


Figure 6-2 – Current bay processing is limited to 1 bay in front of the inner robot platform. Bays are processed in order and the inner platform moves after each bay. The outer robot is only required to move every four bays.

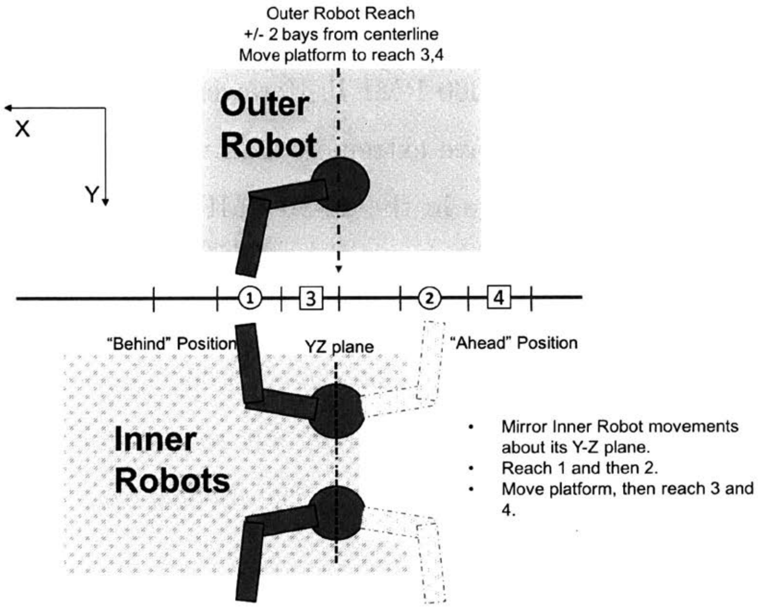


Figure 6-3 – Proposed bay processing with the inner robot work envelope mirrored about the Y-Z plane. Processing occurs in bay 1 and 2, then inner robot and outer robot move and process bay 3 and 4.

Relaxation of this requirement would allow inner robot platform movements to be reduced directly impacting the critical path. Looking to expand the working envelope of the inner robots is difficult, but one option is to allow the robot to not only work one bay in front of itself ("ahead" position), but to mirror the motions about the Y-Z plane and allow it to work in the space behind its base ("behind" position) as well as shown in Figure 6-3. Assuming that the current joint motion ranges are acceptable from an articulation and load perspective, the movements should also work if mirrored about the Y-Z plane. This would mean that every time the robot platform is positioned, the inner robots can reach two work locations instead of one–effectively cutting the number of inner robot platform movements in half. As shown in Figure 6-3 the inner and outer robots would park and then process bays 1 and 2. Then inner and outer robots would both move one bay forward and process bays 3 and 4.

6.4.2 Anticipated Benefit

Using this strategy would reduce the number of inner robot platform moves on the critical path by 18. The best-case predicted scenario is that the inner robot platforms will take 20 RMFT to move from location to location when fully automated. This translates to an overall savings of 360 RMFT. However, the current movements are not only longer on average, they have extremely high variability. Currently average move times for the inner robots are in the 40-80 RMFT range. The assumption is that as automated movement is implemented and as other issues are addressed that impact the positioning accuracy of the platforms, the move time will drop to the best-case time of 20 RMFT. However, if those gains do not materialize and the move times remain high, the time savings from taking out 18 platform moves will be even greater. I.E. the longer moves take, the greater the savings from removing one.

6.4.3 Path to implementation

There are two main barriers to implementation. The first is that the robots have not been used in these positions before. The main engineering groups are in agreement that it should be feasible, but it has not been tested. A more in depth simulation will need to be done to ensure that there are no other constraints that would prevent these types of movements. Assuming that the inner robots are found to be fully capable of these movements, all of the robot platform positions will need to be re-mapped and new OLPs will be required for each of the bays. This change will require the outer robot platform to move more often, but the extra moves will not negatively impact the system since outer platform moves are less frequent than inner platform moves. The other barrier that will need to be addressed is the build requirements. Currently the build requirements require the order of the bay processing to occur in order. Although no issues are anticipated, this new method will require the bays to be processed out order. To make these changes, LSM will need to do proper testing, validation, and qualification to change the current build requirements that state the bays must be processed in order.

6.5 Implementation analysis

Overall, the benefits from all the recommendations represent up to an 8% reduction in machine processing time as shown in Figure 6-4. Note that the times shown in 6-4 do not provide allowances for downtime or a utilization factor. Implementation of these recommendations will require a significant amount of effort to implement. However, none should require physical modification of the equipment and should be within the current capabilities of the system. The challenge that remains however, is that the schedule allocation is 7200 RMFT, and even combining these recommendations, the improvements do not allow production within the schedule time, especially when factoring in items such as equipment failure and maintenance.

6.6 Summary

This chapter presented three possible options for improving the performance of AFS. Each of the improvements will require a cross-functional team effort to implement and some will be more challenging than others. Together, these improvements yield an 8% benefit, however, the total gain not enough for AFS to meet its performance

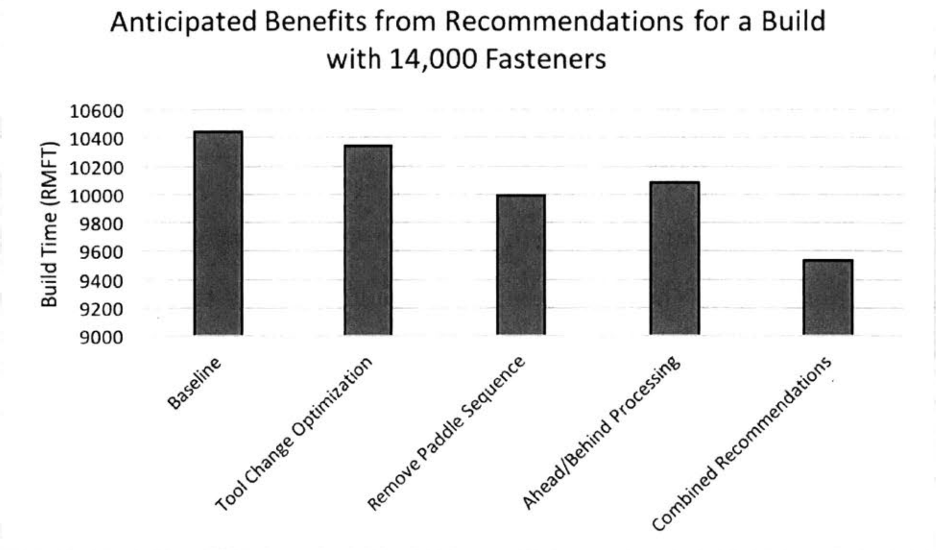


Figure 6-4 – A plot of the expected machine time (in RMFT) to produce a Product A with various recommendations applied. Schedule allocation is 7200 RMFT. Note that machine time shown does not include allowance for downtime or utilization.

goals. Additional improvements will be required.

In the next section I will discuss other areas AFS could consider for improvement as well as how the lessons-learned during this process can be applied to future projects.

Chapter 7

Summary of results

7.1 AFS needs modeling

AFS was never production hardened prior to being implemented in the factory. Instead, the system is still in development, while it is being used in a production environment. Because of this the system development is slow and is often sidelined to production demands. A proper model of the system is critical to making good decisions and driving development forward. Without a reliable means of testing, simulation becomes even more valuable. Additionally, without some sort of model there is no means to predict the final system capacity or to evaluate proposed enablers to the system.

7.2 Readily available data is necessary to make decisions

A model is only as good as its inputs. The model will continue to be subject to questioning and skepticism as long as the data it is based on is shaky. There is no reason for a lack of data in a functioning automated factory. One of the real benefits of automation is the ability to track everything. Currently one of the biggest holes in the data is on move times.

7.3 System capacity summary

First, this thesis shows that AFS has practical limits to its capacity. In fact, without major changes, AFS will not be able to meet its original throughput requirement. It won't even be able to meet its reduced capacity of 14,000 fasteners. The estimated machine time requirements (not including downtime), is 10440 RMFT and the allotted schedule time is 7200 RMFT. If the three recommendations in the previous chapter are implemented, the improved system machine time is expected to be just over 9700 RMFT, so there is still a significant gap that will needs to be addressed.

The time that is encompassed by the OLPs (vision, hole-to-hole movement, is 8580 RMFT. This means that major effort needs go into reducing the OLP time requirements. The vision sequence reduction mentioned in the recommendations will help by reducing time by up to 450 RMFT, but that still leaves a 930 RMFT gap that needs to be addressed either with robot arm movement or the fastening process itself. As a hypothetical scenario, if we were able to be aggressive (and very optimistic) with the arm speed and assume it could be reduced by 50%, that would mean arm movement which is currently 20% of total OLP time would be reduced to 10% which would result in a savings of 858 RMFT. Coupled with the vision reduction, it would still not be enough to get within the time envelope. AFS needs to reduce the fastening process (drilling, hammering, fastener insertion) times themselves to meet their targets.

7.4 Suggested actions for AFS model

Although the model has several useful features, it does not provide an optimal schedule. As the performance of the system is improved, proper scheduling will become more challenging but also more important. A good scheduling algorithm should be developed or the data and model should be combined with the academic model that is being developed by the R&D department. It is likely that the combination of data collected during this project combined with the interface and optimization tools built by the academic team would be extremely powerful. A schedule optimizer would optimize the sequence of bay processing. This will be useful to minimize time that the robots spend blocking one another. Re-planning mid-production could also be useful if a machine goes down. Re-planning scenarios could include finding a schedule that maximizes the number of fasteners installed given that the schedule time remaining is less than the required time to accomplish all work because of an unplanned stoppage.

7.5 Application to future projects at LSM

LSM has been very ambitious in undertaking AFS. There have been many challenges along the way. Hopefully as LSM looks to implement future automation projects, some of the lessons learned from this project can be applied to improve the process. Below are some lessons-learned that should be considered for future projects.

7.5.1 Modeling is critical

Regarding simulation in an article titled "The Purpose of Mathematical Programming is Insight, Not Numbers", operations expert Arthur Geoffrion said:

> The ostensible purpose of a mathematical programming model is to optimize a stipulated objective function subject to stipulated constraints. But its true purpose, at least in strategic applications as every experienced practitioner should know, is to help develop insights into system behavior which in turn can be used to guide the develop- ment of effective plans and decisions. [26]

It is obvious from this work that modeling provides and extremely useful tool to gain insight into a system. For a system as complex as AFS it is critical. All simulations will output numbers; accuracy will vary based on detail and assumptions. Simulation serves as a means for system planning and analysis, but it also serves as a method to force a dialog between different groups in the organization. The process of modeling often brings questions to the surface that otherwise remain buried. By the nature of the process, the more detailed the model, the more detailed the questions are that will be asked. In a situation like AFS where the decision was made implement the system in a production environment without many key features, a fully detailed kinematic model would have been an excellent way to shortcut many of the problems that occurred during implementation.

7.5.2 Early assumptions should be revisited

Every system that has been developed is based on an initial set of assumptions and simplifications. However, as the development progresses, initial assumptions and simplifications need to be revisited and new assumptions made to allow the design to progress. In this case initial assumptions used to build the business case should have been adjusted as the realities of the data became available.

Speaking on traps that companies regularly fall into when implementing robotics, in his handbook Implementation of Robotic Systems, Wilson warns of assumptions:

> The customer has built a justification for the project based on various assumptions. If these are found to be unrealistic, which may be due to the inexperience of the customer, the team should identify the problem and rebuild the business case based on more valid assumptions. Otherwise, the customer is disappointed at the end of the project, which is not beneficial for the vendor or the customer [27].

It is clear from the data that the initial assumptions regarding the cycle times for fastener insertions were extremely optimistic. As the model is refined using more accurate values, it becomes clear that the capacity of the expected system doesn't meet the required rates. This is clear even before reintroducing assumptions that were initially ignored such as downtime and maintenance.

Further, in the rapid development of a system, it is easy to use sub-optimal designs that are easier to implement. The danger of this, however, is that these simplifications get baked into the system and measures that are intended to be temporary become permanent features.

7.5.3 Data from day one

Another major challenge for the AFS program is the lack of good data. Data collection, especially for an automation project, should begin immediately and should be built into the design of the system. For AFS, data collection was planned, but the implementation was much further down the list. It should have been one of the top priorities. Without data, the AFS team was didn't have as much information as they needed when making many their decisions.

7.5.4 Production is not a development environment

One of the biggest challenges that AFS faces is that it is being implemented in a production setting as an incomplete system. AFS had no means to test the system other than in a production setting. While it is likely this was viewed as a means to force the development to happen faster, the reality is that if development is the top priority, trying to do it in a production setting continuously casuses it to be in conflict with other priorities. Day-to-day actions reveal true priorities and development needs were consistently second class to the production schedule. Critical support personel from LSM and RSI often spent time fighting fires instead of finishing development of the system because of production demands. Overall using an only partially implemented new technology caused massive problems in the factory and at the same time slowed development of the system.

One of the principles of lean is not to use unproven technology in the critical path[28]. For future systems, LSM should consider bringing the new system online with all its functionality in its own environment allowing attentions to be fully focused on development. Only after the system is fully functioning and all critical pieces implemented should it be brought in as part of the production process.

7.6 Summary

This chapter summarized the overall results of the project and discussed some of the lessons learned as they apply to AFS, but also as they apply to future automation projects. In order for AFS to meet its business case, it will still need to make significant changes. For future projects, LSM should consider the assumptions their plans are built on, the development process, the technologies used, and the applications selected to ensure a more successful project.

Chapter 8

Conclusion

Complex systems need models, and they need models at every stage of the development process. From conception to full production, the model allows for informed decision making and faster progress. The fidelity of the model and the data that it is built on need to be developed in concert. Good system design should include provisions to easily capture data to assess performance and validate the model.

In the case of AFS developed by LSM, at the conception of the project, an initial model was built with large assumptions and simplifications. However, as the project moved from an early stage concept through later stages of development, the models were neglected and were not updated with more accurate data, nor were early assumptions revisited or extra detail added. Poor modeling, amongst many other things, is one of the reasons that the AFS project has suffered and good decisions have been difficult to make.

This project shows that the value of building a model lies both in the information the model provides, but also in the process of building the model itself. Establishing process steps, collecting data, and validating results builds a vertical knowledge of the system and brings opportunities to revisit assumptions to ensure shortcuts are not buried within the system.

For AFS the model was needed to establish a baseline system capacity. It was then used to identify candidates to improve system performance. Additionally, during the process of building the model, several sub-optimal system decisions were identified that were made early on in an effort to get the system running, but then became entrenched. In the end, the improved model gave a more accurate estimate of end-state capacity and showed that the current system wouldn't meet the business case without significant changes or a major change in the proposed statement of work.

This project contains many valuable lessons as LSM moves forward with future automation projects. Ideally these lessons will help LSM to build a better business case, plan better, choose appropriate implementation targets, and better manage their vendors. Automation will increasingly play a more significant role in their production process, and learning to execute these projects well can become a major competitive advantage.

Bibliography

[1] S. Ljasenko, N. Lohse, and L. Justham, "A Comparison of the Manufacturing Resilience between Fixed Automation Systems and Mobile Robots in Large Structure Assembly", *Procedia cirp*, Factories of the Future in the digital environment - Proceedings of the 49th CIRP Conference on Manufacturing Systems, vol. 57, pp. 235–240, 2016, ISSN: 2212-8271. DOI: `10.1016/j.procir.2016.11.041`. [Online]. Available: `//www.sciencedirect.com/science/article/pii/S2212827116311957` (visited on 01/26/2017).

[2] T. Witzenburg. (2017). Industrial Automation - When and What to Automate, [Online]. Available: `https://belcanengineeringservices.com/index.php/automation-main/191` (visited on 01/26/2017).

[3] M. A. Vonderembse, T. Raghunathan, and S. Subba Rao, "A post-industrial paradigm: To integrate and automate manufacturing", *International journal of production research*, vol. 35, no. 9, pp. 2579–2600, Sep. 1997, ISSN: 0020-7543, 1366-588X. DOI: `10.1080/002075497194679`. [Online]. Available: `http://www.tandfonline.com/doi/abs/10.1080/002075497194679` (visited on 01/25/2017).

[4] K. K. Abd, *Intelligent Scheduling of Robotic Flexible Assembly Cells*, ser. Springer Theses. Cham: Springer International Publishing, 2016, ISBN: 978-3-319-26296-3. [Online]. Available: `http://link.springer.com/10.1007/978-3-319-26296-3` (visited on 01/25/2017).

[5] H. Makino, "The comparison between robotic assembly line and robotic assembly cell.", in *Proceedings of the 10th International Conference on Assembly*

Automation, Place of Publication: Kempston, UK; Kanazawa, Japan. Country of Publication: UK.: IFS Publications Japan Soc. Mech. Eng. Japan Soc. Mech. Eng., Jan. 1, 1989, pp. 1–10.

[6] G. Michalos, N. Kousi, S. Makris, and G. Chryssolouris, "Performance Assessment of Production Systems with Mobile Robots", *Procedia cirp*, Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future - Proceedings of the 48th CIRP Conference on Manufacturing Systems, vol. 41, pp. 195–200, Jan. 1, 2016, ISSN: 2212-8271. DOI: `10.1016/j.procir.2015.12.097`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S2212827115011762` (visited on 01/26/2017).

[7] A. Inman. (2017). Flexible Manufacturing - strategy, levels, system, advantages, type, company, business, competitiveness, [Online]. Available: `http://www.referenceforbusiness.com/management/Ex-Gov/Flexible-Manufacturing.html` (visited on 01/26/2017).

[8] M.-C. Tseng, "Strategic choice of flexible manufacturing technologies", *International journal of production economics*, vol. 91, no. 3, pp. 223–227, Oct. 18, 2004, ISSN: 0925-5273. DOI: `10.1016/j.ijpe.2003.08.010`. [Online]. Available: `//www.sciencedirect.com/science/article/pii/S0925527303002706` (visited on 01/26/2017).

[9] H. J. Boonman, V. Hagspiel, and P. M. Kort, "Dedicated vs product flexible production technology: Strategic capacity investment choice", *European journal of operational research*, vol. 244, no. 1, pp. 141–152, Jul. 1, 2015, ISSN: 0377-2217. DOI: `10.1016/j.ejor.2015.01.007`. [Online]. Available: `//www.sciencedirect.com/science/article/pii/S0377221715000272` (visited on 01/26/2017).

[10] B. J. Durham, *Determining appropriate levels of robotic automation in commercial aircraft nacelle assembly*. 2014.

[11] E. E. Ebner, *A systems approach to procurement of automated technology*. 2015.

[12] M. Wilson, "Chapter 5 - Developing a Solution", in *Implementation of Robot Systems*, Oxford: Butterworth-Heinemann, 2015, pp. 103–131, ISBN: 978-0-12-404733-4. DOI: `10.1016/B978-0-12-404733-4.00005-9`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/B9780124047334000059` (visited on 02/09/2017).

[13] P. Waurzyniak, "Ford's Flexible Push", *Manufacturing engineering*, vol. 131, no. 3, pp. 47–56, Sep. 2003, ISSN: 03610853. [Online]. Available: `http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=aci&AN=500885866&site=eds-live` (visited on 01/26/2017).

[14] J. Markoff, "New Wave of Deft Robots Is Changing Global Industry", *The new york times*, Aug. 18, 2012, ISSN: 0362-4331. [Online]. Available: `http://www.nytimes.com/2012/08/19/business/new-wave-of-adept-robots-is-changing-global-industry.html` (visited on 01/26/2017).

[15] Li Zhao, Junhai Ma, and Yinghua Xue, "Multi-mode Intelligent Storage and Retrieval Systems in Automated Warehoue", *Research journal of applied sciences, engineering and technology , vol 5, iss 14, pp 3721-3726 (2013)*, no. 14, p. 3721, 2013, ISSN: 2040-7459.

[16] A. C. Estes. (2017). 13 Fascinating Farming Robots That Will Feed Humans of the Future, [Online]. Available: `http://gizmodo.com/13-fascinating-farming-robots-that-will-feed-our-future-1683489468` (visited on 01/26/2017).

[17] G. Candan and H. R. Yazgan, "Genetic algorithm parameter optimisation using Taguchi method for a flexible manufacturing system scheduling problem", *International journal of production research*, vol. 53, no. 3, pp. 897–915, Feb. 2015, ISSN: 00207543. DOI: `10.1080/00207543.2014.939244`. [Online]. Available: `http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=bth&AN=99712753&site=eds-live` (visited on 01/26/2017).

[18] Y. Yang, Y. Chen, and C. Long, "Flexible robotic manufacturing cell scheduling problem with multiple robots", *International journal of production research*, vol. 54, no. 22, pp. 6768–6781, Nov. 16, 2016, ISSN: 0020-7543. DOI: `10.1080/00207543.2016.1176267`. [Online]. Available: `http://dx.doi.org/10.1080/00207543.2016.1176267` (visited on 01/26/2017).

[19] J. M. Novas and G. P. Henning, "Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming", *Expert systems with applications*, vol. 41, no. 5, pp. 2286–2299, Apr. 2014, ISSN: 0957-4174. DOI: `10.1016/j.eswa.2013.09.026`. [Online]. Available: `//www.sciencedirect.com/science/article/pii/S0957417413007604` (visited on 01/26/2017).

[20] S. S. Chiddarwar and N. Ramesh Babu, "Conflict free coordinated path planning for multiple robots using a dynamic path modification sequence", *Robotics and autonomous systems*, vol. 59, pp. 508–518, 7–8 Jul. 2011, ISSN: 0921-8890. DOI: `10.1016/j.robot.2011.03.006`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0921889011000583` (visited on 02/10/2017).

[21] M. Gombolay, R. Wilcox, J. Shah, and J. Shah. (2013). Fast Scheduling of Multi-Robot Teams with Temporospatial Constraints.

[22] M. Gombolay, J. Shah, J. Shah, and J. Shah, "Schedulability Analysis of Task Sets with Upper and Lowerbound Temporal Constraints", *Journal of aerospace information systems (jais)*, 2014. DOI: `10.2514/1.I010202`.

[23] ati. (2017). ATI Tool Changer Products: QC-76, [Online]. Available: `http://www.ati-ia.com/products/toolchanger/QC.aspx?ID=QC-76` (visited on 01/28/2017).

[24] m. (www.meacon.de). (2017). Industrial Robots, Controller, Robot Systems, [Online]. Available: `http://www.kuka-robotics.com/taiwan/en/products/industrial_robots/special/shelf_mounted_robots/kr210_r3300_ultra_k/` (visited on 01/28/2017).

[25] abb. (2017). IRB 6640 data - Industrial Robots (Robotics) - IRB 6640 (Industrial Robots - Robotics) — ABB, [Online]. Available: `http://new.abb.com/products/robotics/industrial-robots/irb-6640/irb-6640-data` (visited on 01/28/2017).

[26] A. M. Geoffrion, "The Purpose of Mathematical Programming Is Insight, Not Numbers", *Interfaces*, vol. 7, no. 1, pp. 81–92, Nov. 1976, ISSN: 00922102. [Online]. Available: `http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=bth&AN=6700554&site=eds-live&scope=site` (visited on 03/01/2017).

[27] M. Wilson, "Chapter 8 - Successful Implementation", in *Implementation of Robot Systems*, Oxford: Butterworth-Heinemann, 2015, pp. 163–184, ISBN: 978-0-12-404733-4. [Online]. Available: `//www.sciencedirect.com/science/article/pii/B9780124047334000084` (visited on 01/26/2017).

[28] B. W. Oppenheim, "Lean Enablers for Systems Engineering", in *Lean for Systems Engineering with Lean Enablers for Systems Engineering*, John Wiley & Sons, Inc., 2011, pp. 64–246, ISBN: 978-1-118-06399-6. DOI: `10.1002/9781118063996.ch7`. [Online]. Available: `http://onlinelibrary.wiley.com.libproxy.mit.edu/doi/10.1002/9781118063996.ch7/summary` (visited on 03/05/2017).