Technical Report

Gavin Dormer – S2464806

Introduction

As Systems Integration Manager, for the Type 31 Program, I report to the Head of Manufacturing. I'm charged with overarching responsibility for managing the link between IT and Operations. I manage the agile, effective implementation, installation, and commissioning, of state-of-the-art automated equipment and software. Namely the new £19 million state of the art fully automated PEMA Panel Line, which underpins Babcock's investment in Digital Ship Building. I am responsible for the digitalisation of the Advanced Manufacturing area within the T31 Program, which spearheads Babcock's investment into Digital Shipbuilding, enabling Babcock to provide customers with complex engineering solutions to extremely aggressive schedules. Under the umbrella of Systems Integration Manager, I am the Subject Matter Expert (SME) for the software applications, which drive Advanced Manufacturing Automation on the Type 31 program. Therein I am responsible for the development and customisation of said software with particular focus on tailoring Post Processors for each individual CNC station, ensuring we extract maximum performance and utilisation from our Advanced Manufacturing platform.

I have been lucky enough to be granted a position on this Data Science for Manufacturing Course. Having given a brief description of my job role above, it's clear to understand how beneficial and useful the knowledge obtained throughout the course duration could be to me and the greater business.

Aims & Objectives

After several meetings with my manager and other parties from the senior leadership team, we discussed where we could gain the most benefit from my Assignment 2 investigation, we arrived with a united decision that understanding machine throughput would be the most beneficial to the program.

It was discussed what machine we would like to focus my attention on, initially we had agreed that the HGG Profiler would be targeted, this is the machine that I mentioned in my 'Outline Submission'. However, as I started looking into this in more depth and formulating a plan of attack, I realised we were probably looking at the wrong machine. I approached my boss where I suggested moving the focus onto our ESAB Plate Profiler. The ESAB is a German built machine that is a highly complex Plate Profiler. It comprises the following technologies (a) Gas Torch capable of cutting up to 300mm thick section (b) Plasma Torch capable of cutting up to 50mm thick section, this torch is a full 5 Axis torch, as such it has the functionality to apply bevels of varying configurations on plate edges (c) Inkjet Printing Station, this is used for printing text on plates, for example part numbers etc. (d) Inkjet – Blast Combi Station, this station can shot blast paint primer off plates whilst simultaneously line marking with Ink, thus providing datum lines for child parts to be offered up to and welded in position (e) Drilling & Tapping Station, this is used to drill tolerance holes or holes that are too small for the Plasma or Gas Torch to achieve, it also has the functionality to tap (thread) holes post drilling.

With the extensive range of technologies on this machine I thought it would be an ideal candidate for this project. It is also our most critical asset machine, the ESAB is a feeder machine to all other platforms within Advanced Manufacturing. To that end the ESAB's throughput is intrinsically reflected on all other machines. We are currently going through a huge production ramp up phase. Currently we are measured on cut steel volume, and in the near future we need to cut and process 66 tonnes of steel per week to meet schedule deliverables. Around 85% of steel cut will be processed through the ESAB machine.

Our current cut steel volume per week is around 31 tonnes. The challenge is to streamline production by identifying blockers, highlighting, and understanding bottle necks and choke points, understand what's working well and what's not, identify which processes take the longest and understand what and how we could improve. The goal of this study on the ESAB is to highlight all above with statistical solid facts, present findings in a digestible and understandable manor and produce charts and graph providing visual dynamics.

Data Types & Source

The ESAB machine produces production output files which are stored on the machines C-drive. These production outputs are .csv files. We have worked with .csv files several times throughout the Data Science for Manufacturing Course. I have always known about the creation and existence of the .csv files on the ESAB and have looked at them in the past. However, the files were so big and full of spurious information it was very hard to establish what information the files were providing. Now I'm equipped with the tools this course has provided, I have the knowhow and confidence to pull files from the machine and investigate accordingly. I'm able to pull the .csv files from the machine remotely using a bit of software called VNC Viewer, this software allows me access to the machines control system from my office. The .csv files on the machine are stored for 3 months before being automatically deleted. So, at any given point you always have 3 months' worth of data to work with. For this assignment I decided to look at 1 months' worth of data as a pilot. Going forward and if successful I could investigate bigger chunks of data with confidence.

Data Cleaning & Carpentry

Now that I have identified the data type (.csv) that I'm going to be using for this study, I need to pull said data from the ESAB machine. I accessed the .csv files found on the C-drive of the ESAB machine via VNC Viewer, this software allows me to remotely access the machine control from my PC. Once remoted onto the ESAB I copied the .csv files from the ESAB C-drive to a local folder on my PC.

Now that I have the files saved to my PC, I'm able to open 'Jupyter Notable' and commence work. I start by Importing Python Libraries, in this case 'pandas' and 'numpy'. At this stage I have the opportunity to set some variables that will follow me through the process, I decided to issue a command that provides instruction to allow the viewing of all columns in my data set. Next, I import my chosen data set, which for the study I have identified as 'ESAB_df'.

I have now imported my chosen data set and provided instruction to allow me to view the data for initial inspection. Upon initial inspection I note there is much cleaning to be done, I have columns that contain no data, I have spurious ID names, there are hundreds of duplications and loads of nests that look like they have been cancelled. Before commencing data cleaning and carpentry process, I decide to take a create a 'deep copy' of my data set. This is something we covered in the coursework and is good practice to perform this action.

With the 'deep copy' created I can now start to clean the data. I have executed a command to provide me with the shape/size of my data, in this case it comprises of 23 columns and 605 rows. It's useful to perform this action at the start as you can take another look at the shape/size later in the process, this will provide an indication of size/shape reduction and a reflection on the clean-up process. As mentioned, before I have lots of columns that contain no information, this is the first area I will tidy. I provide arguments to remove unwanted columns, then check to ensure this has been successful. It has been successful, and I'm now left with only the columns that contain data. Moving on to the next task which is identifying and removing all duplications. Using various commands, I have established that there are duplications present within the data held across multiple columns. Now that I have identified all the duplications within the data set, I will go ahead and remove them. I will keep the first instance of each item and delete all duplications that follow. I work through one column at a time and qualify changes have been successful as I go. Also noticed is some spurious ‘Program Names’ which are identified as programs that have not been produced through the CAD/CAM system, they are manual programs that have been produced at the machine by the Operators. These manual programs are generally very simplistic by nature and are configured using Macros. For this study we do not want to include them in our data set, these are also dropped.

With the data set partially cleaned I take another look at it, again using the (.head) and (. shape) commands. I can now clearly see that I have a data set comprising of 13 columns and 94 rows, remembering the original data set was 23 and 605 respectively. The columns I’m left with and what they mean are as follows.

- Program Name – Nest ID
- Start – Nest start time
- End – Nest end time
- Cancelled – Nest cancelled Yes/No
- Total Time – Nest run time from start to end
- Breaking Time – Time to strip finished parts of machine bed
- Number Breaks – Times nest program was paused/stopped and then restarted
- Process 1 Time – Time in seconds the Plasma torch was actively cutting
- Process 1 Length – Length in millimetres the Plasma torch physically cut
- Process 1 Starts – Number of times the Plasma torch was ignited in each nest
- Rapid Time – Time in seconds the Plasma torch spends moving around in fresh air above the nest (link moves)
- Rapid Length – Length in millimetres the Plasma torch travels in fresh air above the nest (link moves)
- Rapid Starts - How many times an air move is actioned (how many link moves in each nest)

The next issue with this data set I notice is the Index is wrong, I want the Index column to contain all the ‘Program Name’ data. To force this action, I use the ‘. set index’ command. Now the data set looks a lot cleaner, has a better structure and is providing me with the desired information in a far less confusing manor. To conclude actions thus far I want to summaries the data set size and shape, there are many ways to do this, in this case I have used the ‘. describe’ argument which provides a matrix of useful statistics and the ‘, info’ argument which great information such as, memory usage, number of columns and data types (i.e., float, integer, object)

Having performed above summarisation I now want to go ahead and convert data types to integers; this will aid me later on when it comes to data analysis and visualisation. To convert data types to integers first we must convert data types to float, this is a Python quirk we were made aware of during some of the course work. Once data types are converted to float, we can go ahead and successfully convert to integers.

Data Analysis

Now that I am happy with the data set, I can begin to look at pulling some data analytics from it. By analysing the data set I will be able to identify weaknesses in our process, machine bottle necks and similarly I’ll be able to establish what is working well. To begin, I will investigate the column ‘Total Time’ this will give me a clear picture of how long each nest has been on the machine. With total time attached to a nest number I’m able to jump into my CAM system, pull up said nest and compare projected times to actuals. When we create a nest within our nesting software (Hexagon SPx Nestix) it produces a nest run time, this time is the theoretical time the nest should take from start through to completion. This is great data for me to draw comparisons, I can easily compare CAM projected times vs. actual times.

I now focus my attention on ‘Breaking Times’, this is the time taken to remove finished parts from the machine bed. Breaking times have produced some very interesting results. Since the start of the Type 31 Program, we have factored breaking times into our processes and standard operating procedures, we have established these times by using a best guess. Not technical or accurate at all. With my newly found knowledge and skill set I’m able to pull actuals from the machine. I have learned actual breaking times are nowhere near where we thought they were, also I note the extreme variations on breaking times. Again, breaking times have an associated nest number attached to them, thus allowing me to examine nests within the CAM system. Performing this exercise, I could very quickly view a pattern and trend. It was easy to establish that nests comprising of lots of small sized components track with huge breaking times. Similarly nests with large and few components have a very short breaking time.

Moving on and continually building a comprehensive picture I look at 'Process 1 Time', this being the time in seconds that the Plasma torch is physically cutting. This was interesting and probably more expected than some of the other findings. 'Process 1 Time' tracks 'Total Time' and 'Breaking Time' This makes sense as the larger more complex nests contain more parts to cut and, take longer to complete the program and take longer to remove from the machine. What was surprising though, was on some nests how short a time the plasma torch was on for, some alarmingly short times were displayed. Interestingly the nests with very short torch time in some cases had respectively large breaking times. Upon further investigation and once again cross-referencing nest numbers in our CAM system with known torch time and breaking times, I discovered some terrible nesting behaviours. For example, there were cases where we had two or three small parts nested onto one huge 13600mm x 3200mm plate. This is clearly not an efficient way to nest; it is not good plate utilisation. Time taken to load and unload the machine far outweighed the torch time. If we are loading a 1360mm x 3200mm plate onto the machine, we must perform as much cutting on this plate as possible before removing it.

Next to be evaluated was Process 1 Length, this is the length in millimetres the Plasm torch travels whilst under cutting conditions. There was not a lot of surprises with this, process 1 length very accurately tracks process 1 time. This is a positive as this is expected behaviour. It is still worth while performing this check and looking at this data for inconsistences as this will aid in highlighting potential process issues.

Lastly, I investigate Rapid Length, this is the length travelled by the Plasma torch whilst not under cutting conditions. Essentially any moves made above the plate (in fresh air). We commonly call said movements fresh air cuts. The figures pulled from the data set on this were extremely interesting. Again, they highlighted some poor-quality nesting practices, where in some cases the plasma torch spent more time in fresh air than it did physically cutting parts. When crossed referencing against CAM systems it was clear to see that a lot of work could be done around the sequencing or ordering of parts being cut. For example, if you have a nest containing 100 parts you would expect the plasma torch to cut one part then move onto its neighbouring part, so on and so forth. Almost like joining the dots. However, what I was seeing was the plasma torch cutting a part on the far-left hand side of plate, once completing that part moving all the way across the plate to the far-right hand side before cutting the next part. This process would be repeated over and over and with plates that are in some cases 13 meters or more long, this is a huge distance to travel whilst the torch is not actually cutting and soaks up valuable time. Fresh air moves (rapid length) should always be kept to a minimum.

It was decided to perform a deep dive investigation into a handful of nests, this was to allow us to fully investigate all aspects in detail and track the process through its life cycle cradle to grave. To do this I decided to look at the top 5 nests in terms of program run time, breaking time, torch cutting time, torch distance travelled, and fresh air torch distance travelled. To identify the top 5 nests for each of aforementioned categories I used the '. nlargest' instruction. This provides said information in descending order, so largest values first and descending in numerical order. This instruction was performed on all 5 aspects mentioned above.

Data Visualisation

Having analysed my data set I now want to create some visual representations of results. To do this I start by importing libraries to enable this functionality. I import 'matplotlib. pyplot', 'seaborn' and 'scipy / stats.

I input some commonly used arguments to look at the correlation of the data set and again the size and shape before moving onto producing a plot graph of results, the plot graph shores up results established during the analysation process. By using the plot graph, it is very easy to digest and understand what activities are taking the most time. I display the same results using a bar graph, again this is a great way of visually understanding and interpreting results from the analysation process. I decided to focus on 4 activities when providing instruction to output the plot and bar graphs. I looked at Process 1 Time, Breaking Time, Process 1 Length and Rapid Length. Of course, I could have included many more activities from the data set, but to keep things simple and to generate a high-level report to share with my managers, I decided to look at 4 of the main activities. Going forward I could and would include further activities, this would provide a very detailed picture of throughput on the ESAB.

Now I decided to split the above mentioned 4 activities into 2 groups, I done this because although plot and bar graphs are great for showing the scale of differences between processes it's not necessarily fair to compare time against distance as they are different units of measurement. To that end I split the data set into a time and distance group. First, I look at breaking time and process time, both of which are measured in seconds. I trialled lots of differing ways of displaying the statistics, but in the end settled for the use of a pie chart. I feel the pie chart lends itself to be easily understood and is a very powerful means of displaying data, especially when comparing only a few processes at one time. Looking at the pie chart, it paints a very clear picture, it shows breaking time to own a huge slice of the pie chart at 81 percent, this leaving only 19 percent for process 1 time. The ESAB is only physically cutting with the Plasma torch for on average 19 percent of machine up time. Incredibly we are spending a huge 81 percent of time stripping the machine bed of parts after program completion. These figures shouldn't come as a surprise as they are pulled directly from the analysation statistics, however when you see it in pie chart form it hits home harder.

Next, I look at process 1 length and rapid length, both of which are measured in linear millimetres travelled. Once again deciding to utilise a pie chart to display this information. Looking at the pie chart, it paints another very clear picture, it shows process 1 length to own less than half of the pie chart, in fact only 38 percent This in turn meaning rapid length is responsible for a massive 62 percent of distance travelled. The ESAB is covering more linear distance with the plasma torch in fresh air (performing link moves) than it is physically cutting parts. This is highly inefficient and an area that is easily addressable. Again, these figures should come as no surprise as they are consistent with analytical statistics.

Lastly, I decided to try and produce a heat map of data set results, we have used this technique many times during the course work. Unfortunately, I could not get this to display correctly. I'm not sure where I'm going wrong, I think it may have something to do with the shading. This is something I can deep dive into and resolve when I have more time, this project is time critical so wont waste any more time on it for now. Furthermore, having the heat map displaying correctly doesn't change any of my results, it's more of a nice to have, than an essential.

Conclusion and Next Steps

This data set has been fascinating to work through. With the data set being my own (Babcock's) all the information I was able to pull from it really makes sense to me and I will be able to make a difference with it. I have already presented this report along with my Power Point presentation to my manager, and from that I was requested to share to a wider audience of the senior leadership team. To that end the evaluation of the ESAB machine over the course of a four-week duration has been invaluable.

Many areas in the Program life cycle on the ESAB have been highlighted and need to be revisited, with the findings provided it better places me and the business to know what to target and go after as a priority. Findings will allow the streamlining of processes, and procedures thus ultimately shaving time and better understanding recourse requirements.

There are many features within the machine that we have not enabled, these features are listed below. Once enabled I could perform a hugely more comprehensive study that would undoubtably highlight further areas in need of improvement.

- Inkjet Station run time
- Inkjet Station linear distance travelled

- Text Marking Station run time
- Text Marking Station linear distance travelled
- Vacuum Blast Station run time
- Vacuum Blast Station linear distance travelled

Above features were displayed in the original ESAB_df.csv data set before I started the cleaning process, I deleted all said columns as they contained no values. Once enabled however these columns will be populated with data. My manager has been so impressed with the results of this study that he has raised a purchase order with ESAB to have addition fields enabled. Enabling these additional features is achieved by a simple software update to the machine. I will revisit this study early next year when by then the additional features on the ESAB will be activated and available to me. Interestingly when this ESAB machine was first procured the business had the chance to include these additional features but were reluctant to purchase this upgrade as we didn't understand it's value. With the learning achieved through this course we now understand its potential and have the tools to utilise and interpret the data outputs to the fullest.

Since completing this study and presenting findings to my manager I was tasked with looking at the main areas highlighted in this report. With the objective of streamlining and improving performance. From the results in these reports, I decided to spend some time with the Nesting Team (the folks that produce the nests) I presented my findings to them, and they were able to change nesting techniques and strategies accordingly. So, what did we change, we now ensure that only nests that fill an entire plate are released to production, this maximises machine run time, whilst minimising manual handling times. There are still on some occasions when we must load a plate onto the machine that is not filled full of parts, this however is the exception of the rule and not a common occurrence anymore. We discovered that an auto nest feature within the nesting application was the root cause of nesting only a few parts onto one large plate. These parameters have been realigned accordingly. Next, we looked at ways of decreasing the time spent removing parts from the machine bed after program completion. Previously if we had a nest containing 200 parts, these 200 parts would need to be removed manually one by one after program completion, then finally the nest carcass would be sectioned up with grinders and removed. We have experimented with bridging all parts, essentially this means we incorporate small tags that join all parts to the carcass, so now there are no loose parts. This enables us to remove the nest in its entirety as one plate with the crane and magnate set up, essentially the reverse of how we load the plate onto the machine. Once the plate is removed it is moved onto a laydown area where the bridges/tags can be cut through with a grinder and parts removed. This allows the machine to be loaded with the next plate and start cutting immediately. When the machine is cutting the next plate, we can work on removing parts from the plate that was removed from the machine. The last thing that we revisited was the sequencing or ordering of cut parts within a nest, we again discovered that the way the nesting software chooses it cut path is an automated process and, in most cases, did not make any sense. After talking to the vendor of the nesting software we realised this auto function can be overruled and the user can instruct the sequence or order of parts to be cut. We have applied this newly found technique and it allows the nesting team to drive the cut path in a very methodical and metronomic fashion. The results of this have been astonishing with massive reduction in fresh air cuts in terms of time and distance travelled.

We have been utilising above for over one week now and can see a marked increase in throughput. We have decided to continue with new ways of nesting for one full month, upon which time I will pull the .csv files from the machine and run them through this very same process. We will then evaluate results and compare with outputs with this report. The expectation is to see a huge difference, including far better material utilisation, vastly reduced breaking times, increased torch active times and decreased torch air moves in terms of time and distance. I'm really looking forward to rerunning the data.

During this study we established from the data that we loaded and successfully processed 94 nests through the ESAB machine. This number may not increase hugely with new nesting practises, it may even decrease as each nest loaded to the machine should be producing far more parts. It will be interesting to see how that pans out.

Moving forward I've been tasked with supporting the ESAB machine until we think we are in a good place and happy with process and procedure and ultimately machine throughput in terms of tonnes of steel cut in week. With the imminent release of further data set within the .csv files as mentioned earlier in this report, this will support me in producing far greater studies, capturing a broader spectrum of data, and enabling me to further drive continuous improvement. I have also been tasked by my manager to conduct like for like studies on other machines within the Advanced Manufacturing platform. As mentioned earlier we have 11 machines automated machines supporting the Type 31 Program. If we can use this data provided by other machines to conduct throughput studies as we have for the ESAB machine, the results and focussed improvements are huge. On this Type 31 Program we have 5 Ships to design, manufacture, and build, with that in mind it is critically essential to extrapolate maximum performance from all machines. There is no better way of doing this than using machine data outputs. To date we have only had one statistic to measure ourselves on, this has been tonnes of steel cut in week. We were unaware of what to focus on or what to target for improvements to increase steel throughput. This report supplies clear direction on what areas to target. In time this process could be refined to produce further detail of the manufacturing process with a focused approach on dedicated areas.

There are a lot of areas I could improve on with this Assignment, I have learned loads and found the process fascinating. Working with my own data although very challenging was worthwhile. It is by far more interesting working on data that you are familiar with and understand. I feel very lucky that the data I have pulled and worked on has already been viewed as a triumph, certainly from a Babcock perspective and actions have already been initiated as a direct result. The nest data set I investigate I intend to build on lessons learned thus far, invest more time into my Jupyter Notable workbook making it more intelligent, slicker, and intuitive. Upon reflection I probably bit off more than I could chew, between my full-on day job and family life it has been stressful getting everything completed on time. This was pointed out to me by course lecturers before I started this journey, lessons learned. What I have managed to produce is far from perfect, but it has provided a fantastic insight to the ESAB performance.

I now look forward to putting my newfound skill set into regular use and build on currency level of knowledge. When I first started this course, I remember thinking what’s the difference between using Python and Excel, now I know. Python is by far more powerful and very user friendly once you get the hang of it and learn how to accurately search google for the commands you really want.

Gavin Dormer.