Luis Arizmendi
CS109

Daily CNC Tool Forecast

Video: https://youtu.be/t7RzWRPja18

Introduction

CNC (Computer Numerical Control) Machining is a popular tool in manufacturing, involving machines such as mills and lathes. (See figure 1). These machines typically have to be programmed before they are run and operated for a particular job.

Figure 1: Peterson

However, these machines, in particular mills, use a variety of tools each of which after some wear and tear from removing material eventually become dull or sometimes even break (See figure 2). From the period that they are first used until this occurs, only counting the time spent being in use, is what we will be calling their lifetime.

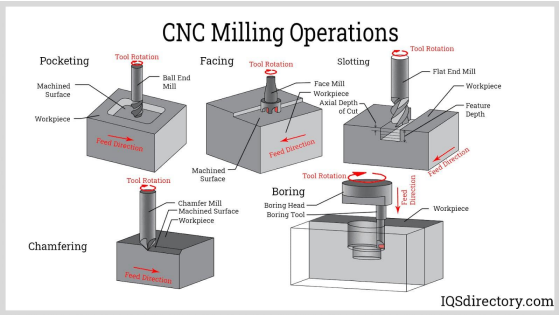

Figure 2: Industrial Quick Search

There are several ways to calculate a machines lifetime, but one way is through an equation know as Taylor's equation: VT^n or T = (C / V)^(1/n) (V = cutting speed, C = machining constant, n = tool life index) (Dwivedi). The numbers C typically comes from a tool's manufacturer, n relates to a job and tool material which are numbers that have been documented for materials, and V will depend on the way one operates and programs a CNC. While many forms of estimating tool life are reliable, this equation does not take into account human errors. These errors can consist of error filled machine code for a job, which is especially potent when one is creating a brand new code. This often occurs when a company may be machining a part for the first time and is in the phase of creating a correct machine code to run on their machines. This process in some machine shops involves some trial and error so there is bound to be many estimated "breaking" tool lifetimes. It may be difficult to calculate how often errors occur, however, probability can perhaps be a useful tool in getting a closer estimate of tool life during these test phases.

It is useful to be able to predict this as productive machine shops would want to have tools available to them always, not waiting on tool orders and purchases, as to prevent down time. At the same time buying a large count of tools for certain parts may be impractical for its cost and potential lack of need for all the tools, especially for

modest shops who's scale may not be able to afford hundreds of extra tool ends.

Project: Creating data

To begin building this project I began by building data. For this implementation it is important to note that I am not using actual data. Instead I will use false data for the sake of an implementation. To do so I created data for one Job/ the machining of one part called Job 1. For a sample of one iteration of this job I collect data for tools used in 480 minutes or runtime for one work day (time variables can be altered to reflect tools used under a different amount of time or time period) , what machine that job was ran on, what brand of tool that machine used, and the amount of coolant (liquid used to cool down parts and tools while the machine is running and gaining heat from friction) used (as a percent). The latter three variables were determined by uniform random variable generator (np.random.choice) while the amount of tools used a version of this but with a poisson. The possion used a lambda which was the amount of tools becoming dull or breaking in one day found by taking the 480 minutes divide by the Taylor Estimate (T=(C / V)^(1/n). After doing this for 10000 times, we have created our data. The code for this part is in makeData.py

Project: Graphical User Interface

The user interface for this project (figure 3), uses the SimpleGui library. It requires the user to first give some csv file with data for different samples of jobs. Then it asks the user for the number of tools the user currently has. This number will be used to see whether or not the user should get more by the end of the day. Next the user should give some information regarding how they will run their job, such as the actual job, the machine, the brand of the tool, and the coolant concentration as a percent (will be rounded to the nearest integer). If the user selects none or leaves collant blank then the application will not consider those factors. Once all is finished being filled out, the user should press ok to run the application. Afterwards, yellow text will appear revealing the probability that the amount of tools used in a day will exceed the amount the user currently has, thus giving the user some information to help advise whether or not a purchase of tools will be necessary for the following day.The cancel button simply closes the application.

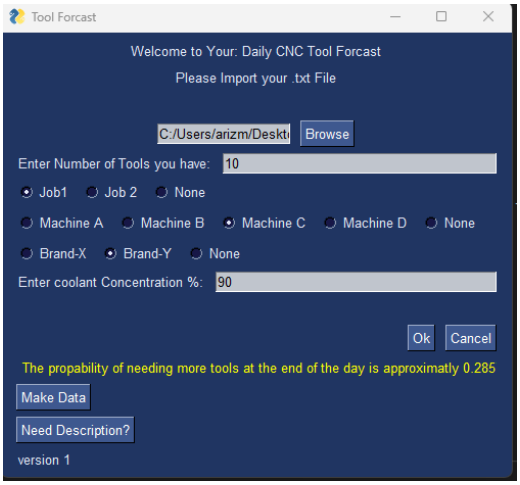

Figure 3: GUI

For the sake of this project the make data function runs the makeData.py code and gives the user some false data in the same folder as makeData.py. The Need description button will give the user this pdf.

Project: Data Analysis

The bulk of the application takes place in analyze.py in which the input csv which will be the data. Ideally this would be data behind actual runs from a machine, but for now it will be a false set of data. First this data is parsed through. Taking in the conditions the user wants to consider, only jobs in which these conditions are true will be observed. From this observed group we can collect all the lifetimes and record the size of the group. Thus we can find a probability mass distribution for this conditional probability: P(X = Tools used in a work day | Conditions). Our application using matplotlib.pyplot produces bar graphs to represent a distribution of used tools as well as a pmf. Please refer to figures 4 and 5 to see these graphs respectively.

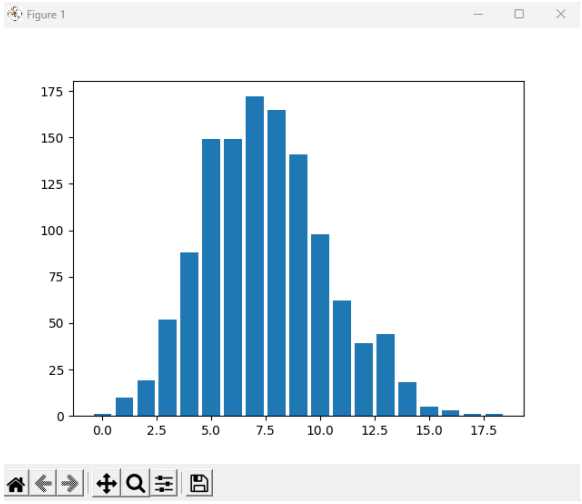

Figure 4:

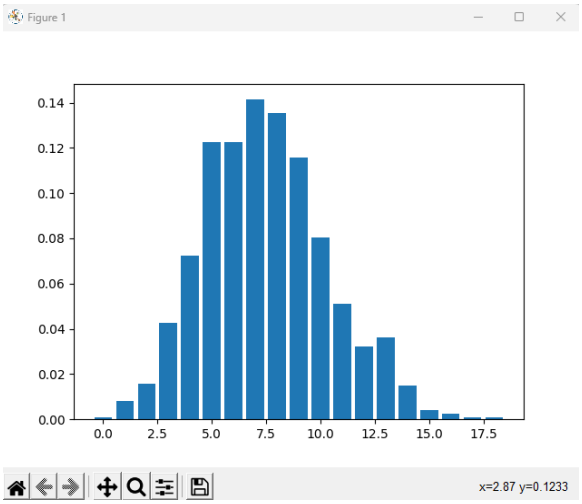

Figure 5:

Additionally, we also record any of the times in which tool use exceeded or met the amount of tools the user currently had, and produced a probability that this would occur. (via division of the number of observed samples) This probability should ideally help a user decide whether or not it would be a good idea to obtain more tools for that day. This probability is displayed in yellow text as described in a previous section. Ultimately the user will be left with a daily (although could be altered to reflect a different time period) CNC tool forecast application.

References

Dwivedi, Kamal. *Tool Life, Taylor's Tool Life Equation, Calculation, Factor*. Mechanical. 21 January 2022. https://www.mechical.com/2022/01/tool-life-taylors-tool-life-equation.html

Industrial Quick Search. *CNC MACHINING.* Industrial Quick Search. https://www.iqsdirectory.com/articles/cnc-machining.html

Peterson, David. *New CNC Mill and LAthe are Ready to Rumble.* The Eagle Online. https://newulmeagle.com/16918/uncategorized/new-cnc-mill-and-lathe-are-ready-to-rumble/