

Speed Up Packer

Engineers from one of the world's best known packaged goods manufacturing companies get a mandate: increase capacity 20-25% - "headquarters" wants more of their high-demand product shipped out the door.

The team is chartered to determine how to eliminate the constraint in the line. If they can, plant engineers believe they have capacity in the system to increase throughput.

Do they?

Introduction

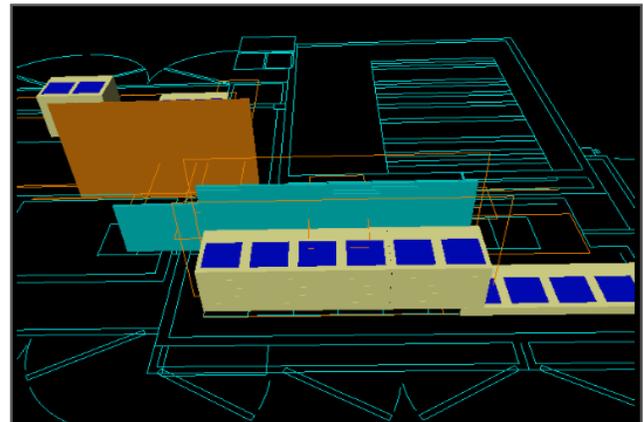
A straightforward request: increase line throughput by 20-25%. Marketing and sales at a large consumer products company have a promotional success on their hands and have increased demand for their product. The packer has long been suspect as being the reason for the line's low throughput. If engineers can coax their packer to pump out product faster, the company can increase sales and profit. And the solution can be deployed to many other lines within their organization.

A brainstorming session is held in which the engineers identify many potential causes for the packer's low performance. They develop options to address these reasons. They land on a few:

- ▶ Prestack Cartons - Burden packer less by pre-staging product
- ▶ Two In-feeds - Create a second in-feed
- ▶ Lengthen Current In-feed - Longer stroke
- ▶ Reconfigure the stacker
- ▶ Replace the Packer
- ▶ None of the Above

It's in the Packer

Plant engineers suspect that if they move the constraint to upstream equipment, they can reach pay dirt. They are convinced that they can speed up the line with alterations to the packer. The problem is that their machine has several different subsystems and there are multiple options to pursue.



Before they invest the money in any one of the solutions, they need to verify that it is viable. One method to do that is to pilot the "best" solution on one machine. This will require capital for engineering and machine modifications. Not to mention, downtime on the machine for the changes - and potential lost production if the solution doesn't work out the way they hope. They call on a team from Haskell to emulate the machine and test changes to see if they work.

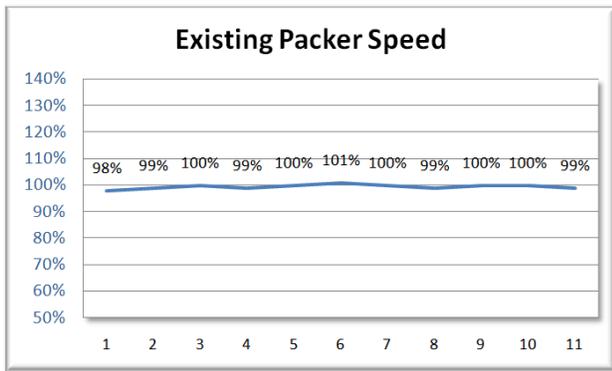
Is it Valid?

How do you know that the model of a machine represents the machine well enough to accurately predict the outcome of the changes? In this case, the packer had multiple subsystems. Each of these subsystems has a dwell and cycle time for each action; moving a cylinder in and out, for example. Not only did each of these actions have to be modeled, but the timing of them had to be accurate.

The Haskell team collected data from the machine for each of the subsystems. Code was added so that each machine action could be timed over a period of multiple shifts. This data was then statistically processed and the result was used for the model.

The team then connected the model to the PLC program running the machine. The most accurate way to model the logic of any system is to use the actual code. This is called an Emulation - the PLC "reads" the input from the model and directs the outputs to cause actions in the model, according to the logic.

Engineers and operations professionals both agreed that the model was spot-on. Everyone could move forward with confidence that changes to the model would accurately reflect change to the real system.



Need for Speed

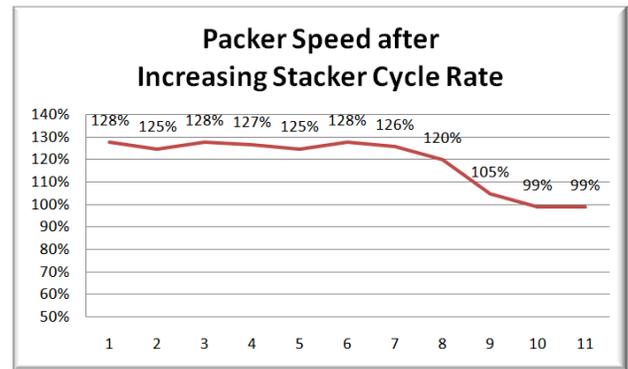
The plant engineers and the Haskell team created four options for a faster in-feed: pre-stack cartons; create a second in-feed; lengthen the current in-feed; and reconfigure the stacker. A fifth option was to review the business case to replace the machine. The sixth option was to do nothing.

The team decided to pick the easiest method to model a faster in-feed. This would help determine if this was the right path. If so, they would then investigate each of the options of how to speed it up. They chose the fourth – reconfigure the stacker. This option, would be the lowest cost and lowest risk to implement.

Each of the cycle and dwell times for all the machine’s actions were variable in the emulation. The engineers could change any of the timing to determine the impact it would have on the operation of the machine. Pushing up a row of cartons could happen quicker; the pusher that moved the stack into the machine could move out and in faster; and the conveyor speed for feeding the cartons could be increased. With these changes complete, the start button was pressed.

The team recognized success instantly. The rate at which the cartons were going into the machine was up by nearly 30%. But, then things started slowing down.

A subsystem in the machine became an internal constraint, and the machine slowed down to its normal design rate. The team moved to this subsystem and sped it up – and another one became the constraint. This continued until the team resolved that a complete machine rebuild was necessary to maintain the 25-30% increase in throughput.



No Business Worth Pursuing

While disappointed that their best efforts to add speed to the line weren’t possible, the team was convinced by the veracity of the data and they now knew that to achieve the results requested, the packer would have to be replaced. They passed on that option for a later time when an overhaul of the line was budgeted for.

Conclusion

Sometimes the best business choices are those not made at all. While the conclusion to do nothing didn’t ring with satisfaction, the team avoided purchases that could have easily exceeded a half-million dollars and, potentially, caused a big interruption to their existing production.