Product driven quality control

Adeline Motte¹, Samuel Bassetto¹

¹Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Montréal, CANADA
(samuel-jean.bassetto@polymtl.ca)

Abstract – Starting with the observation that a great number of defective products are released on the market, we wondered if product driven quality control (PDQC) could be more efficient to prevent such situations. To test this policy, we have built a job-shop manufacturing system using multi-agents technology that enables products to be pro-active both for production and quality controls. In this system, each product is aware of its own quality and takes the initiative of quality measurements. PDQC has been systematically compared to classical frequency based quality control (FQC) policies. The results of the simulation show that there are less defective products in the case of PDQC and a better detection. However the level of uncertain products as the speed of detection are not influenced by this policy.

Keywords – quality, multi-agent system, product driven quality control

I. INTRODUCTION

A loss of one week or more, of production, is a major scrap event that occurs in manufacturing environments. Unfortunately such events are costly nightmares that almost every plant manager has to cope with. Of course, these catastrophic events are not well publicized as they generate losses: manufacturing losses, re-manufacturing costs, logistics costs, systematic shrink of market share, and a severe damage to the brand. The consequences for customers concerned with these events can also be catastrophic, ranging from a production disruption to injury, and even death. For example Thirumalai et al. [1] studied the financial consequences for companies when there are product recalls in the medical device industry. Hora et al. [2] presents consequences for a company in the toy industry.

The policy of quality control is in the front line to prevent such catastrophic events. Massive defects occur due to quality controls inability to detect the failure, its propagation and repetition. At the floor level, quality and production controls interact and impact directly on the level of uncertain products also named Material At Risk (MAR). To illustrate this purpose, let’s present a simplified example. In a factory, two machines produce the same type of items. The control frequency is one out of six. Controls are used to know if both the product and the machine are under control. The dispatching is presented in Fig.1, it is cyclic: 4 on machine 1, 2 on machine 2. So that all the controls are done on products made by machine 2. As a consequence, the status of machine 1 is unknown. After 19 products, the MAR is 13 for machine 1 and 0 for machine 2. It means that 13 products might be defective if there was a problem on machine 1 while it should not take more than 2 items to discover mistakes of machine 2.

Some researches have been done on that topic. Unfortunately, the control of uncertain products is not included in sampling standards. The integration of quality control policy and production control is a research domain that has started with the work of Hsu and Tapiero [3]. Bean [4] realized a tool to adjust the in-line inspections in order to limit the MAR. Bettayeb et al. [5], created algorithms to optimize control plans that limit the maximum number of uncertain products. Sahnoun et al. [6] proposed to analyze risk data in a semiconductor factory to find the best lots to be controlled in order to limit the risk. Recently Colledani and Tolio have presented an integrated design of the quality system and the production system in order to balance adequately the cost of control and the cost of non detected drifts [7]. These works present a common characteristic: the integration is thought in a centralized manner. This leads to a monolithic system often very complicated to handle.

Other researchers like Trentesaux et al. [9] and Sallez et al.[10], investigate new paradigm of production control. Instead of centralizing, they give to products some skills. In their framework products are able to sense, judge and interact with their environment. These researchers use local intelligence, to bring to product enough awareness of their environment to decide of their future in the manufacturing system. These experiments are mainly based on multi agent system (MAS) concepts. They are essentially centered on the production control.

The key idea of this paper is then to test if product driven quality controls (PDQC in the remainder of the article) would retrieve better results than classical frequency based quality control policy (FQC in the remainder). The remainder of the paper presents this investigation. It is structured in three sections. Section 2...
presents the methodology of this research. Section 3 presents the model and the associated simulator. Section 4 presents the results and the associated discussion.

II. METHODOLOGY

To study the system’s behavior in a case of PDQC, the methodology relies on a simulator using the multi-agent system technology. The manufacturing system modeled is a job-shop able to produce a set of products. The simulator has been thought to be adaptable and to take its parameters from an external file. Simulations compare several quality control policies. Five indicators have been used to compare PDQC and FQC policies:

- The number of defective products produced during the simulation.
- The number of defective products that have not been detected.
- The number of drifts (errors) of the machines detected by the control devices.
- The number of finished products recalled to be checked by the measurement devices.
- The maximum values of each machine’s maximum MAR value and the same for the products’ MAR and the detection speed.

A design of experiment on parameters of quality control policies is employed to know their impacts on the indicators above.

III. PROPOSITION

In order to compare PDQC versus FQC policies, in a joint quality and production control, the simulation tool relies on multi-agents technology. NetLogo has been employed for its open and convenience to use architecture. The simulator has been developed under the version 4.1.3 on a MACBookPro (vers. 2011) 4GB RAM. Input and output are summarized in Fig. 2. For research purpose, the simulator is available on demand to authors.

**Simulator structure:** Machines and products are two kinds of agents. They know their own characteristics. Each product carries and knows: its process plan, the number of operations done on it, measures performed at different stages of its process plan, a need of being controlled, its state and information about defectiveness of operations from its process plan. Each machine knows a list of operations it is able to perform (qualification), the time needed to perform it, the maximum number of operations allowed between two controls, the target value of operations, the frequency of drifts and their values. These last two parameters are given for simulation purpose and never retrieved to products for their decision of being controlled or not. Each manufacturing tool embedded also variables to count the number of products waiting in front of it, the number of operations done since the last control, the numbers of mistakes that occurred related to it and the machine’s state. The control limits are the mean value +/- 3σ. The specification limits are given by +/- 4σ where σ is the standard deviation of the normal distribution function. Input files give most of the variables.

In order to simplify the use of the simulator, the choice of the control policy is done directly on the interface. There are four types of quality control. Three are based on the calculation of a given frequency (FQC) another is driven by products (PDQC). They are listed below:

- **FQC by machine:** in this case, each manufacturing tool counts the number of operations since its last control. When this counter reaches a threshold limit, the machine orders the product it is processing, that it has to be controlled immediately.
- **FQC by the number of products of the same type:** A product supervisor counts the products of each type that have been processed. When the counter reaches the threshold, a quality control flag labeled the next product (of this type). In this case products are checked only when they have completed all their operations. The control acts as an outgoing quality test.
- **FQC by the total number of product** is similar to control by number of products of the same type except that it is not counted for each type of product but for the global number of items produced. The control acts as an outgoing quality test.
- **PDQC** is the object of this research. In this quality control policy, products communicate with machines and a global supervisor every time they end an operation. This communication help them to value their own risk of being defective. When this risk reaches a given sensitivity threshold, the product knows it has to be controlled. This behavior is described in detail in this section.
As shown in Fig. 3, the simulation starts by creating the machines, workshops and by creating products. The initiation gives the products and machines their characteristics. The simulation is then run until the stock of orders is empty. The algorithm of control, in Fig. 3, can be presented in two fold: (1) For FQC and (2) for PDQC.

(1) In case of FQC policy:
- If a machine has reached its frequency for drifts, it shifts its mean operating value from a given value ($\delta$). It is assumed that once a drift occurs, it affects every product that is processed on this tool, until one of these products is controlled as faulty.
- Control machines check the products. They have to see if the processes remain under control and if the products are inside the specification limits. A blank noise (random value) is added to measurement to model the alpha and beta risks of control.
- For each machine operating on a product, if the operation is finished, it gives a measure to the product. It is a variable subject to potential drifts. With that value and the standard deviation, the simulator uses the normal distribution function to generate a measure for the product.
- Products can move from one place to another: products that have finished an operation ask a supervisor if they have to be controlled (following the FQC policy). If, yes, it go to the control machine. If not, it asks which machine can make its next operation. It choses the fastest one and the shortest waiting queue. Queues of manufacturing tools behave randomly [6].
- Graphics are drawn to show the MAR of each machine and each type of products and the speed of drifts detection.
- To finish the system writes the new data in the output file.

(2) The PDQC behaves slightly differently. The key idea is to give to the product the ability to decide if it is at risk or not. For this reason a quantified indicator of danger is computed by each product depending on data it can gather from its environment. This indicator is built by summing four sub-indicators:
- **Tool health:** The health of each machine is computed based on the operation value ($op$) retrieved to the product. Three functions have been used for this computation. The first one is $y=x$, the second one is $y=x^2/70$ and the last one is $y=e^{x^2}/x$. This computation is compared to two limits: $op+2\sigma$ and $op+4\sigma$. If the computation is below $2\sigma$, the health is considered as “with no risk”. If it is between $2\sigma$ and $4\sigma$ the risk is considered as “small”. If it is over $4\sigma$ it is considered as a “big”. For instance, the third machine should operates the product N°8. Its $op=17$ with a standard deviation of $\sigma=4$. The actual value retrieved to the product is 37. If the computation function is the first one ($y=x$), the health index is 37, which is greater than $17+4\sigma$ and means the product has a big risk to be defective.
- **Tool history:** The background of the machine gives its trend to produce defective items. It is given by the mean number of defective products divided by the number of controls.
- **Uncertainty produced by a tool:** The number of operations done by the machine since the last time it has been controlled is used to help the system keeping a number of uncertain products below a certain level.
- **Uncertainty over products of the same type:** The number of products of the same type done since the last one was controlled has the same role as the number of operations done by a machine.

Each sub-indicator behaves in the same manner: A small risk adds 1 to the danger level, and a big risk adds 3. Each product has, then a “danger level”, the sum of the sub-indicators given above. When a product reaches a sensitivity threshold, it decides to get controlled. Sensitivity has a great influence on the number of controls done during the simulation. Indeed, if the sensitivity is 1, every time a product has a small risk it will be controlled, in that case an item might be controlled several times. If the sensitivity is 3, a product will need a big risk or 3
small ones to ask for a control and if it is over 12, a product can’t ask to be controlled after just one operation, this is why products with only one operation will never be checked.

The action of control on MAR: Each control has an impact on the evolution of Material At Risk. Each time a control is operated, the state of the controlled product is revealed and clues about quality of precedent products are also unveil.

There are two kinds of MAR: those belonging at machines and those for products. A machine’s MAR represents the number of operations this machine has operated. It increases by one each time a product is operated on this tool. The decrease occurs by controls. If the product controlled doesn’t reveal a drift, all the operations done since the last control (concerning this tool) are supposed to be well done. The MAR will decrease by this number of operations. When a drift appears, it means that operations done since the last control might be impacted by the drift. The concerned products have to be checked. When a drift is detected, the MAR decreases only by one. Other products have to be checked, and the MAR decreases accordingly.

A product’s MAR is the number of products of a given type that are uncertain. It means that the manufacturing system has not monitored if they respect their specification limits. As for the machine’s MAR, a product’s MAR increases by one for each product of its type finished and it is the same mechanism to decrease it.

Technical development: In order to be able to match the simulator to a real shop floor, every parameter is taken from an input file. No limitations have been placed on number of controls, no on manufacturing devices that can be simulated. Parameters are: Number of machines, number of products, type of products and associated process plan, tools qualifications, mean and sigma for every operation. To compare quality control policies a basic manufacturing configuration is given. Simulations have been made with 2 control devices, 10 machines keeping the same characteristics along the simulations and 9 types of products with a constant number of operations to do and a number of items to produce. The first measurement device is dedicated at in-process measurements. The second one is dedicated to recalls measurements and MAR inspection (in case of drift detected by the first measurement tool). A simulation stops when all the products have been produced. It needs about 500 iterations to produce 500 items.

The frequency in FQC policies and the sensibility of PDQC are variables employed in design of experiment. An illustration of the technical interface is presented in Fig.4. The next section will analyze the results of these simulations of the comparison between them.

IV. RESULTS AND DISCUSSION

The simulator has been monitored with: number of controls, number of defective products, number of errors, number of recalls, MAR and detection speed for the different simulations of the experiment plan. This represents a total of 9 indicators (due to 2 control devices). The experiment consists in running the simulator for the four control policies and makes a recursive variation of the quality control parameters. We kept 2 set of four simulations gathered in two comparable groups to analyze them in more details. They are presented in Table I.

Global results: PDQC retrieves two times less defective products, reaching customers, than FQC policies. PDQC retrieves also better performances in the detection of defects. Unfortunately, the MAR for both machines and products are not better with the PDQC policy. Finally the detection speed is not influenced by any of the variations made.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>Extract OF THE RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of control</td>
<td>Controls 2</td>
</tr>
<tr>
<td>By machine</td>
<td>24</td>
</tr>
<tr>
<td>By total number of products</td>
<td>28</td>
</tr>
<tr>
<td>By type of products</td>
<td>25</td>
</tr>
<tr>
<td>Product driven</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of control</th>
<th>Controls 2</th>
<th>Defective</th>
<th>Undetected</th>
<th>Errors</th>
<th>Recalls</th>
<th>Max MAR machine</th>
<th>Max MAR products</th>
<th>MAR speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>By machine</td>
<td>101</td>
<td>47</td>
<td>128</td>
<td>130</td>
<td>95</td>
<td>30</td>
<td>17</td>
<td>28</td>
</tr>
<tr>
<td>By total number of products</td>
<td>99</td>
<td>36</td>
<td>120</td>
<td>102</td>
<td>79</td>
<td>33</td>
<td>5</td>
<td>28</td>
</tr>
<tr>
<td>By type of products</td>
<td>96</td>
<td>34</td>
<td>106</td>
<td>87</td>
<td>86</td>
<td>40</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Product driven</td>
<td>102</td>
<td>31</td>
<td>157</td>
<td>40</td>
<td>72</td>
<td>22</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>

Analysis: An extract of global results is shown in Table I. It presents the indicators for a small number of controls (around 25 products sampled $\Rightarrow$ sampling rate –SR– at 5%) and for a large number (around 100 products sampled $\Rightarrow$ SR at 25%). For each case, every quality control type is presented. From this two extreme situations one can infer following observations:

- **Defective products**: The number of defective products is always smaller in the case of PDQC. Compared to small SR (5%), PDQC generates 183 defects, while FQC policies are between 193 and 238. PDQC improves the detection ratio between 5% to 23%.
Compared to larger SR, there are 57 defective products with PDQC while they are 106 to 128 with FQC policies. The improvement is then between 46% and 55%.

- The non-detection ratio (NDR) (undetected/defective), is also in favor of the PDQC policy. For FQC with low SR, the ratio is respectively 91%, 89%, 94% (line 2 to 4). The PDQC non-detection ratio is lower at 83%. This represents a global improvement ranging between 6% and ~12%. The same observation is done with high SR. PDQC NDR is at 70% while FQC NDR are at 82%, 85% and 86%. This represents an improvement of detection ranging between 14.6% and 18.6%.

- Drift detection (errors): FQC and PDQC have been compared according to drifts that have been detected. PDQC and FQC are equivalent (around 30 drifts) for low SR. The difference is noticeable when the SR increases. PDQC observes 73 errors, while FQC generates between 79 (total number of product), 86 (type of products) and 95 (by machine). This represents a bad performance of PDQC ranging of ~23% to -7.6%. No explanations have been found to explain this situation.

- Recalls: Compared to FQC with low SR, PDQC retrieves a quite comparable amount of recalls: 69. It is less of 0.3% than the FQC policy by total number of products with 71, but far more of 46% of FQC policy by products with 46 products. It is also more of 81% than the FQC by machine, which retrieves only 38 recalls. The performance is better compared to FQC with high SR. The PDQC retrieve 37 recalls. This is more (of 12%) than the FQC by total number of products (33). This is less than other FQC: 37 for the products, 40 for the FCQ “by type of products” and 50 for the FQC by machine. The improvement ranges between -12% to 34%. The recall factor is very complex as it is directly linked at MAR.

- MAR: The maximum machines’ MAR is the biggest in case of “by machine” with 48 (High SR) and 17 (Low SR). It is respectively at 33 and 12 with PDQC. The FQC policy “type of products” is third with 21 and 8. The best policy is FQC policy by “the total number of products” with respectively: 13 and 5. For the products’ MAR the best is always for the FQC policy “type of products” with 18 and 6. The other policy (including PDQC) are worst. For low SR other three policies are around 50. For high SR controls PDQC is the second best one with 13, then it is the FQC policy by “total number of products”: 28 and to finish it is FQC “by machine” with 38. This worst result could be explained with the computation of the “danger index”. The algorithm uses fixed thresholds whatever the sensibility chosen. For simulations where the control frequency is smaller than these thresholds, machines’ and products’ MAR are also smaller for the FQC than for PDQC.

- Detection speed: The PDQC is always second with 15 and 19. This performance is not resolved for the moment.

As the PDQC policy is much better than the others for the defective products and the detection, it could be judicious to investigate further in order to use this kind of controls. To limit the MAR it is possible to give a bigger penalty so sensibility will be reached more easily.

V. CONCLUSION

PDQC is a new control policy where products can compute some pieces of environmental information and decide if they have to be checked. To know if PDQC could be valuable in a job-shop environment, a simulation has been made and a comparison systematically performed with classical FQC. The results show that PDQC helps reducing the number of defective products and the part of non-detection. The PDQC policy reveals also some weaknesses inherent to its constitution. It presents worst results in errors detection and in recall prevention. The detection speed is equivalent to FQC policies. These results are encouraging and pave the way for further tests on: “danger index”, governing rules of agents, other performance indicators, and comparison with dynamic frequency based policies. Finally a scale-1 test in a real factory, could also give some insight in the acceptability of such concepts.

ACKNOWLEDGMENT

We would like to thank Pr. Jean-Marc Frayret from polytechnique Montréal, for his sound participation to the discussion about the MAS model.

REFERENCES