Accelerated Ramp-Up of Assembly Systems through Self-learning

Robert Oates, Daniele Scrimieri, Svetan Ratchev

Accelerated Ramp-up of Assembly Systems through Self-Learning

Robert Oates[1], Daniele Scrimieri[2], and Svetan Ratchev[2]

[1] The School of Computer Science, University of Nottingham
[2] The Precision Manufacturing Centre, University of Nottingham

{robert.oates,daniele.scrimieri,svetan.ratchev}
@nottingham.ac.uk

Abstract. The ramp-up process of assembly systems has a huge impact on both the productivity of those systems and the quality of the output. In this work we present a new technique for accelerating the ramp-up process by automatically capturing knowledge about a machine and subsequently reusing it to inform an engineer performing ramp-up. This technique relies on a novel process called the Knowledge Object Algorithm. The technique is explained and demonstrated using synthetic data, designed to emulate a typical use case of such a system. The future direction for this work is also outlined and further experiments detailed.

Keywords. Self-Learning, Ramp-up, Assembly Systems

1 Introduction

The ramp-up phase of a new assembly device is the time from when the build phase ends, to when the machine reaches maximum productivity. This process includes tuning the machine to reach optimum performance and incrementally enabling components of the machine. Decisions made during this period dramatically change the time it takes to deploy a machine to a site. As a result accelerating this process would represent a significant saving for manufacturing enterprises and systems integrators alike. During ramp-up, all adjustments, whether they directly improve the system or not, add information about the machine. Such data can be analysed and used to provide decision support for the machine they were gathered on and other machines with overlapping functionality.

The structure of this paper is as follows: In Section 2, ramp-up will be discussed in terms of the state-of-the-art in the field of manufacturing and relevant background from the field of machine learning will be introduced; in Section 3 a new technique for capturing and using knowledge from the ramp-up process is presented; in Section 4 the new technique is demonstrated using an artificial case study; and in Section 5, conclusions are drawn from this case study and a discussion about the future of this work is presented.

2 Related Work

2.1 Ramp-Up Acceleration and Performance Indicators

There has been a lot of interest from the research community in reducing the ramp-up phase of production systems. The aim is to bring to manufacturing enterprises a considerable economical advantage by reaching the desired production and quality level as soon as possible, thereby decreasing the time-to-market. A number of investigations have focussed on improving the efficiency of this process [1-4]. The ramp-up process is heavily human-driven and usually requires detailed knowledge about the manufacturing technologies and processes involved. Part of the knowledge is acquired only during the process itself, which is essentially executed by trial and error and, as such, is difficult to plan and predict. All the decisions are therefore taken by shop-floor operators based on their experience. Many researchers have emphasized the need for analysing the ramp-up process empirically, especially in early stages, and acquiring the resulting knowledge [4,5,6]. Knowledge capture and formalisation would avoid human knowledge loss and favour its reuse and sharing. In [4] it is advised that production should be reduced in the early ramp-up stages in favour of learning. This would rapidly increase the production once a proper understanding of the process has been built. In [1] the lack of systematic mechanisms for knowledge capture and reuse has been pointed out as one of the problems affecting ramp-up times in a case study in the automotive sector.

In order to minimise ramp-up time, it is helpful to be able to measure and compare the performance levels reached at specific points. This would permit the quantification of the quality of improvements obtained in each stage and guide the process. The use of measures in manufacturing systems has been analysed in numerous production contexts and many methodologies and techniques have been developed for quantifying different measurable aspects and for designing the appropriate measures [7, 8, 9, 10]. However, none of these methodologies have been specifically defined for ramp-up. In addition, most of the metrics are of a strictly financial nature and do not satisfy all manufacturing needs. In [8], the importance of different factors, such as cost, time, quality, flexibility and productivity are highlighted, while in [7] aspects like availability, reliability and responsibility are proposed as candidates for the design of metrics. In [11] a framework is presented for the systematic measurement of ramp-up performance in order to support a decision-making process which eventually will lead to ramp-up time reduction. The application of such a framework allows for the determination of performance targets and permits shop-floor operators and automatic learning and adaptation systems to get feedback on the effect of performed actions on both specific measured aspects and the overall system performance. The feedback obtained in this way supports the evaluation of the ramp-up process and gives an indication of how far the current system state is from the target.

2.2 Machine Learning and the K-Nearest Neighbour Algorithm

The field of machine learning (ML) is a collection of algorithms and techniques for solving problems which require inference and/or generalization about information. ML is vast and a full review is outside the scope of this paper. Instead a brief intro-

duction to the theory behind the classification problem will be presented, with particular attention paid to the k-Nearest Neighbour algorithm (KNN). The interested reader is directed to [12] for a general introduction to ML.

The classification problem is the general problem of identifying which, of a group of pre-defined sets a given input vector belongs to. It is typically a "supervised" learning problem, i.e. it is assumed that an indicative "training set" of labelled examples for each class are available for analysis, to allow the generalized properties of the class in question to be inferred.

One algorithm designed to solve the classification problem is the KNN. KNN attempts to classify a given input vector using the k samples which are closest to it from the training set. When k is set to one, the class of the training example which is closest to the input vector determines the class assigned to the input. For higher values of k, the neighbours are used as votes, with the most frequent class determining the class assigned to the input. Draws are typically solved by weighting the votes by the inverse of the distances between them and the input vector.

There are a number of common adaptations of the KNN including the use of different distance measurements, different voting schemes and density-based heuristics for dynamically weighting neighbours [13]. As a result the algorithm is highly tuneable and has enjoyed success at a wide range of classification problems including shape recognition [14], protein structure prediction [15] and database retrieval [16].

Of note for this work are techniques for classifying data when the dimensionality (the number of parameters used to describe the vectors) of the input vector is different to the dimensionality of the training set. For the ramp-up problem, the dimensionality can rise, (as new parts are added to the machine) or fall, (as parts are taken offline for investigations). Work has been done in the field of support vector machines to handle moving from high dimensional training sets to lower dimensional cases [17] but the process is one way. Basri et al. make use of a KNN algorithm combined with a surrogate space [16]. For the database retrieval problem the dimensionality, of the database tables (used as a training set) and the input vectors are not always the same, making it difficult to find a metric to measure the distance between the two. The solution to this problem is to transform the input vector and the training set vectors into a surrogate space where both can be represented using the same dimensionality, thus allowing the use of standard distance metrics. The key assumptions being that at least some of the vector dimensions are not entirely orthogonal, (i.e. it is possible to project information from one axis to another) and that a transform exists to move the samples into such a space.

3 Methodology

3.1 The FRAME Project

The FRAME project is aimed at accelerating ramp-up through the development of software tools. Several techniques are being used to achieve this, but here only the techniques involving self-learning will be explored.

The FRAME project uses sensors to capture the current state of the assembly device being monitored. One of the novel features of the FRAME project is the use of a

standardized Key Performance Indicator (KPI) component. This allows a manufacturer-specific measure of performance to be applied to the machine's state and provides crucial feedback to the rest of the system. The mechanics of this component are outside the scope of this paper, but are described in [11]. An "experience recognition component" (ER) monitors adjustments made by the engineer working on the machine, either automatically, (for integrated software adjustments) or through the engineer manually entering the adjustment through the HMI. After the system has settled into its new state, the ER creates an "experience", a data structure which contains an aggregate of the system state prior to the adjustment, a description of the adjustment and an aggregate of the system state after the adjustment. All relevant experiences are stored in an experience base. A self-learning component is responsible for converting these experiences into knowledge, which can be used to support decision making about adjustments in the future. When presented with the current state of the machine, a ranked list of possible adjustments is provided to the engineer. If the engineer chooses to perform the recommended adjustment, information about that adjustment is fed back to the self-learning component, via the ER, allowing more informed decisions to be made in the future.

3.2 The Knowledge Object Algorithm

The challenge for the self-learning component is to provide decision support about the best adjustment to make, based on the provided experiences, and the current state of the machine. There are several features to this problem that make it challenging. Firstly, as the machine is undergoing ramp-up, the dimensionality of the data being provided to the system is constantly changing as sensors and actuators are brought on and offline whilst debugging the machine. Secondly the number of training samples is extremely low for an ML algorithm to generalise from. Thirdly, the problem must be framed in a way that allows the self-learning component to move between machines with little effort. In [18], for example, the ramp-up self-learning problem is treated as a reinforcement learning problem, mapping the adjustments as operations to navigate a space of system states. Unfortunately reinforcement learning techniques not only struggle when the dimensionality of the vectors is large, they also require the individual dimensions to be manually partitioned into states. Arguably, this is self-defeating in terms of accelerated ramp-up, as savings in the commissioning of the machine, have to be balanced against the effort of manually partitioning the system parameters.

These three factors point towards the use of a solution based on the KNN algorithm as it can be adapted to deal with varying dimensionality, provides an answer from the moment it receives its first training example and does not require manual adjustments between problem domains. Framing the ramp-up problem as a classification problem means treating adjustments as classes, where the object is to classify the current state of the machine as being a member of the set of states which would benefit from a given adjustment.

The knowledge object (KO) algorithm is a novel algorithm, based on KNN, specifically designed for the ramp-up problem. The "before" state of each experience in the experience base is used as the training set for a KNN with k set to 1. Repeating

the same adjustment, in the same dimensional space, adds training vectors to the class which represents that adjustment. For new adjustments and for adjustments that have been seen before, but in a different dimensional space, new classes are dynamically generated. Rather than using a simple distance metric, a new similarity metric, termed "affinity" is used, which implicitly performs a non-linear transform into a surrogate space. The affinity between two samples *i* and *j*, A_{ij} is given in equation 1.

$$A_{ij} = \begin{cases} \alpha - (\beta \Delta D_{ij} + \gamma l_{ij}), \alpha \geq \beta \Delta D_{ij} + \gamma l_{ij} \\ 0, otherwise \end{cases} \quad (1)$$

Where α is a starting constant which also defines the maximum affinity, Δ*D* is the difference in dimensionality between the two vectors, β is a constant which scales the penalty for having different dimensions, l_{ij} is the Euclidean distance between *i* and *j* based on the dimensions which exist in both and γ is a constant defining the weight that distance has on the affinity. In cases where there is no overlap between dimensions, the affinity is automatically set to 0.

In addition to affinity, it is also necessary to include an estimate of the effect an adjustment has. Even if the input vector matches the stored state exactly it would be foolhardy to recommend an adjustment which is known to have a negative effect. To achieve this, when an input vector is assigned to a class, an average of the change in KPI caused by that adjustment is returned as an estimate of the reward. The average is weighted by the Euclidean distance between the recorded samples and the input vector. The estimated reward, R_{ic}, for making the adjustment assigned to class *c* given the input vector *i* is given by Equation 2.

$$R_{ic} = \frac{\sum_{j=0}^{n} \frac{\Delta P_j}{l_{ij}}}{\sum_{j=0}^{n} \frac{1}{l_{ij}}} \quad (2)$$

Where Δ*P* is the difference in KPI caused by making adjustment *c* when in state *j* and *n* is the number of samples collected for class *c*.

In order to make the system as user friendly as possible, rather than displaying two separate figures, the classes are displayed ordered by a third metric, "Applicability". This is simply the product of the affinity and reward for a given class.

Classes are separated into "knowledge objects", software objects that encapsulate the training samples which represent a class and keep track of the difference in KPI caused by the adjustment for each sample. This is done partly for ease of implementation, but also because in the future, new KOs can be developed with other ML algorithms or Expert Systems at their core, and integrated into the KO framework with no changes to the interfaces between components. The final stage of the KO algorithm is a filter which automatically reduces the affinity of objects to zero if they are recommending adjustments to change parameters into a state they are already in.

4 Synthetic Case Study

For confidentiality reasons, it is not possible to present data from real manufacturing devices in this arena. Instead, to illustrate this methodology a synthetic case study was developed, designed to be indicative of real-world examples of using FRAME during manufacturing ramp-up. Throughout the case study, the KO algorithm is pa-

rameterised using α =100, β=10 and γ=1, (illustrative parameterisation identified through trial and error).

Table 1. The steps of the case study. ***Bold Italics*** highlight steps recommended by FRAME.

Step	**P1**	**P2**	**T**	**KPI**
1	0	0	-	60
2	1	0	-	70
3	0	0	-	60
4	0	1	-	72
5	0	0	-	60
6	***0***	***1***	-	***72***
7	***1***	***1***	-	***75***
8	1	1	75	75
9	1	1	0	0
10	1	1	75	75
11	1	1	0	0
12	***1***	***1***	***75***	***75***

The case study is based on a simple machine with two adjustable, binary parameters, P_1 and P_2. It is assumed that the two parameters allow an engineer to move the KPIs of the machine from 60 (P1=0, P2=0) up to 75 (P1=1,P2=1). The two intermediate states of P1=1,P2=0 and P1=0,P2=1 give KPIs of 70 and 72 respectively.

Table 1 represents the case study as a series of parameter changes. The rest of this section is an explanation of how these values were reached using the KO algorithm.

It is assumed that the KPIs of the machine in various states are not known in advance and an engineer is attempting to commission the machine incrementally. Steps 1 through 5 represent the engineer exploring the different states of the parameters, but the optimum state, (P1=1, P2=1) is missed out. The FRAME system captures the information from making those adjustments. At Step 5 FRAME is asked to provide decision support. Table 2 contains the values of the knowledge objects at that step. In this case the highest applicability score is given to changing P2 to 2 as it has the best expected increase in reward and the system state is identical to the state FRAME saw the adjustment in last time. Step 6 shows the engineer making the recommended change. In Step 7 FRAME is asked for support again. In this case the adjustment with the highest applicability is P1 to 1. This simple example demonstrates the Knowledge Object algorithm recommending a state that the system has never seen before, which in this case yields the best result for the system.

In Step 8 a transducer (T) is introduced. It has no effect on the KPIs as it is a passive sensor. As this is a structural change to the system, it is not counted as an adjustment and creates no knowledge object. In Step 9 the transducer falls to 0 as the KPI falls to 0. This is representative of a conveyor belt stall. When the engineer asks for support (see Table 2) in Step 9 all of the knowledge objects have applicabilities of 0 or less, indicating that FRAME cannot help in this situation. Note the fall in affinity as the dimensionality of the current system state is different to the previously encountered states. The engineer loosens the conveyor and records this manually as a fix, creating a knowledge object with different dimensionality to the other objects. Step 11 shows a second conveyor stall. In this case the knowledge object with the highest applicability represents loosening the conveyor, restoring the system to full operation.

Table 2. Knowledge Object States at various steps. A dash (-) represents that the object did not exist at that stage.

Desc	Step 5			Step 6			Step 9			Step 11		
	App	Aff	R	App	Aff	R	App	Aff	R	App	Aff	R
P1 -> 0	0	0	-10	0	0	-10	-8.9	89	-10	-8.9	89	-10
P2 -> 0	0	0	-12	-12	100	-12	-10.68	89	-12	-10.68	89	-12
P1 -> 1	10	100	10	9.9	99	10	0	0	10	0	0	10
P2 -> 1	12	100	12	0	0	12	0	0	12	0	0	12
Loosen Conveyor	-	-	-	-	-	-	-	-	-	75	100	75

5 Conclusions and Discussion

The simple case study presented in Section 4 is intended to illustrate the operation of the proposed KO algorithm and the FRAME system. A more rigorous set of experiments need to be performed before any firm statements can be made about the validity of this technique, however, the nature of such experiments is difficult to ascertain. Tests using random data are not representative of the practical nature of the problem being examined. Tests using a real industrial machine pose confidentiality issues and will be limited to a single case. It is proposed that a simulated machine would allow the system to explore many different, practically-grounded scenarios that will allow fair comparisons to be made with other techniques, such as those presented in [18]. Other challenges for the future include providing KOs with the absolute values of the KPIs for a given state, to prevent large improvements which are only possible when the system has failed, (such as the loosening conveyor system) dominating smaller improvements that allow finer tuning at the upper range of the machine's operation. As the algorithm currently stands, this situation is easily rectified by the engineer by simply performing the erroneous adjustment and providing the system with an example of it yielding no improvement.

Future developments of the FRAME system can be separated into algorithm improvements and FRAME developments. In terms of KO algorithm improvements it may be possible to use a matrix calculation for the γl_{ij} term of the affinity function to allow the effects of distance on affinity to be automatically scaled on a dimension by dimension basis. This will avoid problems when the vector includes both binary parameters and continuous variables with large ranges. In terms of developing the FRAME system, with this framework in place, it is now possible to both merge the results from the KO algorithm with a behavioural model component, allowing more advanced techniques to inform the reward estimates and intelligently interpolate between parameterised adjustments to allow the algorithm to make recommendations that have never been seen before.

6 Acknowledgements

This is research has been funded by the European Commission as part of the FP7 NMP FRAME project (CP-FP 229208-2 FRAME).

7 References

1. Fjällström,S., Säfsten,K., Harlin,U. and Stahre,J. (2009) Information enabling production ramp-up. Journal of Manufacturing Technology Management,20(2): p. 178-196.
2. Ceglarek,D.,Huang,W.,Zhou,S.,Ding,Y.,Kumar,R.,and Zhou,Y.(2004) Time-based competition in multistage manufacturing: Stream-of-variation analysis (SOVA) methodology-Review. Journal of Flexible Manufacturing Systems.16(1): p.11-44.
3. Haller,M.,Peikert,A. and Thoma,J. (2003) Cycle time management during production ramp-up. Robotics and Computer-Integrated Manufacturing.19(1-2): p.183-188.
4. Terwiesch,C. and Bohn, R.E.(2001) Learning and process improvement during production ramp-up. International Journal of Production Economics. 70pp: p. 1-19.
5. Carrillo,J. E., and Franza,R.M. (2006). Investing in product development and production capabilities: The crucial linkage between time-to-market and ramp-up time. European Journal of Operational Research, 171, 536-556.
6. Haller,M., Peikert,A., and Thoma, J. (2003). Cycle time management during production ramp-up. Robotics and Computer-Integrated Manufacturing, 19, 183-188.
7. de Ron, A. J. (1995). Measure of manufacturing performance in advanced manufacturing systems. International Journal of Production Economics, 41, 147-160.
8. Hon, K.K.B. (2005). Performance and evaluation of manufacturing systems. CIRP Annals - Manufacturing Technology, 54, 675-690.
9. Kaplan,R.S., and Norton,D.P. (1992). The balanced scorecard--measures that drive performance. Harvard Business Review, 70, 71-79.
10. Neely,A.,Gregory,M.,and Platts,K. (1995). Performance measurement system design: A literature review and research agenda. International Journal of Operations and Production Management, 15, 80-116.
11. Doltsinis,S., Ratchev,S. and Lohse,N. A Framework for Performance Measurement during Production Ramp-up of Assembly Stations. Submitted to the European Journal of Operational Research
12. Bishop, C.M. (2006), Pattern Recognition and Machine Learning, Springer
13. Voulgaris, Z. and Magoulas, G. D. (2008) Extensions of the k Nearest Neighbour Methods for Classification Problems. In Proc. of 26th IASTED International Conference on Artificial Intelligence and Applications, ACTA, 2328
14. Belongie,S., Malik,J., and Puzicha,J. (2001) Shape Matching and Object Recognition Using Shape Contexts.In IEEE Transactions on Pattern Analysis and Machine Intelligence 24:509-522
15. Kim, S. (2003), Protein β-turn Prediction Using Nearest Neighbour Method. In Bioinformatics 20(1):40-44
16. Basri,R., Hassner, T., and Zelnik-Manor,L. (2011) Approximate Nearest Subspace Search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2):266-278
17. Vapnik, V. and Vashist, A. (2009) A new learning paradigm: learning using privileged information. Neural Netw. 22(5-6):544-57
18. Doltsinis, S.C. and Lohse, N. (2012) A Model-Free Reinforcement Learning Approach Using Monte Carlo Method in Production Ramp-Up Policy Improvement. Submitted to the 14th IFAC Symposium on Information Control Problems in Manufacturing