Reconfigurable Multi-Agent Manufacturing through Deep Reinforcement Learning: A Research Agenda

Mohsen Moghaddam [a], Amro M. Farid [b, c]

[a] *Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115 USA (Tel: 617-373-6256; e-mail: mohsen@northeastern.edu)*
[b] *Thayer School of Engineering, Dartmouth College, Hannover, NH 03755, USA*
[c] *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: amfarid@dartmouth.edu)*

Abstract: Multi-agent systems are a key enabler of reconfigurable manufacturing; a key feature of Industry 4.0 intended to reduce time-to-market and enable mass-individualization of products. While significant contributions have been made to the development of multi-agent paradigms and reference architectures in recent years, there has been a relative absence of methodologies for optimization of agent policies and behaviors. This article formulates a methodological framework based on multi-agent deep Q-networks (DQN), identifies research opportunities and gaps, and proposes an agenda for future investigation on the optimal, data-driven control of multi-agent systems behavior through deep learning.

Keywords: Reconfigurability; Distributed control; Industry 4.0; Reinforcement learning; Deep Q-network

1. INTRODUCTION

Cyber-physical production systems (CPPS) have emerged over the past decade to enable reconfigurable manufacturing through the integration of computing, communication technologies, and artificial intelligence (Wang, Torngren and Onori, 2015; Monostori *et al.*, 2016). The premise is to reduce time-to-market and enable mass-individualization (Koren *et al.*, 2013) through enhanced *reconfigurability* of systems and processes (Farid and Mcfarlane, 2007). A reconfigurable manufacturing system refers to a system "designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or regulatory requirements" (Koren *et al.*, 1999). The evolution of CPPS has thus coincided with the transformation of the topology of manufacturing control architectures from hierarchy (*e.g.*, the automation pyramid) to network (Riedl *et al.*, 2014). CPPS roots in earlier paradigms for multi-agent control of shop-floor systems (Nof, 2007; Monostori *et al.*, 2015), supported by recent communication standards such as OPC-UA (Mahnke, Leitner and Damm, 2011) and reference models such as the Industry 4.0 component (DIN, 2016).

Multi-agent manufacturing has received significant recent attention (Leitão, 2009; Trentesaux, 2009; Monostori *et al.*, 2015). This has led to the emergence of paradigms such as *bionic* (Ueda, 1992), *reconfigurable* (Koren *et al.*, 1999), and *holonic* (Van Brussel *et al.*, 1998) manufacturing, along with several reference architectures such as PROSA (Van Brussel *et al.*, 1998) and ADACOR (Leitão and Restivo, 2006). In spite of great success in system-specific applications, a major limitation that hinders the full potential and practical application of these advances in multi-agent manufacturing is the lack of robust quantitative methods for optimal control of multi-agent manufacturing systems behavior. Although recent studies have developed formal quantitative methodologies for the design of multi-agent system architectures (Farid and Ribeiro, 2015; Dias-Ferreira *et al.*, 2018), the mapping between such architectures and agent behavior has yet to be explored. That is, it is not clear how these architectures translate into distributed and asynchronous interaction and negotiation mechanisms that yield the best performance in terms of reconfigurability (or any other design criterion of interest).

This article investigates the optimal control of agent behavior in multi-agent manufacturing systems. First, the problem is formalized as a multi-agent reinforcement learning problem, and the limitations of current methods in this area are discussed (Section 2). A deep Q-network (DQN) formalism is then proposed to address the outlined limitations of current methods (Section 3), followed by a summary of challenges and an agenda for future research (Section 4).

2. MULTI-AGENT SYSTEM BEHAVIOR

The design of a multi-agent system is typically comprised of five stages (Farid and Ribeiro, 2015): (1) paradigm and high-level design principles (*e.g.*, holonic manufacturing), (2) reference architecture (*e.g.*, PROSA), (3) system-specific architecture, (4) multi-agent system behavior control, and (5) implementation. While extensive work has been done with regard to Stages 1-3, devising principles and mechanisms to enable optimal agent behavior and interaction (*i.e.*, Stage 4) is still an unsolved problem (Monostori *et al.*, 2015; Nof *et al.*, 2015). "Optimality" in this context can be formalized through Markov decision process (MDP):

> For an agent in state s, what is the optimal policy π^* that leads to the selection of action a, which in turn maximizes the expected reward r?

The problem of modeling agent behavior has been approached from different angles over the past few decades (Baker, 1998;

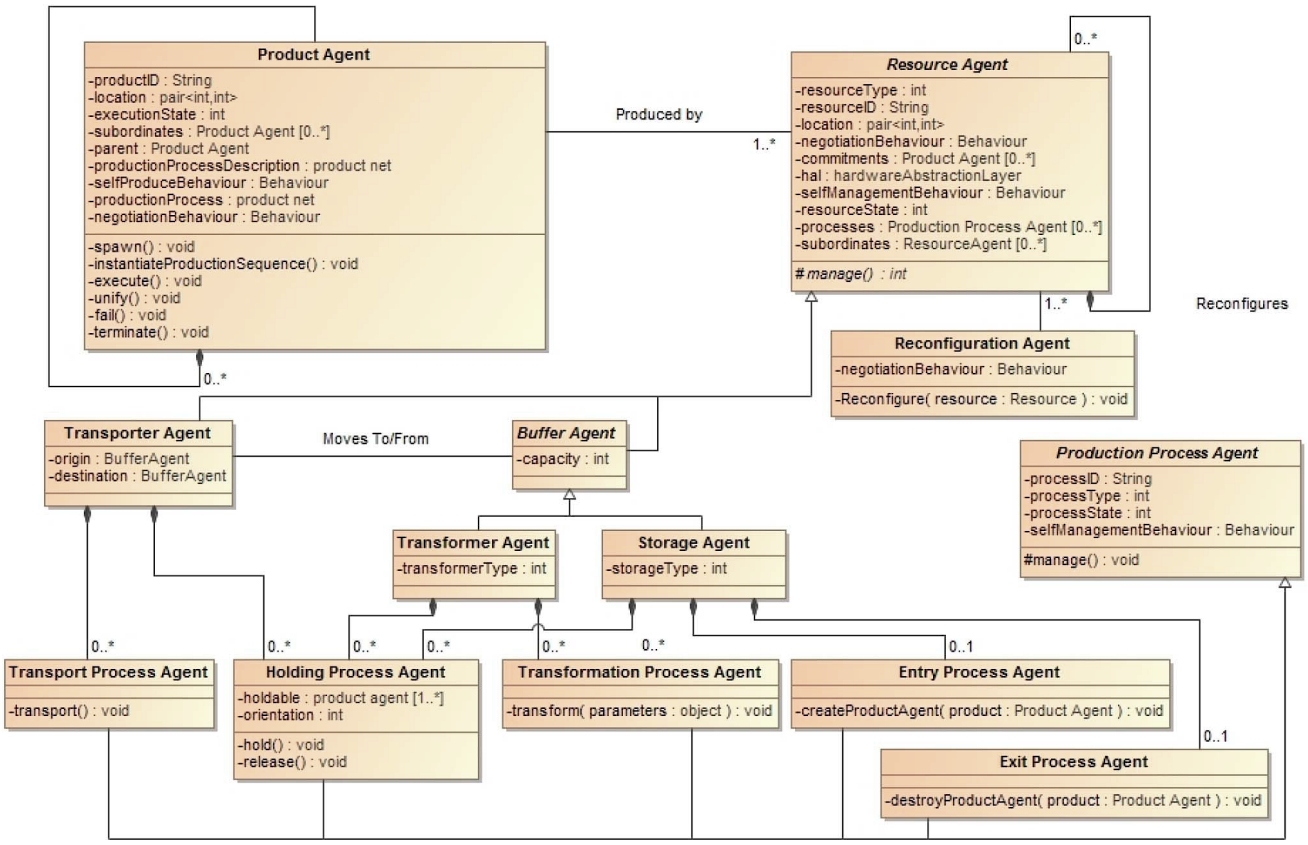

Fig. 1. The ADMARMS architecture (Farid and Ribeiro, 2015).

Shen and Norrie, 1999; Nof, 2007; Trentesaux, 2009; Monostori *et al.*, 2015). Examples include market-based design (Márkus, Kis Váncza and Monostori, 1996), cooperative and consensus-seeking games (Saber and Murray, 2003; Shamma, 2007), and bio-inspired swarm intelligence (Hadeli *et al.*, 2004; Dias-Ferreira *et al.*, 2018), among others. In spite of remarkable contributions to system-specific applications, none of these developments fully addresses the challenge of performance guarantee that stems from the "myopic behavior" of agents (Trentesaux, 2009). That is, given the autonomy of agents as well as their lack of global perspective, it is hard to guarantee that a chosen state-action pair (s, a) would lead to the performance (*i.e.*, reward r), from both local and system-level perspectives.

Consider tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ where the elements respectively denote the set of possible states, the set of possible actions, the expected reward function of each possible state-action pair, the transition probability between states, and a discount factor. Considering the current state $s \in \mathcal{S}$ and the action taken $a \in \mathcal{A}$, the agent receives reward $r \sim \mathcal{R}(.|s, a)$ and next state $s' \sim \mathcal{P}(.|s, a)$. A policy π is defined as a mapping from $\mathcal{S}$ onto $\mathcal{A}$ which specifies the action to be taken at any given state, *i.e.*, $a \sim \pi(.|s)$. An optimal policy π^* is thus a policy that maximizes the total discounted reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_t (\gamma^t r_t | \pi)\right], \quad (1)$$

where t denotes the iteration index. This problem is referred to as reinforcement learning (Sutton and Barto, 1998; Szepesvári, 2010), which involves an agent interacting with a stochastic environment that provides feedback to the agent's actions in the form of reward signals. In order to discuss how reinforcement learning may contribute to resolving the aforementioned issues related to Stage 4 (*i.e.*, multi-agent system behavior control), besides providing formal definitions of agent, state, action, reward, policy and other design considerations in the context of manufacturing, a generalization to multi-agent reinforcement learning is required.

The generalization to multi-agent MDP can be recast as a *stochastic game* (Busoniu, Babuska and Schutter, 2008) in which given a shared state $\mathcal{S}$ and transition probability $\mathcal{P}$, each agent i $(i = 1, \dots, n)$ operates with a different set of actions $\mathcal{A}_i$ and expected reward function $\mathcal{R}_i$. Accordingly, a fully-cooperative environment implies $\mathcal{R}_1 = \mathcal{R}_2 = \cdots = \mathcal{R}_n$ while in a fully-competitive environment, $\sum_i \mathcal{R}_i = 0$. A CPPS typically lies somewhere in the cooperation-competition spectrum. That is, despite the cooperative nature of shop floor components, for example, they sometimes compete for limited resources due to certain conflicts between their individual goals. Among the challenges of multi-agent reinforcement learning are poor scalability and the curse of dimensionality along with the non-stationarity of policies due to the inherent interdependencies between the agents' actions and rewards. Next section elaborates on this problem in the context of CPPS and outlines potential solutions based on the emerging notion of DQN (Mnih *et al.*, 2015) and its extensions to multi-agent environments.

3. DEEP Q-NETWORK FOR OPTIMAL CONTROL OF AGENT BEHAVIOR

In this section, a formalism of multi-agent manufacturing systems is provided, the challenges of distributed control through reinforcement learning are further elaborated, and a framework based on multi-agent DQN is proposed to potentially address the identified gaps associated with agent behavior optimization and performance guarantees.

3.1 Multi-Agent System Architecture for Reconfigurable Manufacturing

A reference architecture for axiomatic design of a multi-agent reconfigurable mechatronic systems (ADMARMS) is depicted in Fig. 1 (Farid and Ribeiro, 2015). To recast the problem of control of agent behavior as a reinforcement learning problem, the design elements of reinforcement learning are defined in this section based on the ADMARMS formalism. Interested readers are referred to (Farid and Ribeiro, 2015) for details on ADMARMS.

Within the domain of manufacturing execution system and shop-floor control (*i.e.*, Levels 0-3 of ISA-95), agents can be classified into two major categories:

- *Manufacturing resources*. Each physical manufacturing resource is associated with a resource agent (RA) as its cyber mirror image. This class of agents corresponds with the role-based equipment hierarchy model of ISA-95 (IEC, 2013). In the Industry 4.0 standard (DIN, 2016), RA may refer to the administration shell of an I4.0 (Industry 4.0) component which is responsible for virtual representation, interaction with the system, and resource management. Resource agents are classified into three main categories of transformer agent (*e.g.*, a CNC machine), transporter agent (*e.g.*, an AGV), and storage agent (*e.g.*, a smart pallet), which are also further classified into subcategories (see Fig. 1).

- *Manufacturing processes*. Each manufacturing process, a logical grouping of functions with specified outcomes, is associated with a process agent (PA) as its informatic counterpart. This class of agents corresponds with the functional hierarchy model of ISA-95 (IEC, 2013), especially the Level 3-0 activities. The processes range from workflow control, detailed scheduling, and reliability assurance (Level 3) to supervisory control sensing and process manipulation, and physical manufacturing processes (Levels 2-0). Similar to resource agents, process agents are also classified into categories such as entry agent, exist agent, transformation process agent, and transportation process agent (see Fig. 1).

Considering the *independence axiom* (Suh, 2001), which characterizes the mutual exclusiveness and collective exhaustiveness of manufacturing processes, each match between a pair of resource and process is regarded as an event describing a distinct step in the production process (Farid and Ribeiro, 2015). This matching process can be represented using a *manufacturing system knowledge base* (Farid and McFarlane, 2008), a binary matrix $\mathbf{K}$ of size $n(PA) \times n(RA)$, where $K[i,j] = 1$ if process agent $i \in PA$ is matched to resource agent $j \in RA$, and $K[i,j] = 0$, otherwise. PA and RA denote the sets of process agents and resource agents, respectively, and $n(\cdot)$ measures the cardinality of a set.

The following definitions result from this formalism. The state of process agent i (resource agent j) is determined by the i-th row (the j-th column) of the manufacturing system knowledge base $\mathbf{K}$. In case of unavailability of an agent due to error or conflict, the elements of its respective row or column in $\mathbf{K}$ become zero during the unavailability period, in accordance with the *scleronomic constraints matrix* of ADMARMS (Farid and Ribeiro, 2015). Further, an action is attributed to a pair of process and resource agents $(i,j) \in RA \times PA$, if $K[i,j] \neq K'[i,j]$, where $'$ denotes the next iteration. The expected reward of each state-action pair for an agent equals the resulting expected changes in a set of performance metrics (*e.g.*, makespan), considering the current states and actions of other agents coexisting in the same multi-agent system.

3.2 DQN

Given the reinforcement learning formulation of the problem, the optimal policy π^* (see Eq. 1) for each agent i that would individually maximize the expected reward r_i for a state-action pair (s_i, a_i) can be obtained by solving the *Bellman equation* (Szepesvári, 2010):

$$Q^*(s_i, a_i) = \mathbb{E}_{e_i \sim \mathcal{D}_i}\left[r_i + \gamma \max_{a_i'} Q^*(s_i', a_i') \mid s_i, a_i \right], \quad \forall i. \tag{2}$$

$\mathcal{D}_i = \{e_{i1}, \dots, e_{im}\}$ denotes the *experience replay* dataset (Mnih *et al.*, 2015); the m most recent experiences of agent i, where $e_i = \langle s_i, a_i, r_i, s_i' \rangle$. Due to the randomness associated with the MDP, Eq. (2) aims at maximizing the expected cumulative rewards in transitioning from one state to the next. The optimal policy π^* is thus achieved by taking the best action at any given state. The challenging, however, is to estimate Q^* considering the large state-spaces in a manufacturing environment. Deep learning can be applied as a powerful tool for approximation of the Q-function $Q^*(s,a)$. The application of deep learning in this context is referred to as deep reinforcement learning (Mnih *et al.*, 2015; Li, 2017). Accordingly, the Q-function can be approximated by a deep neural network characterized by weight parameters θ_x as $Q(s,a;\theta_x)$ where x denotes iteration. The forward propagation thus involves calculating a loss function for each training iteration x as:

$$L_{xi}(\theta_{xi}) = \mathbb{E}_{e_i \sim \mathcal{D}_i}[(r_i + \gamma \max_{a_i'} Q(s_i', a_i'; \theta_{xi}^-) - Q(s_i, a_i; \theta_{xi}))^2], \tag{3}$$

where θ_{xi} and θ_{xi}^- are respectively the weight parameters of the Q-network and a target network. Backpropagation is thus conducted by minimizing the loss function with respect to parameters θ_{xi} by solving for the gradient function $\nabla_{\theta_{xi}} L_{xi}(\theta_{xi})$. Learning can take place following an ε-greedy policy, which applies the DQN greedy policy with probability $1-\varepsilon$, and applies a random action with probability ε. DQN and its extensions have shown remarkable, human-level performance at tasks such as playing Atari games (Mnih *et al.*, 2015).

However, these developments have been limited to single-agent reinforcement learning ($i = 1$). Further, and their generalization to multi-agent environments, especially in mixed environments where both cooperation and competition are present (*i.e.*, $\exists\, i, j, \mathcal{R}_i \neq \mathcal{R}_j$) is a challenging task (Castañeda, 2016; Egorov, 2016; Gupta, Egorov and Kochenderfer, 2017).

3.3 Multi-Agent DQN

Generalization to multi-agent reinforcement learning ($i > 1$) requires reformulation of the DQN with respect to the characteristics of the stochastic game between the agents (Busoniu, Babuska and Schutter, 2008). Therefore, in addition to learning the structure of the environment, agents also have to learn about the existence, actions, and goals of other agents (Castañeda, 2016). In the case of absolute cooperation, agents can independently optimize their own policies while collectively maximizing a joint reward function and possibly sharing their policy parameters with other agents of the same class (Gupta, Egorov and Kochenderfer, 2017). In competitive or mixed scenarios, however, joint strategies must be obtained in various fashions such as the minimax principle, stochastic game Nash equilibrium, and Win-or-Learn-Fast Policy Hill-Climbing, among others (Busoniu, Babuska and Schutter, 2008). Nevertheless, the challenges associated with the interdependencies between agent actions, policies, and rewards and the non-stationarity of policies have yet to be explored in the context of multi-agent DQN (Castañeda, 2016). Further, in spite of some recent success in multi-agent DQN such as learning to play the game of Pong (Tampuu *et al.*, 2017), the scalability to nontrivial numbers of agents is identified as a challenging research problem in multi-agent DQN.

4. CONCLUSION AND DISCUSSION

The manufacturing industry and research community can leverage two unprecedented opportunities to tackle the challenging problem of *optimal* multi-agent control realize reconfigurable manufacturing systems. First is data availability. One of the few limitations of deep learning is the dependency on the size of training data. In DQN, that implies the availability of data on the reward $r \sim \mathcal{R}(.\,|\,s, a)$ and transition probability $s \sim \mathcal{P}(.\,|\,s, a)$ associated with each state-action pair, considering current state-action pairs of other agents in the same environment. That necessitates the availability of a massive dataset that could not have been captured a few years ago with acceptable frequency and fidelity. The extensive adoption of sensing and communication technologies in shop-floor systems, however, is changing this game. Second is the rapid growth in the areas of cloud computing and deep learning. Traditionally, *scalability* has been a major limitation of MDP-based reinforcement learning methods, simply because iteratively searching the enormous state-action space for a multi-agent system is practically impossible. To capture the magnitude, consider a simple example: "*one hundred agents, each with ten behaviors, require the programming of only 1000 individual behaviors, yet provide a behavior space on the order of 10^{100}, a number far larger than the total number of elementary particles in the universe*" (Van Dyke Parunak, 1997). The rapid advances in cloud-based platforms and compute services (*e.g.*, AWS, Azure) together with the recent developments in the field of deep learning can potentially address these scalability challenges.

An interesting property of Q-learning for policy optimization is that it only depends on the immediate reward assigned to a state-action pair of the agent. Although the reward may be influenced by the environment, the states/actions of other agents, or other factors, the data-driven nature of Q-learning eliminates the need for incorporating such factors in the model. That is, the "decision myopia", or the lack of global view of an agent, does not affect the learning performance. The challenge, however, is to devise a mechanism to simultaneously optimize the policies of a large number of agents using DQN. Future investigations will be centered on formal mapping of the AD-MARMS onto a multi-agent MDP framework and developing a multi-agent DQN methodology to optimize the policies of manufacturing agents (*e.g.*, resources; processes; products) in a mixed cooperative-competitive environment.

Another challenge is the need for experimental validation. The performance of a distributed control method cannot be fully demonstrated and validated unless implemented and tested on a real testbed. That is one of the major barriers to industrial adoption of multi-agent manufacturing systems. Future investigations will thus be built upon a physical testbed to experiment and validate the optimal control via multi-agent DQN and compare the performance with other existing methodologies for distributed control of multi-agent manufacturing systems.

5. REFERENCES

Baker, A. D. (1998) 'A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling, and Pull', *Journal of Manufacturing Systems*, 17(4), pp. 297–320. doi: 10.1016/S0278-6125(98)80077-0.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998) 'Reference architecture for holonic manufacturing systems: PROSA', *Computers in Industry*, 37(3), pp. 255–274. doi: 10.1016/S0166-3615(98)00102-X.

Busoniu, L., Babuska, R. and Schutter, B. De (2008) 'A Comprehensive Survey of Multiagent Reinforcement Learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* 38(2), pp. 156–172. doi: 10.1007/978-94-007-1162-4.

Castañeda, A. O. (2016) *Thesis: Deep Reinforcement Learning Variants of Multi-Agent Learning Algorithms*. University of Edinburgh. doi: 10.1016/j.scienta.2006.03.011.

Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P. and Onori, M. (2018) 'BIOSOARM : a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors', *Journal of Intelligent Manufacturing*. Springer US, 29, pp. 1659–1682. doi: 10.1007/s10845-016-1258-2.

DIN (2016) *DIN SPEC 91345:2016-04, Reference Architecture Model Industrie 4.0 (RAMI4.0)*. Berlin.

Van Dyke Parunak, H. (1997) 'Go to the ant: Engineering principles from natural multi-agent systems', *Annals of Operations Research*. Kluwer Academic Publishers, 75(0), pp. 69–101. doi: 10.1023/A:1018980001403.

Egorov, M. (2016) *Multi-Agent Deep Reinforcement Learning*. doi: 10.1109/IJCNN.2010.5596468.

Farid, A. M. and Mcfarlane, D. C. (2007) 'A Design Structure Matrix Based Method for Reconfigurability Measurement of Distributed Manufacturing Systems', *International Journal of Intelligent Control & Systems*, 1(1), pp. 1–12.

Farid, A. M. and McFarlane, D. C. (2008) 'Production degrees of freedom as manufacturing system reconfiguration potential

measures', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. SAGE PublicationsSage UK: London, England, 222(10), pp. 1301–1314.

Farid, A. M. and Ribeiro, L. (2015) 'An Axiomatic Design of a Multiagent Reconfigurable Mechatronic System Architecture', *IEEE Transactions on Industrial Informatics*. IEEE, 11(5), pp. 1142–1155. doi: 10.1109/TII.2015.2470528.

Gupta, J. K., Egorov, M. and Kochenderfer, M. (2017) 'Cooperative Multi-agent Control Using Deep Reinforcement Learning', in Rodriguez-Aguilar and A., G. S. J. (eds) *AAMAS 2017 Best Papers*, pp. 66–83. doi: 10.2174/978160805824211401000 3.

Hadeli, Valckenaers, P., Kollingbaum, M. and Van Brussel, H. (2004) 'Multi-agent coordination and control using stigmergy', *Computers in Industry*. Elsevier, 53(1), pp. 75–96. doi: 10.1016/S0166-3615(03)00123-4.

IEC (2013) *IEC 62264:2013: ISA95 – Enterprise-Control System Integration*.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H. (1999) 'Reconfigurable Manufacturing Systems', *CIRP Annals - Manufacturing Technology*, 48(2), pp. 527–540. doi: 10.1016/S0007-8506(07)63232-6.

Koren, Y., Hu, S. J., Gu, P. and Shpitalni, M. (2013) 'Open-architecture products', *CIRP Annals - Manufacturing Technology*, 62(2), pp. 719–729. doi: 10.1016/j.cirp.2013.06.001.

Leitão, P. (2009) 'Agent-based distributed manufacturing control: A state-of-the-art survey', *Engineering Applications of Artificial Intelligence*. Pergamon, 22(7), pp. 979–991. doi: 10.1016/J.ENGAPPAI.2008.09.005.

Leitão, P. and Restivo, F. (2006) 'ADACOR: A holonic architecture for agile and adaptive manufacturing control', *Computers in Industry*. Elsevier Science Publishers B. V., 57(2), pp. 121–130. doi: 10.1016/j.compind.2005.05.005.

Li, Y. (2017) 'Deep Reinforcement Learning: An Overview'. Available at: http://arxiv.org/abs/1701.07274 (Accessed: 27 January 2019).

Mahnke, W., Leitner, S.-H. and Damm, M. (2011) *OPC Unified Architecture*. Springer-Verlag Berlin Heidelberg.

Márkus, A., Kis Váncza, T. and Monostori, L. (1996) 'A Market Approach to Holonic Manufacturing', *CIRP Annals*. Elsevier, 45(1), pp. 433–436. doi: 10.1016/S0007-8506(07)63096-0.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015) 'Human-level control through deep reinforcement learning', *Nature*. Nature Publishing Group, 518(7540), pp. 529–533. doi: 10.1038/nature14236.

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. and Ueda, K. (2016) 'Cyber-physical systems in manufacturing', *CIRP Annals - Manufacturing Technology*, 65(2), pp. 621–641. doi: 10.1016/j.cirp.2016.06.005.

Monostori, L., Valckenaers, P., Dolgui, A., Panetto, H. and Brdys, M. (2015) 'Cooperative control in production and logistics', *Annual Reviews in Control*, 39, pp. 12–29. doi: 10.1016/j.arcontrol.2015.03.001.

Nof, S. Y. (2007) 'Collaborative control theory for e-Work, e-Production, and e-Service', *Annual Reviews in Control*, 31, pp. 281–292.

Nof, S. Y., Ceroni, J., Jeong, W. and Moghaddam, M. (2015) *Revolutionizing Collaboration through e-Work, e-Business, and e-Service*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Riedl, M., Zipper, H., Meier, M. and Diedrich, C. (2014) 'Cyber-physical systems alter automation architectures', *Annual Reviews in Control*. Elsevier Ltd, 38(1), pp. 123–133. doi: 10.1016/j.arcontrol.2014.03.012.

Saber, R. O. and Murray, R. M. (2003) 'Consensus protocols for networks of dynamic agents', in *Proceedings of the 2003 American Control Conference, 2003.* IEEE, pp. 951–956. doi: 10.1109/ACC.2003.1239709.

Shamma, J. S. (ed.) (2007) *Cooperative Control of Distributed Multi-Agent Systems*. Chichester, UK: John Wiley & Sons, Ltd. doi: 10.1002/9780470724200.

Shen, W. and Norrie, D. H. (1999) 'Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey', *Knowledge and Information Systems*. Springer-Verlag, 1(2), pp. 129–156. doi: 10.1007/BF03325096.

Suh, N. P. (2001) *Axiomatic design : advances and applications*. New York: Oxford University Press. Available at: https://www.worldcat.org/title/axiomatic-design-advances-and-applications/oclc/44128159 (Accessed: 26 January 2019).

Sutton, R. S. and Barto, A. G. (1998) *Reinforcement learning : an introduction*. MIT Press. Available at: https://mitpress.mit.edu/books/reinforcement-learning (Accessed: 26 January 2019).

Szepesvári, C. (2010) *Algorithms for Reinforcement Learning*, *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers. doi: 10.2200/S00268ED1V01Y201005AIM009.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J. and Vicente, R. (2017) 'Multiagent cooperation and competition with deep reinforcement learning', *PLoS ONE*, 12(4), pp. 1–12. doi: 10.1371/journal.pone.0172395.

Trentesaux, D. (2009) 'Distributed control of production systems', *Engineering Applications of Artificial Intelligence*. Elsevier, 22(7), pp. 971–978. doi: 10.1016/j.engappai.2009.05.001.

Ueda, K. (1992) 'A Concept for Bionic Manufacturing Systems Based on DNA-type Information', *Proceedings of the IFIP TC5 / WG5.3 Eight International PROLAMAT Conference on Human Aspects in Computer Integrated Manufacturing*. North-Holland, pp. 853–863.

Wang, L., Torngren, M. and Onori, M. (2015) 'Current status and advancement of cyber-physical systems in manufacturing', *Journal of Manufacturing Systems*. The Society of Manufacturing Engineers, 37, pp. 517–527. doi: 10.1016/j.jmsy.2015.04.008.