32nd European Modeling & Simulation Symposium
17th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-44-7 © 2020 The Authors.
DOI: 10.46354/i3m.2020.emss.002

A simulation of autonomous robot movement directed by reinforcement learning

Andrew Greasley

Aston University, Birmingham, B4 7ET, UK

Email address: a.greasley@aston.ac.uk

Abstract

As companies embrace Industry 4.0 and embed intelligent robots and other intelligent facilities in their factories, decision making can be derived from machine learning algorithms and so if we are to simulate these systems we need to model these algorithms too. This article presents a discrete-event simulation (DES) that incorporates the use of a reinforcement learning (RL) algorithm which determines an approximate best route for robots in a factory moving from one physical location to another whilst avoiding collisions with fixed barriers. The study shows how the object oriented and graphical facilities of an industry ready commercial off-the-shelf (COTS) DES software package enables an RL capability without the need to use program code or require an interface with external RL software. Thus the article aims to contribute to the methodology of simulation practitioners who wish to implement AI techniques as a supplement to their input modelling approaches.

Keywords: discrete-event simulation; reinforcement learning; autonomous robots

1. Introduction

Autonomous robots which can perform complex tasks are one of the key elements of Industry 4.0 (Rüsmann et al., 2015) and it is expected that Artificial Intelligence (AI) will be used to facilitate the 'intelligent' capabilities of these robots. Reinforcement learning (RL) is a subfield of machine learning (ML) that uses algorithms that explore options and when they achieve their aim, deduce how to get to that successful endpoint in the future. This approach can be implemented by the use of a reward and penalty system to guide a choice from a number of random options. Hosokawa et al. (2014) discuss the difficulty of designing controllers for autonomous mobile robots which can be adaptable to different environments and propose the use of reinforcement learning which makes it possible to automatically acquire a robot controller only from the results of the robots behaviour and thus does not require detailed teaching signals by a human. Benotsmane et al. (2019) discuss the role of simulation tools and AI to improve the efficiency of the smart factory. Simulation is used to predict, evaluate and validate system behaviour to transform the basic concepts of the smart factory into reality. However if we wish to model and predict the performance of systems incorporating AI techniques such as reinforcement learning we will need to incorporate RL algorithms within our simulations. A widely used simulation technique is discrete-event simulation (DES) which Law (2015) defines as "the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time. These points in time are the ones at which an event occurs, where an event is defined as an instantaneous occurrence that may change the state of the system." DES is the most used operational research technique in practice (Brailsford et al., 2014) with Rockwell Automation claiming that 52,000 students are trained in its Arena DES software each year (www.arenasimulation.com).

Elbattah and Molloy (2018) state that simulation modelling and machine learning although long-established have both tended to be employed in separate territories with limited, if any, integration which might help simulations attain a higher level of model realism. Creighton and Nahavandi (2002) state that an obstacle to using RL in conjunction with DES is the challenge in interfacing the agent with commercial of the shelf (COTS) simulation software. Currently the main approach for doing this is to use the library-based application programming interfaces (APIs) provided in COTS DES packages. For example the C interface of the Tecnomatix Plant Simulation can be used to access the functions of MatLab (Bergmann et al., 2017), the Simio software offers Visual C# user extensions in areas such as user defined model selection rules and AnyLogic offers Java user extensions that can make use of Java-based libraries such as Deeplearning4j (https://deeplearning4j.org/). However in the light of the development of drag and drop interfaces in such tools as Arena, DES users may find it a particular challenge to adapt to the need for program coding when developing a machine learning algorithm. Thus in order to ensure that the large industrial user base of COTS DES software (such as Arena, Simio and Witness) are able to implement this capability we would like to employ the current facilities provided by COTS DES software without recourse to programming code such as Java or requiring an interface with external RL software. This article presents the use of a DES that incorporates the use of a reinforcement learning algorithm which determines an approximate best route for an autonomous robot in a factory moving from one physical location to another whilst avoiding collisions with fixed barriers. The capability is implemented entirely within the COTS DES software Simio v11 (Smith et al., 2018) using the software's standard process logic facilities.

The article is organized as follows. A literature review provides an overview of the context of the problem in terms of the Industry 4.0 concept and the role of autonomous robots within that initiative. This study involves the use of simulation to model decision processes, in this case those involved in directing robot movement. The literature review covers the use of simulation to model decision processes using the approaches of machine learning and reinforcement learning. A case study is then presented of the use of the q-learning RL algorithm to enable the robot to learn the approximate best route between two locations in a factory.

2. Literature Review

A review is undertaken to show the increased relevance of the use of autonomous robots in the context of Industry 4.0. The remainder of the literature review will thus cover the use of the AI techniques of machine learning and reinforcement learning to enable autonomous decision making by robots. The focus of this study refers to the decision processes undertaken by a robot when moving between locations so the use of DES to model decision processes using these AI techniques will be considered.

2.1. Industry 4.0

The Industry 4.0 initiative supports the view that interoperability, virtualization, decentralisation, real-time capability and modularity must be present in the production systems of the future (Benotsmane et al., 2019). These features are based on the pillars of the Industry 4.0 concept the first of which is summarised by Rüsmann et al. (2015) as Multi-Agent Systems (MAS) which can be intelligent smart machines, collaborating robots, sensors, controllers etc., that are communicating with the production control system and the smart workpieces so that machines coordinate, control, and optimise themselves and the whole production process. The two main elements of multi-agent systems are described as Autonomous Robots which can cooperate and collaborate with each other during manufacturing in order to perform more complex tasks with higher efficiency and Artificial Intelligence which is the ability of robots to learn and think logically and autonomously, not only depending on the programs written by people. Thus autonomous robots are both seen as one of the pillars of the Industry 4.0 concept in terms of the implementation of multi-agent systems and a technological component that is seen as one of the key parts of Industry 4.0 success. A further pillar are simulation tools and these are considered as a key component for the success of Industry 4.0 (Posada et al., 2015).

2.2. DES and Reinforcement Learning

Reinforcement learning (RL) can be classified as a third paradigm of machine learning, not within the supervised and unsupervised learning categories, but as a technique that looks to maximise a reward signal instead of trying to find hidden structure. Thus RL can be considered a type of machine learning in which the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them (Sutton and Barto, 2018). RL algorithms are relevant when both the input variable and output variable are uncertain and are also used in sequential decision making scenarios (Kumar, 2017). This makes them relevant for problems when moving between two previously unknown locations in a number of sequential movements. However learning algorithms in general face a dilemma in that they seek to learn action values conditional on subsequent optimal behaviour, but they need to behave non-optimally in order to explore all actions (to find the optimal actions). To achieve this an 'off-policy' strategy uses two approaches, one that is learned about and becomes the optimal policy, and one that is more exploratory and is used to generate behaviour. The policy being learned about is called the target policy, and the policy used to generate behaviour is called the behaviour policy (Sutton and Barto, 2018). In terms of DES and reinforcement learning, Creighton and Nahavandi (2002) use DES with RL to identify optimal operating policies that minimise the cost of operation of a multi-part serial production line. The RL was implemented in MatLab with communication over a server to the DES software. Waschneck et al. (2018) outline an application targeted at the Industrie 4.0 vision for production control of a decentralised, self-learning and self-optimising system. Here they apply deep reinforcement learning using deep neural networks implemented using the discrete event facilities of MatLab and the Google DeepMind DQN agent.

2.3. DES, Robots and Reinforcement Learning

General guidance on using DES to model robots in the form of automatic guided vehicle systems (AGVS) is provided in Harrell and Tumay (1995). Studies that have taken place which use DES to model robots in an industrial setting include He et al. (2016) who use simulation to investigate robot scheduling in a flexible manufacturing system (FMS). Ono and Ishigami (2019) simulate multiple mobile robots in a warehouse that receive a shipping list that specifies the number of different shipping destinations, the number of product items for each destination, and stock locations of the items in the warehouse. In terms of the use of reinforcement learning to inform the movement of autonomous robots, Khare et al. (2018) present the use of reinforcement learning to move a robot to a destination avoiding both static and moving obstacles. Chewu and Kumar (2018) show how a modified Q-learning algorithm allowed a mobile robot to avoid dynamic obstacles by re-planning the path to find another optimal path different from the previously set global optimal path. Troung and Ngo (2017) show how reinforcement learning can incorporate a Proactive Social Motion Model that considers not only human states relative to the robot but also social interactive information about humans.

3. The Simulation Study

The aim of the simulation study is to implement a RL algorithm to train and test an autonomous robot moving around a factory. A traditional approach to controlling the movement of entities in a DES would be to repeatedly assess the Euclidean distance as the robot progresses towards a target location. However the use of RL offers the potential to provide a more efficient path between locations by considering a strategy for traversing the path in one go rather than moving in the general direction of the target to only be obstructed by barriers within the factory.

A simulation model was built using the Simio v11 discrete-event simulation software using an object oriented approach to modelling. Entities can have their own behaviour and make decisions and these capabilities are achieved with the use of an entity token approach. Here a token is created as a delegate of the entity to execute a process. Processes can be triggered by events such as the movement of entities into and out of objects or by other processes. In this case, what Simio terms 'Add-on Processes', are incorporated into the ModelEntity1 (Robot) object definition allowing data and processes to be encapsulated within each entity definition within the simulation. This means each entity (robot) runspace object simulated in the model will have its own process execution and data values associated with it. Most DES software packages allow data to be associated with an entity (through what are usually termed attribute values) but do not provide the ability to embed process definitions within the entity. The process logic (algorithms) for the simulation contained in the add-on processes train and move each robot through a number of predefined pick and deliver locations.

The main method used for validation of the RL algorithms in Simio was to project the q value transition matrix for a robot on to the Simio animation display of the factory (figure 1). The user can then observe the q-values updating on each learning pass and confirm the path derived from the RL algorithm and to ensure that the robot moves along this path.

With validation complete the model can be run with the training mode operating in the background and the animation showing the movement of the trained robots between pick and deliver stations. Figure 2 shows the model running with 2 autonomous robots with each robot moving in response to its individual schedule of pick and deliver stations and transition matrix of q values. In terms of performance, currently when running the simulation for each robot when a movement has been completed, the RL algorithm is

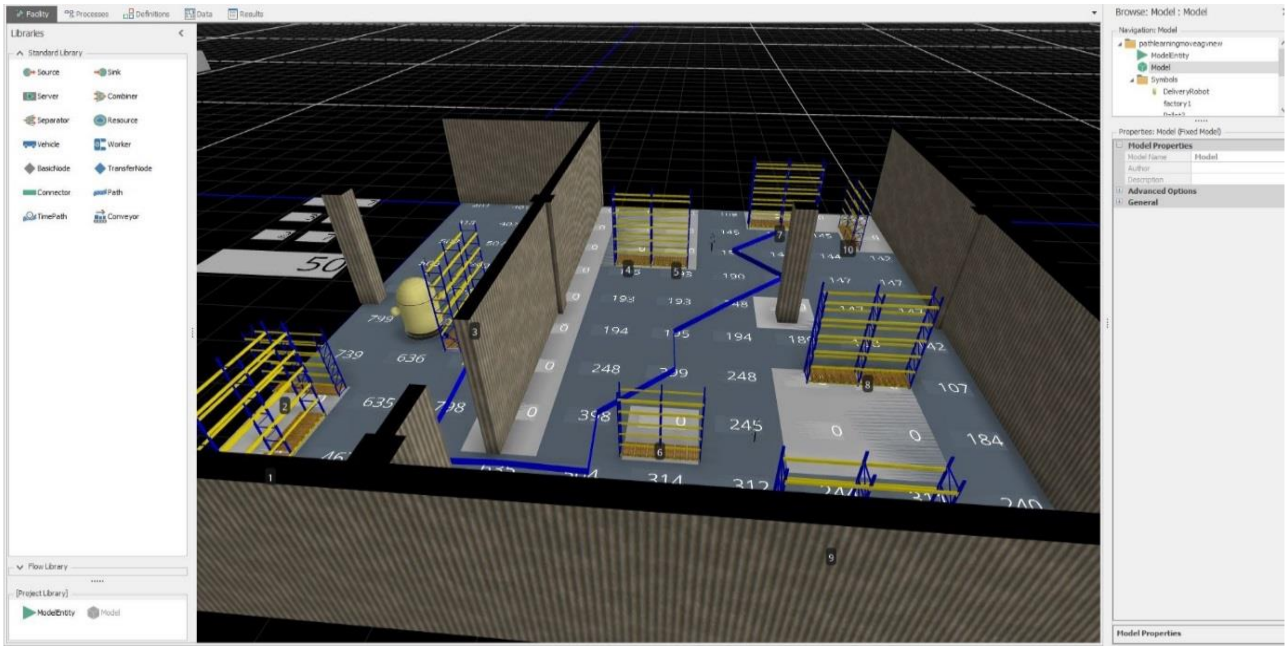

Figure 1. Robot route from station 7 to station 3 after training using RL. The grid cells show the current q values and the line shows the path taken by the robot

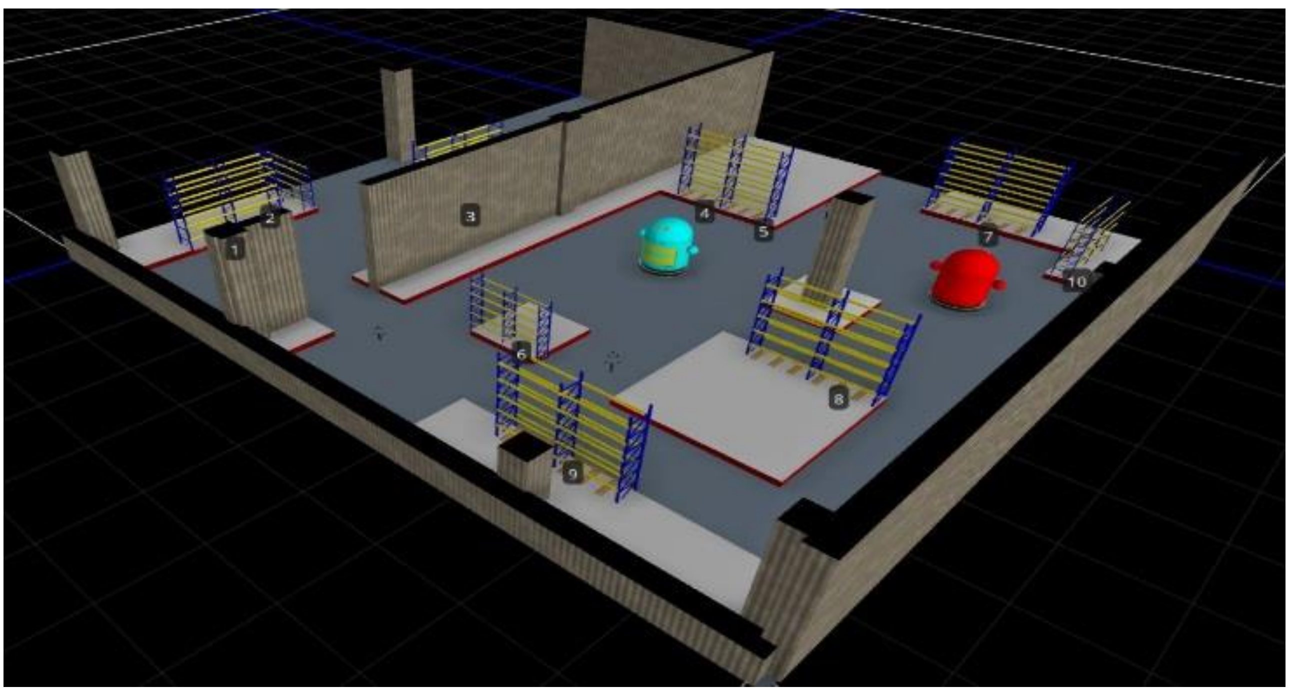

Figure 2. Simio display of 2 robots moving using RL between pick and deliver locations

executed in training mode to find the 'best' path to the next destination. In the current layout configuration a processing delay of around 1 second was apparent when running the simulation in animation mode on a Lenovo ThinkPad with an Intel Core i7-6500U CPU @ 2.50 GHZ with 8.00GB RAM. This delay time could increase with a larger grid size, more complex layout design or increased learning passes. This issue however only affects the smoothness of the animation display as simulation time is not progressed during the training phase. Thus when the simulation is run in fast-forward mode (without animation) for the compilation of results then the delay has only a small effect on runtime speed.

Further research is possible in terms of the operation of the RL algorithm by investigating the effect of adjusting the discount factor and number of learning passes on the generation of an approximate best route strategy. In terms of the model design the effect of robot travel speed (which can vary according to loading) and the incorporation of acceleration and deceleration of the robot could be investigated. The model could also be developed to incorporate collision detection with dynamic (moving objects) as although the method of pre-computing paths avoids the problem of incremental planning in a complex layout there is still a requirement for checking at each robot move for other moving objects such as other robots or people. There are a number of ways of doing this, for example Klass et al. (2011) put forward three rules to prevent collision between 2 AGVs when they get into proximity. The model could also be tested for larger industrial applications that require a larger gridworld, a greater number of robots and thus place higher processing demands on the simulation.

4. Discussion

DES represents a mature tool using a graphical interface to produce an industry strength process modelling capability. To reflect thus maturity this investigation covers the use of a commercial off-the-shelf (COTS) DES software which provides a relatively fast and easy model development for practitioners in an organisational setting. DES practitioners typically combine the technical knowledge required to undertake DES such as model building and statistical methods with an understanding of an application domain such as manufacturing or healthcare. Although many experienced simulation practitioners began their simulation careers coding models in simulation languages such as SIMAN and using languages such as FORTRAN for file processing, in the light of the development of drag and drop interfaces in such tools as Arena, recent users may find it a particular challenge to adapt to the need for coding when developing a machine learning algorithm in tools such as Matlab, R or Python. In this article, using the facilities of a relatively new COTS DES package of Simio, the aim has been to demonstrate the possibility of incorporating a subfield of machine learning, namely reinforcement learning into a COTS DES using embedded process logic. The DES software make this approach feasible with its ability to animate entities by x,y,z coordinate in 2D or 3D space and thus eliminate the need to predefine every possible route taken by the entity in advance. The software also implements an object-oriented approach and allows encapsulation of both the data and process logic definitions within the entity object. Encapsulation of data allows each robot to generate its own q value matrix and if required each robot's own static obstacle or 'no-go' locations can be defined. Encapsulation of process logic through the use of add-on processes allows multiple entities (robots) to each follow their individual training and move cycles.

This article aims to contribute to the methodology of simulation practitioners who wish to implement AI techniques as a supplement to their input modelling approaches. The work should also be of interest to analysts involved in reinforcement learning applications as simulation can provide a virtual environment in which the reinforcement training and testing can take place safely and far quicker than in a real system. In terms of a practical contribution it is claimed that a number of other features of Simio COTS DES have been built into the software to enable Industry 4.0 applications including integration with external data sources such as MES, data generated model facilities and specialised reporting (Zaayman and Innamorato, 2017). Whether under the umbrella of Industry 4.0 or as a reflection of the greater use of AI techniques generally, more and more industrial systems will incorporate AI techniques and so this article provides guidance to those practitioners which require valid and efficient methods of generating simulation models of these systems.

References

Benotsmane, R., Kovács, G., Dudás, L.: Economic, Social Impacts and Operation of Smart Factories in Industry 4.0: Focusing on Simulation and Artificial Intelligence of Collaborating Robots, Social Sciences, 8(143), (2019).

Bergmann, S., Feldkamp, N., & Strassburger, S.: Emulation of control strategies through machine learning in manufacturing simulations, Journal of Simulation, 11(1), 38-50, (2017).

Brailsford, S.: Theoretical comparison of discrete-event simulation and system dynamics. In S. Brailsford, L. Churilov, & B. Dangerfield (Eds.), Discrete-Event Simulation and System Dynamics for Management Decision Making, John Wiley and Sons Ltd, Chichester (2014).

Chewu, C.C.E. and Kumar V.M.: Autonomous navigation of a mobile robot in dynamic indoor environments using SLAM and reinforcement learning, IOP Conf. Series: Materials Science and Engineering, 402, 012022, (2018).

Creighton, D.C. and Nahavandi, S.: Optimising discrete event simulation models using a reinforcement learning agent, Proceedings of the 2002 Winter Simulation Conference, pp. 1945-1950, (2002).

Elbattah, M. and Molloy, O.: Analytics Using Machine Learning-Guided Simulations with Application to Healthcare Scenarios IN Analytics and Knowledge Management (S. Hawamdeh and H.C. Chang (eds.)), (2018).

Greasley, A.: Simulating Business Processes for

Descriptive, Predictive and Prescriptive Analytics, DeGruyter Press (2019).

Harrell, C. and Tumay, K.: Simulation Made Easy: A Manager's Guide, Industrial Engineering and Management Press, Norcross, USA (1995).

He, Y., Stecke, K.E., Smith, M.L. 2016. "Robot and machine scheduling with state-dependent part input sequencing in flexible manufacturing systems", *International Journal of Production Research*, Vol. 54, pp. 6736-6746.

Hosokawa, S., Kato, J., Nakomo, K.: A reward allocation method for reinforcement learning in stabilizing control tasks, Artif Life Robotics 19, 109-114, (2014).

Khare, A., Motwani, R., Akash, S., Patil, J, Kala, R.: (2018) Learning the goal seeking behaviour for mobile robots, 3rd Asia-Pacific Conference on Intelligent Robot Systems, IEEE, pp. 56-60, (2018).

Klass, A., Laroque, C., Fischer, M., Dangelmaier, W.: Simulation aided, knowledge based routing for AGVs in a distribution warehouse, Proceedings of the 2011 Winter Simulation Conference, IEEE, pp. 1668-1679, (2011).

Kumar, U.D.: Business Analytics: The Science of Data-Driven Decision Making, New Del-hi: Wiley, (2017).

Law, A.M.: Simulation Modeling and Analysis, 5th Edition, New York: McGraw-Hill Edu-cation, (2015).

North, M.J. and Macal, C.M.: Managing Business Complexity: Discovering Strategic Solutions with Agent-based Modeling and Simulation, Oxford University Press (2007).

Ono, Y. and Ishigami, G.: Routing problem of multiple mobile robots with human workers for pickup and dispatch tasks in warehouse, Proceedings of the 2019 IEEE/SICE International Symposium of System Integration, IEEE, pp. 176-181, (2019).

Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., de Amicis, R., Pinto, E. B., Eisert, P., Dollner, J., Vallarino, I.: Visual Computing as a Key Enabling Technology for Industrie 4.0 and Industrial Internet, IEEE Computer Graphics & Applications, 35(2), 26-40, (2015).

Robinson, S.: Simulation: The practice of model development and use, Second Edition, Palgrave Macmillan (2014).

Rüsmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P. Harnisch, M.: In-dustry 4.0: The Future of Productivity and Growth in Manufacturing Industries, The Boston Consulting Group, Boston (2015).

Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T.K.S., Koenig, C., Choset, H.: PRIMAL: Pathfinding via Reinforcement Learning and Imitation Multi-Agent Learning, IEEE Robotics and Automation Letters, 4(3), 2378-2385, (2019).

Seifert, R.W., Kay, M.G., Wilson, J.R.: Evaluation of AGV routeing strategies using hierarchical simulation, International Journal of Production Research, 36(7), 1961-1976, (1998).

Smith, J.S., Sturrock, D.T., Kelton, W.D.: Simio and Simulation: Modeling, Analysis, Applications, 5th Edition, Simio LLC, (2018).

Sutton, R.S. and Barto, A.G.: Reinforcement Learning: An Introduction, Second Edition, The MIT Press (2018).

Truong, X.T., Ngo, T.D.: Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model, IEEE Transactions on Automation Science and Engineering, 14(4), 1743-1760, (2017).

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., Kyek, A.: Optimization of global production scheduling with deep reinforcement learning, Procedia CIRP 72, 1264-1269, (2018).

Watkins, C.J.C.H. and Dayan, P.: Q-learning, Machine Learning, 8(3-4), 279-292, (1992).

Zaayman, G. and Innamorato, A.: The application of Simio scheduling in Industry 4.0, Proceedings of the 2017 Winter Simulation Conference, pp 4425-4434, (2017).