Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 63 (2017) 107 – 112

www.elsevier.com/locate/procedia

The 50th CIRP Conference on Manufacturing Systems

Motion Planning for Industrial Robots using Reinforcement Learning

Richard Meyes[a,*], Hasan Tercan[a], Simon Roggendorf[b], Thomas Thiele[a], Christian Büscher[c], Markus Obdenbusch[b], Christian Brecher[b], Sabina Jeschke[a], Tobias Meisen[a]

[a] *Institute for Information Management in Mechanical Engineering (IMA) of RWTH Aachen University, Dennewartstr. 27, 52064 Aachen, Germany*
[b] *Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Steinbachstr. 19, 52074 Aachen, Germany*
[c] *Saint-Gobain Sekurit Deutschland GmbH & Co. KG, Glasstr. 1, 52134 Herzogenrath, Germany*

* Corresponding authors. Tel.: +49-241-80-91146; fax: +49-241-80-91122. *E-mail address:* richard.meyes@ima-zlw-ifu.rwth-aachen.de

Abstract

A major challenge of today's production systems in the context of Industry 4.0 and Cyber-Physical Production Systems is to be flexible and adaptive whilst being robust and economically efficient. Specifically, the implementation of motion planning processes for industrial robots need to be refined concerning their variability of the motion task and the ability to adaptively deal with variations in the environment. In this paper, we propose a reinforcement learning (RL) based, cognition-enhanced six-axis industrial robot for complex motion planning along continuous trajectories as e.g. needed for welding, gluing or cutting processes in production. Our prototype demonstrator is inspired by the classic wire loop game which involves guiding a metal loop along the path of a curved wire from start to finish while avoiding any contact between the wire and the loop. Our work shows that the RL-agent is capable of learning how to control the robot to successfully play the wire loop game without the need of modeling the wire or programming the robot motion beforehand. Furthermore, the extension of the system by a visual sensor (a camera) allows the agent to sufficiently generalize the learning problem so that it can solve new or reshaped wires without the need of additional learning. We conclude that the applicability of RL for industrial robots and production systems in general provides vast and unexplored potential for processes that feature variability to some extent and thus require a general and robust approach for process automation.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems

Keywords: Reinforcement Learning; Cyber-Physical Production Systems (CPPS); Self-Optimization

1. Introduction

In the era of the fourth industrial revolution, frequently noted as Industry 4.0, one of the four key requirements of Cyber-Physical Production Systems is the ability to react adaptively to dynamic circumstances of production processes [1,2,3].

Motions of industrial robots that are part of a bigger production process are commonly programmed in a non-flexible way and require exact control over the circumstances of the motion task. For instance, the motion of a simple pick-and-place process requires exact knowledge about the position of the object to be picked up and about the container in which the object is to be placed in. Small deviations of either the object or the container would result in process failure, as the non-flexible programming of the motion is not able to deal with small variations of the environment. This general problem gives rise to the question how robots can be enabled to deal with such variations of the environment and autonomously adapt to them to plan their motion in a flexible way.

In this paper, we address this question and present a proof of concept for an augmentation of an industrial robot with cognitive capabilities. This concept is based on the idea to provide a robot with sensor technology that allows it to observe its environment and to complement it further by an operating agent that is able to control the robot and gather experiences about its interaction with the environment. Based on those experiences, the agent seeks to adapt its behavior and control the robot in such a way that the motion task is performed as intended.

We implemented this concept in an exemplary use case scenario in which a six-axis industrial robot (UR5 Robot from Universal Robots) is controlled by an agent that learned to play the wire loop game [4]. The robot autonomously guides a metal

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.095

loop along the path of a curved wire from start to finish while avoiding any contact between the wire and the loop. It is enhanced by a visual sensor (a camera) that provides vision of the agent's environment. The agent's algorithm learns to play the game which is modelled as a Markov Decision Process (MDP) by means of reinforcement learning (RL) [5,6] and Q-learning [7] without any domain knowledge, i.e. it does not know the concept of a loop or a wire and it does not know what its actions do exactly.

1.1. State of the Art

The successful application of RL and a variation of Q-learning, was previously demonstrated by enabling an agent to play board games, e.g. backgammon [8] or even Atari games directly from visual sensory input [9,10]. MDPs and partially observable MDPs (POMDPS) have been largely used in for motion planning in mobile robotics [11,12], autonomous planning for unmanned ground and aerial vehicles [13,14] and human assisted teleportation [15].

Recent research has attempted to utilize deep RL to tackle a wide variety of continuous motor control problems, e.g. motion planning for industrial robots directly from sensory input [16,17]. Although these attempts demonstrated the conceptual usability of the methods they rely on heavy computational effort both in terms of the number of used cores (either CPUs or GPUs) and the required computation time which is of the order of several days.

With our approach, we avoid directly processing sensory input to decrease the computational effort of the learning problem significantly. We utilize a well-approved approach from the mobile robotics domain that relies on the formalization of the planning problem as an MDP and transfer this approach to the stationary robotics domain. We show that our agent is able to learn how to play the wire loop game within only a few minutes of computation time on a single CPU. Furthermore, it is capable of generalizing the learning problem within a few hours so that it is able to play the game with wire configurations that were not used during the training phase. Our results imply that RL-based augmentations for robots provide a feasible way to deal with processes that feature variability to some extent and thus, in order to be automated, require a general and robust approach.

2. Background

The presented concept for an augmentation of robots is based on RL and the condition to model the learning task at hand as a Markov Decision Process allowing to utilize a variation of Q-learning, a learning paradigm that has been successfully applied to various virtual learning scenarios.

2.1. Reinforcement Learning and Markov Decision Processes

Reinforcement learning (RL) is a machine-learning paradigm inspired by behaviorist psychology and addresses the procedure of how an agent (an animal, a human or even a machine) interacts with its environment [5]. It describes the problem of an agent that tries to develop a behavioral strategy in order to maximize some notion of cumulative reward as a result of taking the right actions in any state of its environment [6]. Figure 1, left hand side illustrates the underlying state-action-reward principle of RL problems. In general, these kinds of problems can be formalized as MDPs.

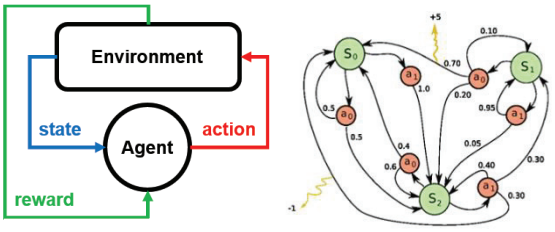

Figure 1: (left) Schematic illustration of the RL paradigm. An agent interacts with its environment and receives a reward for each action that is taken in a specific state of the environment. **(right)** Schematic illustration of a simple, discrete MDP with three states s_0, s_1, s_2 and two actions a_0, a_1. The probability of reaching a state s_i by taking an action a_i is given by the black number next to the black transition arrows. The reward that is given after taking certain actions in certain states is represented by yellow arrows. Figure adopted from [18].

An MDP is a mathematical framework for modeling decision making as a discrete time stochastic control process [19]. It assumes that the modelled stochastic process possesses the Markov property, i.e. the conditional probability distribution of each future state depends only on the present state and not on the sequence of events that preceded it [20]. Figure 1, right hand side illustrates a simple example of an MDP with three states and two actions. In order for an agent to maximize its reward in the exemplary MDP in Figure 1, right hand side, the agent needs to learn that the cumulative reward over time can only be maximized when temporary punishments, i.e. negative rewards, are accepted. Thus, in general, an agent needs to take into account not only immediate rewards but also possible future rewards. A single episode e_{MDP} of any given MDP forms a finite sequence of states, actions and rewards and can be expressed as:

$$e_{MDP} = \{s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots, s_{n-1}, a_{n-1}, r_{n-1}, s_n\} \quad (1)$$

where s_i, a_i, r_i represent the i-th state, action and reward that is received after performing the action, respectively. The total future reward from any time point t is given by:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-t} r_n \quad (2)$$

where $\gamma \epsilon (0, 1)$ is the discount factor and models how strongly the agent takes future rewards into account. Values close to 0 will represent a short-sighted strategy as higher-order terms for rewards in the distant future become negligible. If the environment is deterministic, γ can be set to 1 as the same actions always result in the same rewards. A good strategy for an agent trying to maximize its discounted future reward can be learned by means of the Q-learning paradigm.

2.2. Q-Learning

Q-learning is a paradigm that can be used to allow an agent to find an optimal policy for choosing an action for any given finite MDP. In general, a policy is a deliberate system of principles to guide decisions or more specifically, a decision function that specifies what the agent will do for each possible value that it can sense [21]. In contrast to other learning paradigms such as SARSA (state-action-reward-state-action), Q-learning is an off-policy learner and allows learning from