Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
*Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird*
*Annals & Proceedings of DAAAM International 2011*

# EFFECTS OF INITIAL KNOWLEDGE ON REINFORCEMENT LEARNING BASED CONTROL

**ALBERS, A[lbert]; YU, X[i] & SOMMER, H[ermann]**

***Abstract:*** *Reinforcement learning methods have been involved in novel approaches to control nonlinear dynamic systems. The large amount of necessary interactions leads to lengthy learning durations. To accelerate the learning process, the one way of providing initial knowledge to the learning agent is introduced. The aim of this paper is to discuss the effects on the reinforcement learning based motion control by providing initial knowledge obtained by the simulation process. Experimental results from the learning process with and without initial knowledge are presented and compared.*
***Key words:*** *machine-learning, robotics, dynamics, control*

## 1. INTRODUCTION

### 1.1 Reinforcement Learning

Introducing reinforcement learning (RL) is one of the novel approaches in motion control of nonlinear and flexible systems, especially when the control agent is confronted with an individual task that is neither investigated before nor provided with existing examples. By RL method the learner (agent) must discover how to map a situation to the optimal action, by understanding the environment, trying different actions, gaining experiences based on the feedback from the environment (usually expressed as numerical reward signals) and hence result in an optimized solution.

There are different RL methods. In most of them the experiences are stored as quality values related to different state-action pairs in a matrix representation. Each selection of the agent is based on the comparison among quality values of all the state-action pairs regarding the current state. (Denzinger & Laureyns, 2008) have concluded that SARSA algorithm (Sutton & Barto, 1998) is a possible solution in the motion control of a 2-DOF manipulator system.

However, certain complexity in the RL method has narrowed its application. For example, a ' wise ' selection from available actions made by the RL agent is ensured only after the agent has gained enough experiences. In other words, the agent must investigate all the possible choices before it can tell if any action is ' better ' than another one. Additionally, the number of states increases exponentially by adding states or actions to the system. This ' curse of dimensionality ' (Bellman, 1957) brings in huge numbers of state-action pairs and therefore a long time of investigation before the agent figuring out an optimal solution.

One possible solution to accelerate the learning process is to provide the agent with existing quality values as initial knowledge. There can be various sources of the initial knowledge, e.g. experiences gained from similar experiments, or simulation. As the purpose of introducing RL methods is to deal with unfamiliar problems, which may have no similarity to existing examples, simulation is a more reliable source.

The motion control based on RL methods has been realized in simulated environment (Martin & De Lope, 2007). In this article, the related approach is launched on a real system. Results gained from learning experiments with/without initial knowledge are presented.

### 1.2 The 2-DOF Manipulator

In (Denzinger & Laureyns, 2008) a 2-DOF planar robot manipulator was presented. The RL experiments on a real system are operated on a model built up corresponding to the same details. To limit the computation time, a simulation model is established referring to the sketch of the manipulator (Fig. 1).

Experiments with and without initial knowledge are operated on the same manipulator under the same experiment condition with the same RL parameters, e.g. discounting rate, exploration rate, etc.

### 1.3 The Control Agent

The RL algorithms are programmed under LabVIEW environment. The program including all parameters are edited in a main PC, and sent to a target PC after each experiment is started. The target PC acts as the learning agent during experiments. The manipulator is controlled by the target PC through FPGA to limit the execution time.

## 2. RESEARCH METHODS

### 2.1 Experiments Design

The RL experiments are designed in an episodic way. At the very beginning the manipulator is reset, (by which the current position of the robot will be set as the origin), and all the quality values are initialized as zero. In the beginning of each episode the robot starts at the origin and begins attempting and learning step by step. In each learning step the RL algorithm is as described in (Yan et al., 2009). When the robot reaches the target or exceeds the maximum step number of one episode, this episode is stopped and the robot moves back to the origin. Meanwhile, the updated quality values are stored and passed over to the next episode. Before the first episode is executed on the real manipulator, the agent chooses whether to do a prior simulation or not. With a prior simulation the experiments on the manipulator start with quality values initialized as updated data from simulation. Otherwise the agent starts operating the manipulator directly with all quality values initialized as zero.

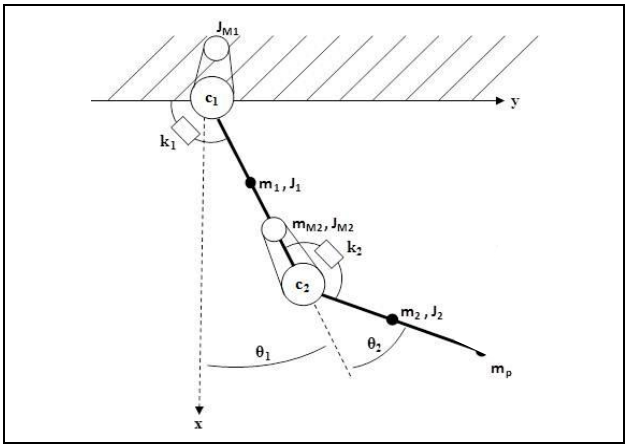


Fig. 1. General sketch of the manipulator

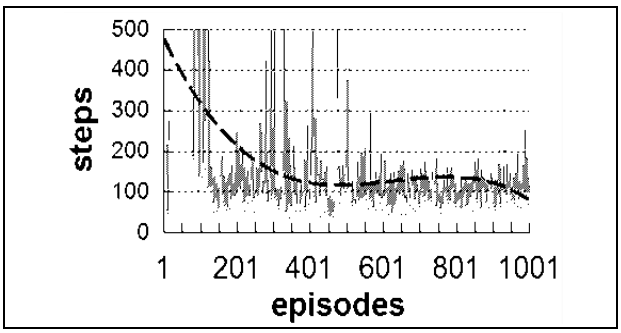

Fig. 2. Learning curve in direct learning

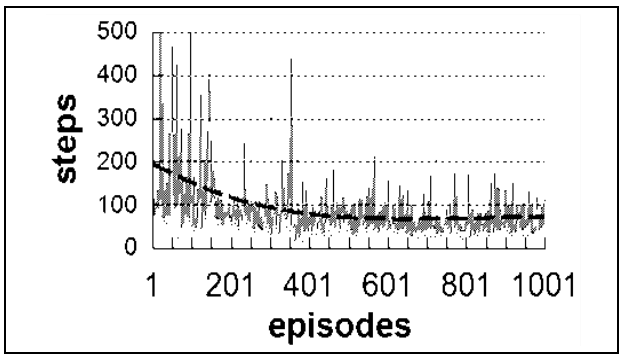

Fig. 3. Learning curve with initial knowledge

**2.2 Experimental Setup**

The RL agent uses the angular speed and the offset (in degrees) between the current position and the target to describe a state. In the relative experiments with and without a prior simulation the targets are always set as (50˚, -30˚). When the sum of squared offsets is smaller than one and the angular velocities in both joints are slower than 1˚/s, the robot will be regarded as ' reach the target '.To reduce the number of states each of the offsets and the velocities is divided into seven groups. For each group the agent is able to choose from five possible torques exerted on the joint. The torque list for the shoulder joint is [-0.145, -0.005, 0, 0.005, 0.145] Nm, and the list for elbow joint is [-0.077, -0.001, 0, 0.001, 0.077] Nm. Each experiment runs 1000 episodes on the manipulator. If a prior simulation is ordered, the agent runs 1000 episodes under simulated environment and another 1000 episodes on a real system. In each episode there are to the maximum 500 steps, with an interval of 0.1 s.

**2.3 Learning Curve**

The agent records the step number in each episode on the manipulator and plots them at the end of the experiment. It is the learning curve reflecting the learning efficiency and result. The dashed curves in Fig. 2 and 3 are the trend analysis represented in a third-order polynomial. A learning curve converging to a small step number within fewer episodes and with less fluctuation indicates a learning process with higher efficiency and a better respectively more stable final solution.

## 3. RESULTS

Fig.2 shows a typical learning curve gained from a RL process directly launched on the robot manipulator. In the first 80 episodes the robot rarely reaches the target within 500 steps. The learning process converges to stable solutions of about 120 steps to reach the target after 570 episodes. Before the agent is able to obtain a stable solution to cover the offset, there are a cumulative total of 121189 steps, which leads to a learning process up to three hours and 22 minutes. In contrast, the learning curve in Fig.3, which depicts the RL process with initial knowledge, shows a more efficient behavior. The robot is able to reach the target frequently even in the initial episodes. The agent comes up with a more stable solution leading to the target in about 80 steps after 350 episodes. The accumulative learning duration (before the learning curve converges) is 45939 steps, or one hour and 17 minutes. The time required to obtain a comparatively stable solution could be reduced by 62%. The shortest episodes in the experiments contain 22 steps for a direct RL process, while 16 steps for learning with initial knowledge. However, this may accounted to the inevitable differences between the simulation model and the real system. In both types of experiments the robot's behavior varies at the beginning due to the exploration rate (epsilon) of the learning algorithm allowing the occasional selection of random actions(Sutton & Barto, 1998). By making a lucky guess the robot can reach the target in a very short time, e.g. around 20 steps; but it leads to difficulties in reaching the target by making unlucky guesses, even though the initial knowledge is enough to suggest an optimal choice. That may explain the conspicuous confusion in the robot's decisions in the beginning of the RL process with initial knowledge. By determining an optimal epsilon to the RL process, a more effective and efficient learning behavior is expectable.

## 4. CONCLUSION

First of all, the agent is able to discover a solution for the robot to reach the target on a real system by introducing SARSA algorithms into the learning process. Secondly, providing initial knowledge stimulates the RL process. Although the simulated environment is simplified and unable to describe the model exactly, the initial knowledge provided by the simulated RL process positively decrease the learning duration for the agent. However, the agent still takes a long time while seeking for an optimal solution. The approach introduced in this paper is an improvement to the existing RL methods in motion control but it is still limited by the computational effort; therefore further development on the algorithm is necessary before extending this method to a model with a higher DOF. Future effort could be focused on epsilon determination, by which more experiments are required to find out the optimal parameters related to different cases. Another point of interest for future research is to provide other sources of initial knowledge, e.g. quality values gained by repeated experiments. Ongoing research will determine the effect of the implementation of averaged Q-Tables. An optimal method to obtain the initial knowledge will be suggested correspondingly.

## 5. REFERENCES

Denzinger J.; Laureyns I. & et.al. (2008). A Study of Reward Functions in Reinforcement Learning on a Dynamic Model of a Two-link Planar Robot, The 2nd European DAAAM International Young Researchers´ and Scientists´ Conference

Martin & De Lope, (2007) A Distributed Reinforcement Learning Architecture for Multi-Link Robots. 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO). Angers, France.

Peters J. (2008). Machine Learning for Robotics. VDM Verlag Dr. Müller, Saarbrücken, ISBN 978-3-639-02110-3

Sutton R.S & Barto A.G. (1998). Reinforcement Learning, an introduction, *The MIT press*, MA, ISBN: 978-026-2193986

Bellman,R.E. (1957). Dynamic Programming, Princeton University Press, Princeton, U.S., ISBN: 978-069-107951-6

Yan, W & et. al. (2009). Application of reinforcement learning to a two DOF Robot arm control, Annals of DAAAM for 2009 & Proceedings of 20th DAAAM