Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium, Volume 20, No. 1, ISSN 1726-9679
ISBN 978-3-901509-70-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2009
*Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird*
*Annals of DAAAM International*

# APPLICATION OF REINFORCEMENT LEARNING TO A TWO DOF ROBOT ARM CONTROL

**ALBERS, A[lbert]; YAN, W[enjie] & FRIETSCH, M[arkus]**

***Abstract:*** *Automatic manipulators control poses a complex challenge for control systems, which have to deal with various dynamic and nonlinear effects. This paper presents a novel approach for motion control of a two DOF robot arm based on reinforcement learning. A new method to reduce the high computational efforts, which come along with this method, is presented. In order to accelerate the convergence of the learning process, a fuzzy logic system is integrated in the reward function. For further optimization of the implemented algorithm a library of already learned motions is created. The experimental result shows a significant improvement of learning efficiency.*

***Key words:*** *reinforcement learning, robotics, computational intelligence*

## 1. INTRODUCTION

One of the biggest challenges in current research in robotics is, that robots “leave” their well structured environment and are confronted with new tasks in a more complex environment. Due to this, it can only be successful resp. useful, when it is able to adapt itself and learn from experiences. Reinforcement Learning (RL), a branch of machine learning (Mitchell & Tom, 1997), is one possible solution. RL is a learning process, which uses reward and punishment from the interaction with environment to learn a policy for achieving tasks. Various RL methods e.g. Q-learning (Watkins, 1989) have been studied in the recent decades and it is shown that two problems must be considered. At first, the high computational efforts: RL is disturbed by the “curse of dimensionality” (Bellman, 1957), which refers to the tendency of a state space to grow exponentially in its dimension, that is, in the number of state variables (Sutton & Barto, 1998). Secondly, a Q-table is created for one specific task. It requires an extreme large space to store policies for all possible tasks, which restricts the practical application of this learning method strongly.

In (Martin & De Lope, 2007), the author presents a distributed architecture in RL to solve the first problem, which uses some small, low dimensional Q-tables instead of a global high-dimensional Q-table with the evaluations of actions for all states. In this paper, another ways based on an optimized state space representation are proposed for improving the learning ability of RL

.

## 2. APPROACH

### 2.1 Overview

The schematic of robot arm with two degree of freedom is illustrated in Fig. 1. The system is dynamic and nonlinear because the inertia of the upper arm changes due to a variable angle $\theta_2$ (Denzinger & Laureyns, 2008).

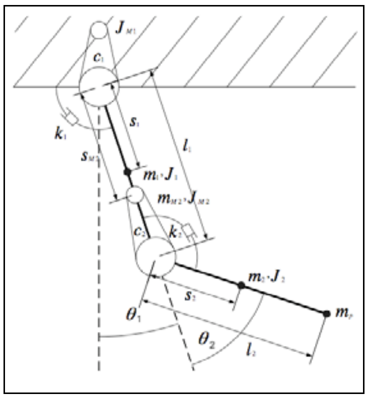


Fig. 1. Robot arm model

Since the redundancy of the inverse kinematic, the angle set which is closer to the target state will be selected as end position. Because our work is focused on how to reach the goal, the motion accuracy of intermediate states is not interested thus the interval of a state close to the target could be much smaller than of a state far from it. A relative position is adopted to normalize the distribution of state space, whose values are always changed depending on the target position. Moreover, a fuzzy-logic system is integrated in the reward function for evaluating the executed actions and a coordinate transformation is applied to represent different tasks with one Q-table in library.

### 2.2 Unevenly distribution of state space

The state space of robot arm consists of its angles and angle velocities. A Q-table in this case is a 6-dimensional hyperspace, here means $\theta$ angle, $\dot{\theta}$ velocity and $a$ action:

$$Q = \{\theta_1,\, \dot{\theta}_1,\, \theta_2,\, \dot{\theta}_2,\, a_1,\, a_2\} \qquad (1)$$

According to the approach of (Martin & De Lope, 2007), we can decompose the Q-table in $Q_1 = \{\theta_1, \dot{\theta}_1, a_1\}$ and $Q_2 = \{\theta_2, \dot{\theta}_2, a_2\}$. With the help of the fixed target position, each position can be calculated as a constant value $\theta_t$ plus a difference $\Delta\theta$ as shown in the follow equations:

$$\begin{aligned} (\theta_1, \theta_2) &= (\theta_{1,t} + \Delta\theta_1, \theta_{2,t} + \Delta\theta_2) \\ \rightarrow (\Delta\theta_1, \Delta\theta_2) &= (\theta_1 - \theta_{1,t}, \theta_2 - \theta_{2,t}) \end{aligned} \qquad (2)$$

If the state space is not built according to $\theta$ but to$\Delta\theta$, the target position can always be located at $(\Delta\theta_1, \Delta\theta_2) = (0,0)$. In this case, the state space can be easily divided with different intervals.

$$\begin{aligned}\Delta\theta &= [0,\ \pm 0.1,\ \pm 0.45,\ \pm 1] \quad (rad)\\ \Delta\dot{\theta} &= [-1,\ -0.5,\ 0,\ 0.5,\ 1] \quad (rad/s) \qquad (3)\end{aligned}$$

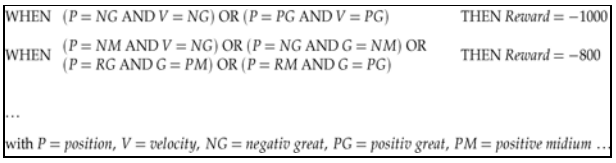
A state space for a joint can be created with $5 \cdot 7 = 35$ elements and for both joints with $2 \cdot 5 \cdot 7 = 70$ elements. In comparison, $(2\pi/0.1) \cdot 5 \cdot 2 \approx 628$ elements are needed to generate a conventional state space with fixed increments (we assume that $\Delta\theta = 0.1$ so as to get the same accuracy).

### 2.3 Q-table library

As mentioned before, the movement based on the relative position $\Delta\theta$ is independent of the absolute position $\theta$. For two movements with the same $\Delta\theta$, we can change the coordinate system so as to make both of them the same. Because the dynamic model changes with $\cos\theta_2$, it is necessary to save the Q-tables after $\theta_2$ or $\cos\theta_2$. By using ten discrete values for every dimension, a cubic shaped library with 1000 elements is created for different $\Delta\theta_1$, $\Delta\theta_2$, $\theta_2$. The library is searched through these three values for a new task and the Q-table is loaded as initial policy in reinforcement learning.

### 2.4 Reward function with fuzzy logic system

A fuzzy logic (Michels & Kai, 2002) is an extension of boolean logic to multi-valued logic, which can be used to approximate complex function with simple and well understood control conditions. Because a reward function is composed of different parameters and challenging to describe in mathematic, a fuzzy logic system is integrated in the reward function so as to reduce the modeling expenses. The angle and velocity of each joint are defined as input fuzzy sets and the reward value as output. The reward values for both joints are calculated before summed to one. The fuzzy rule is depicted in Fig. 2 and the reward function of single joint in Fig. 3.


WHEN $(P = NG \text{ AND } V = NG) \text{ OR } (P = PG \text{ AND } V = PG)$ THEN $Reward = -1000$

WHEN $(P = NM \text{ AND } V = NG) \text{ OR } (P = NG \text{ AND } G = NM) \text{ OR }$ $(P = RG \text{ AND } G = PM) \text{ OR } (P = RM \text{ AND } G = PG)$ THEN $Reward = -800$

…

with $P = position,\ V = velocity,\ NG = negativ\ great,\ PG = positiv\ great,\ PM = positive\ midium$ …

Fig.2. Control rule for fuzzy logic system

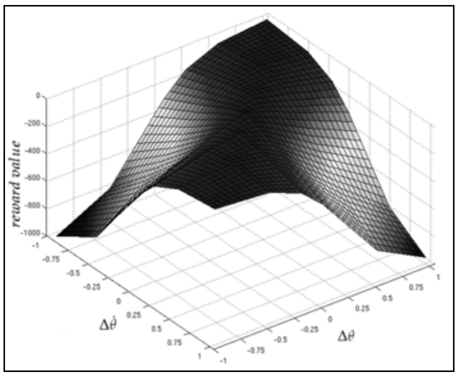


Fig. 3. Reward function for single joint

## 3. EXPERIMENTAL RESULTS

The experiments confirm the success of our approaches. The figures show the results of the operating process of performed simulations:

- State space with unevenly and evenly distribution of angle position with an increment of $0.1\ rad$ (Fig. 4)
- Learning with and without the use of learned policy (Fig. 5)

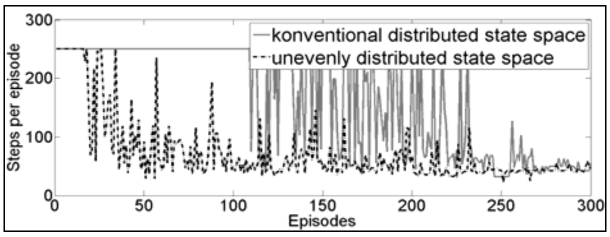


Fig. 4. Test results with different state space

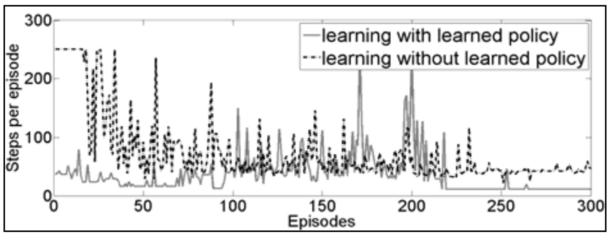


Fig. 5. Test results with and without policy file

## 4. CONCLUSION

This paper introduced novel approaches for optimizing the computational efforts in reinforcement learning by means of an unevenly distribution of state space and building the policy library so as to accelerate the learning process for new tasks. The experimental results showed that the learning performance can be improved significantly with this new approach. At the same time, the quality of calculated solutions is better than the solutions with conventional methods. For these reasons, we conclude that our approach is a suitable technique to enhance the learning ability of reinforcement learning for the given application. However, this method can be only used for point-to-point motion control and still difficult for online learning. The presented methods are currently being implemented on a higher DOF robot arm.

## 5. REFERENCES

Bellman, R. E. (1957). *Dynamic Programming*, ISBN: 978-0691079516, Princeton University Press, Princeton, NJ

Denzinger, J. & Laureyns, I. (2008). A study of reward functions in reinforcement learning on a dynamic model of a two-link planar robot, *Proceedings of DAAAM 2008,* ISBN: 978-3-901509-68-1, Vienna, Austria, Oct 2008

Martin H, J. A. & De Lope, J. (2007). A Distributed Reinforcement Learning Architecture for Multi-Link Robots, *Proceedings of ICINCO 2007,* Angers France, May 2007

Michels & Kai (2002). *Fuzzy-Regelung: Grundlagen, Entwurf, Analyse*, Springer, ISBN: 3540-43548-4, Berlin, Heidelberg

Mitchell, Tom M. (1997). *Machine learning*, McGraw-Hill, ISBN 0-07-042807-7, New York

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, MIT Press, ISBN: 978-0-262-19398-6, Cambridge