Mechatronics 24 (2014) 1021–1030

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Comparison of model-free and model-based methods for time optimal hit control of a badminton robot

B. Depraetere [a,*], M. Liu [b], G. Pinte [a], I. Grondman [b], R. Babuška [b]

[a] Flanders' Mechatronics Technology Centre vzw (FMTC), Celestijnenlaan 300 D, B-3001 Leuven, Belgium
[b] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 30 September 2013
Revised 30 July 2014
Accepted 22 August 2014
Available online 10 September 2014

Keywords:
Robot control
Time optimal motion
Optimization
Reinforcement learning
Natural actor-critic

ABSTRACT

In this research, time optimal control is considered for the hit motion of a badminton robot during a serve operation. Even though the robot always starts at rest in a given position, it has to move to a target position where the target velocity is not zero, as the robot has to hit the shuttle at that point. The goal is to reach this target state as quickly as possible, yet without violating the limitations of the actuator. To find controllers satisfying these requirements, both model-based and model-free controllers have been developed, with the model-free controllers employing a Natural Actor-Critic (NAC) reinforcement learning algorithm. The model-based controllers can immediately achieve the desired motions relying on prior model information, while the model-free methods are shown to yield the desired robot motions after about 200 trials. However, in order to achieve this result, a good choice of the reward function is essential. To illustrate this choice and validate the resulting controller, a simulation study is presented in which the model-based results are compared to those obtained with two different reward functions.

1. Introduction

This paper considers time optimal motion to a non-equilibrium target state, with the goal of using this to perform serve motions of a badminton robot. For such a serve, the robot always starts from an initial position, after which it has to accelerate the racket to intercept a shuttle, which means it has to arrive at a prescribed hit time and point with a prescribed, non-zero hit velocity. Note that because of this non-zero hit velocity the target is a not an equilibrium state, as the robot cannot remain in the desired position with the desired non-zero velocity. While performing In this case, to allow the opponent less time to prepare, we want to perform this motion in a time optimal manner, so that the robot starts moving as late as possible taking into account the actuator limitations. In general, similar time optimal motion control problems are relevant for a wide variety of mechatronic applications, where being able to generate faster motions typically means more units can be produced or more output can be generated within a given time span. Research into time optimal motion has therefore already received considerable attention [2,5,8,10,17,21,22,24,25], but this has mostly been from the field of model-based motion control, while model-free methods are rarely considered. For model-based methods, an optimal control problem is directly formulated and solved numerically, explicitly minimizing the motion time or using approximate costs yielding simpler optimization problems [21,24]. While these model-based approaches yield good results, they do have some associated drawbacks. A first drawback is that model-based methods rely on an accurate model of the system to be controlled, while in many cases such models are not available or are difficult and time-consuming to obtain. Another drawback is that these methods require the solution of a numerical optimization problem, which can be a daunting task, especially when nonlinear models are needed that lead to non-convex problems, although this problem is less significant if the calculations can be performed offline. A final drawback of model-based methods is that in general they do not adapt or learn from past experience. As a result, if the model used for the optimization is not perfect, the motions obtained will not be time optimal and will remain as such, unless learning laws are added as for example in [8].

Model-free methods can address most of these drawbacks, since they operate by interacting with the environment and by learning from these interactions. As a result, they can be employed with very little prior information, and adapt automatically if the circumstances are altered. To investigate whether such methods can directly be applied to a real mechatronic system and how well they perform, a model-free approach is implemented in this paper, as well as a classical model-based one, and their performance is

* Corresponding author.
E-mail address: bruno.depraetere@fmtc.be (B. Depraetere).

http://dx.doi.org/10.1016/j.mechatronics.2014.08.001
0957-4158/

compared for the specific task of the badminton robot's serve operation.

As stated, learning approaches have already been developed for time optimal motion [8,17], but these rely on explicit models of the controlled system or assume linear dynamics and perform an implicit identification. In contrast, a model-free reinforcement learning (RL) algorithm is employed in this paper, which directly learns a control policy for the considered task, without the knowledge of a model of the system. Among the model-free methods, RL is a framework in which an agent learns an optimal policy (control law) to control its environment (the system considered) by using experience obtained from interacting with it. This interaction is characterized by two relevant aspects, performing actions and observing the resulting system behaviour [9]. Each time an action is taken (driving the actuators), a state transition can occur and a scalar reward is calculated using the observed results (evaluation of the quality of the action). The agent then adapts the mechanism it uses to select its actions, aiming to maximize the reward received in the future. Since the goal is to maximize these rewards, it is essential to select a reward function corresponding to the specifications, which in this case means a time optimal motion.

A popular class of RL algorithms are the actor-critic methods [12], where the actor is equivalent to the control policy and the critic is a value function used to evaluate the policy's quality. Actor-critic methods can deal with continuous state and action spaces and, in general, have good convergence properties and performance if a gradient-based policy improvement is used [12]. The critic provides a low-variance value function estimate based on which the gradient with respect to the policy can be computed, so the actor can be updated in the direction of performance improvement indicated by the gradient. It has also been suggested to improve the learning performance by using the natural policy gradient, which gives the steepest ascent direction with respect to the Fisher information matrix, instead of the standard gradient [1]. We therefore use the Natural Actor-Critic (NAC) algorithm as described in [14], which is widely used in robotics and often yields a good learning performance [4,11,15]. Some modifications are proposed here though to make it work in an episodic instead of continuous framework, since this is more natural for the robot's serve motion.

The remainder of the paper is organized as follows. First, the badminton robot is introduced in Section 2, and the task to be performed is defined. Next, the model-based and model-free approaches are developed, in Sections 3 and 4 respectively. Simulation results for both controllers are then presented in Section 5, including a detailed comparison and discussion, before conclusions and topics for future work are suggested in Section 6.

2. Badminton robot

2.1. Overview of badminton robot

The application considered in this work is the badminton robot developed by FMTC. A schematic overview of the robot and its mode of operation are given in Fig. 1. Besides the mobile robot platform on which the racket is mounted, there is also a 3-dimensional camera system that detects shuttles flying to and from the robot and estimates their expected further trajectories. The possible interception points at which the robot can hit the shuttles are then calculated and the robot is actuated towards these points to perform the hit motion. A more detailed description of the badminton robot can be found in [18], and a movie of the robot is available at http://www.fmtc.be.

A closer look at the mobile platform is shown in Fig. 2. It is essentially a serial robot with 3 degrees of freedom. The first two are driven by the linear and rotational motor (Rot. motor in the figure), and they respectively allow the robot to move along the linear guide and to rotate in a plane perpendicular to a typical shuttle trajectory. To perform the effective hit, the important degree of freedom is the one that allows the racket to rotate backward and forward about the joint near its handle, which is driven by the hit motor. The construction is such that at the time contact is made with the shuttle only the hit motor should have a non-zero velocity, while the linear and rotational motor can both be at standstill. As a result, the linear and rotational motor are easier to control, and in the remainder of the paper we only focus on controlling the hit motor towards the non-zero target velocity.

2.2. Serve operation of badminton robot

In this paper only the serve motion of the robot is considered, since this motion is always performed in a similar manner. As a result, learning becomes possible, and it is straightforward to compare the results obtained during different hits and using different strategies. In order to perform a serve, a shuttle is dropped from a mechanism placed at a fixed point above the robot, and the instant $t = t_r$ it starts to fall is detected using an optical sensor, as shown in Fig. 3. Since the drop mechanism always releases the shuttles in the same manner, the point where the robot has to hit the shuttle is always approximately the same, as is the period of time T_{drop} between the shuttle's release t_r and the hit $t_h = t_r + T_{drop}$. To obtain a time optimal motion, the goal is then to start moving as late as possible, at $t = t_h - T$, minimizing the motion's duration T. For the ease of notation, however, it is assumed in the remainder of the paper that the hit motion starts at $t = 0$ and is completed at $t = T$.

Summarizing the specifications for the serve motion, the hit motor has to start at an angle q and angular velocity $\dot{q}$

$$(q(0), \dot{q}(0)) = (-\pi/4, 0) \text{ rad}, \tag{1}$$

and has to move to the desired hit point with

$$(q(T), \dot{q}(T)) = (0, 3) \text{ rad/s}, \tag{2}$$

while minimizing T so that the motor reacts as late as possible, and without violating bounds on the motor. In this case, the controller sends a voltage signal u proportional to the current applied to the motor. The allowable range for these controller voltages u and hence also the controller outputs is

$$-0.2 \leqslant u \leqslant 0.2 \text{ V}. \tag{3}$$

3. Model-based control

For the model-based approach, a parametric model is needed to predict the behaviour of the racket. For this, a frequency response function (FRF) expressing the dynamic relation between the controller output u and the racket's angle q is first estimated, as shown in Fig. 4. The racket generally behaves like a double integrator, as would be expected for an inertia driven by a motor. Other dynamics are observable as well however, at low frequencies due to friction in the bearing, and around 10 Hz, where the resonance frequency of the racket is found. In a next step, a parametric model is fitted to this FRF, using a least-squares model-fitting procedure [16]. In this case, a linear state space model is found, with matrices A, B, C and D.

Using the estimated prediction model and the specifications, an optimal control problem can be formulated, searching for the optimal motion profiles and the corresponding control signals. The following problem is then obtained: