A User Model for Adaptation of Task Parameters in Robot-Assisted Exercise

Nicola Lotti(✉), Davide Piscopiello, and Vittorio Sanguineti

Department of Informatics, Bioengineering, Robotics and Systems Engineering,
University of Genoa, Genoa, Italy
nicola.lotti@edu.unige.it

Abstract. Robot-assisted exercises often use controllers which automatically adapt task parameters to the user's performance. One problem with these controllers is how to accommodate the wide variety of degrees of impairment and to properly track the user's improvement in spite of the inherent variability of performance that is typical of these tasks. Here we describe an adaptive controller model which uses reinforcement learning to maintain a model of user's performance and uses it to continuously regulate the task parameter. We show that the model rapidly identifies the user's model parameters and then smoothly tracks performance improvements.

1 Introduction

Robots are now widely used in the treatment of persons with neuromotor impairments. In robot-assisted exercise, robots provide assistive or perturbing forces which either facilitate or challenge user's movements [1]. Several studies also showed that robots can be used to facilitate learning a new motor skill [2] and to transfer a skill from expert to naïve subjects [3]. One of the challenges is that the difficulty level of the exercise should match the user's residual motor capabilities. An exercise that is too difficult or too easy may reduce user's involvement with negative effects on recovery/learning. A way to prevent this is to adjust the exercise's difficulty level – for instance, the magnitude of assistive forces to the user's For this reason, robot-assisted exercises often include controllers which automatically adapt task parameters to the user's performance. One problem with these controllers is how to accommodate the wide variety of degrees of impairment and to properly track the user's improvement in spite of the inherent variability of performance that is typical of these tasks. Here we describe an adaptive controller model which uses reinforcement learning to maintain a model of user's performance and uses it to continuously regulate the task parameter.

This work is partly supported by a grant from the Italian Ministry of Education, University and Research – Research Projects of National Interest (PRIN 2015).

L. Masia et al. (Eds.): ICNR 2018, BIOSYSROB 21, pp. 200–204, 2019.
https://doi.org/10.1007/978-3-030-01845-0_40

2 Materials and Methods

2.1 Adaptive Control Model

The controller uses reinforcement learning to estimate, trial by trial, the user's model and to calculate the task parameters at the next trial. The user model plays the role of the 'critic'. An 'actor' calculates the next task parameters. The reward provided at the end of each trial is typically a complex function of the user's motor action and is affected by the task parameters specified by the robot, u_R. We assume that, for a given user's skill level, the reward depends monotonically on the task parameters. We specifically use a logistic function: $r(t) = 1/1 + e^{-\beta(u_R(t)-K)} + v(t)$, where $v(t) \sim N(0, R)$ reflects the observation that due to performance variability the movement score may fluctuate from trial to trial even if the task parameters remain the same. This model is general enough to accommodate a large variety of situations.

(1) User Model (Critic): We take the unknown parameters as the user model's state vector: $x \triangleq [K, \beta]^T$. We also assume that the temporal evolution of the user model is described by: $x(t+1) = x(t) + w(t)$, where $w(t) \sim N(0, Q)$ is process noise. This is interpreted as a smoothness constraint on the temporal evolution of the model parameters. The critic aims at maintaining an estimate $\hat{x}(t)$ of the state vector, given the task parameter, $u_R(t)$, and the observed reward, $r(t)$. This is done through and Extended Kalman filter algorithm, in which the correction step is defined as:

$$\begin{cases} W(t) = P(t)^- \cdot \hat{C}(t)^T \cdot \left[\left(\hat{C}(t) \cdot P(t)^- \cdot \hat{C}(t)^T\right) + R\right]^{-1} \\ \hat{x}(t)^+ = \hat{x}(t)^- + W(t) \cdot [r(t) - \hat{r}(t)] \\ P(t)^+ = \left(I - W(t) \cdot \hat{C}(t)\right) \cdot P(t)^- \end{cases} \tag{1}$$

where the expected reward, $\hat{r}(t)$ is defined as

$$\hat{r}(t) = 1/\left\{\{1 + e^{-\hat{x}_2(t)^-\left[u_R(t)-\hat{x}_1(t)^-\right]}\right\} \tag{2}$$

and: $\hat{C}(t) = \left[-\hat{x}_2^- \cdot \hat{r}(1-\hat{r}),\ (u_R - \hat{x}_1^-) \cdot \hat{r}(1-\hat{r})\right]$. The prediction step is defined as:

$$\begin{cases} \hat{x}(t+1)^- = \hat{x}(t)^+ \\ P(t+1)^- = P(t)^+ + Q \end{cases} \tag{3}$$

(2) Action Selection (Actor): Action selection aims at selecting the next robot action, $u_R(t+1)$ in order to obtain the target reward r^*:

$$u_R(t+1) = \hat{x}_1(t+1)^- - \frac{1}{\hat{x}_2(t+1)^-} \cdot \log\left(\frac{1}{r^*} - 1\right) + \eta(t) \tag{4}$$

where $\eta(t) \sim N(0, E)$ is exploration noise.

(3) Model Parameters: The model has three parameters, R, Q, E. In addition, we need to specify the initial values of estimated state, its covariance, and robot

input, i.e. $\hat{x}^+(0) = x_0$, $P^+(0) = V_0$ and $u_R(0) = u_0$. As a general procedure we assume that the task parameter, u_R, ranges from $u_{\min}$ to $u_{\max}$. As a consequence, $x_2^{\min} = 10/\left[9(x_{\max} - x_{\min})\right]$ and $x_2^{\max} = 30x_2^{\max}$. We then take $R = 0.01$, $\sqrt{Q} = \text{diag}\left[0.4(u_{\max} - u_{\min}), 0.005(x_2^{max} - x_2^{min})\right]$, and $\sqrt{E} = 0.05(u_{\max} - u_{\min})$. We also set $x_0 = [u_m\, x_{2m}]$ where $u_m = (u_{\min} + u_{\max})/2$ and $x_{2m} = (x_2^{\min} + x_2^{\max})/2$.

2.2 Experimental Apparatus and Task

We used a planar robot manipulandum [4] specifically designed for motor learning studies and robot therapy. Subjects sat in front of a 43" LED monitor and grasped the handle of a planar manipulandum [4] with their dominant hand. Torso and shoulder were restrained by means of suitable holders. The forearm was supported to compensate gravity and a wrist band reduced wrist movements. The task consisted of controlling a virtual 'tool', consisting of a simulated mass ($m = 5$ kg) connected to the robot handle through a linear spring. Subjects were instructed to move the virtual mass as fast as possible toward a target. To do so, subjects must learn to control the mass-spring dynamics and the internal degrees of freedom of the virtual tool [5]. The spring stiffness, K, determines task difficulty. With a high stiffness, the task is little different from simple reaching. With a low stiffness, the task is very challenging because the mass is very hard to control. After each trial, the subjects received a 0–1 score, calculated in terms of movement time and curvature of the trajctory of the virtual mass. In a previous study, training with this task led to an improved sensorimotor coordination in persons with Multiple Sclerosis [6]. To validate the model, five healthy subjects (3 male and 2 female, age 25 ± 2) underwent a 300-trial training section. We took $K_{max} = 200$ N/m and $K_{min} = 50$ N/m as the ranges of variation of the task difficulty. We compared learning performance with that of five control subjects (3 male and 2 female, age 25 ± 2) performing the same exercise protocol, but with a constant stiffness value ($K = 100$ N/m).

3 Results

Figure 1 shows the time course of the model parameters estimates, calculated for all subject (mean ± SE). In the very early trials the model identifies the user model. After that, the model parameters change gradually as the performance improves.

Figure 2 shows the temporal evolution of the score, r, calculates for all subjects (mean ± SE). After the initial model exploration trials, r reaches the target score r^*, changing the stiffness value and modifying the difficulty of the task.

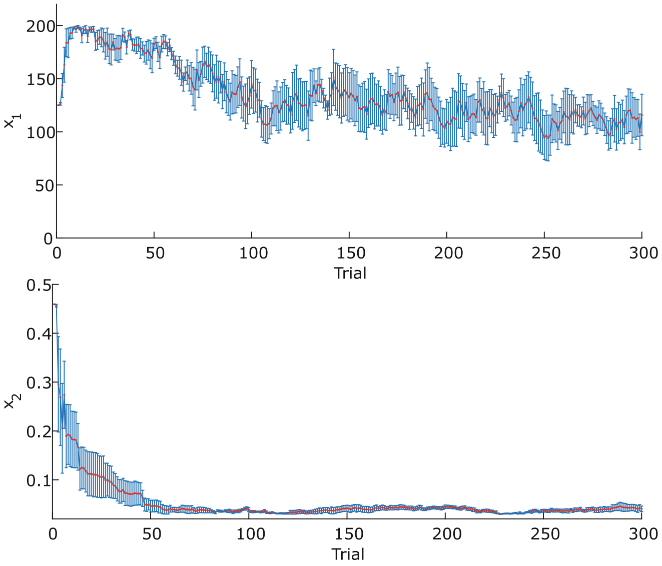

Fig. 1. Temporal evolution of the model parameters, x_1 (top) and x_2 (bottom), averaged over all subjects (Mean ± SE)

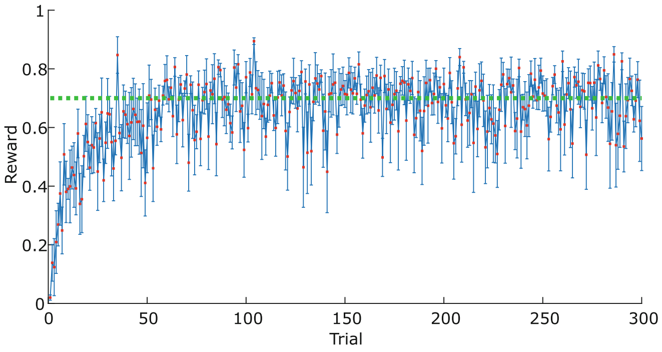

Fig. 2. Temporal evolution of the score, averaged for all subjects (mean ± SE). The green dotted line indicates the target score r^* which was set to $r^* = 0.75$

4 Discussion and Conclusions

We designed an adaptive controller of task difficulty or assistance level which is general enough to work with any exercise and robust enough to deal with the variability typically observed in motor learning and/or rehabilitation trials. Early tests – to be confirmed in a larger experiment – suggest that adaptive control of task difficulty leads to faster and more stable learning.

References

1. Marchal-Crespo, L., Reinkensmeyer, D.J.: Review of control strategies for robotic movement training after neurologic injury. J. Neuroeng. Rehabil. **6**(1), 20 (2009). https://doi.org/10.1186/1743-0003-6-20
2. Basteris, A., Bracco, L., Sanguineti, V.: Robot-assisted intermanual transfer of handwriting skills. Hum. Mov. Sci. **31**(5), 1175–1190 (2012)
3. Galofaro, E., Morasso, P., Zenzeri, J.: Improving motor skill transfer during dyadic robot training through the modulation of the expert role. In: 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 78–83. IEEE (2017)
4. Casadio, M., Sanguineti, V., Morasso, P.G., Arrichiello, V.: Braccio di Ferro: a new haptic workstation for neuromotor rehabilitation. Technol. Health Care **14**(3), 123–142 (2006)
5. Dingwell, J.B., Mah, C.D., Mussa-Ivaldi, F.A.: Experimentally confirmed mathematical model for human control of a non-rigid object. J. Neurophysiol. **91**(3), 1158–1170 (2004)
6. Basteris, A., De Luca, A., Sanguineti, V., Solaro, C., Mueller, M., Carpinella, I., Cattaneo, D., Bertoni, R., Ferrarin, M.: A tailored exercise of manipulation of virtual tools to treat upper limb impairment in multiple sclerosis. In: 2011 IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–5. IEEE (2011)