A Two-level Model of Anticipation-based Motor Learning for Whole Body Motion

Camille Salaün, Vincent Padois, Olivier Sigaud

Université Pierre et Marie Curie - Paris6
Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS FRE 2507,
4 place Jussieu, F-75005 Paris, France
Camille.Salaun@isir.fr, Vincent.Padois@isir.fr, Olivier.Sigaud@isir.fr

Abstract. We present a model of motor learning based on a combination of Operational Space Control and Optimal Control. Anticipatory processes are used both in the learning of the dynamic model of the system and in the coordination between both types of control. In order to illustrate the proposed model and associated control method, we apply these principles to the control of a simplified virtual humanoid performing a standing task starting from a crouching posture.

1 Introduction

There is a wide consensus on the fact that human motor control is model-based, combines feedforward control processes and feedback control processes and calls upon optimization processes [1, 2]. From a robotics control stand point, two classes of methods are used to synthetize such a type of controller:

- projection methods, such as Operational Space Control (OSC) [3] perform their computation in the so-called operational space, a space relative to the task, which is usually smaller than the articular space. They can thus be applied to large robotic systems [4]. They also give rise to a mathematically straightforward way of decoupling a set of tasks ranked by priority, but are sensitive to these priorities and cannot directly perform a global optimization of a cost function associated to the motion of the system over time;
- optimization methods, such as Optimal Control (OC) [5, 6] perform their computation in the articular space, but benefit from the easy definition of the constraints and performance criteria [7] of the tasks that do not need to be ranked.

Recently, Stochastic Optimal Feedback Control (SOFC) methods [5, 8] have received a wide agreement as a good model of human motor control for elementary tasks. The model is based on the idea that human motor control gives rise to some noise in proportion to the amplitude of the control input. Thus a "minimal intervention principle" specifies that, in order to be accurate, the muscular inputs must be as small as possible to reduce the noise [9]. However the computational cost of OC methods make them unsuitable to solve larger problems such

as the synthesis of whole body motion for humanoid robots. OSC approaches are thus still appealing in such contexts.
In this contribution, we propose a hierarchical model combining the assets of both control methods. Our model is consistent with Shadmehr's view [10] according to which the central nervous system starts with computing a global trajectory in the operational space and then dynamically realizes it at the articular level with OC tools. Besides, these control methods call upon a kinematic model and a dynamic model of the system, respectively. We show how these models can be learnt from experience. Finally, the coordination of both control levels also calls upon an anticipatory process, the dynamic control set point being chosen as a point that is forward in time with respect to the current time.
In section 2, we briefly explain how one can learn the kinematic and dynamic models of a system using self-supervised (anticipatory) processes. Then, in section 3, we introduce our controller structure and present some preliminary results about the control of a simplified virtual humanoid (3 DOFs, planar, fixed foot) performing a standing task starting from a crouching posture (obtained using Arboris, a Matlab dynamic simulator [7]).

2 Model learning based on anticipation

The equation of motions of a (bio)mechanical system can be described by a set of state dependent non linear functions relating the future state of the system $\mathbf{x}_{k+1}$ to its current state $\mathbf{x}_k$[1] and control input $\mathbf{u}_k$[2]. This *state-space* representation is written in its discrete and matrix form : $\mathbf{x}_{k+1} = A(\mathbf{x}_k)\mathbf{x}_k + B(\mathbf{x}_k)\mathbf{u}_k$. Given a set of instances ($\mathbf{x}_k$,$\mathbf{u}_k$,$\mathbf{x}_{k+1}$), this non linear model can be approximated linearly using a Least Square procedure.
The Recursive Least Square (RLS) algorithm performs this computation incrementally from experience. RLS is an instance of anticipatory supervised learning process, where the supervision signal is the error between the anticipated output $\hat{\mathbf{x}}_{k+1}$ and the actual ouptut $\mathbf{x}_{k+1}$. Matrices $\hat{A}$ and $\hat{B}$ which are required to perform OC can trivially be extracted from such a process.
When a linear model is not accurate enough or when the number of dimensions of the state space becomes large, a more powerful anticipatory supervised learning algorithm such as LWPR [11] can be used. Unfortunately, extracting matrices $\hat{A}$ and $\hat{B}$ from a model of the dynamics learnt by LWPR is not direct since LWPR calls upon several projection operators. As a matter of fact, in the control method we propose, we use RLS to learn $A(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$.
From an estimation of $A(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$ matrices, we can compute the control input of the system as a function of the joint space error with respect to a target. However, the target being specified in operational space, it is also necessary to

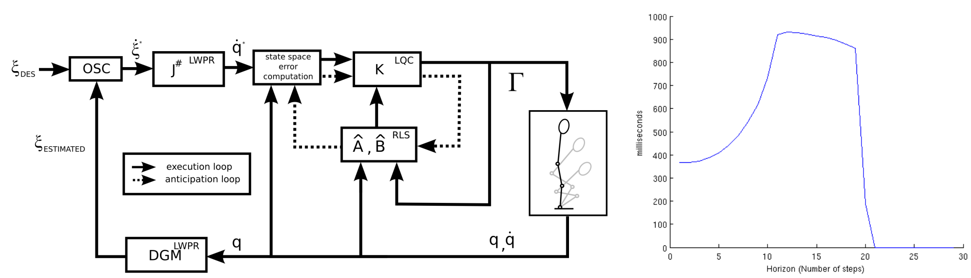
[1] The state of a mechanical system is uniquely defined when the velocity and position of all the parts of the system are known without any ambiguity. An intuitive choice of state variables is $\mathbf{x}_k = \left[\mathbf{q}^T, \dot{\mathbf{q}}^T\right]^T$ where $\mathbf{q}$ is the joints angles vector.

[2] The control input in such systems can be modeled as a vector of motor torques $\boldsymbol{\Gamma}$ applied at the articular level.

perform a transformation from the operational space set point to the joint space. This transformation relies on the kinematics of the system which at the velocity level is modeled as a set of state dependent non linear functions relating the velocity of the operational coordinates $\boldsymbol{\xi}$ to control (the 2D position of the top of the head in our example) to the joints velocity $\dot{\mathbf{q}}$. In its matrix form, this relation can be written $\dot{\mathbf{q}} = J(\mathbf{q})^\sharp \dot{\boldsymbol{\xi}}$, where $J(\mathbf{q})$ is the jacobian matrix associated to the considered task and $J(\mathbf{q})^\sharp$ is a weighted pseudo-inverse of $J(\mathbf{q})$. LWPR is used to learn an approximation of this (pseudo)-inversed jacobian matrix in a way similar to [12]. It is also used to estimate the Direct Geometric Model.

3 Controller structure

Our low level controller (cf. Figure 1 *(a)*) is based on Linear Quadratic Controller (LQC) [13]. In order to learn estimations of $A(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$ matrices with RLS, we first bootstrap these matrices with a standard Least Square method until we get enough data. The learnt state space model is then used to generate with LQC a feedback control gain matrix K_{LQC} applied to the state space error and resulting in an optimal motor input (*ie.* that minimizes the norm of the input vector as well as the state space error).
The high level controller is based on the learnt relation $J(\mathbf{q})^\sharp$ between the velocity of joint space coordinates $\dot{\mathbf{q}}$ and the velocity of the operational space coordinates $\dot{\boldsymbol{\xi}}$ used to describe the task to be performed. Given the desired position of the head in cartesian space, joint space velocity set points are computed (using a Proportional controller on the head position error) and then servoed using the low level controller.



(a) *(b)*

Fig. 1. *(a)*: Our control scheme. *(b)*: Results: stability of the mannequin as a function of the anticipation horizon.

If the servoing is based on the current point on the trajectory, the global controller is not stable enough to accurately maintain the posture of our mannequin: the low-level controller oscillates around the reference trajectory. In order to

stabilize it, we introduce a further anticipation process. The low level controller tracks points that are up to H steps in the future, calling upon the learnt dynamic model to simulate its state up to these points in the future and determine the corresponding control vector as a degressive weighted sum of the anticipated controls (anticipation loop) : $\mathbf{u} = \frac{2\sum_{h=1}^{H} h\mathbf{u}_{H+1-h}}{H\,(H+1)}$.
Figure 1 *(b)* shows how many milliseconds the mannequin keeps standing with its head above 1.4 meters, given different anticipation horizons H. One can see that an horizon between 11 and 13 times steps ahead is optimal. Thus we have shown that anticipation can be used at the same time to learn the kinematic and dynamic models of the system and to stabilize the interactions between both control levels.
These results are preliminary. In the future, we would like to learn $A(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$ with LWPR on a larger system with more degrees of freedom and then combine several tasks at the high level so as to demonstrate the benefits of using OSC at that level.

References

1. Wolpert, D.M., Ghahramani, Z.: Computational principles of mouvement neuroscience. Nature Neuroscience **3** (2000) 1212–1217
2. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks **11**(7-8) (1998) 1317–1329
3. Khatib, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE Journal of Robotics and Automation **3**(1) (1987) 43–53
4. Sentis, L., Khatib, O.: Control of free-floating humanoid robots through task prioritization. In: ICRA. (2005)
5. Todorov, E.: Optimality principles in sensorimotor control. Nature Neurosciences **7**(9) (2004) 907–915
6. Guigon, E., Baraduc, P., Desmurget, M.: Computational motor control: Redudancy and invariance. Journal of Neurophysiology **97**(1) (2007) 331–347
7. Barthelemy, S., Bidaud, P.: Stability measure of postural dynamic equilibrium. In: Romansy. (2008)
8. Todorov, E., Jordan, M.I.: Optimal feedback control as a theory of motor coordination. Nature Neurosciences **5**(11) (2002) 1226–1235
9. Todorov, E., Jordan, M.: A minimal intervention principle for coordinated movement. In: NIPS. (2003) 27–34
10. Shadmehr, R., Wise, S.: The Computational Neurbiology of Reaching and Pointing. MIT Press (2005)
11. Vijayakumar, S., Schaal, S.: Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space. In: International Conference In Machine Learning (ICML). (2000) 1079–1086
12. D'Souza, A., Vijayakumar, S., Schaal, S.: Learning inverse kinematics. In: International Conference on Intelligent Robots and Systems **1** (2001) 298–303
13. Bertsekas, D.P.: Dynamic Programming and Optimal Control. Athena Scientific, Belmont, MA (1995)