An application of receding-horizon neural control in humanoid robotics

Back

Serena Ivaldi[†‡], Marco Baglietto[‡], Giorgio Metta [†‡], and Riccardo Zoppoli[‡]

[†] *Robotics, Brain and Cognitive Science Department, Italian Institute of Technology* {serena.ivaldi,giorgio.metta}@iit.it
[‡] *Department of Communications, Computer and System Sciences, University of Genoa, Italy* {mbaglietto,rzop}@dist.unige.it

Keywords : ERIM, robotics, receding horizon.

Abstract : Optimal trajectory planning of a humanoid arm is addressed. The goal is to make the end effector reach a desired target or track it when it moves in the arm's workspace unpredictably. As a reference setup, we considered a 7 degrees of freedom humanoid robot arm. Physical constraints require the on-line computations to be very quick. Following previous studies [1], a receding-horizon method is proposed that consists in assigning the control function a fixed structure (e.g., a feedforward neural network) where a fixed number of parameters have to be tuned. More specifically, in the off-line phase, a set of neural networks (corresponding to the control functions over a finite horizon) is optimized using the Extended Ritz Method. The training set corresponds to a sampling of the arm and target position and velocity configuration space. The expected value of a suitable cost is minimized with respect to the free parameters in the neural networks. Therefore, a nonlinear programming problem is addressed that can be solved by means of a stochastic gradient technique. The resulting approximate control functions are sub-optimal solutions, but (thanks to the well-established approximation properties of the neural networks) one can achieve any desired degree of accuracy [5]. Once solved the off-line finite-horizon problem, only the first control function is retained in the on line phase: at any sample time t, given the system's state and the target's position and velocity, the control action is generated with a very small computational effort.

1 Introduction

In robotics, the task of positioning the end effectors is fundamental: whenever a robot has to move its arm in order to grasp an object, track a moving target, avoid collision with the environment or just explore it, reaching is involved. Given the target position, estimated for example by a vision system, it is common practice to plan a suitable trajectory in the cartesian space and then find the corresponding joint and torque commands. To this end, a Finite Horizon (FH) optimal control problem can be addressed to improve the robot's motion performance, but it is scarcely useful as generally the duration of the movements cannot be predicted *a priori*. Moreover, moving through a fixed horizon strategy could lead to a lack of responsiveness, whenever the target dynamics is

Int. Workshop on Assessment and Future Directions of NMPC
Pavia, Italy, September 5-9, 2008

too fast and no previous information about it are available to make some prediction. In the classical Receding Horizon (RH) approach, at each time instant t, when the state of the system is $\underline{x}_t$ a FH optimal control problem is solved and a sequence of T optimal control actions is computed, $\underline{u}^{FH}_{t|t}, \underline{u}^{FH}_{t+1|t}, \ldots, \underline{u}^{FH}_{t+T-1|t}$, which minimize a suitable cost function affecting the motion performances; then only the first control vector is applied: $\underline{u}^{RH}_t = \underline{u}^{FH}_{t|t}$. This procedure is repeated stage after stage, thus yielding a feedback control law. Stabilizing properties of RH control have been shown for case of both linear and nonlinear, continuous and discrete time systems, using the terminal equality constraints [8]. Hard constraint $\underline{x}_{t+T} = \underline{0}$ was relaxed by requiring a regulator to drive the system to a neighborhood of the origin [9] in combination with a switching regulator; in [1] the attractiveness of the origin was imposed by means of a penalty function in the cost function. Thereafter the RH control paradigm has been extended to tracking, nonlinear control, and has been widely accepted for industrial applications. The RH classical technique assumes the control vectors to be generated after the solution of a nonlinear programming problem at each time instant: this assumption is unrealistic in the case of humanoid robotics, as the robot's and the target's dynamics are very fast. In this paper we will design a feedback RH regulator for reaching tasks, that must reveal itself to be quick and reactive to changes, in particular able to track a target moving in the robot's workspace in an unpredictable way. We will also describe a technique which concentrates in a off line phase the computation of a time-invariant feedback optimal control law, for every possible system and target states belonging to the set of admissible ones. The proposed algorithm consists of two steps. In the first one, a suitable sequence of neural networks is trained off line, so that they can approximate the optimal solutions of a stochastic FH control problem, which is generalized for every possible state configuration. In the second (online phase), only the first control law is applied, at each time instant. The Extended RItz Method (ERIM) [6] is chosen as a functional approximation technique. The use of feedforward neural networks (thanks to their well known approximation capabilities [7]) guarantees that the optimal solutions can be approximated at any desired degree of accuracy. We remark that the computation is concentrated in the off line phase, while in the on line phase only the computation of the current control is performed, thus yielding a quick response to unpredictable changes in the target's state. The feasibility of this approach has already been tested on the control of a nonholonomic robot [2].

2 The robotic platform

The object of our control scheme is the left arm of the humanoid robot James [4], which is being developed by the University of Genoa and the Italian Institute of Technology. James is a 22 Degrees Of Freedom (DOF) torso, with the overall size of a 10 years old boy and a total weight of about 8 kg. Torque is transmitted to the joints by plastic toothed belts and stainless-steel tendons, actuated by rotary motors. The head, equipped with two eyes, is mounted on a 3 DOF neck. The arm has 7 DOF: 3 in the shoulder, one for the elbow and 3 in the wrist. In particular, the shoulder consists of 3 rotative joints, actuated through tendons and pulleys by 3 motors located in the torso: 2 joints (the ones yielding abduction) are mechanically coupled, so as to gather the shoul-

der a wider range of motion. Low level motor control is distributed on twelve Digital Signal Processing (DSP) cards, communicating with a PC cluster via CAN bus. The robot's motion can be controlled by sending position and velocity commands to the DSP. Most of the motors are then directly controlled by standard PID controllers, except for the shoulder, neck and eyes motors which require different control strategies to handle various mechanical constraints. In the following we shall only focus on the arm motion planning and control, in particular from the shoulder up to the wrist, which will be considered as the end effector of the kinematic chain. We neglect the rotation of the hand. The accurate description of the control architecture and the different low level control strategies fall outside of the scope and the available space of this paper.

2.1 Kinematic model

James robot's model is an open kinematic chain. Let us denote by $\underline{x}_c^r$ the cartesian coordinates of the end effector in the robot's workspace, with respect to a fixed reference frame, and by $\underline{q}^r$ the vector whose components are the joints' coordinates of the arm. Then the forward kinematics $\underline{x}_c^r = f_{arm}(\underline{q}^r)$ can be easily found by Denavit-Hartenberg convention [3]; $f_{arm} : \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_c}$ where $n_q = n_c = 2$ if we consider the arm as a two-link rigid body moving on a planar surface. In the following, we will assume the robot model to be known and kinematic singularities be avoided such that the jacobian matrix $J(\underline{q}^r) = \partial f_{arm}(\underline{q}^r)/\partial \underline{q}^r$, being $\dot{\underline{x}}_c^r = J(\underline{q}^r)\dot{\underline{q}}^r$, is always non-singular. The reaching control problem consists in finding the optimal controls $\underline{u}^\circ$ in the cartesian space, and the corresponding velocity controls in the joint space $\dot{\underline{q}}^{r\circ}$, so that the end effector can reach or track a target, moving unpredictably in the robot's workspace while minimizing some suitable cost function. We shall denote be $\underline{x}^r$ the robot's end effector state vector, and by $\underline{x}^g$ the target's one. A discrete-time (first order Euler) model has been considered. We denote by $\underline{x}_t$, at time instant t, the difference between the end effector and the target cartesian coordinates and velocities ($\underline{x}_t \triangleq \underline{x}_t^g - \underline{x}_t^r$). The goal of the control problem, at time instant t, is to minimize a suitable cost function, which is chosen so as to characterize the trajectories of the end effector. A common choice in humanoid robotics [11] is represented by the minimum jerk principle, describing the criterion used by human beings during movements, that is (in continuous form, for planar movements of the end effector):

$$J = \int_0^{t_f} \left[\left(\frac{d^3 x^r}{dt^3} \right)^2 + \left(\frac{d^3 y^r}{dt^3} \right)^2 \right] dt \ . \tag{1}$$

A more common function cost in automatic controls is instead:

$$J = \sum_{i=t}^{t+T-1} \underline{c}(\underline{u}_i) + \left\| \underline{x}_{i+1} \right\|_{V_{i+1}}^2 \tag{2}$$

where the criterion for the task accomplishment is a tradeoff between the minimization of the energy consumption (for physical limits, it is important not to exceed in current absorption) and the "best" end-effector proximity to the target at the end of the manoeuvre (it could not be able to reach it perfectly, as a consequence of the unpredictable behavior of the target or the robot's intrinsic

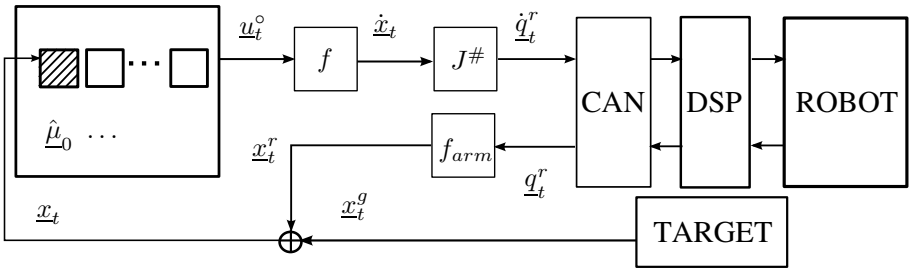

Figure 1: James's arm control scheme. Velocity commands are sent through a CAN bus, while direct motor control is performed by DSP cards. The retrieving of the target's cartesian coordinates is not modeled, as it would require to discuss the robotic visual system.

physical limits). Weight matrices V_i are chosen such as to obtain reasonable compromise between the attractiveness of the target and the energy consumption, whereas $c(u_t^j), j = x, y$ is a nonlinear but convex function (see, e.g., [2]). We remark that once the optimal control $\underline{u}_t^\circ$ is found, then the velocity control in the joint space can be easily computed with standard formulations, i.e., $\dot{\underline{q}}_t^r = J^\#(\underline{q}_t^r)\dot{\underline{x}}_t^{r\circ}$, where $J^\#$ denotes the Moore-Penrose pseudo-inverse of the jacobian matrix. The velocity commands in the joint space, computed by a common PC, are sent through the CAN bus to the DSP cards, where a low level control loop is performed. The control scheme is shown in Figure 1.

3 Receding horizon regulator: a neural approach

Now let us represent the previous equations in the more general and compact form

$$\underline{x}_{t+1} = \underline{f}\,(\underline{x}_t, \underline{u}_t) \ , \ t = 0, 1, \ldots, T-1 \tag{3}$$

where at the time instant t, $\underline{x}_t$ is the state vector, taking values from a finite set $X \subseteq \mathbb{R}^n$, and $\underline{u}_t$ is the control vector, constrained to take values from a finite set $U \subseteq \mathbb{R}^m$. The desired target state, at instant t, is then $\underline{x}_t^* = 0$, meaning that the goal is to bring the difference between the end effector and the target to zero. We remark that by setting the target state to zero, we implicitly apply a certainty equivalence principle: at time instant t, it is supposed that the target vector $\underline{x}^g$ will remain constant for T time instants, that is: $\underline{x}_{t+i+1}^g = \underline{x}_{t+i}^g, i = 0, \ldots, T-1$. We can now state a RH control problem.

Problem 1. *At every time instant $t \geq 0$, find the RH optimal controls $\underline{u}_t^\circ \in U$, where $\underline{u}_t^\circ$ is the first vector of the control sequence $\underline{u}_{t|t}^\circ, \ldots, \underline{u}_{t+T-1|t}^\circ$ that minimize the FH cost functional*

$$J(\underline{x}_t) = \left\{ \sum_{i=t}^{t+T-1} h_i(\underline{x}_i, \underline{u}_{i|t}) + h_T(\underline{x}_{t+T}) \right\} \ . \tag{4}$$

The classical RH control assumes that at each time instant of control a FH control problem is solved, and a sequence of T optimal controls is found. This approach is not suitable in humanoid robotics, as there is no sufficient time to compute the optimal control sequence at each sampling time. Therefore we will change the problem's formulation so as to be able to compute the control laws in an off line phase.

Problem 2 (RH)**.** *For every time instant $t \geq 0$, find the RH optimal control law $\underline{u}_t^\circ = \underline{\mu}_t^\circ(\underline{x}_t) \in U$, where $\underline{\mu}_t^\circ$ is the first control function of the sequence $\underline{\mu}_{t|t}^\circ, \ldots, \underline{\mu}_{t+T-1|t}^\circ$ that minimize the FH cost functional*

$$\bar{J} = \underset{\underline{x}_t \in X}{E} \left\{ \sum_{i=t}^{t+T-1} h_i(\underline{x}_i, \underline{\mu}_{i|t}^\circ(\underline{x}_i)) + h_T(\underline{x}_{t+T}) \right\} \tag{5}$$

Thanks to the time invariance of the systems dynamics and of the cost function, $t = 0$ can be considered as a generic time instant. Then, a single (functional) FH optimization problem is addressed.

Problem 3 (FH)**.** *Find a sequence of optimal control functions $\underline{\mu}_0^\circ, \ldots, \underline{\mu}_{T-1}^\circ$, that minimize the cost functional*

$$\bar{J} = \underset{\underline{x}_0}{E} \left\{ \sum_{i=0}^{T-1} h_i(\underline{x}_i, \underline{\mu}_i^\circ(\underline{x}_i)) + h_T(\underline{x}_T) \right\} \tag{6}$$

subject to the constraints $\underline{\mu}_i^\circ \in \underline{U} \subseteq \mathbb{R}^m$ and (3).

The RH control strategy will correspond to use $\underline{\mu}_0^\circ$ as a time invariant control function, i.e., to apply $\underline{u}_t = \underline{\mu}_0^\circ(\underline{x}_t)$.

3.1 From a functional optimization problem to a nonlinear programming one

In order to solve Problem FH we shall use the ERIM [6], which turns the functional optimization problem into a nonlinear programming one. More specifically, we constrain the admissible control functions $\underline{\mu}_0, \underline{\mu}_1, \ldots, \underline{\mu}_{T-1}$ to take on a fixed parametrized structure, in the form of one-hidden-layer (OHL) networks:

$$\underline{\hat{\mu}}(\underline{x}, \underline{\omega}_\nu) = col \left[\sum_{i=1}^{\nu} c_{ij} \varphi_i(\underline{x}, \underline{\kappa}_i) + b_j \right] \tag{7}$$

where $\underline{\hat{\mu}}(\cdot, \underline{\omega}_\nu) : \mathbb{R}^n \times \mathbb{R}^{N(\nu)} \mapsto \mathbb{R}^m$, $c_{ij}, b_j \in \mathbb{R}, \underline{\kappa}_i \in \mathbb{R}^k, j = 1, \ldots, m$. The finite number of free parameters, N (called *basis cardinality number of the OHL network*), grows linearly with ν, that is the number of *neurons* constituting the network. By substituting (7) into (6), calling $\underline{\omega}_i$ the parameters of the i-th OHL network $\underline{\hat{\mu}}_i(\underline{x}_i, \underline{\omega}_i)$ (for the sake of simplicity we write simply $\underline{\omega}_i$ instead of $\underline{\omega}_{\nu_i}$), the general functional cost $\bar{J}(\underline{\mu}_0, \underline{\mu}_1, \ldots, \underline{\mu}_{T-1})$ is turned into a function which is only dependent on a finite number of real variables, $\hat{J}_\nu(\underline{\omega})$, where $\underline{\omega} = col(\underline{\omega}_i, i = 0, 1, \ldots, T-1)$ is the vector in which all the parameters to be optimized are collected. We can now restate Problem 3 as:

Problem 4 (FH_ν). *Find the optimal parameters* $\underline{\omega}_0^\circ, \ldots, \underline{\omega}_{T-1}^\circ$ *that minimize the cost functional*

$$\hat{J}_\nu = \underset{\underline{x}_0}{E} \left\{ \sum_{i=0}^{T-1} h_i(\underline{x}_i, \hat{\underline{\mu}}_i^\circ(\underline{x}_i, \underline{\omega}_i)) + h_T(\underline{x}_T) \right\} \tag{8}$$

subject to the constraints $\hat{\underline{\mu}}_i^\circ(\underline{x}_i, \underline{\omega}_i) \in \underline{U} \subseteq \mathbb{R}^m$ *and (3).*

Then, for every time instant t the time-invariant RH control law corresponds to $\underline{u}_t^\circ = \hat{\underline{\mu}}^{RH}(\underline{x}_t, \underline{\omega}_0^\circ) = \hat{\underline{\mu}}_0^\circ(\underline{x}_t, \underline{\omega}_0^\circ)$ (see Problem FH_ν).

The computation of the time invariant feedback control law is concentrated in the off line phase. Hence, at any time instant t only the first computed neural network is used. This approach is particularly efficient in real time, because the computation of the new control action is quick, consisting only in few mathematical operations; moreover, it can be performed by memory-limited electronic boards, as it only requires the memorization of the parameters of one single network. Of course, the constraints on the admissible values of $\underline{x}_t$ and $\underline{u}_t$ are taken into account. To be more precise, the classical OHL networks were slightly modified, specifically by adding two sigmoidal functions $\sigma(z) = \tanh(z)$, bounded by the values $[-U, U]$, to the final output layer: with this choice, the constraints on the control values can be removed from the problem formulation since the neural networks already embed them.

3.2 Solution of the nonlinear programming problem by stochastic gradient

The optimal parameters in the OHL control functions can be found by a usual gradient algorithm, i.e.

$$\underline{\omega}_i(k+1) = \underline{\omega}_i(k) - \alpha(k) \nabla_{\underline{\omega}_i} \underset{\{\underline{x}_0\}}{E} \left\{ \hat{J}_\nu \left[\underline{\omega}_i(k), \underline{x}_0(k) \right] \right\}, \; k = 0, 1, \ldots \tag{9}$$

where at step k the stochastic variable $\underline{x}_0$ is generated randomly on the basis of the known probability density functions. Within this context, it is impossible to calculate exactly all the gradient components, because of the stochastic nature of $\underline{x}_0$; then, instead of the gradient $\nabla_{\underline{\omega}} E \left[\hat{J}_\nu(\underline{\omega}, \underline{x}_0) \right]$ a "realization" $\nabla_{\underline{\omega}} \hat{J}_\nu(\underline{\omega}, \underline{x}_0)$ is computed, where the stochastic variable $\underline{x}_0$ is generated according to its probability distribution. Then a simple gradient steepest descent algorithm can be applied:

$$\underline{\omega}_i(k+1) = \underline{\omega}_i(k) - \alpha(k) \nabla_{\underline{\omega}_i} \hat{J}_\nu \left[\underline{\omega}(k), \underline{x}_0(k) \right] + \eta(\underline{\omega}(k) - \underline{\omega}(k-1)) \tag{10}$$

for $k = 0, 1, \ldots$, where we added a regularization term, weighted by $\eta \in [0, 1]$, as it is usually done when training neural networks. The convergence of the method, which is known as *stochastic gradient*, is assured by a particular choice of the step size $\alpha(k)$, that must fulfill a set of conditions [10]. Of course, one has to compute the partial derivatives of the cost $\hat{J}_\nu$ with respect to the parameters to be optimized, $\underline{\omega}_i$:

$$\frac{\partial \hat{J}_\nu}{\partial \underline{\omega}_i} = \frac{\partial \hat{J}_\nu}{\partial \underline{u}_i} \frac{\partial \hat{\underline{\mu}}_i(\underline{x}_i, \underline{\omega}_i)}{\partial \underline{\omega}_i} \; . \tag{11}$$

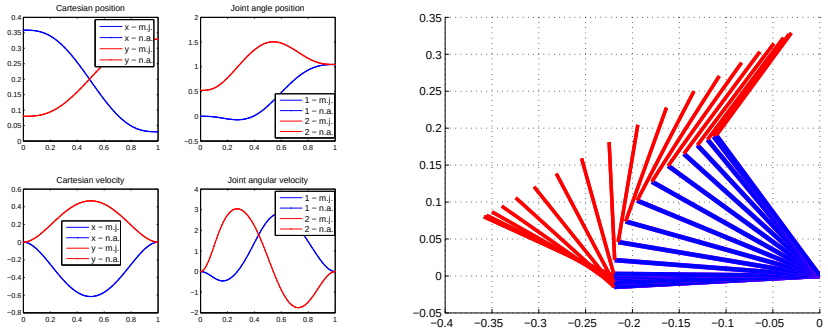

Figure 2: A minimum jerk movement of James'arm: cartesian and joints position and velocity are shown, as well as samples of the plaanr trajectory.

The proposed algorithm for the computation of the optimal parameters consists in two phases, a forward and a backward one, and in a backpropagation technique. In the *forward phase* we "unroll" the system and the neural controllers in time, making the feedback explicit. At iteration step k, given the initial state $\underline{x}_0$, we compute all the state and controls generated by the sequence of OHL networks that is $\underline{u}_t = \hat{\underline{\mu}}_t(\underline{x}_t, \underline{\omega}_t(k))$, given $\underline{x}_0, \underline{x}_t = f(\underline{x}_{t-1}, \underline{u}_{t-1})$, $t = 1, \ldots, T$. Then we can compute all the partial costs $h_t(\underline{x}_t)$, $h_T(\underline{x}_T)$. In the *backward phase*, we compute all the gradient components and "back-propagate" them through the networks' chain. The recursive propagation is described by the following equations, for $t = T-1, T-2, \ldots, 0$:

$$\frac{\partial \hat{J}_\nu}{\partial \underline{u}_t} = \frac{\partial h_t(\underline{x}_t, \underline{u}_t)}{\partial \underline{u}_t} + \frac{\partial \hat{J}_\nu}{\partial \underline{x}_{t+1}} \frac{\partial f(\underline{x}_t, \underline{u}_t)}{\partial \underline{u}_t} \tag{12}$$

$$\frac{\partial \hat{J}_\nu}{\partial \underline{x}_t} = \frac{\partial h_t(\underline{x}_t, \underline{u}_t)}{\partial \underline{x}_t} + \frac{\partial \hat{J}_\nu}{\partial \underline{x}_{t+1}} \frac{\partial f(\underline{x}_t, \underline{u}_t)}{\partial \underline{x}_t} + \frac{\partial \hat{J}_\nu}{\partial \underline{u}_t} \frac{\partial \hat{\underline{\mu}}_t(\underline{x}_t, \underline{\omega}_t)}{\partial \underline{x}_t} \tag{13}$$

initialized by $\partial \hat{J}_\nu / \partial \underline{x}_T = \partial h_T(\underline{x}_T) / \partial \underline{x}_T$.

4 Results

Numerical results have been found for both aforementioned cost functions: (1) and (2). Some FH results are shown in Figure 2, while RH trajectories during a tracking/reaching task are shown in Figure 3. In particular, "neural" trajectories minimizing (1) were compared with the analytical solution of the minimum jerk trajectories (that is one of the main reasons we have firstly addressed the control problem for a two-link planar arm), proving the effectiveness of the proposed approach. We point out that the property of computing controls in real time as fast as possible is a strict requirement, because in the future the integration with a visual feedback will be addressed. In the proposed framework, singularities and redundancies of the kinematic chain have been neglected. In the future they will be taken into account; moreover, direct control in joint space will be addressed and delays will be modeled.

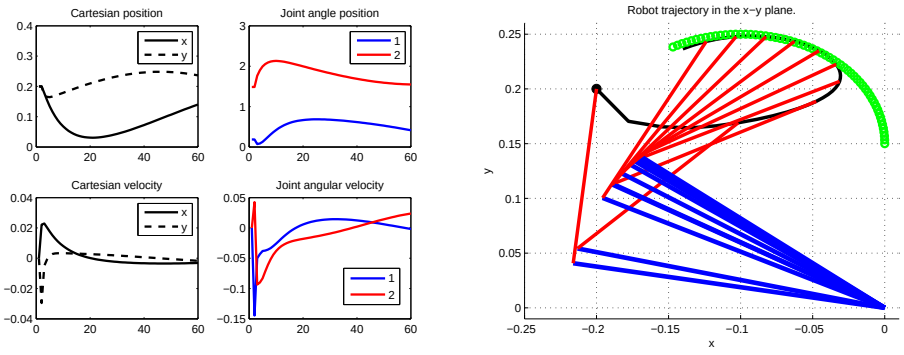

Figure 3: James' left end effector (black) tracking a target (green) moving in an unpredictable way, according to cost function (2).

5 Acknowledgment

This work is supported by the ROBOTCUB project (IST-2004-004370), funded by the European Commission through the Unit E5 "Cognitive Systems".

References

[1] T. Parisini and R. Zoppoli. A receding horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31:1443–1451, 1995.

[2] S. Ivaldi, M. Baglietto and R. Zoppoli. Finite and receding horizon regulation of a space robot. *Int. Conf. on Mathem. Probl. in Engin., Aerospace and Sciences*. Genoa, Italy, 2008.

[3] L. Sciavicco and B. Siciliano. Modeling and Control of Robot Manipulators, 2nd Edition. *Springer-Verlag*. London, UK, 2000.

[4] L. Jamone, G. Metta, F. Nori and G. Sandini. James: a humanoid robot acting over an unstructured world. *Int. Conf. on Humanoids Robots*. Italy, 2006.

[5] V. Kurkova and M. Sanguineti. Error estimates for approximate optimization by the Extended Ritz method. *SIAM J. on Optimization*, 15:461-487, 2005.

[6] R. Zoppoli, M. Sanguineti and T. Parisini. Approximating networks and Extended Ritz Method for the solution of functional optimization problem. *Journ. of Optim. Theory and Applications*, 112:403-439, 2002.

[7] A. Barron. Universal approximation bounds for superposition of a sigmoidal function. *IEEE Trans. on Information Theory*, 39:930–945, 1993.

[8] S. Keerthi and E. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *Journ. of Optim. Theory and Applications*, 57:265–293, 1988.

[9] H. Michalska and D. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. on Automatic Control*, 38:1623–1633, 1993.

[10] H.J. Kushner and J. Yang. Stochastic approximation with averaging and feedback: rapidly convergent "on-line" algorithms. *IEEE Trans. on Automatic Control*, 40:24–34,1995.

[11] M.J. Richardson and T. Flash. On the Emulation of Natural Movements by Humanoid Robots. *Int. Conf. on Humanoids Robots*. Boston, USA, 2000.