

Generating Human-like Reaching Movements with a Humanoid Robot : A Computational Approach[☆]

Michel Taïx^{a,b}, Minh Tuan Tran^a, Philippe Souères^a, Emmanuel Guigon^c

^aLAAS-CNRS, Univ. of Toulouse, 7 Av. du Colonel Roche, F-31077 Toulouse, France

^bUniv. Paul Sabatier, 118 Route de Narbonne, F-31062 Toulouse, France

^cISIR, UPMC/CNRS, 4 place Jussieu, University Paris 6, F-75252 Paris, France

Abstract

This paper presents a computational approach for transferring principles of human motor control to humanoid robots. A neurobiological model, stating that the energy of motoneurons is minimized and that dynamic and static efforts are processed separately, is considered. This paradigm is used to produce humanoid robots reaching movements obeying the rules of human kinematics. A nonlinear programming problem is solved to determine optimal trajectories. The optimal movements are then encoded by using a basis of motor primitives determined by principal component analysis. Finally, generalization to new movements is obtained by solving of a low-dimensional optimization problem in the operational space.

Keywords: human motor control, motor primitives, humanoid robot, nonlinear programming, optimization.

1. Introduction

Even though today's humanoid robots roughly have the same shape as humans, they are strongly different in their mechanical structure, their sensing and actuation capabilities and the way they process data. However, despite these differences, we want to show in this paper that it is possible to make these machines

[☆]An overview of this collaborative work between roboticists and neuroscientists was presented to the International Symposium on Scientific Computing for the Cognitive Sciences, in Heidelberg, Germany, in October 2010

Email addresses: taix@laas.fr (Michel Taïx), soueres@laas.fr (Philippe Souères), emmanuel.guigon@upmc.fr (Emmanuel Guigon)

move in a human-like way by applying computational principles of human motor control. We focus on the control of reaching movements, which is the topic of an extensive literature in computational neurosciences [39].

In humans, the question is to determine how the central nervous system (CNS) processes to control the arm muscles in order to drive the hand to a given target position. As initially pointed out by Bernstein [3], the difficulty of this question comes from the fact that, due to the numerous degrees of freedom (DoF) of the arm and the important number of muscles involved in the reaching movement, the system is highly redundant with respect to the task. As a consequence, infinitely many control can be used to perform the reaching movement. The question is then to determine how the CNS process to choose one specific control solution.

In humanoid robotics, as the arm joints are usually driven by DC motors, the redundancy does not yet concern muscle actuation but the redundancy still exists at the kinematic and the dynamic level. Indeed, only 3DoF are necessary for positioning the hand at the target position whereas the robot arm usually includes at least 6 DoF. The question is then do determine how to control these joints in order to make the humanoid robot move in a human-like way.

To answer this question different kinds of biologically inspired approaches have so far been proposed in robotics. Some authors made a special effort to reproduce the mechanical musculoskeletal structure of the human arm and develop control strategies involving the dynamics of internal forces [42], or replicate biological patterns of muscle activity [32]. Khatib et al. [23] used a prioritized task-level control framework to generate whole-body movements with a dynamic musculoskeletal model of human. Based on simulations performed with this approach they suggested that human postural motion minimize a muscle effort potential. Other authors tried to characterize trajectories by minimizing a particular cost function such as variance [40], torque [27], jerk [38], or energy [1]. A trajectory generation method based on a neural time-base generator was proposed in [43] to reproduce constrained reaching movements having human-like velocity profiles for rehabilitation. A biologically inspired hybrid robot controller allowing to regulate the movement both in the joint space and in the Cartesian space was proposed in [19] for executing reaching movements. This concept of multi-referential controller was more recently improved in two directions by Pattacini et al. [34]. These authors replaced the vector integration to endpoint (VITE) model with a bio-inspired generation scheme to obtain trajectories with minimum-jerk profile, and used an interior point optimization technique to solve the inverse kinematics under constraints. This approach was successfully experiments on the iCub to execute reaching movements having the main characteristics of human move-

ments. Finally, motion capture and retargeting techniques were used by numerous authors to execute human-like motions [26], [35], [13]. However, beyond imitation, the application of biological motor principles seems to be a way to produce canonical movements in any situation.

In this paper we consider a new approach based on a recent computational theory of motor control [14], [15]. We propose to apply this theory, which can accurately account for kinematics, kinetics, muscular, neural, and stochastic characteristics of redundant movements, to control the humanoid robot HRP-2. The approach is grounded on the idea that the CNS processes dynamic efforts (inertial, velocity dependent) and static efforts (elastic, gravitational) separately, and that the energy of motoneurons is continuously minimized during motion. In order to apply these principles to the control of HRP-2, a global model is considered, which contains the dynamics of the six degrees-of-freedom (DoF) robot arm and includes, for each DoF, an additional filter simulating the dynamics of a pair of virtual antagonist muscles. The computation of optimal trajectories is based on nonlinear programming. A direct transcription method is used to transform the original problem into a discretized version that is solved by using the interior point method implemented in the Ipopt software [52]. We show that the robot arm movements obtained with this approach exhibit the main kinematic features of human motions, namely: quasi rectilinear hand trajectories and bell-shaped single peak velocity profiles.

The strengths and weaknesses of the method are then considered in the mid-term outcome analysis section. Though the method allows to generate human-like reaching movements in a canonical way, the high computational time is pointed out as an important drawback of the method. Indeed, several minutes are necessary to compute certain movements. To cope with this problem, we show in the second part of the paper that the theory of motor primitives can be applied to reduce the computation time. This theory, which is one of the most important in motor control neurosciences, conjectures that, instead of recomputing each time a complex optimization problem, the CNS uses a finite basis of functions to generate movements [37, 29, 8, 50]. In order to develop a sufficiently generic method we based our study on two databases of movement. The first one was obtained from simulation by applying Guigon's model [14] on the dynamic model of HRP2's arm. The second database was obtained by recording the reaching movements of different human subjects by using a motion capture system. From both databases, primitives are extracted by using principal component analysis (PCA). The reconstruction process, which enables to express each movement as a linear combination of these primitives, is analyzed. We prove that twenty primitives al-

low to represent the movements of both databases with a very good accuracy. An original generalization method is then proposed to generate new movements from this basis of primitives. The method consists in solving a low-dimensional minimization problem to determine the weighting coefficients for which the movement approaches the minimum-jerk trajectory with the best precision. Depending on the number of primitives, the computation-time and the precision of the reconstruction processes, are discussed. We show that the use of motor primitives allows to strongly reduce the computation time of trajectories. Examples of reaching movements computed with this method and executed on HRP-2 are finally presented.

2. The biological principles of motor control

Breakthroughs into the understanding of motor functions have generally been brought about by computational studies that disclose functioning principles independent of brain structures or neural processes. The model proposed by Guigon et al. [14], [15] provides a unified account of motor behavior by making the hypothesis that motor control is mainly governed by the following four principles.

Separation principle: There is a substantial experimental evidence to support the idea that static and dynamics forces are processed separately by the CNS. Psychophysical studies have shown that velocity profiles remain unchanged when moving in a known constant or elastic force field, but that they are in general modified by time and amplitude scaling in velocity-dependent and inertial fields [36]. EMG studies have revealed additive velocity-independent, tonic and velocity-dependent, phasic components which have been related to the generation of anti-gravity torques and dynamic torques, respectively [53]. A similar additive combination between tonic and phasic activity was observed in neurons of primate motor cortex [20]. Finally, experiments have shown that the terminal posture of 3D redundant movements is independent of velocity [30]. As the relative contribution of anti-gravity and dynamic torques varies with velocity, optimization of the total torque pattern would predict variations of terminal posture with velocity. This result suggests that dynamic forces are optimized independently of static forces.

Optimal feedback control principle: The CNS processes an optimal control of dynamic forces that is appropriate for an online regulation of movements. Though optimal control has been repeatedly used to account for many aspects of motor control, e.g. trajectory formation, muscular redundancy, postural control, locomotion, etc. [11], [51], [16], it has rarely been applied to the case of nonlinear redundant systems [47]. The difficulty is to account for the simultaneous con-

trol of posture and movement. Most studies did not consider the case of static forces due to the difficulties to solve optimal control problems in the presence of gravitational forces [45]. When a movement consists of a transition between two equilibrium postures, the boundary conditions of the optimal control problem should specify terminal equilibrium signals, e.g. muscle forces which compensate for applied static (elastic, gravitational) forces. The idea to add to the cost function a term which enforces given initial and final equilibrium postures should lead to solutions which depend on the level and nature of the static forces [16]. In contrast, the previously introduced separation principle provides a way to apply optimal feedback control to kinematic redundancy problems with static forces, as there is no a priori specification of the final posture of the limb.

Maximum efficiency: The energy of the signals of motoneurons, that eventually generate the dynamic forces, is continuously minimized along the motion. The system attempts to reach the goal with zero error and minimal control signals. Note that, compared to other cost functions encountered in the motor control literature (jerk [11], torque change [51], variance [16], energy [1], etc.) this cost function is easily measurable by the CNS. Furthermore, the constraint functions used in this model are the initial and final boundary conditions that lead to an univocal description of motor control, contrary to models involving related cost function such as error/effort minimization [47].

Constant effort principle : The idea of motor behavior being associated with the minimum of a cost function is appropriate when both movement amplitude and duration are specified. Otherwise, infinitely slow/fast or infinitely short/long movements could result. The constant effort principle states that a given set of instruction is equivalent to a level of effort. For these instructions, movements of different amplitudes, directions, or against different loads are executed with the same effort.

Modeling studies dealing with 3D movements are rare [48], [6], and address only the kinematic level (trajectory formation). The main interest of the present approach is to consider optimization directly at the level of motor commands, which provides a proper predictive account of motor control [47, 14, 15].

3. Model and Problem Statement

The separation principle stated in the preceding section can be easily applied to control robots such as HRP-2, in which each joint is independently regulated to a reference angular value. Indeed, thanks to the robustness of the low-level control

of each joint we can make the hypothesis that static forces are exactly counterbalanced. The synthesis of a controller can then be obtained by disregarding static effects. In this way, the optimization problem can be solved on the basis of a simplified model describing the phasic component only. In order to implement this strategy on the robot, we developed a global model that contains the dynamics of the $n = 6$ DoF arm and includes, for each DoF, the dynamics of the neuromuscular system associated to a pair of virtual antagonist muscles. In this way, the considered system input is the vector of signals of the virtual motoneurons. Each muscle i ($1 \leq i \leq 2n$) is controlled by a motoneuron and, according to [18], the set constituted by the motoneuron and the muscle (neuromuscular system) can be described by a second-order low-pass filter having the neural control signal u_i as input and the muscular force F_i as output, according to the following scheme

$$\begin{aligned} v(de_i/dt) &= -e_i + u_i \\ v(da_i/dt) &= -a_i + e_i \\ F_i &= \eta(a_i), \end{aligned} \quad (1)$$

where v is a time constant. The function η (used to express the fact that a muscle exerts a pulling force only) is defined by $\eta(z) = z$ if $z > 0$; otherwise, $\eta(z) = 0$. The variables e_i and a_i correspond to excitation and activation parameters¹ respectively. As $\eta(z)$ is not differentiable at the origin, it can be replaced by the function $z \rightarrow \log[1.0 + \exp(\kappa z)]/\kappa$, with $\kappa > 0$. Torques τ_k were calculated at each DoF from the difference between the forces generated by antagonist muscles scaled by a coefficient γ_k (in meters), as described in (2).

$$\tau_k = \gamma_k(F_{2k-1} - F_{2k}), \quad k = 1, \dots, n. \quad (2)$$

The robotics arm of HRP-2 (see figure 1) has six DoF: three at the shoulder, two at the elbow and one at the hand (see [22] for details), the grasping DoF of the hand being not considered here. We used the Lagrangian approach to express

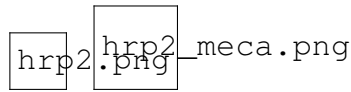


Figure 1: Humanoid robot HRP-2 and its mechanical structure.

¹The Electromyographic activity (EMG) usually corresponds to $\eta(e_i)$

the equation of dynamics of this rigid, multi-linked, articulated system [41]. The computation was done through the symbolic calculus tools in Matlab. On this basis, the equivalent C code was generated for the optimization program described in section 4. The arm dynamics can be expressed under the usual form

$$\tau = M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) \quad (3)$$

where M is the inertia matrix, N is a nonlinear vector including Coriolis effects, G is a vector related to gravity efforts, τ is the torque vector that generates the movement, q , \dot{q} , \ddot{q} are the vectors of angular position, velocity and acceleration of successive joints respectively. According to the separation principle, for the computation of optimal trajectories, the term $G(q)$ which corresponds to static efforts can be removed from (3). So, the relationship between angular accelerations and torques can be expressed as follows

$$\ddot{q} = M^{-1}(\tau - N\dot{q}) \quad (4)$$

By gathering the arm dynamics described by (4) with the actuation dynamics associated to each of the six pairs of virtual muscles described by (1) we obtain a 36 dimensional dynamical system of the form

$$\dot{x} = f(x(t), u(t)) \quad (5)$$

whose state is defined by

$$\begin{aligned} x &= (x_1, \dots, x_{36})^T \\ &= (q_1, \dots, q_6, \dot{q}_1, \dots, \dot{q}_6, a_1, \dots, a_{12}, e_1, \dots, e_{12})^T \end{aligned}$$

where q_1, \dots, q_6 are the joint coordinates, $\dot{q}_1, \dots, \dot{q}_6$ are the angular velocities, and for $i = 1, \dots, n$, (a_{2i-1}, a_{2i}) and (e_{2i-1}, e_{2i}) represent the activation and excitation parameters respectively, associated to the pair of antagonist muscles corresponding to the i th DoF. The control vector, $u = (u_1, \dots, u_{12})^T$ is of dimension $2n = 12$. For $i = 1, \dots, n$, (u_{2i-1}, u_{2i}) represent the motoneurons signals associated to the pair of antagonist muscles corresponding to the i th DoF. Note that the derivatives of the state variables x_i are computed as follows: for $i = 1, \dots, 6$, $\dot{x}_i = \dot{q}_i = x_{i+6}$, for $i = 7, \dots, 12$, \dot{x}_i are given by (4) and for $i = 13, \dots, 36$, \dot{x}_i are deduced from (1).

Remark 1. *The control vector $u = (u_1, \dots, u_{12})^T$ is then the input to the global system that includes, in cascade, the upstreaming low-pass filtering action of the virtual muscles and the dynamics of the robot arm.*

Given an initial arm configuration $x^0 = x(t_0)$, at time t_0 , and a target position of the hand at time final T , and according to the optimal feedback control principle and the maximum efficiency principle, the trajectory of the global system must minimize the energy of motoneurons along the time interval $[t_0, T]$. This problem can be stated as an optimal control problem as follows: find a deterministic control $u(t) = \{u_i(t)\}$ ($1 \leq i \leq 2n$) over $[t_0, T]$ such that $x(t)$ is a solution of (5) satisfying the following boundary conditions

$$x(t_0) = x^0 \text{ and } \psi(x(T)) = 0, \quad (6)$$

while minimizing the quantity

$$E = \sum_{i=1}^{2n} \int_{t_0}^T u_i^2(t) dt \quad (7)$$

In relation (6), $\psi(x(T))$ expresses the constraint on the state at time T , which corresponds to the specified final hand position at which $\dot{q}(T) = a(T) = e(T) = 0$. As explained in the next section, this final condition can be deduced from the expression of the direct kinematics at time T . Considering that the robot arm is at rest at t_0 , the initial condition is given by

$$x^0 = (q_1^0, \dots, q_6^0, 0, \dots, 0)^T \quad (8)$$

where q_1^0, \dots, q_6^0 are the initial angular values of the robot joints at t_0 .

4. Optimization

To solve the two-point boundary-value problem stated by (5), (6), (7) we used a direct transcription method [5]. Following this approach, the original problem can be transformed into a discretized version which then can be solved by a large-sparse NonLinear Programming (NLP) method, such as the interior point method implemented in the Ipopt software [52]. The main steps of the method are presented in the sequel, more details can be found in [5].

4.1. Direct Transcription

The first step of the direct transcription approach is to determine the vector of variables of the corresponding NLP problem. It consists in discretizing the time interval of movement duration $[t_0, T]$ into $m + 1$ time-points as follows

$$t_0 < t_1 < t_2 < \dots < t_J = T$$

At each time-point, the state vector and the control vector are both specified. As a consequence, the discretized version of the variable vector has the form

$$\begin{aligned} \chi^T = & (x_1^0, \dots, x_{36}^0, u_1^0, \dots, u_{12}^0, \\ & x_1^1, \dots, x_{36}^1, u_1^1, \dots, u_{12}^1, \\ & \dots, \\ & x_1^J, \dots, x_{36}^J, u_1^J, \dots, u_{12}^J) \end{aligned} \quad (9)$$

where, for $i = 1, \dots, 36$; $k = 1, \dots, 12$ and $j = 0, \dots, J$; x_i^j and u_k^j are respectively the state variables and the control inputs at time t_j . This vector χ includes $l = (J + 1) \times (36 + 12)$ variables.

The next step is to represent the constraints. Here we have two types of constraint, one on the dynamics (5) and another one on the ψ function (6). The description of variables in (9) conducts to the following trapezoidal representation of dynamical constraints

$$\begin{aligned} x_i^{j+1} - x_i^j - \frac{h^j}{2}(f_i^j + f_i^{j+1}) &= 0 \\ i = 1, \dots, 36, j = 0, \dots, J-1 \end{aligned} \quad (10)$$

where $h^j = t_{j+1} - t_j$ is the duration of the time interval $[t_j, t_{j+1}]$. So, the number of dynamical constraints is equal to $d = m \times 36$. The condition $\psi(x(T)) = 0$, which expresses the fact that the hand must reach the target at the final time $t_j = T$, is defined by the following equations

$$|g(x_1^J, x_2^J, x_3^J, x_4^J, x_5^J, x_6^J) - P_{target}| = 0 \quad (11)$$

$$x_7^J = x_8^J = \dots = x_{36}^J = 0 \quad (12)$$

where g is the direct kinematics function which calculates the position of the hand in Cartesian coordinates from the angular value of each link, and P_{target} is the desired target point. The constraints corresponding to the initial boundary conditions (8) are represented by

$$\begin{cases} \text{for } i = 1, \dots, 6, & x_i^0 = q_i^0 \\ \text{for } i = 7, \dots, 36, & x_i^0 = 0 \end{cases} \quad (13)$$

One important advantage of the direct transcription method is that additional constraints can be easily taken into account by introducing bounds on the state variables. For the case of the HRP-2 arm, the angular constraints are as follows

$$\text{for } i = 1, \dots, 6, \quad \underline{x}_i < x_i^0, x_i^1, \dots, x_i^J < \bar{x}_i \quad (14)$$

Finally, the cost to minimize in (7) can be expressed as a finite sum computed at all discretized time points

$$E(\chi) = \sum_{i=1}^{2n} \sum_{j=0}^J (u_i^j)^2 \quad (15)$$

4.2. NLP Solving

By (9)–(15) we have formulated a large-sparse NLP problem from an optimal control problem, which has the form

$$\begin{aligned} \min_{\chi \in \mathbb{R}^l} E(\chi) & \quad (16) \\ \text{such that : } \underline{c} \leq c(\chi) \leq \bar{c} \\ \underline{\chi} \leq \chi \leq \bar{\chi} \end{aligned}$$

where $E(\chi) : \mathbb{R}^l \rightarrow \mathbb{R}^+$ is the objective function, and $c(\chi) : \mathbb{R}^l \rightarrow \mathbb{R}^d$ are the constraint functions. The vectors \underline{c} and \bar{c} denote the lower and upper bounds on the constraints, and the vectors $\underline{\chi}$ and $\bar{\chi}$ are the bounds on the variables χ . For the considered problem, the vector $\bar{\chi}$ is defined by (9), $E(\chi)$ is given by (15), $c(\chi)$ correspond to the functions in (10) and (11), where $\underline{c} = \bar{c} \equiv 0$, and the constraints on χ are represented by (12) and (13).

To solve the NLP problem, we used the Ipopt solver [52], which turned out to be efficient in terms of accuracy and convergence time. In practice, for the direct transcription, we took $J = 50$ or 100 for movements of 1 second and $J = 100$ or 200 for movements of 2 seconds, that correspond to 2448, 4848 and 9648 variables respectively.

5. First simulation results

In order to generate a movement, the algorithm requires as input the initial angular configuration of the arm, the movement duration, and the position of the desired target expressed in a system of spherical coordinates (r, θ, ϕ) centered at the initial hand position. In this definition, r is the movement amplitude, θ and ϕ are the azimuth and elevation angles respectively. This representation is well appropriate for the description of movements starting with the same initial arm configuration and directed toward peripheral hand target positions. The muscle parameters provided to the system were chosen in the same way as in [14]:

$\nu = 0.05s$ (time constant of muscle filtering), and $\kappa = 10$ (force generation characteristics of the muscle). Moreover, according to (2), at each DoF the link between the force generated by the pair of muscles and the torque is related to a coefficient γ_k . The choice of these coefficients is somewhat arbitrary and discussed in [14]. In our program, we did not try to find the best values of γ_k for all movements. The durations of movements were adapted to the robot dynamics, they were a little bit increased with respect to durations in [14]. Figures 2 and 3 represent the shape of trajectories of amplitude 25cm and duration 1 second, ending at ten evenly spaced peripheral hand positions in the frontal and in the sagittal plane respectively. These target positions correspond to nonsingular configurations of the robot. The initial configuration of the robot arm is $(5^\circ, -15^\circ, 30^\circ, -110^\circ, 0^\circ, 0^\circ)$. Figure 4 shows

Figure 2: Movements of amplitude $r = 25cm$, starting from the same arm configuration, and ending at ten evenly spaced peripheral targets in the frontal plane. The same color is used to represent the hand trajectory (left picture) and the corresponding velocity profile (right picture).

Figure 3: Hand movements of amplitude $r = 25cm$, starting from the same arm configuration, and ending at ten evenly spaced peripheral targets in the sagittal plane. The same color is used to represent the hand trajectory (left picture) and the corresponding velocity profile (right picture).

in detail the variation of motion parameters during a simulated movement of the hand of amplitude $r = 25cm$ to the direction $(\theta, \phi) = (-45^\circ, 45^\circ)$. As a result, it

Figure 4: Evolution of motion parameters during a movement of amplitude $r = 25cm$ to the direction $(\theta, \phi) = (-45^\circ, 45^\circ)$. Left up: Cartesian coordinates of the hand; right up: angular trajectory; left middle: hand trajectory; right middle: hand velocity; left down: control signals; right down: torques.

is interesting to remark that the robot trajectories obtained through this approach exhibit the principal characteristics of human motions, as reported in [33], [9],

[14]. Indeed, the hand trajectories are almost rectilinear and the corresponding velocity profiles are single-peak and bell-shaped. Figure 5 is from [9]. It shows two sets of human hand paths starting with the same arm configuration and ending at peripheral targets locations in the sagittal and the frontal plane respectively. These trajectories look very similar to the ones executed by HRP2, which are represented in figures 2 and 3. Furthermore, the slight curvature variation associated to the circular distribution of targets seems to obey the same rule in both cases. Concerning the accuracy of our result, the average difference between the final robot hand position and the target is very small, around 1 mm, and the average of maximum distance to a virtual reference straight line trajectory is about 1 cm.

In order to demonstrate that the control approach is not limited to short motions, the method has also been tested for movements of amplitude 50 cm with a duration of 2 seconds. Single-peak bell-shaped velocity curves still occur, though the trajectory is unsurprisingly more curved (the average distance to the straight line is about 4 cm). The target-hand distance error at final time is still small, about 1 mm. Figure 6 shows two examples of such motions obtained with the simulation software OpenHRP and figure 7 shows corresponding hand trajectories and velocities. To simulate motions with OpenHRP, we took the angular trajectory of the robot joints obtained from the optimization program and then we interpolated it to determine the input data. The simulator follows this input to display the movement by taking into account the exact parameters of the robot. Such a validation on OpenHRP guarantees that the movement will be correctly executed by the real robot.

Finally, to illustrate the generality of the approach, we applied it to generate upper-body reaching movements. We considered a 7 DoF system including 2 DoF at the torso (yaw and pitch angles) and 5 DoF at the arm (3 at the shoulder, and 2 at the elbow, the last rotation at the wrist being fixed). The same motion generation method was applied by considering the dynamical model of the upper body of the robot. The passage from 6 to 7 DoF induced a significant increase of the computation time. Figure 8 shows the robot executing a downward reaching movement by using its upper body joints. Strikingly, when the torso joints are used, the robot seems to behave more closely like a human. Here again the hand trajectory is almost straight and the hand velocity curve is bell-shaped. Figure 9 shows the variation of parameters during this movement of the upper-body. It appears that the highest torques are produced at the torso, more important mass displacements being actuated by these joints.

In our simulations, a movement generation normally took from 1 to 4 minutes in the case of the 6 DoF arm, depending on the amplitude of the motion and the

choice of parameters. Whereas, up to 8 minutes were necessary in the case of the 7DoF upper body model. The program Ipopt converged to satisfy the constraints with very good accuracy after around 200 to 300 iterations but it did not always finish properly. This may be due to numerical errors in the calculation of the dynamics or to the approximation of the Hessian in the program. This phenomenon was also pointed out in [28]. Finally, the performance of the program could be improved by completing the refinement step in the direct transcription approach. Videos showing different motions of HRP-2 are available at:

<http://homepages.laas.fr/taix/neuro/>

6. Outcome analysis

This first part of the study shows that the principles of motor control proposed by Guigon et al [14] can be applied to humanoid robot control to generate reaching movements having the kinematic characteristics of human movements. Beyond providing a generic method to produce realistic movements, the proposed approach has two main interests from the control point of view. First, the separation of dynamic and static efforts simplifies the optimization problem which leads to the characterization of trajectories. The hypothesis that static gravitational efforts are continuously compensated by the tonic control is particularly well grounded in the case of robotic systems such as the humanoid HRP2, for which each joint is individually regulated to a reference value. Second, the redundancy problem can be solved implicitly without solving the inverse kinematics for the 6 DoF dynamic model of the arm and even for the 7 DoF dynamic model of the robot upper-body. Thanks to the direct transcription method and the Ipopt software, which give performance and flexibility to our optimization program, further constraints such as bounds on articular joints, or collision avoidance can be easily added to the program. However, the main drawback of the approach is the computational cost of the optimization problem resolution. Indeed, depending on the amplitude of the movement, the computation of an optimal trajectory required from 1 to 4 minutes for the 6 DoF arm and up to 8 minutes for the 7 DoF model of the upper-body . The heaviness of this computation constitutes a strong limitation that prevents the method from being used online to control the robot, in particular within a feedback process.

This limitation led us to ask whether the information contained in a database of reaching trajectories could be used to generate, at low cost, new trajectories still having the kinematic characteristics of human movements. Many specialists of motor control in neuroscience agree that the CNS does not solve such a

complex optimization problem each time it generates a movement. Instead, they believe that the knowledge acquired during the execution of previous movement is stored by the brain and used to generate new movements. Among the theories based on this hypothesis is the theory of motor primitives or synergies. This theory, which was suggested by biological evidences, conjectures that the CNS uses a finite set of elementary motor components, called primitives, to generate movements [37, 29, 8, 50]. At the kinematic level, the primitives are sometimes described as joint covariations. For example, covariations of ankle, knee and hip joints during bending movements and walking were described in [2] and [24]. A similar coactivation between whole-body joints during reaching movements was reported in [44] and [21]. Motor primitives or synergies were also pointed out at the muscular level, from EMG measurements. For instance, five muscular synergies involved in postural control were described in [49]. The encoding of motor primitives was also shown at the neural level. The well known experiment by Mussa-Ivaldi and Bizzi [29], showing that local stimulations of the spinal chord induce different leg movements in frogs, is a good illustration of this phenomenon. Recently, an attempt at modeling these experimental results in the control theory framework was proposed in [31]. For roboticists, reducing the complexity of control by using a finite set of movement primitives is a very attractive idea. It offers a promising alternative to the computation of inverse kinematics and cost minimization to cope with the high redundancy of anthropomorphic structures. This idea has already motivated some applications. Two primitives were used by Hauser et al. [17], for controlling the balance of a small size humanoid robot. Lim et al. [25], used Principal Component Analysis (PCA) to extract primitives from the captured movement of a human arm modeled as 4 DoF chain. These primitives were used as basis functions for parameterizing new realistic robot movements. In the study of Chhabra et. al. [7], the authors used a nonnegative matrix factorization method to extract primitives from a database of control signals to control a 2 DoF arm. Interestingly, it was shown that new movements can be learned faster in the primitive space than in the control space [4]. In order to introduce feedback in the control, Todorov et al. [46] considered sensorimotor primitives. This kind of primitive was applied to the control a 2 DoF arm.

Following this trend, the remaining part of the paper presents an original method for encoding the reaching movements from a small number of motor primitives in order to generate realistic human-like movements at lower computational cost.

7. Encoding reaching movements with motor primitives

In order to show that the method is sufficiently general and can be applied both to simulated movements, computed from Guigon's model, and to recorded human movements, two databases of reaching movements were considered.

- The first database was obtained by simulation by applying to the dynamic model of HRP-2's arm the principled approach of motor control following the method described in sections 3 to 5.

- The second database was obtained by recording human movements using a motion capture system. Arm motions of 3 participants were recorded using infrared markers attached to the shoulder, the elbow and the wrist. Accuracy of capture was less than $1mm$, with a frequency of $100Hz$. Participants were asked to perform a set of reaching movements while standing up. We considered 32 target positions, regularly-spaced in two parallel grids located in front of the participant's shoulder (voir figure 10). The target was a small ball at the end of a stick that was manipulated by the experimenter. It was placed at one of the 32 positions. The participants heard a sound to indicate that they should start moving their hand to the target. Upon reaching the current target, the next target was randomly chosen from the other 31 positions, and the experimenter moved the stick to this new position. Movements were executed until the 32 targets were reached from each other, giving a total of 993 movements per participant.

The same sequence of task was used for simulating the reaching movements with the dynamic model of HRP-2, in order to construct the first database. However, in the case of the robot, a scaling factor was applied to the dimension of the target setup and its distance to the robot in order to obtain comparable values. For both the human subjects and the robot, the databases of movements contain the variation of six joints: three at the shoulder, two at the elbow and one at the wrist. Note that the remaining DoF of the human arm were not considered for this application. In order to simplify the primitive extraction, the robot movements were all executed with a duration of $1s$ discretized in 100 time values. In the same way, the human reaching trajectories were normalized to $1s$ within the same sampling of $0.01s$. So, for both the robot and the humans, each reaching movement was described by 600 values encoding the variation of 6 angles.

8. Primitives extraction

We are looking for open-loop primitives with no sensory feedback. In the literature, such primitives are defined as time functions to be modulated in amplitude

by weighting coefficients [25], [17], [50]. The most general and simplest way of modeling the problem is to consider that movements are expressed as linear combination of these primitives. For our problem, each movement of the 6 DoF arm can be represented by a joint trajectory $U(t) \in \mathbb{R}^6$. A database of M movements is then described by a set of such trajectories:

$$U_m(t) = (U_{m1}(t), U_{m2}(t), \dots, U_{m6}(t))^T \in \mathbb{R}^6, m = 1, \dots, M, t \in [0, T].$$

For this database, the primitive extraction problem is to determine K time-functions: $\Phi_k(t) = (\Phi_{k1}(t), \Phi_{k2}(t), \dots, \Phi_{k6}(t))^T \in \mathbb{R}^6, k = 1, \dots, K, t \in [0, T]$, such that, for all $m = 1, \dots, M$, it is possible to determine k real coefficients α_{mk} , verifying:

$$U_m(t) = \sum_{k=1}^K \alpha_{mk} \Phi_k(t) \quad (17)$$

In this expression, the functions $\Phi_k(t)$ represent the expected primitives and the coefficients α_{mk} are weighting the contribution of each primitive $\Phi_k(t)$ into the movement $U_m(t)$. Each $\Phi_k(t)$ has then the same dimension as $U_m(t)$, and constitutes a particular joint trajectory.

Remark 2. *According to the considered definition, each motor primitive is the kinematic representation of a basic movement. Such primitives can be viewed as basis functions from which reaching movement can be reconstructed by linear combination to reach any target position in a region of space. However, it is important to note that these primitives do not constitute universal units that can be used to compose any kind of movements.*

The determination of Eqn. (17) states two problems. The first one is to determine the number K of primitives that are necessary to reach the expected movement precision and the second one is to characterize these K primitives. To reduce the complexity of the control problem, K needs to be as small as possible, compared to the number of parameters necessary to encode the trajectory $U_m(t)$. For a continuous-time problem, this number is infinite. However, in practice, the problem is solved by considering a sampling of the time interval $[0, T]$ in J elements. For the considered 6 DoF arm, K needs to be small with respect to the $6J$ angle values that encode each movement. In other terms, once the K primitives are determined, it is sufficient to compute K coefficients $\alpha_{mk}, k = 1, \dots, K$, to generate the movement $U_m(t)$ over $[0, T]$, instead of $6J$ variables. The second problem is to compute the canonical time-functions Φ_k . Considering the notation of Eqn. (17), each movement U_m and each primitive Φ_k is defined by a $6 \times J$ matrix as follows:

$$U_m = \begin{bmatrix} u_{m1}^1 & u_{m1}^2 & \dots & u_{m1}^J \\ u_{m2}^1 & u_{m2}^2 & \dots & u_{m2}^J \\ \dots & \dots & \dots & \dots \\ u_{m6}^1 & u_{m6}^2 & \dots & u_{m6}^J \end{bmatrix}, \Phi_k = \begin{bmatrix} \phi_{k1}^1 & \phi_{k1}^2 & \dots & \phi_{k1}^J \\ \phi_{k2}^1 & \phi_{k2}^2 & \dots & \phi_{k2}^J \\ \dots & \dots & \dots & \dots \\ \phi_{k6}^1 & \phi_{k6}^2 & \dots & \phi_{k6}^J \end{bmatrix} \quad (18)$$

where, u_{mh}^j and, ϕ_{kh}^j , $h = 1, \dots, 6$, $m = 1, \dots, M$, $j = 1, \dots, J$ are integers. Using this notation, the reconstruction error relative to the M movements of the database, $\{U_m, m = 1, \dots, M\}$ is defined by:

$$E = E(\alpha_{mk}, \Phi_k) = \sum_{m=1}^M \sum_{h=1}^6 \sum_{j=1}^J (u_{mh}^j - \sum_{k=1}^K \alpha_{mk} \phi_{kh}^j)^2 \quad (19)$$

Solving this problem comes then to determining K primitives, with K as small as possible, ($K \ll 6J$), but sufficiently large to guarantee that the reconstruction error will be lower than the expected precision threshold. This compromise will be discussed in the next section. Among the existing techniques that allow to cope with this kind of problem (see [12] for an overview), Principal Component Analysis (PCA), is intrinsically well adapted, simple and well-performing. It has already been used for such data representation problems in neurosciences and in robotics [2], [44], [37] [25]. In order to apply PCA, it is more convenient to use a vector notation of data. So, instead of using the matrix notation (Eqn. (18)), each movement U_m , $m = 1, \dots, M$, will be described by the following vector expression:

$$U_m = (u_{m1}^1, u_{m1}^2, \dots, u_{m1}^J, u_{m2}^1, \dots, u_{m2}^J, \dots, u_{m6}^1, \dots, u_{m6}^J)^T \quad (20)$$

where, each u_{mh}^j , $h = 1, \dots, 6$, $j = 1, \dots, J$, is the angular value of joint h at time t_j . The M column vectors U_m of dimension $N = 6J$ are then gathered to compose a $M \times N$ matrix. The $N \times N$ covariance matrix is then computed. The eigenvectors e_i , $i = 1, \dots, N$ and eigenvalues λ_i , $i = 1, \dots, N$ of this covariance matrix are then determined and ordered. Each eigenvalue λ_i represents the variance of data in the direction of the corresponding eigenvector e_i . The larger λ_i , the more its associated component e_i is dominant in the representation of data. The question is then to determine the number K , such that the first K principal components e_k , $k = 1, \dots, K$ constitute a basis of primitives from which the whole database can be expressed with the required precision. Recall that each primitive e_k is itself the representation of a six-joints trajectory. Finally, for each movement U_m , the coefficient corresponding to the primitive e_k is determined by computing the scalar product between these two vectors: $\alpha_{mk} = \langle U_m, e_k \rangle$, $m = 1, \dots, M$, $k = 1, \dots, K$.

Note that the application of PCA requires the data to be centered. This condition was almost exactly satisfied for each database.

PCA was separately applied to the robot database and the human databases described in section 7. In each case, we observed that the variance accounted for (VAF) by the $K = 8$ first primitives was more than 96%, whereas it was more than 99.5% with the $K = 20$ first primitives. Using the previously described approach, we determined the coefficients that enable us to reconstruct the whole set of movements by a linear combination of these primitives. These reconstructed trajectories were then compared with the original ones. To compute the reconstruction error for a whole database, we used the Root Mean Squared Error (RMSE) expressed by:

$$RMSE = \sqrt{\frac{E}{(M \times 6 \times J)}}$$

where, E is defined by Eqn. (19). Figure 11 shows the decay of the RMSE as a function of the number K of primitives, for each database. The decay rate of the RMSE is similar for the three human subjects, while it is slightly different for the robot arm. For human subjects, beyond the first 8 primitives, each addition of a new primitive induces almost the same error reduction. Whereas, for the robot model, the important decay rate between the 8th and the 16th primitive shows that these primitives still contain an important part of information. Interestingly, the curves corresponding to the human subjects and the robot become roughly parallel after the 18th primitive, showing a certain level of similarity between both databases. Beyond this level, the regular gap between the curves seems to be due to the difference in the kinematics and the dynamics of the arm structures. As human subject are taller and heavier than the robot, the effect of inertia and masses are higher on the human arm than on the robot arm. This might induce a higher reconstruction error for human movements. In each case, the primitives computed with PCA enable to represent the original data with a good precision. Depending on the expected level of precision, the number of primitives can be easily chosen. For instance, with $K = 16$ primitives the robot database can be reproduced with a mean error of 0.02 radians, (1.2°). The same number of primitives enable to represent the database of subject 1 with a mean error of 0.03 radians, (1.7°). As an example, Fig. 12 shows two trajectories reconstructed from these primitives and Fig. 13 shows the first 8 primitives for each database: the robot, subject 1, 2 and 3. The primitives represent the principal variations of the movements, classified by decreasing order of dominance variance. The first primitives are rather regular, while the last ones, more oscillatory, enable to capture subtle variations of

movements. It is interesting to remark in Fig. 13, that the primitives corresponding to the different database are not necessarily similar. This difference is due to the definition of primitives given by (17). Indeed, according to this relation, each primitive represents the simultaneous temporal evolution of the six arm joints, and these six trajectories are weighted by the same coefficient α_{mk} . This choice is biologically plausible, but leads to a different primitive basis for each database. Indeed, even though different subjects do the reaching movements in a similar way, the variations due to each subject may radically modify the expression of primitives.

It is important to recall that some authors used PCA to extract primitive for each joint separately [25]. As a consequence, the modulation of primitives is decoupled. In that case, it was shown that a lower number K of primitives is sufficient for each link (usually between 2 and 4), but the number of coefficients is multiplied by N . Thus, if 3 primitives are necessary for each link of a 6 DoF arm, then 18 primitives will be necessary for representing the arm movements. We also tested this approach for our problem and we found that the variance accounted for by 3 primitives on each link was more than 96%. In particular, we found a similarity between the primitives for the subject and the robot arm. However, there is no fundamental difference between the two methods of primitive extraction. The number of variables are almost identical in both cases. Recall that, by looking for primitives containing the six arm joints, our goal was to capture the joints co-activation. In this way, we expected to reduce the complexity, increase the precision of reconstructed movements, and find a basis of primitives well adapted to generalization. This last step is considered in the next section.

9. Generalization

So far, we have shown that large databases of reaching movements can be expressed as linear combinations of a small number of primitives. The question is now to determine if it is possible, from these primitives, to generate new movements having the characteristics of human movements. In other terms, we want to solve a generalization problem which can be stated as follows: *Considering a set of primitives Φ_k , given the initial arm configuration and a reaching task defined by the target position and the movement duration, determine the coefficients α_k that allow to express the trajectory as a linear combination of the primitives.*

However, recall that the objective is to generate realistic reaching movements rapidly, without having to solve each time the complex optimization problem described in section 4. Our first attempt to solve the generalization problem was to

apply learning methods in order to characterize fast input-output interface. We tested different architectures of multi-layer feedforward neural networks which received as input the six values of initial arm joints, the three Cartesian coordinates of the target and the movement duration, and were expected to give the α_k weighting coefficients of the K primitives as output. For some of these networks, the learning process on the 993 input-output pairs of each database took several hours. However, none of them succeeded in characterizing the α_k in a sufficiently generic way. In many cases, the reaching error between the final hand position and the target was too large and the movements were often not realistic. Increasing the number K of primitives did not improve the quality of the result. It seemed that the information learned by the networks was not sufficiently rich to capture the movement characteristics. Furthermore, feedforward neural network does not seem to be well appropriate to cope with the sensibility of the weighting of motor primitives. Then, as the learning techniques we considered did not provide us with satisfying results, we developed an original approach which consists in constraining the hand trajectory in the operational space. This method is described in the following section.

9.1. Method of trajectory constraint in the operational space

We want to determine the weighting coefficient α_k to generate new movements that still have the characteristics of human movements. To this end, we propose to determine the α_k for which the hand trajectory approximates at best a reference trajectory in the Cartesian space [10]. As we know that human trajectories are almost rectilinear and have bell-shaped velocity profiles, we have chosen to consider the minimum-jerk criterion to compute the reference trajectories. However, though this criterion can be used to demonstrate the feasibility of the generalization process, it is not fully satisfying. Indeed the main limitation is that the minimum-jerk criterion does not allow to take into account the dynamics of the controlled object and that it fails to account for some characteristics of reaching movements such as the slight curvature of the trajectory. We could as well use any other criteria that provides a more realistic account of experimental observations in the Cartesian space. The minimum kinetic energy criterion considered by Biess et al. [6] could be an interesting alternative solution. In order to implement our reasoning, we formulate an optimization problem whose objective is to characterize the α_k coefficients that minimize the error between the hand trajectory and the minimum-jerk curve in the Cartesian space. As before, the time interval $[0, T]$ is sampled into J elements $0 = t_1, \dots, t_J = T$. For $j = 1, \dots, J$, let us denote by $q^j = (q_1^j, q_2^j, q_3^j, q_4^j, q_5^j, q_6^j)^T$ the arm configuration at time t_j and $H^j = f(q^j)$ the

corresponding hand position in the Cartesian space, where f represents the direct kinematics associated to the 6 DoF arm model. Using this notation, the optimization problem can be stated as follows: *Given K primitives Φ_k , $k = 1, \dots, K$, each one representing a trajectory of the six arm joints, the initial arm configuration q^1 , and a reaching task defined by the target position H_{target} , find the value of K real coefficients α_k , $k = 1, \dots, K$, minimizing the distance E_{jerk} , defined below as the sum of the successive gaps between the hand position and the reference minimum-jerk reference curve computed at each time t_j :*

$$E_{jerk} = \sum_{j=1}^J \|f(q^j) - g(\tau_j, H^1, H_{target})\| \quad (21)$$

In this expression, τ is the normalized time and $\tau_j = t_j/T$ the discrete normalized value that corresponds to time t_j in $[0, T]$, and g is the time-function representing the hand position along the reference minimum-jerk trajectory expressed by:

$$g(\tau, H^1, H_{target}) = H^1 + (H^1 - H_{target})(15\tau^4 - 6\tau^5 - 10\tau^3), \quad (22)$$

Recall that, according to the definition of primitives, the angular value q_h^j of joint h at time t_j , that appears in the above definition, is computed from the relation $q_h^j = q_h^1 + \sum_{k=1}^K \alpha_k \phi_{kh}^j$, where ϕ_{kh}^j , $j = 1, \dots, J$, is the angular value of the h th joint of the k th primitive at time t_j . In this problem, the joint limits on the q_h^j during the movement are also considered. They are defined by inequalities of the type:

$$l_h \leq q_h^j \leq u_h \quad \text{for } h = 1, \dots, 6, j = 1, \dots, J, \quad (23)$$

where l_h and u_h are respectively the upper and lower bounds of the joint h at t_j .

Note that this problem is different from the one considered in [25]. Here, the objective is not to minimize a kinematic or dynamic criterion to determine the control signals, but instead to approximate a reference curve, having the characteristics of the human movement, in the Cartesian space. This problem is also different from the usual minimum-jerk problem in the sense that the joint trajectory of the arm is directly deduced from the knowledge of the α_k coefficients. No additional computation is then required to compute the joint trajectories; a step which usually requires to solve the inverse kinematics problem with the minimization of some additional criterion to cope with redundancy. Furthermore, note that the generalization problem described by Eqn. (21), (22) and (23) is a simple optimization problem involving K real variables and $6J$ linear constraints, which can be solved by using standard techniques. In this work, we used the Matlab *fmincon* solver. The results are described in the next section.

Table 1: Mean value of the distance E_{jerk} computed over the whole set of new generated movements and corresponding computation time for different values of K

K	Mean value of E_{jerk} (cm)	Computation time (s)
6	3.6504	5.5549
8	1.6120	11.1249
10	1.3822	14.1024
12	1.1417	20.5878
14	1.1169	28.3012
16	1.0842	35.5586
18	1.0415	41.1163
20	0.9676	48.6022

9.2. Results

In order to test the proposed generalization approach, a large number of new movements were generated by considering the model of the robot arm. To this end, a shift between $-10cm$ and $+10cm$ was applied at random in each direction of the initial target grid (see Fig. 10) to generate new targets, and angular increments between -5° and $+5^\circ$ were randomly added to each arm joint to specify new initial arm configurations. In this way, we generated a new large set of reaching movements with different arm configurations. Note that the largest amplitude movements were about 1 meter long.

In order to evaluate the compromise between computation time and distance to the reference minimum-jerk trajectory, different values of K between 6 and 20 were considered. Table 1 shows the mean value of the distance E_{jerk} computed over the whole set of new generated movements and the corresponding computation time, for each value of K . Logically, for a higher K the computation time increases, while the distance to the reference trajectory decreases. It is important to recall that the distance E_{jerk} defined by (21) measures the sum of the gaps between the hand movement and the reference minimum-jerk trajectory, at each t_j , all along each movement. Figure 14 shows the characteristics of movements obtained with the robot arm, for $K = 6, 8, 10, 12, 20$. Clearly, the hand trajectories and the velocity curves are not realistic for $K = 6$ and 8. However, as K increases, the trajectory becomes straighter and the velocity curve becomes smoother and more regular. For $K = 12$ the movement starts being very realistic. The improvement obtained by adding a new primitive is very small. For $K = 20$ the velocity profiles are perfectly bell-shaped and the hand trajectories are almost straight with a continuous weak curvature. Compared to our initial approach presented in sec-

tion 3 to 5, where reaching trajectories were obtained by minimizing each time the energy of virtual motoneurons, the computation time was reduced by a factor six by using the motor primitives. Note that this time could be further reduced by encoding the algorithm in C++ instead of using Matlab.

10. Conclusion

The interest of applying neurobiological principles to the control of humanoid robots has been illustrated in the case of reaching movements. Though the mechanical structure of today's humanoid robots strongly differs from the structure of the human body, applying biological principles provides a canonical way to generate human-like movements. From a computational point of view, the application of such principles is very interesting for roboticists as it provides new strategies for synthesizing the controller. The separation of static and dynamic efforts strongly simplifies the optimization problem as it removes the difficult boundary condition. Indeed, the dynamic controller vanishes at the beginning and at the end of the movement and the compensation of gravity is considered separately. This hypothesis is well adapted to the control of humanoid robots such as HRP-2, in which the low-level regulation of joint positions guarantees the compensation of gravity during reaching movements. It is then possible to focus on the dynamic part of the control for designing movements resembling those of humans. Another interesting advantage of the proposed approach is that it does not need the computation of inverse kinematics. As the Jacobian matrix is not square, such an inversion would require the computation of a generalized inverse and implicitly the choice of an additional cost function which is unknown. It was pointed out that the main drawback of the initial optimization approach is the computation time. To cope with this problem, motor primitives were designed to encode the movements. Thanks to this strategy it was possible to reduce the computational cost by a factor six. It turns out that the use of movement primitives coupled with an appropriate generalization process provides an efficient way to store the complex information of human movements and quickly generate new trajectories. Although it has been suggested that the CNS generates movements by solving a complex optimization problem, such as minimizing the energy of motoneurons, it is likely that the use of a reduced number of canonical synergies allows to simplify the computation. This study proves that this biological solution can be efficiently transferred to robotics. In future works, we plan to address the problem of combining such open-loop primitives to produce more complex whole-body movements.

11. Acknowledgment

This work was supported by the Zeuxis project funded by the EADS foundation and by the ROMEO project funded by the French FUI.

References

- [1] R. Alexander, Optimization and gaits in the locomotion of vertebrates, *Physiological Reviews* 69 (1989) 1199–1227.
- [2] A. Alexandrov, A. Frolov, J. Massion, Axial synergies during human upper trunk bending, *Experimental Brain Research* 118 (1998) 210–220.
- [3] N. Bernstein, *The coordination and regulation of movements.*, Oxford: Pergamon press, 1967.
- [4] B. Berret, F. Bonnetblanc, C. Papaxanthis, T. Pozzo, Modular control of pointing beyond arm’s length, *Journal of Neuroscience* 29 (2009) 191–205.
- [5] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, 2001.
- [6] A. Biess, D. Liebermann, T. Flash, A computational model for redundant human three-dimensional pointing movements: Integration of independent spatial and temporal motor plans simplifies movement dynamics, *Journal of Neurosciences* 27 (2007) 13045–13064.
- [7] M. Chhabra, R. Jacobs, Properties of synergies arising from a theory of optimal motor behavior, *Neural Computation* 18 (2006) 2320–2342.
- [8] A. d’Avella, P. Saltiel, E. Bizzi, Combinations of muscle synergies in the construction of a natural motor behavior, *Nature Neurosciences* 6 (2003) 300–308.
- [9] M. Flanders, J. Pellegrini, S. Geisler, Basic features of phasic activation for reaching in vertical planes, *Experimental Brain Research* 110 (1996) 67–79.
- [10] T. Flash, B. Hochner, Motor primitives in vertebrates and invertebrates, *Current Opinion in Neurobiology* 15 (2005) 660–666.

- [11] T. Flash, N. Hogan, The coordination of arm movements: An experimentally confirmed mathematical model, *Journal of Neuroscience* 5 (1985) 1688–1703.
- [12] I. Fodor, A survey of dimension reduction techniques, Manuscript (2002).
- [13] Y. Fujita, Y. Nakamura, K. Yamane, S. I., Somatosensory computation for man–machine interface from motion-capture data and musculoskeletal human model, *IEEE Transaction On Robotics* 21 (2004) 58–66.
- [14] E. Guigon, P. Baraduc, M. Desmurget, Computational motor control: Redundancy and invariance, *Journal of Neurophysiology* 97 (2007) 331–347.
- [15] E. Guigon, P. Baraduc, M. Desmurget, Optimality, stochasticity, and variability in motor behavior, *Journal of Computational Neuroscience* 24 (2008) 57–68.
- [16] C. Harris, D. Wolpert, Signal-dependent noise determines motor planning, *Nature* 394 (1998) 780–784.
- [17] H. Hauser, G. Neumann, A.J. Ijspeert, W. Maass, Biologically inspired kinematic synergies provide a new paradigm for balance control of humanoid robots, *IEEE-RAS Int. Conf. on Humanoids* (2007) 73–80.
- [18] F. Van der Helm, L. Rozendaal, Musculoskeletal systems with intrinsic and proprioceptive feedback, *Neural control of posture and movement* (2000) 164–174.
- [19] M. Hersch, A. Billard, A biologically-inspired controller for reaching movements, *IEEE/RAS-EMBS Inter. Conf. on Biomedical Robotics and Biomechatronics* (2006) 1067–1072.
- [20] J. Kalaska, C. D.A.D., M. Hyde, M. Prud’homme, A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task, *The Journal of Neuroscience* 9 (1989) 2080–2102.
- [21] T.R. Kaminski, The coupling between upper and lower extremity synergies during whole body reaching, *Gait Posture* 26 (2007) 256–262.

- [22] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, T. Isozumi, Humanoid robot hrp-2, *IEEE Inter. Conf. on Robotics and Automation* (2004) 1083–1090.
- [23] O. Khatib, J. Warren, V. De Sapio, L. Sentis, Human-like motion from physiologically-based potential energies, *Advances in Robot Kinematics* (2004) 149–163.
- [24] F. Lacquaniti, R. Grasso, M. Zago, Motor patterns in walking, *News in Physiological Sciences* 14 (1999) 168–174.
- [25] B. Lim, S. Ra, F.C. Park, Movement primitives, principal component analysis, and the efficient generation of natural motions, *IEEE Int. Conf. on Robotics and Automation* (2005) 4630–4635.
- [26] M. Mataric, Getting humanoids to move and imitate, *IEEE Intelligent Systems and their Applications* 15 (2000) 18–24.
- [27] T. Matsui, M. Honda, N. Nakazawa, A new optimal control model for reproducing human arm’s two-point reaching movements: A modified minimum torque change model, *IEEE Inter. Conf. on Robotics and Biomimetics* (2006) 1541–1546.
- [28] S. Miossec, S. Yokoi, K. Kheddar, Development of a software for motion optimization of robots - application to the kick motion of the hrp-2 robot, *IEEE Inter. Conf. on Robotics and Biomimetics* (2006) 299 –304.
- [29] F.A. Mussa-Ivaldi, E. Bizzi, Motor learning through the combination of primitives, *Phil. Trans. R. Soc. Lond., Biological Sciences* 355 (2000) 1755–1769.
- [30] K. Nishikawa, S. Murray, M. Flanders, Do arm postures vary with the speed of reaching, *Journal of Neurophysiology* 81 (1999) 2582–2586.
- [31] F. Nori, R. Frezza, A control theory approach to the analysis and synthesis of the experimentally observed motion primitives, *Biological Cybernetics* 93 (2005) 323–342.
- [32] S. Northrup, N. Sarkar, K. Kawamura, Biologically-inspired control architecture for a humanoid robot, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2001) 1100–1105.

- [33] M. P., Spatial control of arm movements, *Exp. Brain Research* 42 (1981) 223–227.
- [34] U. Pattacini, F. Nori, L. Natale, G. Metta, G. Sandini, An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots, *IEEE/RSJ Int. Conf. on Intelligent Robot ad Systems*, Taipei, Taiwan (2010).
- [35] N. Pollard, J. Hodgins, M. Riley, A. C.G., Adapting human motion for the control of a humanoid robot, *IEEE Int. Conf. on Robotics and Automation*, Washington D.C. (2002).
- [36] M. Rand, Y. Shimansky, G. Stelmach, J. Bloedel, Adaptation of reach-to-grasp movement in response to force perturbations, *Experimental Brain Research* 154 (2004) 50–65.
- [37] T.D. Sanger, Human arm movements described by a low-dimensional superposition of principal components, *The Journal of Neuroscience* 20 (2000) 1066–1072.
- [38] H. Seki, S. Tadakuma, Minimum jerk control of power assisting robot based on human arm behavior characteristics, *IEEE Int.Conf. on Syst, Man and Cybernetics* (2004) 722–727.
- [39] S.P. Shadmehr, R.and Wise, *The computational neurobiology of reaching and pointing*, MIT Press, 2005.
- [40] G. Simmons, Y. Demiris, Optimal robot arm control using the minimum variance model, *Journal of Robotic Systems* 22 (2005) 677–690.
- [41] M. Spong, M. Vidyasagar, *Robot Dynamics and Control*, Wiley, 1989.
- [42] K. Tahara, Z. Luo, S. Arimoto, On control mechanism of human-like reaching movements with musculo-skeletal redundancy, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2006) 1402–1409.
- [43] Y. Tanaka, T. Tsuji, Bio-mimetic trajectory generation using a neural time-based generator, *Journal of Robotic Systems* 22 (2005) 625–637.
- [44] J.S. Thomas, D.M. Corcos, Z. Hasan, Kinematic and kinetic constraints on arm, trunk, and leg segments in target-reaching movements, *Journal of Neurophysiology* 93 (2005) 352–364.

- [45] K. Thoroughman, K. Feller, Gravitational effects on torque change and variance optimization in reaching movements program no. 492.20. abstract viewer/itinerary planner, Washington, DC: Society for Neuroscience, Online (2003).
- [46] E. Todorov, Z. Ghahramani, Unsupervised learning of sensory-motor primitives, IEEE Int. Conf. Engineering in Medicine and Biology Society (2003) 1750–1753.
- [47] E. Todorov, M. Jordan, Optimal feedback control as a theory of motor coordination, *Nature Neuroscience* 5 (2002) 1226–1235.
- [48] E. Torres, D. Zipser, Simultaneous control of hand displacements and rotations in orientation-matching experiments., *Journal of applied physiology* 96 (2004) 1978–1987.
- [49] G. Torres-Oviedo, J.M. Macpherson, L.H. Ting, Muscle synergy organization is robust across a variety of postural perturbations, *Journal of Neurophysiology* 96 (2006) 1530–1546.
- [50] M. Tresch, V. Cheung, A. d’Avilla, Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets, *Journal of Neurophysiology* 95 (2006) 2199–2212.
- [51] Y. Uno, M. Kawato, R. Suzuki, Formation and control of optimal trajectory in human multijoint arm movement - minimum torque change model, *Biological Cybernetics* 61 (1989) 89–101.
- [52] A. Wächter, L. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, *Mathematical Programming* 106 (2006) 25–57.
- [53] T. Welter, M. Bobbert, Initial muscle activity in planar ballistic arm movements with varying external force directions, *Motor Control* 6 (2002) 32–51.

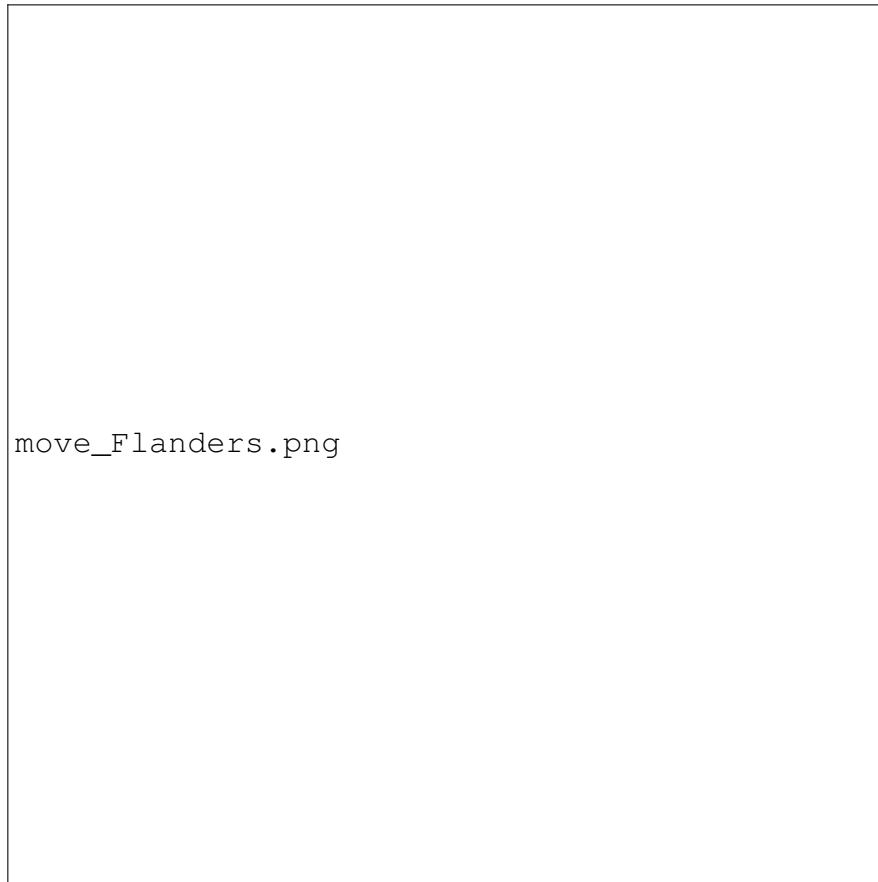


Figure 5: Human reaching hand paths as reported by Flanders et al. in [9]. The situation is similar to the one described in figures 2 and 3: targets are arranged at twenty peripheral positions located at 30cm from the initial hand location. F, M and L stand for forward, right and left respectively.



Figure 6: From left to right : four successive snapshots of a reaching movements of the 6 DoF arm of HRP-2, simulated with OpenHRP. This movement of amplitude $r = 50cm$ is directed to the target position $(-35^\circ, 35^\circ)$.

Figure 7: Hand trajectories (left image) and velocities (right image) corresponding to the two reaching movements represented in figure 6.

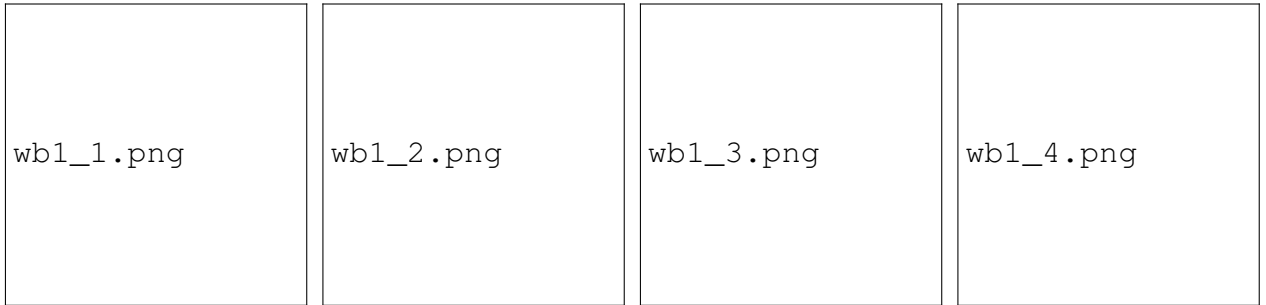


Figure 8: From left to right : four successive snapshots of a reaching movement involving the upper-body joints : 2 DoF at the torso (pitch and yaw) and 5 DoF at the arm.

Figure 9: Variation of parameters during an upper-body reaching movement of the robot of amplitude $50cm$, toward the direction $(\theta, \phi) = (-45^\circ, 45^\circ)$. Left up: Cartesian coordinates of the hand; right up: angular trajectory; left down: control signals; right down: torques.



Figure 10: (a) Illustration of the human experiment during successive reaching movements. The arm segments are in red and examples of hand trajectories are drawn in blue (b) The humanoid robot HRP-2 (Kawada Industries Inc., Japan). (c) Representation of the 32 target positions on two parallel virtual grids. These target positions were used to drive the human reaching movements



Figure 11: Root Mean Square Error between the original trajectories and the reconstructed ones, as a function of the number of primitives K . The curves corresponding to the three human subject and the robot arm are represented. In addition, the blue curve represents the mean of the three human subjects.



Figure 12: Example of joint trajectories reconstructed from $K = 16$ primitives. The curves in the left picture were obtained with the robot database, whereas the curves in the right picture were obtained with the database of subject 1. The original trajectories are in dotted lines and the reconstructed trajectories in full line.



Figure 13: Description of the first 8 primitives obtained with PCA on the different databases. From left to right the column correspond to the robot, subject 1, 2 and 3 respectively. From top to down, primitives are classified by variance dominance. In each scheme, the abscissa axis represents the time in seconds and the ordinates represent the joint angles in degrees. A different color is used for each of the 6 arm joints



Figure 14: For a same reaching task of amplitude 58cm , description of the movement successively generated by using an increasing number primitives $K = 6, 8, 10, 12,$ and 20 from top to down. From left to right, each horizontal set of 4 pictures corresponds to a particular value of K and represents: the Cartesian hand coordinates, the hand velocity profile, the joint trajectories and the hand trajectory. The reaching errors in cm are respectively: $4.07, 1.14, 0.36, 0.12$ and 0.11 .