A Software Toolbox to Carry-out Virtual Experiments on Human Motion

Franck Multon, Jean-Luc Nougaret, Gérard Hégron, Luc Millet, Bruno Arnaldi

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Software Toolbox to Carry-out Virtual Experiments on Human Motion

Franck Multon, Jean-Luc Nougaret, Gérard Hégron, Luc Millet and Bruno Arnaldi

N° 3303

Novembre 1997

THÈME 3

Rapport de recherche

ISSN 0249-6399

A Software Toolbox to Carry-out Virtual Experiments on Human Motion

Franck Multon, Jean-Luc Nougaret, Gérard Hégron*, Luc Millet*
and Bruno Arnaldi

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet SIAMES

Rapport de recherche n3303 — Novembre 1997 — 20 pages

Abstract: We present a simulation toolbox to carry-out virtual experiments on human motion. 3D visualization, automatic code generation and generic control design patterns bring dynamic simulation tools into the hands of biomechanicians and doctors.

(Résumé : tsvp)

This work was supported by the European community.

We would like to thank Gilles Dietrich from INSEP for his help and advises in the conception of this tool. A special thank to Walter Maurel from EPFL for his image of the topological model of arm.

* École des Mines de Nantes, FRANCE

Unit´e de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
T´el´ephone : 02 99 84 71 00 - International : +33 2 99 84 71 00
T´el´ecopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Une Boîte à Outils Logiciels Permettant de Mettre en Œuvre des Expérimentations Virtuelles sur le Mouvement Humain

Résumé : Nous présentons, dans ce rapport, une boîte à outils logicielle permettant de mettre en œuvre des expérimentations virtuelles sur le mouvement humain. Visualisation 3D, generation automatique de code de simulation et définition de classes de contrôleurs génériques sont mis à la disposition des biomécaniciens et des docteurs.

□ Studying the human motion is a challenging task. Several experiments can be carried-out to investigate the complex phenomena that are involved in human motion. Experimental methods techniques suffer important limitations which considerably restrict the scope and accuracy of the analysis. Indeed, the human body is a complex dynamic system which is characterized by several hidden parameters. These parameters are not easily accessible by conventional measurement equipments. Motion capture technology, optics-based or magnetics-based, provide only superficial and inaccurate information. A complete range of external and invasive sensors (force plates, accelerometers, X-ray or fluoroscopic imaging and EMG) provide additional information. Nevertheless, when they are considered independently, these techniques provide a limited and partial view of the motion dynamics, in addition to being cumbersome to implement. Sensor fusion and clever experimental protocols only partially overcome those limitations.

To validate their theoretical models of human motion, biomechanicians carry-out experiments on real subjects by using one or several of the above techniques. As a complementary tool, virtual experiments would be fast and economical methods to try some ideas of motion control without the tedious preparation of a real one. In the next few years, simulation will be central to medicine. Real-time, interactive realistic simulation systems will routinely be a major integral component in physician and paramedical personnel training, pretreatment planning, instrument/device design and the evaluation and approval process. Nowadays, teleimaging, simulations, image-guided robot-assisted therapies [8, 3] and several methods to model human figures (including bones and muscles) [9, 10] are widely investigated. But, up to our knowledge, no tool has been proposed to biomechanicians in order to experiment their ideas of motion control strategies.

We have defined an open platform in order to experiment motion control strategies on a virtual human body (see figure 1). First, medics and biomechanicians have to describe the skeleton of the virtual body by giving its anatomical and topological description. These descriptions are entered as a parameter file so that a simulation code for the equivalent mechanical structure is automatically generated. Then, to test a control strategy, biomechanicians just have to define the controller in the simulation platform without taking care of graphics and simulation problems. These tools have been developed during

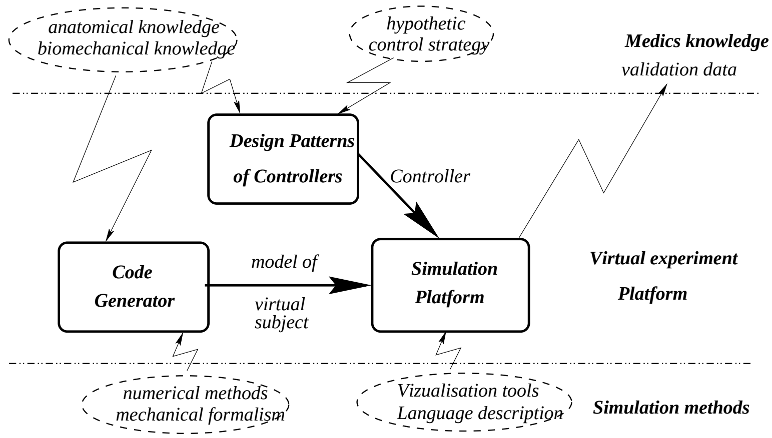


Figure 1: Overview of the virtual experiment platform.

the CHARM project[1] and applied to the human arm (including the scapula). Partners of the project have provided us with the topological and biomechanical [6] description of the skeleton (see figure 2 for snapshots of this skeleton model). These data have been obtained by processing the resulting images from the *Visible Human* project[2]: an anatomic and geometric model have been reconstructed from medical images, and have been used to define a topological and biomechanical model of the arm (mass, inertia, line-action attachments). From these data, we have developed a tool to experiment several motion control strategies of this synthetic human arm. To this end, we have studied and applied several control strategies (currently proposed in biomechanics) to this model. These studies make it possible to design a general software architecture (in which modules are mainly defined by their inputs and outputs) that allow implementation of these control strategies. The output of this system are angular values provided to the hierarchical geometric model at each time step of simulation (25Hz for real-time visualization). Fast computation algorithms are required to compute the arm motion at this frame rate.

[1]CHARM: Comprehensive Human Animation Resource Model. European Esprit-Bra n^o9036. For more information, see **http://ligwww/ ~maurel/CHARM/Home_Page.html**

[2]see **http://www.nlm.nih.gov/research/visible/visible_human.html** for more information

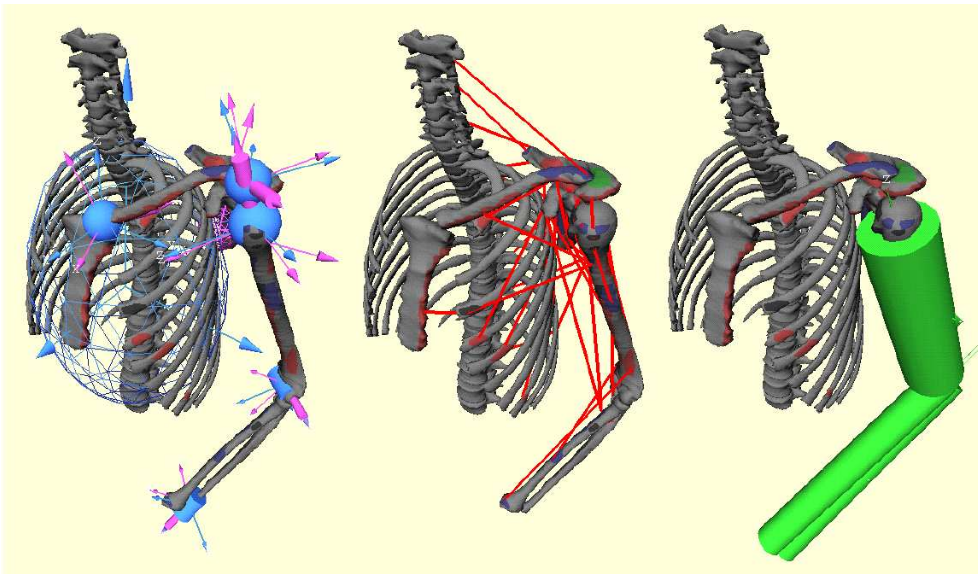


Figure 2: Reconstructed biomechanical model of the human arm. Courtesy of W. Maurel from EPFL, Switzerland.

This paper describes the general toolbox and simulation platform to practice virtual experiments and explains how it has been used in the CHARM project. The article is organized as follows. First, section 1 presents an overview of our simulation platform. The components of this platform are described beginning with the model of arm in section 2. Finally, several possible control design patterns are proposed in section 3. Section 4 describes the user interface. Then, section 5 gives some results by describing two experiments.

1 Overview of the simulation toolbox

The simulation toolbox has been designed to simplify the preparation of virtual experiments by providing software components that automatically generate the simulation code. Thus, the experimenter just has to focus on the design of controllers and of a virtual model. This toolbox is composed of three main parts, corresponding to the components of the process depicted in figure 1:

- a software package automatically generates the C^{++} simulation code of the model from its energetic description;
- several design patterns of controllers commonly studied in biomechanics are provided as abstract classes. The experimenter can design its own control strategies by choosing the required pattern and by writing the computation code of the controller;
- a simulation platform with visualization and user-interfaces libraries is provided to embed the two above-mentioned modules (the simulation code of the model and the computation code of the controller).

To experiment an embedded motion control strategy, the user first asks the model to execute several movements. Once these orders are analyzed, the system calls the simulation code of the model and its controller to execute the desired motion. A simulation step is divided in three parts:

- numerical integration of the actuator actions depending on the control parameters,
- computation of the differential equations of the mechanical model,
- computation of the changes in the geometric structure.

To this end, user's orders are translated into functions that modify the control parameters. First, a planner decomposes complex motions into elementary motions and translates orders expressed in the Cartesian frame into orders in the articulation frame (or the reverse-way) depending on the pattern of the controller. The controller takes angles or Cartesian ending positions provided by the user and makes the arm move to this goal. The global synopsis of the simulation platform is described in figure 3. A database containing the state of this model and its controller is shared between the controller and the planner. The controller fills-in this shared database during the simulation to make the other components of the platform know the state of the system. T

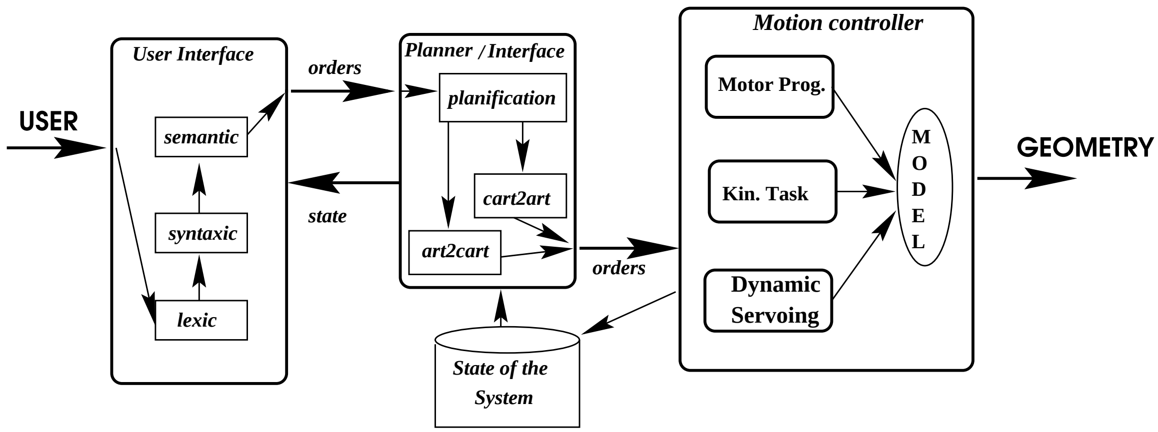


Figure 3: Global synopsis of the virtual experiment system.

2 Code Generation from a mechanical description

In this section, we describe how an experimenter can specify his virtual human model to automatically generate its simulation code. First, the complex architecture of real muscles is approximated as a appendix A. So, the human body can be considered as a mechanical model of linked rigid bodies with rotary joints. Characteristics of position, mass, inertia and topology (transform matrix from one body to another) are given by the experimenter to describe his virtual subject. Then, from the Lagrangian formulation, a set of second-order nonlinear differential equations can be automatically obtained. Despite its generality this formalism wasn't widely used previously, because of its inefficiency compared to its competitors. Using theoretical results coming from the research's field about differentiation we are now able to use it for obtaining a very efficient dynamic model [4] (whose complexity is the same than if using Newton-Euler). The motion equations were generated using NMecam a dynamic simulator developed over several years at Irisa. NMecam main features are:

- the dynamic model and all data (like kinetic energy or kinematic values) are computed in a symbolic way,

- generation of efficient numerical C/C^{++}-code.

Additional constraints (kinematic closed-loop structures) imply that additional forces (exerted at the contact point) must be taken into account. In the case of the 11 DOFs human arm model, they consist in specific kinematic (holonomic) constraints: ellipsoidal constraint of the scapulo-thoracic contact (see appendix B), closed-loop structure for which one joint is replaced by a constraint,etc. Two ways can be used to take the constraints into account:

- the Lagrange multipliers:

$$\begin{cases} \mathcal{Q}_i - \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{C}}{\partial \dot{q}^i}\right) + \dfrac{\partial \mathcal{C}}{\partial q^i} + \mathcal{L}_i \\ \mathcal{L}_i = \lambda . \frac{\partial f}{\partial q_i} + \mu . \frac{\partial g}{\partial \dot{q}_i} \end{cases} \tag{1}$$

- a penalty scheme:

$$\begin{cases} \mathcal{Q}_i - \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{C}}{\partial \dot{q}^i}\right) + \dfrac{\partial \mathcal{C}}{\partial q^i} + \mathcal{L}_i \\ \mathcal{L}_i = k \ \sum_h \ f \ \frac{\partial f}{\partial q^i} + k \ \sum_h \ g \ \frac{\partial g}{\partial q^i} = 0 \end{cases} \tag{2}$$

where $q = (q^i)_{i=1,n}$ is the set of the Lagrangian parameters of the multibody system, $\mathcal{C}$ is the kinetic energy, $\mathcal{Q}_i$ is the generalized given effort relative to q^i. L represents the sum of kinetic and potential energies and Q_i represents generalized external efforts that are exerted to the system or forces that do not derivate from a scalar potential function. The constraints are denoted by f (holonomic) and g (non holonomic), and k is a chosen penalty constant.

If the Lagrange multipliers ensure that we satisfy exactly the constraints at each time step, this method has the major drawback of adding some unknowns to the system. Consequently, the dimension of the system increases with the number of constraints. Moreover this method fails when antagonist constraints are applied to the mechanism. Then, we have chosen a penalty scheme method in order to maintain a constant size of the system and to deal with antagonist constraints (the constraints may then be viewed like springs whose stiffness is k). One can note that the penalty scheme is equivalent to the Lagrange multiplier's method when k tends toward infinity (but so the stiffness of the system).

3 Design Patterns for human motion control

First, our work has been focusing on gathering information and building-up knowledge about the various control schemes that may be employed for producing physically correct of the synthetic human arm. To this end, we made a list and a classification of the various motions that may be performed by the human arm, namely *kinematic tasks, reflex-like behavior* and *dynamic servoing* and for each of these, a workable control scheme has been suggested (see figure 4). In this classification, three main control strategies have been identified and can be considered as design patterns of human motion controllers. These design patterns are then described as abstract classes that the experimenter can overload with his own computation methods or gain matrix. The resulting classes are used to make plug-ins to the simulation platform. We now describe more precisely each of these control patterns.

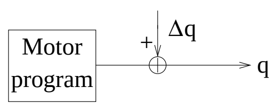
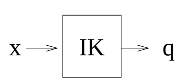
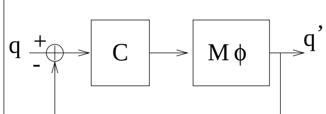
Control structure	Controller Aim	Scope	Main feedback	Neurophysiological background
	Prerecorded motion playback	Reflex behaviour (playing table-tennis)	open loop (no feedback)	Motor programs (Schmidt93)
	Redundancy solving	precision grasping, manipulation (playing chess)	end-effector external feedback (planning-level)	Neural-wired IK (Soechting, Flanders89)
 C : linear / nonlinear	Compensation of dynamic effects	servo-controlled motion (windsurfing)	proprioceptive state feedback	Cerebellum servo-control (Kawato87)

Figure 4: Movement classification and proposed control schemes

Kinematic tasks

For the last controller type, slow motion that require accurate positioning of the hand, dynamic parameters of the arm do not matter in the sense that any intermediate configuration of the arm's limbs can be considered to be quasi-static. Given the instant position and orientation of the end-effector, an (constrained) inverse kinematic [1] (IK) algorithm can be used to determine the actual configuration of the arm (by computing the DOF of all the remaining limbs). To solve for the intrinsic kinematic redundancy of the arm, a secondary task can be optimized. Equation 3 expresses the relation between the variation of the effector's position and the angular values.

$$\Delta q = J^+ \Delta x + \alpha (I - J^+ J) \ grad \ C \tag{3}$$

where q represents the vector collecting the angular coordinates of the arm, and x represents the 6 DOF vector of the hand. J^+ is the so-called *pseudo-inverse* of the non-square Jacobian matrix (hence the redundancy). The term "secondary task" refers to the fact that minimization of C do not alter fulfillment of the primary task (positioning of the hand). Depending on the arm's redundancy, cascaded criteria can be used ... Various criteria C, such as *minimum jerk*, *maximum comfort* [2] or *minimum torque variations* [5], have been suggested for expressing the natural motion of the human arm.

In some cases, we can apply kinematic constraints to the dynamic model to obtain a functionality similar to inverse kinematic. This solution was applied in the CHARM project (constraints-based controller) by overloading the computation methods. The inverse kinematic computation module is then replaced by an activation or deactivation of the kinematic constraint.

Reflex-like behavior

The so-called *Motor Programs* have been theorized to be the underlying representation of rapid movement since the beginning of the century. More recently, generalized motor programs [11] have been proposed to account for the production of human movements with qualitative invariant features. Evidence for units of motor control and provision for motor programs is that the time response of sensory feedback loops are too slow to be involved in the production of rapid movements. Hence the idea that some central neural structure must be responsible for storing and "flushing" the gesture. One important feature of these motor program units is that the engendered motions are able to expand or contract in a number of dimensions (time, amplitude) while maintaining some invariant characteristics (relative timing). Motor programs can be taken into account by approximating prerecorded angular trajectories that can be modified using, for example, signal processing techniques. In that case, the entries of the design pattern are the identifier of the motion and its parameters. The outputs are the angular values of all the DOFs that are just provided to the visualization module. As a matter of fact, it can also be argued that motor programs are stored as torque information. In this case, the mechanical model works as a dynamic filter. The actual reflex motion is obtained by applying this filter to the input torque sequence associated with the motor program.

Dynamic servoing

Whenever movement implies significant velocities or loading of the arm, dynamic effects affect motion in a noticeable way. In this case the mechanical equations that govern motion and the actual arm's dynamic parameters (inertia matrices) must be taken into account. For this purpose, a set of differential equations governing dynamic motion will be automatically built from the arms' geometric description, using Lagrange formalism (see section 2).

From the point of view of control theory, the synthetic human arm is a multi-input multi-output (MIMO) dynamic system with high-dimensionality defined by its state-space equation $\dot{x} = h(x, u)$ where x represents the system's state vector (built-up by collecting all DOF together with their respective angular velocities), and u the vector made of all externally applied generalized forces.The system exhibits coupling between DOF as well as strong nonlinearities. As such, the model is a a tough system to control although nonlinear control theory provides alternative and workable control strategies. One solution to the problem would be to consider a set of operating points around of which a linearized model of the arm could be extracted:

$$\dot{x} = Ax + Bu \tag{4}$$

where matrices A and B are calculated from the Jacobian $\frac{\partial h}{\partial x}$ and $\frac{\partial h}{\partial u}$. By doing so, the initial nonlinear control problem can make use of the powerful tools provided by the well-established theory of linear control. For instance, for each operating point of the human arm one may consider designing an optimum time-invariant linear feedback controller, using standard design methods such as H_2 design. Around each operating point, the so-designed H_2 controller would minimize a weightable tradeoff between energy consumption and tracking accuracy. An alternative approach is to recast the nonlinear control problem as a H_∞ problem (also referred-to as *robust control*) by regarding the nonlinear plant as a linear plant associated with an uncertainty due to its nonlinear components. In any case, such solutions may yield to combinatory explosion in the number of operating points to be sampled, unless a very simple model of the arm is considered.

Thanks to the local linearization schemes described above, the state feedback law $u = -G(x)$ reduces to a local matrix gain $u = -Gx$. Some form of motion coordination is achieved when off-diagonal terms of G are different from zero. Proportional derivative (PD) control occurs when G is diagonal, in which case the individual degrees of freedom are treated separately. As gravity exerts a constant disturbance, simple proportional derivative controllers cannot ensure that the steady error be zeroed once the dynamic model has settled. In this case, additional integral terms are needed to annihilate the effects of gravity (in the non-coordinated case, we compute input torques as $\tau = k_p(\theta - \theta_d) - k_d\dot{\theta} - k_i \int(\theta - \theta_d)dt$. Of course, such considerations are given from the perspective of control theory. This would be up to biomechanicians and doctors to decide what form of control law $u = -G(x)$ should be implemented for their virtual experiments.

4 User Interface

As this tool is destinated to biomechanicians and medics, a specific user-friendly interface has been defined. Orders are entered in natural language by using usual medical vocabulary (such as pronation, abduction,etc. for the arm motion). Once these orders have been analyzed, the motion is displayed on the visualization window (see figure 5). To this end, the analysis of the

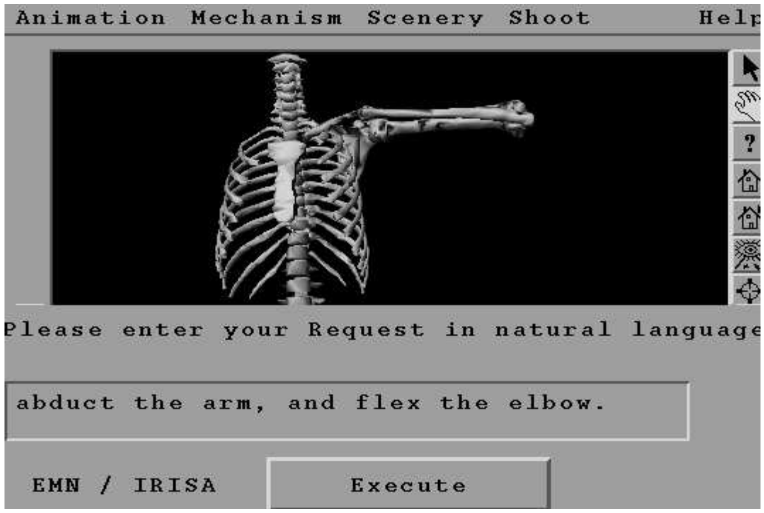


Figure 5: Snapshot of the Interface.

user's sentences produces function calls applied to the embedded controller. But, for complex motions, the planner has first to subdivide them into elementary motions and then invoke the required functions. These functions are divided into 3 main types:

- order in the Cartesian frame: the controller drives a selected body part in order to reach a target in the Cartesian frame (for the human arm, the wrist is driven);
- order in the articular frame: a set of end-angle positions is given by the user for a specific DOF;
- database query: a knowledge of the state of the system is required to deduce the exact will of the user (ie abducting the arm whereas the arm is flexed).

The functions are computed by the *Planner* that communicates with the controller and the database to understand and satisfy the user's orders.

The natural language interface has been designed to make the user choose one of the several predefined motions (currently used by medics) or drive a part of the skeleton to a given Cartesian position (for example: “Move the wrist to the position 1.0 0.5 0.1”). In the particular case of the arm, specific motion vocabulary has been used, such as *pronation, supination, abduction*,etc. Each predefined motion is defined by body parts, amplitude, velocity, direction and subsequences of elementary motions. Thus, an abduction (“abduction.”), which roughly corresponds to a raise of the arm in the body plane is described by a predefined final angular position of the gleno-humeral joint, and its average velocity during the motion. The previous predefined values can be tuned by the user using fuzzy quantifiers. Then, automatic translation of a qualitative specification (as a reference trajectory $q_{ref}(t)$) is achieved using tools coming from fuzzy logic theory. The process of deriving numerical values from qualitative attributes is referred-to as a stage of defuzzification. For example, it is possible to ask the system to “raise the arm rapidly.” where the qualitative attribute “rapidly” is assigned to the action of raising the arm. The attributes for speed and amplitude can take a qualitative value belonging to a so-called *fuzzy set*, such as $\{slowly, normally, rapidly, strongly, completly, moderatly, slightly\}$. Each of these attributes corresponds to a function (called the *membership function*) which expresses the degree of fitness to the qualitative attribute, as described in figure 6. These processes are performed after a three-stage lexical,

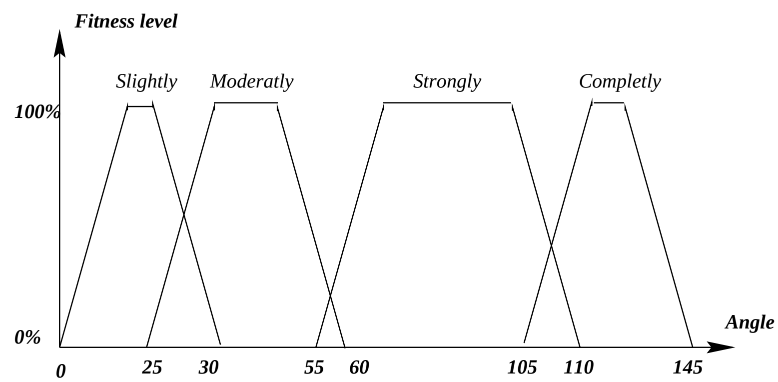


Figure 6: Linguistic variables used for bending the elbow.

syntaxical and semantical analysis of the sentence entered by the user. In his sentences, the user can combine several basic motions into a more complex one (e.g. ”abduct the arm, and flex the elbow.”).

5 Results

In this section, we present two virtual experiments (control strategies) on the human arm. In the CHARM project, the resulting sequences were provided to other partners in order to compute muscle deformation (University of Lisbon) and to perform high-quality rendering (University of Karlsruhe). To this end, two main control strategies were tested: a simplified dynamic servoing (PID) control strategy, and a kinematic task (constraints-based controller) strategy. In the future, several other control strategies could also be tested, depending on medics requirements.

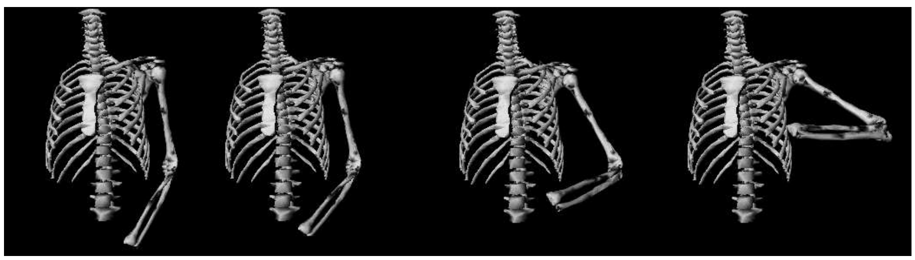


Figure 7: Sequence of abduction on the 11 DOF model of arm.

Upon completion of the embedded mechanical simulator of the human arm, a PID controller was designed for each DOF. In this design, the 11 DOFs mechanical model exhibits strong coupling and non-linearities. Within our control design software workshop, the mechanical model of the arm appears as a black-box where the inputs are the driving angular torques, and where the outputs are the arms degrees of freedom (DOF), as required for the simulation platform. A resulting sequence is given in figure 7 in which an abduction of the arm is performed.

A constraints-based controller has also been developed. This control strategy makes it possible to attach virtual contact points to the arm skeleton. Hence, the constraints-based control system provides a tool that exhibits similarity with inverse-kinematics, except that the motion is generated by the mechanical simulation. In addition constraints-based controllers also makes it possible to get valuable information about the actual torques that need to be generated in order to produce the constrained motion (see figure 8).

For the 11-DOF model, however, the penalty scheme that is employed to ensure constraint-fulfillment at runtime cannot provide such torque information. Hence a simplified 5 DOF model with no closed kinematic loop, had to be designed for this purpose. These data are produced by the simulation tool.

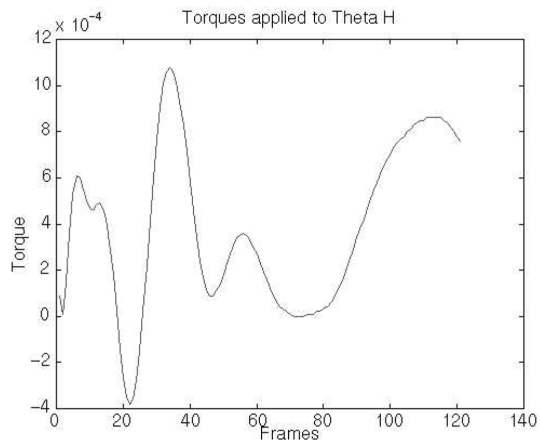


Figure 8: Torque computed by the inverse dynamic module.

Once a constraints-based controller with the 5 DOF model is chosen, the experimenter interactively specified the motion of the wrist by entering its goal Cartesian position in natural language. The torques are computed during the simulation, depending on the played motion and stored in a data file. In the particular case of the constraint-based controller, the definition of the control strategy is included into the definition of the mechanical model by inserting the constraints expressions. Thus, the controller module only consists in activating or not the constraints and providing the goals in the Cartesian frame.

6 Conclusion

We have described a comprehensive software workshop, to be used by biomechanicians and medics to test and validate various hypotheses about the dynamics and control of arm motion. In order to illustrate the possibilities of such a system, a simplified model of the human arm shoulder was designed. The non-linear equations of multi-body rigid motions were obtained by applying Lagrange mechanics to an energetic description. Alternative control strategies were suggested, as a set of tools. Although the proposed model takes into account a complex gliding contact at the shoulder, it is still only a starting point, that will need to be refined and made more accurate. Nevertheless, such refinements are well beyond our objective because we consider that they should be under the sole responsibility of biomechanicians and medics. As computer scientists, our goal was restricted to provide a flexible tool with which the experts of natural motion could play with. Therefore, openness and user-friendliness of the system were our main focus. With that respect, the system fulfills the goals by allowing biomechanicians to work on specific components of the motion process (mechanics, muscle models, and control strategies), while getting the whole picture about how motion is affected by their choices. In addition, a full suite of software tools was provided, to generate and update embedded models of arm dynamics. Finally, our job ends-up at the point where the system is ready to be handed-out to biomechanicians and medics.

7 Acknowledgments

We would like to thank Gilles Dietrich from INSEP[3] for his help and advises in the conception of this tool. We also thank the European community for supporting this work. A special thank to Walter Maurel from EPFL for his image of the topological model of arm.

A Equivalent muscles

As mentioned in [12], most studies on multi-muscle coordination assume frictionless pin and ball-and-socket joints. Since anatomic data on muscle attachment are not precise enough, simulating the arm motion by computing the torques delivered by each line action is not actually possible with a good accuracy. Moreover, as their is not enough data on the understanding of the strong

[3]INSEP: Institut National des Sports et de l'Education Physique. A biomechanical lab. working on sportive motions.

coupling of line action, it is more relevant to choose another muscle representation. So, muscle joint torques are linearly related to muscle moments. In the following, we refer to the *equivalent torque* as the torque which is engendered by the set of muscle lineic forces acting on a particular joint.

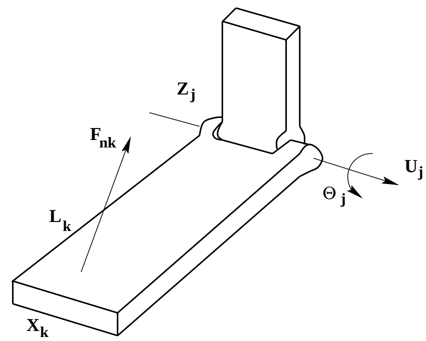


Figure 9: Muscle acting about a revolute joint.

Muscles can have a very wide insertion on the skeleton. Computation and application of the muscle effort on the skeletal model requires that muscles be discretized as a set of fibers, or macrofibers, having one identified path. The conversion of muscle forces to joint torques requires the determination of the levering effects, i.e. the determination of the effective moment arms. Such a transformation can be put in matrix form:

$$\tau = \mathcal{A}(q)F \tag{5}$$

where the matrix $\mathcal{A}$, referred-to as the anatomy matrix by [7] contains the instantaneous moment arms of the muscles about the joints:

$$\mathcal{A}_{jk} = (x_k - z_j) \times l_k u_j \tag{6}$$

where x_k is the point of attachment of the k-th muscle, z_j locates in space the unit vector of revolution u_j of the j-th joint, and l_k is a unit vector of the effective line of action of the k-th muscle. When several muscles act together about the same joint j, their action must be combined using summation over all muscles (see figure 9). As reported by [12], sensitivity analysis studies have shown that the behavior of the musculoskeletal model "tends to be very sensitive to the assumptions used to define the path (and consequently, moment arm) near the joint".

B Kinematic constraint at the shoulder

As opposite to typical robotic manipulators, the human arm is characterized by a complex closed-loop chain at the shoulder, which constraints the evolution of the mechanical system. The gliding contact of the scapula over the thorax can be modeled as follows: a fixed point P_c of the scapula is constrained to belong to an ellipsoid fitted to the thorax. See figure 2 for the geometric description of the ellipsoidal constraint. It's equation can be expressed in the principal frame of the ellipsoid $\mathcal{R}_e$, as follows:

$$\frac{x_e^2}{a^2} + \frac{y_e^2}{b^2} + \frac{z_e^2}{c^2} - 1 = 0 \tag{7}$$

Hence, the instant coordinates of P_c, (x_s, y_s, z_s), of fixed value in the scapula fixed frame R_s, must be transposed in the ellipsoidal frame, by computing symbolically the frame transfer matrix P, from $\mathcal{R}_e$ to $\mathcal{R}_s$:

$$\begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = P \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} \tag{8}$$

where P is evaluated numerically at each time step.

References

[1] R. Boulic and R. Mas. Inverse kinetics for center of mass position control and posture optimization. Technical report, EPFL, Lausanne, Switzerland, 1994.

[2] M. Bruwer and H. Cruse. The human arm as a redundant manipulator: the control of path and joint angles. *Biol. Cybern.*, (54):125–132, 1987.

[3] DM. Chi, JR. Clarke, BL. Webber, and NI. Badler. Casualty modeling for real-time medical training. *Presence*, 5(4):359–366, 1996.

[4] Rémi Cozot and Dominique Villard. Formal computation of non-rigid multibody systems dynamics. In SCS, editor, *10th European Simulation Multiconference*, Budapest Hungary, June 1996.

[5] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki. Trajectory formation of arm movement by cascade neural networks model based on minimum torque-change criterion. *Biol. Cybern.*, pages 275–288, 1990.

[6] W. Maurel, D. Thalmann, P. Hoffmeyer, P. Beylot, P. Gingins, P. kalra, and N. Magnenat-Thalmann. A biomechanical musculoskeletal model of human upper limb for dynamic simulation. In R. Boulic and G. Hegron, editors, *Proceedings of the Sixth EUROGRAPHICS Workshop on animation and simulation*, pages 121–136, 1996.

[7] S.G. Meek, J.E. Wood, and S.C. Jacobsen. *Modeling Musculoskeletal Movement Systems : Joint and Body Segmental Dynamics, Musculoskeletal Actuation, and Neuromuscular control; in Multiple Muscle systems, J.M Winters and S.L-Y Woo editors.* Springer Verlag, New York, 1990.

[8] S. Pieper, D. Zeltzer, and J. Rosen. Interactive computer graphics in surgical simulation. In *Proceedings of NICOGRAPH*, Tokyo, Japan, 1991.

[9] F. Scheepers, R.E. Parent, W.E. Carlson, and S.F. May. Anatomy-based modeling of the human musculature. In *Proccedings of the ACM-SIGGRAPH*, 1997.

[10] J. Wilhelms and A.V. Gelder. Anatomically based modeling. In *Proceedings of the ACM SIGGRAPH*, 1997.

[11] D.E. Young and R.A. Schmidt. Motor programs as units of movement control. *Making them move : mechanics, control and animation of articulated figures / edited by N. Badler, B. Barsky, D. Zeltzer*, pages 129–155, 1991.

[12] F.E. Zajac and J.M. Winters. *Modeling Musculoskeletal Movement Systems : Joint and Body Segmental Dynamics, Musculoskeletal Actuation, and Neuromuscular control; in Multiple Muscle systems, J.M Winters and S.L-Y Woo editors.* Springer Verlag, New York, 1990.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399