Comparison of control and optimization approaches for trajectory tracking in the forward dynamic simulation of biomechanical multibody systems

Á1varo Noriega[*], Urbano Lugrís[#], Javier Cuadrado[#]

[*] Department of Construction and Manufacturing Engineering
University of Oviedo
Campus de Gijón, C/Pedro Puig Adam, Edf. Dep. Oeste, Módulo 5, 33203 Gijón, Spain
noriegaalvaro@uniovi.es

[#] Laboratory of Mechanical Engineering
University of La Coruña
Escuela Politécnica Superior, Mendizábal s/n, 15403 Ferrol, Spain
ulugris@udc.es

ABSTRACT

In the study of biomechanical systems, it is common to have a parametric model of a multibody system which should perform a certain movement, usually defined by a motion capture of the system in real conditions. From these capture data, the inverse dynamic analysis provides a solution for drive torques that, when introduced into a forward dynamic analysis, generates a movement which is different from the one previously captured. In this paper, a simple multibody model and a movement are proposed to benchmark two approaches for the trajectory tracking problem: a control approach featuring a PD control with computed feedforward and PD feedback, and an optimization approach based on the parameterized histories of drive torques with and without computed feedforward. Results show that the proposed multibody model generates the same problems than more complex models, despite its apparent simplicity. It is also shown that the PD control with computed feedforward is faster and more accurate than any of the tested optimization methods.

Keywords: Biomechanics, Multibody system, Forward dynamics simulation, Control, Optimization.

1 INTRODUCTION

In the study of biomechanical systems and, more specifically, in the study of human gait, it is common to have a model of multibody dynamics for the individual whose motion is to be analyzed. This model can have more or less detail, but full fidelity is impossible since many biomechanical parameters are not directly measurable and must be estimated by means of scaling of tabulated data obtained from other analyzed individuals [1, 2].

It is also common to have an optical motion capture system as indicated in [3]. This system captures the position histories of a group of markers that are assumed rigidly attached to the bones of the individual. Data from this capture are filtered to remove noise and, from them, the coordinates that define the instantaneous position of the individual are calculated.

These histories of coordinates have no kinematic consistency since markers, in reality, are not rigidly attached to the bones and there may exist inaccuracies in the measurement. Therefore, data from these histories must undergo a process to ensure their kinematic consistency [4].

Once the histories of the coordinates that define the position of the biomechanical model are available, the histories of the degrees of freedom of the model (usually associated with rotations in joints) can be obtained.

After that, an approximation of these histories is carried out, for instance, by cubic splines. The histories of the coordinates can be differentiated with respect to time with different numerical techniques to obtain the velocity and acceleration, but the use of cubic splines is advantageous since it allows to implement the analytical differentiation very easily [3].

Then, if an inverse dynamic analysis is made with the histories of movement in the degrees of freedom calculated with the previously indicated procedure, the drive torques in the joints can be obtained.

To validate the histories of the drive torques so obtained, they can be introduced in the biomechanical model and perform a forward dynamic analysis to check whether the original captured movement is recovered. Conversely to what could be expected, it is found that the resulting gait is unstable [3].

The origin of this divergence of behaviours can be multiple. Possible errors in measurements, filtering, the process to ensure the kinematic consistency or the use of splines can lead to minor differences between the actual data and the data used to solve the inverse dynamic analysis prior to forward dynamic analysis. These differences are amplified during the process of integration of the differential equations of the model. The differences may also come from the mathematical modelling of real contacts between the biomechanical system and the environment. Finally, the integration process itself assumes that functions are continuous, which can lead to differences in points of rapid change or even discontinuity of these functions.

The interest of using forward dynamic analysis in simulation and control of biomechanical systems is due to the fact that it naturally enables the introduction of muscle models thus leading to a more realistic behaviour. To address the trajectory tracking problem various approaches have been proposed in the literature [5], being the most important ones control and optimization. Both have advantages and disadvantages, as indicated by Xiang [5], apparently being optimization a better choice for motion prediction, while control is preferable to track a known movement. However, it is interesting to quantify these advantages and disadvantages in a practical way.

To that end, the use of a simple multibody model with a single degree of freedom which allows fast simulation, but where typical problems of complex multibody models can occur, is proposed. A predefined movement is set to avoid the process of motion capture, filtering and kinematic consistency, which adds nothing to this research. The objective is that the multibody model tracks the predefined movement as accurately as possible within a forward dynamic analysis.

Two approaches are proposed to do this. On the one hand, a forward dynamic analysis with PD control and computed feedforward from the solution obtained in the inverse dynamic analysis is performed. On the other hand, a forward dynamic optimization where the design variables are the histories of the drive torques is carried out. These histories will be approximated by three techniques: Artificial Neural Networks (ANN), cubic splines (CS) and parametric functions (PF). The optimization is performed with and without computed feedforward and several optimization algorithms are used to solve the problem.

The performance of the different resulting methods is measured by means of the following three indicators:

- Root of mean squared error with respect to the desired rotation ($RMSE_{rot}$).
- Root of mean squared error with respect to the drive torque obtained in the inverse dynamic analysis ($RMSE_{torque}$).
- Runtime (t).

The paper is organized as follows. The second section describes the multibody model and the formulation for inverse dynamic analysis. The third section shows the control approach through a PD controller with computed feedforward. The fourth section presents the optimization approach with and without computed feedforward. The fifth section compares the results obtained by the different methods. Finally, the sixth section gathers the conclusions.

2 FOREARM MODEL AND INVERSE DYNAMIC ANALYSIS

A planar forearm multibody model with 1 degree of freedom in the elbow is proposed as case study. The schematic is shown in Figure 1.

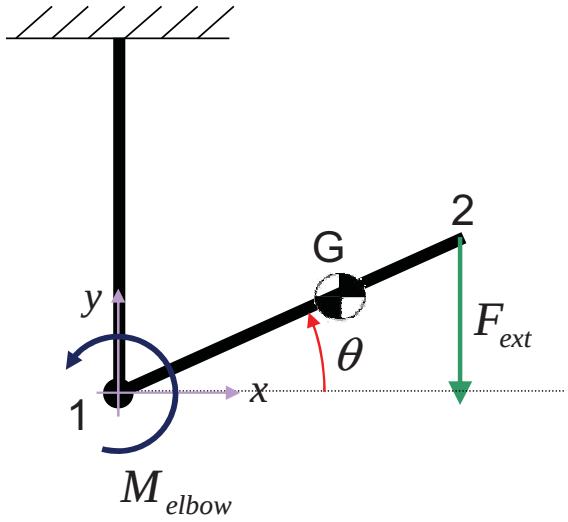

Figure 1. Schematic of 1-DOF forearm model.

The kinematic model is implemented by means of the mixed coordinates [6],

$$\mathbf{q} = \begin{bmatrix} x_2 & y_2 & \theta \end{bmatrix}^T \tag{1}$$

with the elbow rotation as independent coordinate,

$$\mathbf{z} = \begin{bmatrix} \theta \end{bmatrix}^T \tag{2}$$

The formulation to solve the inverse dynamic analysis at each instant of time and to calculate the drive torque at the elbow (M_{elbow}) is defined by expression (3) and its terms are explained in (4). Constants are indicated in Table 1.

$$\mathbf{Q}_m = \mathbf{R}^T \cdot \left(\mathbf{M} \cdot \ddot{\mathbf{q}} - \mathbf{Q}_{ext} \right) \tag{3}$$

$$\mathbf{M} = \begin{bmatrix} \frac{I_1}{L_{12}^2} & 0 & 0 \\ 0 & \frac{I_1}{L_{12}^2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{Q}_{ext} = \begin{bmatrix} 0 \\ -F_{ext} \\ 0 \end{bmatrix} + \mathbf{C}_\mathbf{G}^\mathbf{T} \cdot \begin{bmatrix} 0 \\ -m \cdot g \\ 0 \end{bmatrix} \qquad \mathbf{Q}_m = \begin{bmatrix} 0 \\ 0 \\ M_{elbow} \end{bmatrix} \tag{4}$$

$$\mathbf{C}_\mathbf{G} = \frac{L_{1G}}{L_{12}} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{R} = \dot{\mathbf{q}}\left(\mathbf{z}(\theta), \dot{\mathbf{z}}(\theta, 1) \right)$$

Table 1. Constant values.

Constant	Value
L_{12}	0,33 m
L_{1G}	0,11 m
m	1 kg
I_1	0,04 kg·m^2
F_{ext}	15 N
g	9,81 m/s^2

The simulation is performed in the time range [0,1] s, extracting values every 0.001 seconds.

The desired movement in the elbow is defined as is shown in (5):

$$\begin{aligned} t^*(t) &= a_1 \cdot t + (1-a_1)\cdot t^2 \\ \theta(t^*) &= L\cdot\left(t^* - \frac{1}{2\cdot\pi}\cdot\sin(2\cdot\pi\cdot t^*) - 1\right) \\ \dot{\theta}(t^*) &= L\cdot\left(1-\cos(2\cdot\pi\cdot t^*)\right) \\ \ddot{\theta}(t^*) &= 2\cdot\pi\cdot L\cdot\sin(2\cdot\pi\cdot t^*) \end{aligned} \tag{5}$$

Being $a_1 = 0.05$ and $L = \pi/4$.

These expressions allow to define the vector of independent coordinates and its derivatives as,

$$\mathbf{z} = \left[\theta(t)\right]^T \qquad \dot{\mathbf{z}} = \left[\dot{\theta}(t)\right]^T \qquad \ddot{\mathbf{z}} = \left[\ddot{\theta}(t)\right]^T \tag{6}$$

The equations to calculate the position, velocity and acceleration of the dependent coordinates are,

$$\begin{aligned} x_2 &= L_{12}\cdot\cos(\theta) \\ y_2 &= L_{12}\cdot\sin(\theta) \\ \dot{x}_2 &= -L_{12}\cdot\sin(\theta)\cdot\dot{\theta} \\ \dot{y}_2 &= L_{12}\cdot\cos(\theta)\cdot\dot{\theta} \\ \ddot{x}_2 &= -L_{12}\cdot\left(\cos(\theta)\cdot\dot{\theta}^2 + \sin(\theta)\cdot\ddot{\theta}\right) \\ \ddot{y}_2 &= L_{12}\cdot\left(-\sin(\theta)\cdot\dot{\theta}^2 + \cos(\theta)\cdot\ddot{\theta}\right) \end{aligned} \tag{7}$$

These expressions allow to define the vector of dependent coordinates and its derivatives,

$$\begin{aligned} \mathbf{q} = \begin{bmatrix} x_2 & y_2 & \theta \end{bmatrix}^T = \mathbf{q}(\mathbf{z}) \qquad \dot{\mathbf{q}} = \begin{bmatrix} \dot{x}_2 & \dot{y}_2 & \dot{\theta} \end{bmatrix}^T = \dot{\mathbf{q}}(\mathbf{z},\dot{\mathbf{z}}) \\ \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{x}_2 & \ddot{y}_2 & \ddot{\theta} \end{bmatrix}^T = \ddot{\mathbf{q}}(\mathbf{z},\dot{\mathbf{z}},\ddot{\mathbf{z}}) \end{aligned} \tag{8}$$

In Figure 2, the forearm rotation and the drive torque in the elbow obtained by inverse dynamic simulation are shown.

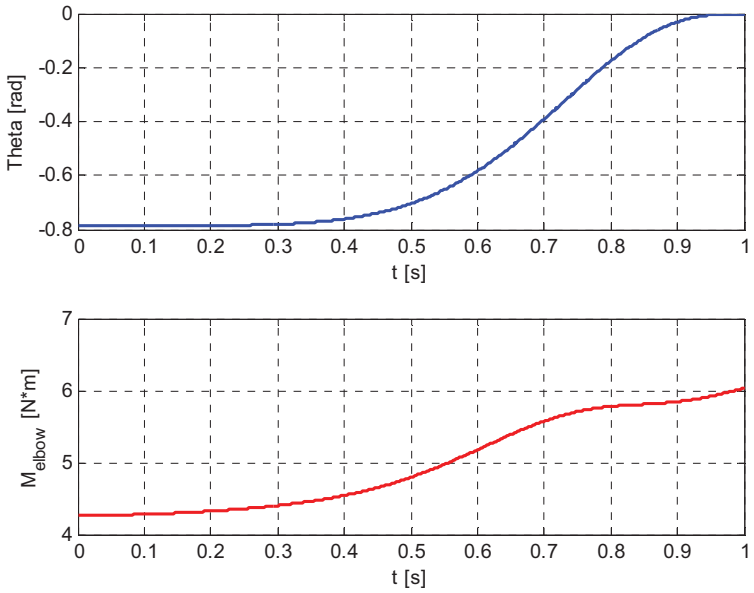

Figure 2. Forearm rotation and drive torque in the elbow obtained from inverse dynamic simulation.

3 FORWARD DYNAMIC ANALYSIS WITH PD CONTROL AND COMPUTED FEEDFORWARD

For the coordinates defined in (1), the set of kinematic constraints is specified in (9), the Jacobian matrix of the kinematic constraints is shown in (10), and its time derivative in (11).

$$\mathbf{\Phi}(\mathbf{q}) = \begin{bmatrix} (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_{12}^2 \\ (y_2 - y_1) - L_{12} \cdot \sin(\theta) \end{bmatrix} = \mathbf{0} \tag{9}$$

$$\mathbf{\Phi}_{\mathbf{q}}(\mathbf{q}) = \begin{bmatrix} 2 \cdot (x_2 - x_1) & 2 \cdot (y_2 - y_1) & 0 \\ 0 & 1 & -L_{12} \cdot \cos(\theta) \end{bmatrix} \tag{10}$$

$$\dot{\mathbf{\Phi}}_{\mathbf{q}}(\mathbf{q}) = \begin{bmatrix} 2 \cdot \dot{x}_2 & 2 \cdot \dot{y}_2 & 0 \\ 0 & 0 & L_{12} \cdot \sin(\theta) \cdot \dot{\theta} \end{bmatrix} \tag{11}$$

For the forward dynamic analysis, a penalty formulation is used [6]. It is defined by the following system of equations,

$$\left(\mathbf{M} + \alpha \cdot \mathbf{\Phi}_q^T \cdot \mathbf{\Phi}_q\right) \cdot \ddot{\mathbf{q}} = \mathbf{Q} - \alpha \cdot \mathbf{\Phi}_q^T \cdot \left(\dot{\mathbf{\Phi}}_q \cdot \dot{\mathbf{q}} + 2 \cdot \xi \cdot \omega \cdot \dot{\mathbf{\Phi}} + \omega^2 \cdot \mathbf{\Phi}\right) \tag{12}$$

The vector of generalized forces is described in (13) and the term due to the PD control is shown in (14).

$$\mathbf{Q} = \mathbf{Q}_{ext} + \mathbf{Q}_m + \mathbf{Q}_{control} \tag{13}$$

$$\mathbf{Q}_{control} = K_P \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left(\mathbf{q}_{desired} - \mathbf{q}(\mathbf{z})\right) + K_D \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left(\dot{\mathbf{q}}_{desired} - \dot{\mathbf{q}}(\mathbf{z}, \dot{\mathbf{z}})\right) \tag{14}$$

4 FORWARD DYNAMIC OPTIMIZATION

4.1 Without computed feedforward

The forward dynamics formulation for the forearm, with the drive torque defined by an Artificial Neural Network (ANN), a Cubic Spline (CS) or a Parametric Function (PF), is the one indicated in Section 3, except that the vector of generalized forces is the expression given in (15), where the term corresponding to the additional torque is shown in (16).

$$\mathbf{Q} = \mathbf{Q}_{ext} + \mathbf{Q}_{motor} \tag{15}$$

$$\mathbf{Q}_{\text{motor}} = \left[0 \quad 0 \quad ANN\left(t, \mathbf{p}_{ANN}\right) \text{ or } CS\left(t, \mathbf{p}_{CS}\right) \text{ or } PF\left(t, \mathbf{p}_{PF}\right)\right]^T \tag{16}$$

being

$ANN\left(t, \mathbf{p}_{ANN}\right)$ a parametric Artificial Neural Network which describes the history of drive torque in time;

$CS\left(t, \mathbf{p}_{CS}\right)$ a parametric Cubic Spline which describes the history of drive torque in time;

$PF\left(t, \mathbf{p}_{PF}\right)$ a Parametric Function which describes the history of drive torque in time.

If the history of the torque applied to the elbow is defined by an ANN, the artificial network has the structure shown in Figure 3.

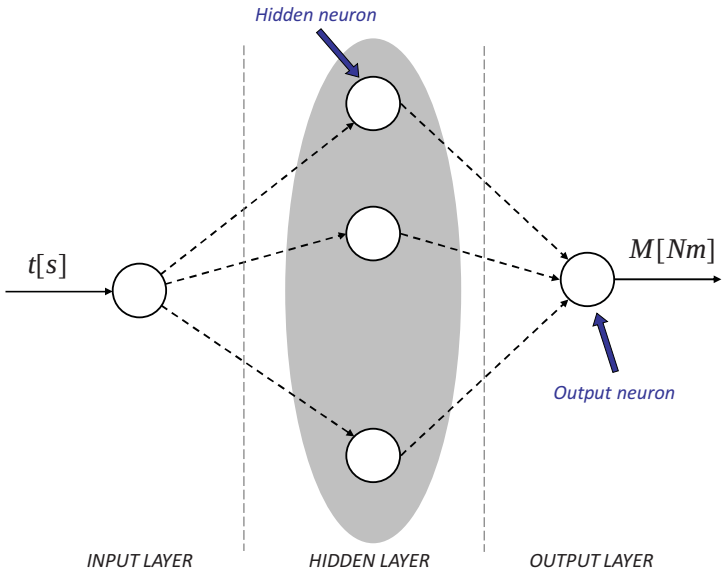

Figure 3. Structure of the ANN used.

ANN seems a good choice because of its flexibility to precisely adapt to non-regular shapes. The ANN is, in fact, a parametric model where the parameters are the weights of the synapses that connect neurons and the bias in these neurons. For the ANN shown in Figure 3, the number of parameters is $3 \cdot n_n + 1$ if the number of neurons in the hidden layer is n_n.

The number of neurons in the hidden layer determines the complexity that the torque history can have. The higher the number, the greater the complexity. The optimal number of hidden layer neurons generates a fit to the training data with sufficient accuracy while minimizing the number of neurons (which, in turn, minimizes the number of parameters), and is determined with the method shown in [7]. In this case, the training data come from the inverse dynamic

analysis, because it is assumed that the optimum history of drive torque for the forward dynamic analysis is quite similar. Therefore, a prior training of the ANN must be made with these data to determine the optimal number of neurons for the hidden layer, and the weights of synapses and bias for this optimal number of neurons. These values of the parameters for the ANN serve as initial guess for the optimization problem to be solved later. This process has been carried out and the optimal number of 4 neurons in the hidden layer has been obtained. The weights of synapses and bias that adjust the ANN to the results of the inverse dynamic simulation have also been obtained.

In case the history of the drive torque applied to the elbow is defined by a cubic spline, a set of time instants (here, every 0.1 s) are defined, being the parameters of the spline the values of torque at those instants, as shown in Figure 4.

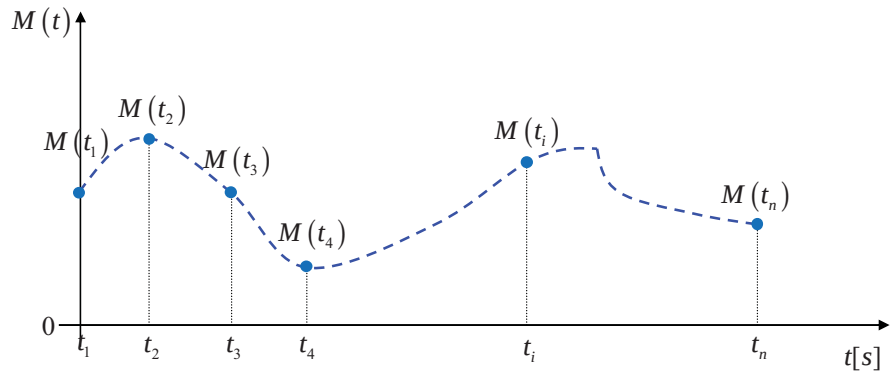

Figure 4. Cubic spline to describe the history of the drive torque at the elbow.

A cubic spline is very flexible to adapt itself to non-regular shapes with precision, but this adaptation is achieved by increasing the density of points that define the history, which means an increase in the number of required parameters.

Finally, if the history of drive torque in the elbow is defined by a parametric function, its structure must be defined as well as their parameters and the ranges in which they can move while maintaining the validity of the function. In this study, many postulated functions governed by simple parameters and sums of such functions have been tried, but to determine a parametric function that adapts itself with enough accuracy to the history of the torque obtained from the inverse dynamic simulation is almost impossible. Furthermore, the postulated function would only adapt to a very specific shape. If the movement changes, the drive torque also changes and then it is very probable that the former parametric function does not serve. For this reason, the PF without computed feedforward is not included in the benchmark.

Once the torque history is available by any of the described methods, the forward dynamic formulation explained in (12) can be implemented and the simulation of the movement with a set of parameters ($\mathbf{p}_{ANN}$, $\mathbf{p}_{CS}$ or $\mathbf{p}_{PF}$) carried out. The movement obtained from this simulation can be compared with the desired movement in order to obtain $RMSE_{rot}$. Then, the following optimization problem can be stated,

$$\min\ f(\mathbf{p}) = \min\ RMSE_{rot}(\mathbf{p}) \tag{17}$$

Being $f(\mathbf{p})$ the objective function that calculates the error indicated depending on the parameters that define the history of elbow torque ($\mathbf{p}$). This function internally makes the forward dynamic simulation of the motion with the formulation indicated in (12) and with the history of elbow drive torque defined by the ANN, the CS or the PF.

As the optimization algorithm requires that the objective function be surjective, the handling of cases where the integrator cannot complete the forward dynamic simulation in the complete range of time must be considered. To that end, the objective function is modified by adding a term which is proportional to the not completed range of time, $(1-t_{end})$, as shown in (1)8. Thus, sets of parameters that generate unstable simulations are penalized, being the penalty proportional to the degree of instability.

$$\min f^{*}(\mathbf{p}) = \min \left(RMSE_{rot}(\mathbf{p}) + \omega_{penal} \cdot (1-t_{end}) \right) \tag{18}$$

To select the optimization algorithm, the features of the problem and of the objective function must be taken into account:

- It is a minimization problem of a single objective.
- It has a large number of continuous variables.
- The derivative of the objective function with respect to the variables is unknown.
- It is unknown whether there will be one or more optima, and their relative values are also unknown.
- An approximation of the optimal solution is known (from inverse dynamic analysis).

Based on these features, two algorithms of different types are selected. On the one hand, evolutionary strategy CMA-ES [8] is chosen because it natively works with continuous variables, besides showing good performance in functions with several optima. On the other hand, a BFGS quasi-Newton method is selected, for which derivatives are numerically estimated . Both algorithms typically require few objective function evaluations. This is an interesting property to keep runtime within acceptable values.

Finally, the weight ω_{penal}, used to modulate the relevance of the set of parameters being capable of generating a stable simulation with the integrator used, is set to1.

4.2 With computed feedforward

The formulation with computed feedforward is the same indicated in Section 3, adding the feedforward term to the vector of generalized forces as shown in (19).

$$\mathbf{Q} = \mathbf{Q}_{ext} + \mathbf{Q}_{m} + \mathbf{Q}_{motor} \tag{19}$$

This approach is operationally similar to that described in Section 4.1 except that, in this case, the elbow drive torque is added to the one coming from inverse dynamic analysis. This makes that, most of the time, the value of the additional torque should tend to 0 and it only grows when there are significant variations in the error in position or its derivatives. Consequently, the initial guess for the parameters of the ANN and the CS is 0.

For que case of PF, the parametric function proposed in [2] is considered. This function has $3 \cdot n_t$ parameters, being n_t the number of terms. Its expression is shown in (20).

$$PF(t, \mathbf{p}_{PF}) = \sum_{i} A_i \cdot e^{-C_i \cdot (t-T_i)} \tag{20}$$

Being $\mathbf{p}_{PF} = [A_1 \quad C_1 \quad T_1 \quad \cdots]$

5 RESULTS

The code has been programmed in MATLAB® on a PC with Intel® Core Duo 2.67 GHz, 3.5 GB RAM and Windows 7.

The inverse dynamic simulation has a computation time of 0.12 s. Forward dynamic simulation with PD control and computed feedback was performed with a gain of $K_P = 80$ and $K_D = 0.3$ which have been adjusted by trial and error method. Simulation time was 0.075s and generates errors $RMSE_{rot} = 0.116$ and $RMSE_{torque} = 0.149$.

The forward dynamic optimization poses multiple options whose results are shown in Tables 2 and 3. Table 2 shows the case without computed feedforward and Table 3 shows the case with computed feedforward. The optimization problems have been solved allowing a maximum of 200 evaluations of the objective function in the optimization algorithms.

Table 2. Results of the forward dynamic optimization without computed feedforward.

History of drive torque defined by …	Optimization method used	
	CMA-ES	BFGS Quasi-Newton
ANN (4 hidden neurons)	$RMSE_{rot} = 1.250$ $RMSE_{torque} = 0.149$ $t = 411.56s$	$RMSE_{rot} = 0.310$ $RMSE_{torque} = 0.121$ $t = 449.37s$
CS (11 points)	$RMSE_{rot} = 0.558$ $RMSE_{torque} = 0.118$ $t = 29.03s$	$RMSE_{rot} = 0.197$ $RMSE_{torque} = 0.116$ $t = 41.59s$

Table 3. Results of the forward dynamic optimization with computed feedforward.

History of drive torque defined by …	Optimization method used	
	CMA-ES	BFGS Quasi-Newton
ANN (4 hidden neurons)	$RMSE_{rot} = 27.383$ $RMSE_{torque} = 0.366$ $t = 2439.15s$	$RMSE_{rot} = 2.079$ $RMSE_{torque} = 0,009$ $t = 1482.89s$
CS (11 points)	$RMSE_{rot} = 1.284$ $RMSE_{torque} = 0.105$ $t = 35.53s$	$RMSE_{rot} = 0.696$ $RMSE_{torque} = 0.046$ $t = 35.67s$
PF (4 terms)	$RMSE_{rot} = 1.829$ $RMSE_{torque} = 0.012$ $t = 24.11s$	$RMSE_{rot} = 1.816$ $RMSE_{torque} = 0.095$ $t = 17.09s$

In Figure 5, we can appreciate the discrepancy between the desired motion and the results provided by control-based approach and the optimization-based without feedforward approaches. In Figure 6, we can see the same comparison made with the optimization-based with feedforward approaches.

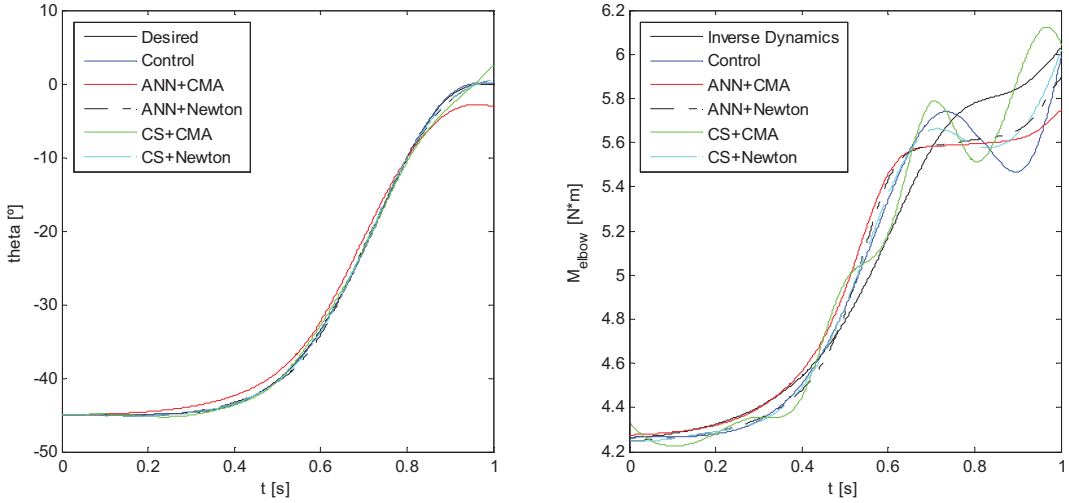

Figure 5. Comparison among control-based approach and optimization-based approaches without feedforward.

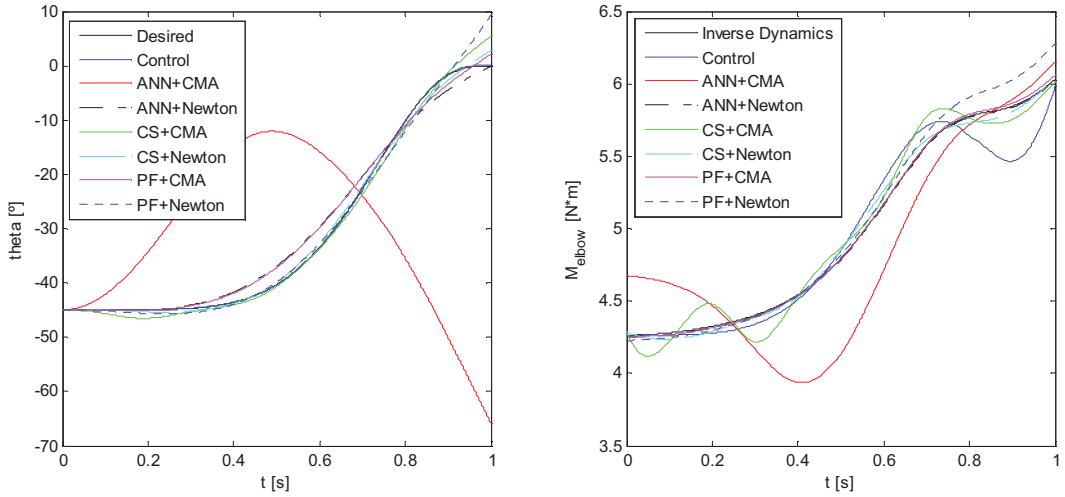

Figure 6. Comparison among control-based approach and optimization-based approaches with feedforward.

In general, optimization-based approaches work better without feedforward. Furthermore, the results show that the optimal solution is very sensitive to the optimization method used, with the BFGS quasi-Newton method obtaining the best results in all cases.

6 CONCLUSIONS

First, it has been shown that despite the simplicity of the proposed multibody model, it can generate the same problems as more complex models in the forward dynamic simulation.
Several approaches have been proposed to parameterize the history of drive torque in a flexible way.
The control-based approach is almost 3 or 4 orders of magnitude faster and more accurate than the optimization-based approaches.

Optimization-based approaches involve a high number of function evaluations, i.e. forward dynamic simulations. Moreover, convergence to a solution with an accuracy comparable to that obtained with the PD controller and within a reasonable time cannot be ensured. For this reason, the computational effort was limited in all the optimization algorithms.
In optimization-based approaches, the CS with and without computed feedforward has shown the best behaviour. Surprisingly, optimization-based approaches without feedforward have better behaviour than their counterparts with computed feedforward.
Regarding efficiency, the ANN is slower than the other two methods due to two reasons. First, the ANN needs a long time to be evaluated (around tenths of seconds), and this process must be repeated at every simulation instant (every 0.001s) or even more, depending on the integrator. Second, the drive torque history may include sudden variations that hinder the integration of the equations of motion, making the integrator to reduce the time-step size.
Finally, the results of optimization-based approaches have shown a greater or lesser dependence on the optimization methods used with them and even on the parameters of these methods.

ACKNOLEDGEMENTS

The support of this work by the Spanish Ministry of Economy and Competitiveness (MINECO) under project DPI2012-38331-C03-01, co-financed by the European Union through EFRD funds, is greatly acknowledged.

REFERENCES

[1] C.L. Vaughan, B.L. Davis, J.C. O'Connor. Dynamics of human gait, 2nd ed. Kiboho Publishers, 1999.

[2] J.A.C. Ambrosio, A. Kecskemethy. Multibody dynamics of biomechanical models for human motion via optimization, Multibody Dynamics – Computational Methods and Applications, J.C. Garcia Orden, J.M. Goicolea, J. Cuadrado, pp. 245-272, Springer, 2007.

[3] R. Pamies-Vila, J.M. Font-Llagunes, U. Lugris, J. Cuadrado. Forward Dynamics of Human Gait based on Control Techniques, In Proceedings of 3rd Joint Int. Conference on Multibody System Dynamics (IMSD) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), pp. 61-62, Busan, Korea, 2014.

[4] U. Lugris, J. Carlin, A. Luaces, J. Cuadrado. Gait analysis system for spinal cord injured subjects assited by active orthoses and crutches. Journal of Multi-body Dynamics, Vol 227, No 4, pp. 363-374, 2013.

[5] Y. Xiang, J.S. Arora, K. Abdel-Malek. Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches. Structural and Multidisciplinary Optimization, Vol 42, pp. 1-23, 2010.

[6] J. García de Jalón, E. Bayo. Kinematic and Dynamic Simulation of Multibody Systems – The real-time challenge. Springer-Verlag, 1994.

[7] A. Noriega, D. Blanco, B.J. Álvarez, A. García. Dimensional accuracy improvement of FDM square cross-section parts using artificial neural networks and an optimización algorithm. International Journal of Advanced Manufacturing Technology, Vol. 69, pp. 2301-2313, 2013.

[8] N. Hansen, A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. Evolutionary Computation, Vol 9, No 2, pp. 159-195, 2001.