Global *vs.* Local Nonlinear Optimization Techniques for Human-Like Movement of an Anthropomorphic Robot

Eliana Costa e Silva[*], M. Fernanda Costa[†], Wolfram Erlhagen[†] and Estela Bicho[**]

[*]*CIICESI, ESTGF, Polytechnic Institute of Porto, Portugal*
[†]*Centre of Mathematics/Dept. of Mathematics and Applications, University of Minho, Portugal*
[**]*Centre ALGORITMI/Dept. of Industrial Electronics, University of Minho, Portugal*

Abstract. In this paper a comparison between using global and local optimization techniques for solving the problem of generating human-like arm and hand movements for an anthropomorphic dual arm robot is made. Although the objective function involved in each optimization problem is convex, there is no evidence that the admissible regions of these problems are convex sets. For the sequence of movements for which the numerical tests were done there were no significant differences between the optimal solutions obtained using the global and the local techniques. This suggests that the optimal solution obtained using the local solver is indeed a global solution.

Keywords: Large scale nonlinear optimization, Ipopt, Nlopt, bimanual human-like movements, anthropomorphic robot
PACS: 02.60.Pn

INTRODUCTION

With the intent to use robots in environments initially design for human use, in the past few years there has been an increased interest for the field of dual-arm anthropomorphic robotic systems. Although there are many recent works on autonomous binamual manipulation in robotics (for a review see [1]), there is still a clear need for the development of new planning and control methods for human-like bimanual actions in humanoid robots [2].

Taking inspiration from the Posture-Based Motion Planning Model (PBMPM) of Rosenbaum and colleagues (e.g. [3]; [4]) which was proposed to explain how humans plan goal-directed upper limb movements, in [5] it was presented a model for the generation of human-like bimanual movements, including obstacle avoidance. The movement generation in general involves the selection of a final and a bounce posture. These two problems are modeled as nonlinear constrained optimization problems. The numerical results in [5] were obtained using Ipopt [6], proved to be quite adequate for solving the large scale problems that arise in movement generation. However, since the optimization problems may be nonconvex ones, many local minima may exist in the feasible region. When using the Ipopt solver (which is a local optimizer) there is only guarantee that (local) optimal solutions are obtained. In this paper we make a comparison between the results obtained using this local solver and the global solver provided by Nlopt in the MATLAB OPTimization Interface (OPTI) Toolbox.

THE MODEL

This section presents a summarized description of the model that was presented in [5].

The anthropomorphic robot, **ARoS**, has two anthropomorphic arms and hands, which can be represented as a series of links connected by joints. The number of joints which can be independently actuated define its DOFs. Each of **ARoS**' anthropomorphic robotic arm has 7 DOFs, $\theta_1^a, \ldots, \theta_7^a$, and each hand has 4 DOFs, $\theta_8^a, \ldots, \theta_{11}^a$. Therefore, the arm and hand configuration in joint space is defined by the vector $\boldsymbol{\theta}^a = (\theta_1^a, \theta_2^a, \ldots, \theta_{11}^a)^\top$, where $a = R$ or $a = L$, for the right or left arm and hand, respectively. The movement of the joints of each robotic arm and hand, $a \in \{L, R\}$, is given by: $\boldsymbol{\theta}^a(t) = \mathscr{T}^a(t, \boldsymbol{\theta}_f^a, \boldsymbol{\theta}_b^a) = \boldsymbol{\theta}_0^a + \mathscr{T}_{direct}^a(t, \boldsymbol{\theta}_f^a) + \mathscr{T}_{bk}^a(t, \boldsymbol{\theta}_b^a)$, where $\mathscr{T}_{direct}^a$, is a *direct movement* describing a bell-shaped unimodal velocity profile, from the initial posture, $\boldsymbol{\theta}_0^a$, to a final posture, $\boldsymbol{\theta}_f^a$; and $\mathscr{T}_{bk}^a$ is a *back-and-forth movement* from initial to a bounce posture, $\boldsymbol{\theta}_b^a$, intended to avoid collision with obstacles in the robot's workspace. In general, the movement planning of arm-hand $a \in \{R, L\}$ involves the resolution of two problems: $\mathbf{Pa}^a$ - determining the appropriated final posture that allows, for e.g., **ARoS** to grasp a given object or to achieve a specific location, with a particular grip type; $\mathbf{Pb}^a$ - determining a bounce posture, that serves as a sub-goal for a *back-and-forth* movement.

Problems $\mathbf{Pa}^a$ and $\mathbf{Pb}^a$ were modeled as nonlinear constrained optimization problems, with bounds, equality and inequality constraints. The **Pa** is solved first and its solution $\boldsymbol{\theta}_f$ is used as a parameter in corresponding $\mathbf{Pb}^a$ that is solved next. We use a direct transcription method, therefore, $t \in [0, T_\mathrm{d}]$ is discretized in N_T equally spaced points $t_i = i\Delta$, where $\Delta = \frac{T_\mathrm{d}}{N_T}$ is the step size and $i = 0, 1, \ldots, N_T$. Our convention is that $\mathscr{T}_i^a \equiv \mathscr{T}^a(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ represents $\mathscr{T}^a(t, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ at time t_i. Mathematically we formulate these problems as:

$$
\begin{array}{lll}
\mathbf{Pa}^a & \min\limits_{\theta_1^a,\ldots,\theta_7^a} & \sum\limits_{k=1}^{7} \left(\theta_{0,k}^a - \theta_{f,k}^a\right)^2 \\
 & \text{s.t.} & \boldsymbol{h}_1^a(\theta_{f,1}^a, \ldots, \theta_{f,7}^a, \theta_{f,9}^a) = \mathbf{0} \\
 & & \boldsymbol{h}_2^a(\theta_{f,1}^a, \ldots, \theta_{f,7}^a) = \mathbf{0} \\
 & & \boldsymbol{h}_f^a(\theta_{f,1}^a, \ldots, \theta_{f,11}^a) \leq \mathbf{0} \\
 & & \theta_{m,i}^a \leq \theta_{f,i}^a \leq \theta_{M,i}^a, \quad i = 1, \ldots, 7
\end{array}
$$

$$
\begin{array}{lll}
\mathbf{Pb}^a & \min\limits_{\theta_{b,1}^a,\ldots,\theta_{b,n_j^a}^a} & \sum\limits_{k=1}^{n_j^a} \left(\theta_{0,k}^a - \theta_{b,k}^a\right)^2 \\
 & \text{s.t.} & \boldsymbol{\theta}_m^a \leq \mathscr{T}^a(t_i, \boldsymbol{\theta}_f^a, \boldsymbol{\theta}_b^a) \leq \boldsymbol{\theta}_M^a \\
 & & \underline{\boldsymbol{h}}_b^a(\mathscr{T}^a(t_i, \boldsymbol{\theta}_f^a, \boldsymbol{\theta}_b^a)) \leq \mathbf{0} \\
 & & \overline{\boldsymbol{h}}_b^a(\mathscr{T}^a(t_i, \boldsymbol{\theta}_f^a, \boldsymbol{\theta}_b^a), \varepsilon(t_i)) \leq \mathbf{0}, \quad t_i = 0, \ldots, T_\mathrm{d} \\
 & & \boldsymbol{\theta}_m^a \leq \boldsymbol{\theta}_b^a \leq \boldsymbol{\theta}_M^a
\end{array}
$$

where $\theta_{m,i}^a$ and $\theta_{M,i}^a$ are constants that represent the lower and upper joint limits of each arm $a \in \{R, L\}$ respectively; $\boldsymbol{h}_1^a$ and $\boldsymbol{h}_2^a$ are nonlinear functions (of target pose and joint angles) concerning the position and orientation of the robot hand relatively to the target, respectively; $\boldsymbol{h}_f^a$ are nonlinear functions of the obstacles pose and arm-hand angles, and is concerned with collision avoidance at the moment of grasp, with all the obstacles in the workspace, $\varepsilon(t_i)$ is a function of time representing the clearance distance, and $\underline{\boldsymbol{h}}_b^a$, $\overline{\boldsymbol{h}}_b^a$ are nonlinear functions of the obstacles pose and of the arm-hand angles. $\underline{\boldsymbol{h}}_b^a$ represents collision avoidance for all the time instants in the movement. Finally, $\overline{\boldsymbol{h}}_b^a$ deals with collision avoidance with the object to be grasped.

RESULTS

Here the focus is on the sequence of movements presented in [5], involving both arms and hands of the anthropomorphic robot: *Movement 1* - reaching and grasping a column from the table with the left arm-hand; *Movement 2* - transporting the column from the left to the right hand; *Movement 3* - reaching and grasping the column using the right arm-hand; *Movement 4* - transporting the column and and plugging it into a specific hole in the round base. Table 1 presents the problems corresponding to each of these movements.

TABLE 1. Problems description.

Movement	Arm-hand	Final Posture Selection	Bounce Posture Selection	Movement	Arm-hand	Final Posture Selection	Bounce Posture Selection
1	left	$\mathbf{P1a}^L$	$\mathbf{P1b}^L$	3	right	$\mathbf{P3a}^R$	$\mathbf{P3b}^R$
2	left	-	$\mathbf{P2b}^L$	4	right	$\mathbf{P4a}_1^R$+$\mathbf{P4a}_2^R$	$\mathbf{P4b}^R$

All optimization problems, $\mathbf{P\#a}^a$ and $\mathbf{P\#b}^a$ ($a \in \{R, L\}$), were coded in AMPL modeling language and solved using Ipopt 3.11 and Nlopt 2.3, both available from OPTI Toolbox [7]. Ipopt implements an interior point filter line search method [6] that aims to find a (local) solution of a twice continuously differentiable nonlinear problem. The initial guess for Ipopt was generated inside the simple bounds of the variables and is the same for all problems. For finding a global solution the implementation of the Improved Stochastic Ranking Evolution Strategy (ISRES) algorithm provided by the Nlopt solver was used. This population-based stochastic derivative-free algorithm implements a evolution strategy that is based on a combination of a mutation rule and a Nelder-Mead-like update rule (for details see [8]).

The numerical results were obtained using a core2Duo-2.66GHz, 4Gb de RAM. In the implementation of $\mathbf{Pa}^a$ and $\mathbf{Pb}^a$, the equality constraints were transformed into inequality constrained considering its squared euclidean norm and using δ = 0.001. OPTI Toolbox was used with its default options, with the exception of the maximum number of function evaluations, the maximum number of iterations and the maximum computational time, set to 1.00e+10, 3.00e+03 and 3.00e+02 seconds, respectively. Ipopt was run with the default options, with the exception of the second order derivatives information that were approximated using a limited-memory Broyden-Fletcher-Goldfarb-Shanno method and using 'acceptable_tol' equal to 1e-5. In the ISRES algorithm the size of the population is made to depend on the problem dimension and is set to $20(N+1)$. For each optimization problem 30 independent runs were carried out for comparison with the results obtained for a single run using Ipopt.

The numerical results are presented in Tables 2 and 3, where N is the number of variables, M is the total number of constraints, obj* is the optimal objective function value, viol* is the infinite norm of the constraint violation vector in the optimal solution, CPU is the computational time in seconds, and nfe is the number of function evaluations. The results from the 30 runs using Nlopt were ranked according to viol*. In order to show more details concerning the quality of the obtained solutions, in the tables, the results for the best run, the average (between brackets) and the standard deviation of the objective function values, St.D, for the 30 runs are presented.

TABLE 2. Comparison of the results obtained for the $\mathbf{P}^a$#a using Ipopt *verus* Nlopt with ISRES.

			Ipopt				Nlopt using ISRES				
Prob.	N	M	obj^*	$viol^*$	CPU	nfe	obj^*	$viol^*$	CPU	nfe	St.D
$\mathbf{P1a}^L$	7	85	6.347e00	0.00e00	3.819e-01	353	6.347e00 (6.412e00)	0.00e00 (7.129e-11)	3.046e+01 (2.001e+01)	197273 (129595.4)	6.850e-02
$\mathbf{P3a}^R$	7	90	3.012e00	0.00e00	1.66e00	985	3.012e00 (3.051e00)	0.00e00 (5.329e-10)	2.687e+01 (2.812e+01)	288596 (328550.4)	3.464e-02
$\mathbf{P4a}_1^R$	7	75	6.849e-01	0.00e00	4.046e-01	250	6.849e-01 (6.918e-01)	0.00e00 (9.876e-10)	8.114e00 (7.807e00)	100177 (96268.83)	2.150e-02
$\mathbf{P4a}_2^R$	7	75	5.358e-02	1.246e-09	2.894e-01	114	5.359e-02 (5.381e-02)	0.00e00 (0.00e00)	6.641e00 (1.562e+01)	81832 (243375.03)	3.267e-04

TABLE 3. Comparison of the results obtained for the $\mathbf{P}^a$#b using Ipopt *verus* Nlopt with ISRES.

			Ipopt				Nlopt using ISRES				
Prob.	N	M	obj^*	$viol^*$	CPU	nfe	obj^*	$viol^*$	CPU	nfe	St.D
$\mathbf{P1b}^L$	9	1423	1.453e-01	0.00e00	1.006e00	58	8.074e-02 (8.079e-02)	0.00e00 (0.00e00)	2.734e+02 (2.293e+02)	91746 (76838.07)	6.522e-05
$\mathbf{P2b}^L$	7	718	1.051e-01	0.00e00	7.237e-01	97	1.051e-01 (1.052e-01)	0.00e00 (0.00e00)	3.15e+01 (4.368e+01)	74284 (67393.73)	6.784e-05
$\mathbf{P3b}^R$	9	458	7.679e-16	0.00e00	4.740e-01	42	2.562e-07 (4.136e-06)	0.00e00 (0.00e00)	1.846e+01 (1.596e+01)	49871 (43066.9)	4.647e-06
$\mathbf{P4b}^R$	7	773	3.208e-02	0.00e00	9.142e-01	135	3.209e-02 (1.003e-01)	0.00e00 (0.00e00)	7.172e+01 (6.917e+01)	74230 (71601.53)	2.364e-02

For each problem, the results of the 30 runs of Nlopt show consistence as can be seen by the St.D which are very close to zero. This strongly suggests that, for each problem, the best optimal solution found was indeed the global one. Moreover, the optimal values obtained using Ipopt and the best solution found by Nlopt, presented in Tables 2-3, are approximately the same in all problems. The same happens for the minimizer found by each solver, as can be seen by the trajectories of the robotic arms-hands and the tangential hand velocities for each of the movements, in Figures 1, 2 and 3, where the solutions obtained using Ipopt and Nlopt are quite similar for movement 1, and almost indistinguishable for the remaining movements. As expected the CPU and the nfe for Nlopt are larger than the ones of Ipopt.

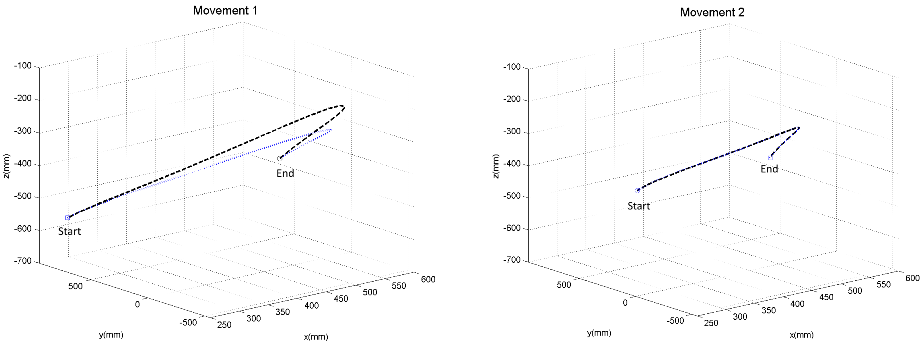

FIGURE 1. Hand trajectory in 3d for movements 1 and 2, for Ipopt (in black) and for Nlopt (in blue).

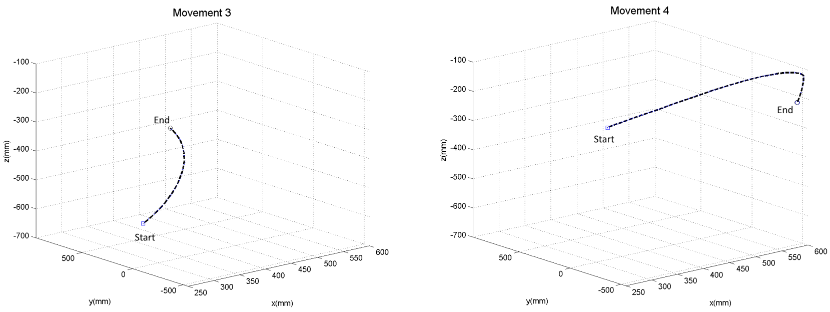

FIGURE 2. Hand trajectory in 3d for movements 3 and 4, for Ipopt (in black) and for Nlopt (in blue).

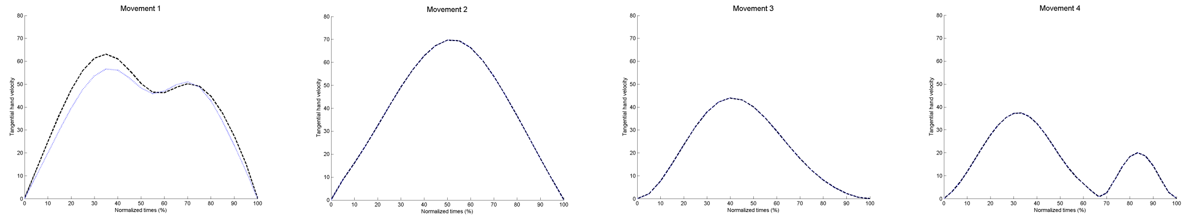

FIGURE 3. Tangential hand velocity for Ipopt (in black) and for Nlopt (in blue).

CONCLUSIONS

A comparison between global and local optimization techniques for solving the problem of movement generation of an anthropomorphic dual arm robot was made. The algorithm ISRES from Nlopt and Ipopt were used. No significant differences between the optimal solutions obtained using both solvers were found. The same happens for the generated trajectories. The consistent results found by Nlopt, in the 30 runs of each problem, strongly suggest that the global solution was found, and therefore the solution obtained using Ipopt was in fact the global one.

ACKNOWLEDGMENTS

This work has been supported by FCT through projects: PEst-OE/EEI/UI0319/2014 and PEst-OE/MAT/UI0013/2014, and by the EU Project PF7 Marie Curie: NETT-Neural Engineering Transformative Technologies.

REFERENCES

1. C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D.V. Dimarogonas and D. Kragic, Dual arm manipulation - A survey, Autonomous Systems **60**, 1340–1353 (2012).
2. F. Zacharias, C. Schlette, F. Schmidt, C. Borst, J. Rossmann and G. Hirzinger, Making planned paths look more human-like in humanoid robot manipulation planning, Proceedings of IEEE Inter. Conf. on Robotics and Automation, 1192–1196 (2011).
3. D. Rosenbaum, R. Meulenbroek, J. Vaughan and C. Jansen, Coordination of reaching and grasping by capitalizing on obstacle avoidance and other constraints, Exp Brain Res **128**, 92–100 (1999).
4. D. Rosenbaum, R. Meulenbroek, J. Vaugham and C. Jansen, Posture-based Motion planning: Applications to grasping, Psychological Review **108**, 709–734 (2001).
5. E. Costa e Silva, F. Costa, Z.P. Araújo, D. Machado, L. Louro, W. Erlhagen and E. Bicho, Towards human-like bimanual movements in anthropomorphic robots: a nonlinear optimization approach, Appl. Math. Infor. Sci., Vol. 9, **2** (2015).
6. A. Wächter and L. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming **106**, 25–57 (2007).
7. J. Currie, and D. I. Wilson, OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User, in Foundations of Computer-Aided Process Operations, edited by N. Sahinidis, and J. Pinto, Savannah, Georgia, USA, 2012.
8. T. Philip Runarsson, and X. Yao, Search biases in constrained evolutionary optimization, IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews, vol. 35, **2**, 233–243 (2005).