2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Computed Muscle Control for an Anthropomimetic Elbow Joint

Michael Jäntsch, Steffen Wittmeier, Konstantinos Dalamagkidis and Alois Knoll

***Abstract*—The soft robotics approach is widely considered to enable human-friendly robots which are able to work in our future homes and factories. Furthermore, achieving the smooth and natural movements of humans has become a hot topic in robotics, especially when robots are supposed to work in close proximity to humans. The anthropomimetic principle aims at mimicking not only the outside but also the inner mechanisms of the human body in humanoid robots. However, for this class of robots there exist as yet no scalable controllers that might make it possible to control a full body, or even several joints. A very similar problem is ongoing research in biomechanics which is the computation of muscle excitation patterns for coordinated movements. For this purpose, biomechanicists have developed *computed muscle control* which has proven a very scalable technique.**

In this paper, we demonstrate the adaptation of *computed muscle control* for a tendon-driven robot, comparing different methods for obtaining the muscle kinematics, as well as different low-level controllers. Results are shown for the implementation on a distributed control architecture and a single revolute elbow joint.

I. Introduction

Standard industrial manipulators offer extremely precise performance for repetitive tasks for which trajectories can be planned ahead, as would be the case in the well-structured environments of a factory. However, when unstructured environments are considered, humans still outperform robots in almost every aspect. This is due to the uncertainty that comes with these environments which cannot be sufficiently perceived and modeled, leading to possible collisions. In case of a rigid impact, however, a stiff robot is very likely to damage itself or its surroundings. The soft robotics approach is therefore widely considered to enable human-friendly robots which will be able to work in our modern homes and factories [1]. Humans display extremely smooth and effortless motor control over a certainly very complex body, which is still far from being matched by any robot. Achieving these smooth and natural movements has become a hot topic in robotics, especially when robots are supposed to work in close proximity to humans [2]. We can try to achieve this by actively computing human-like trajectories, but as long as the internal mechanisms of the robot are so different from the ones in the human body, this may never be truly achieved. One way towards this goal is to incorporate more and more of the mechanisms that can be found in humans for robots.

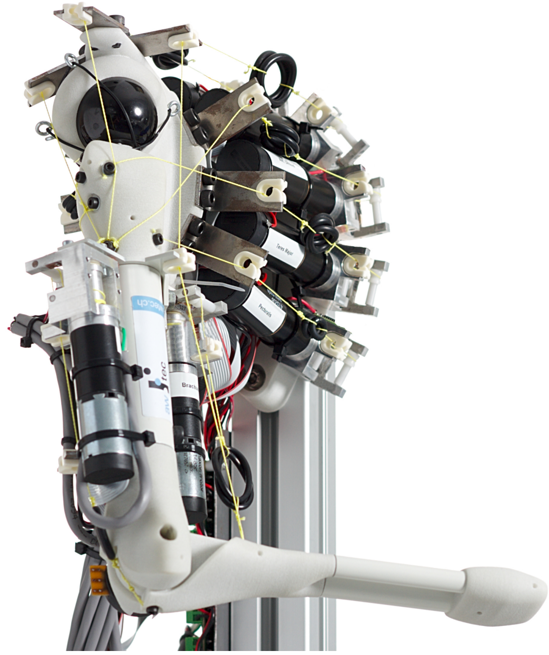

Fig. 1. ANTHROB. A 3D printed anthropomimetic robot with a spherical shoulder, and a revolute elbow joint. Eleven tendon-driven compliant muscles are realized with DC motors, kite line and elastic elements.

The anthropomimetic principle [3] aims at mimicking not only the outside but also the inner mechanisms of the human body in humanoid robots. However, for this class of robots there exist as yet no scalable controllers that might make it possible to control a full body, or even several joints. With the design come several challenges which are not present in other robotic systems, like complex joint types (e.g. ball-and-socket joints), multi-articular muscles, and most importantly muscles that collide with the rigid structure of the robot as well as with other muscles.

Tendon-driven robots can be differentiated by the tendon configuration which is the number of tendons per *degrees of freedom* (DoF) in the joints. Three main configurations exist, N, N+1, and 2N [4], while models of human bodies as well as anthropomimetic robots have to be classified as 2N+, denoting many more muscles per DoF in the joints than 2N. Control approaches from the robotics community like [4–7] deal only with the 2N configuration. For robots of the 2N+ configuration with complex joint types and colliding muscles, on the other hand, a control problem has to be

The authors are with the chair for Robotics and Embedded Systems, Department of Informatics, Technische Universität München, Munich, Germany
michael.jaentsch@in.tum.de

solved that is also ongoing research in biomechanics. Here, the goal is to compute muscle excitation patterns for coordinated movements [8]. In biomechanics the motivation is to find out how different muscles are involved when the human body performs certain tasks, where only a few states, like joint angles or excitation patterns of some muscles, can be measured non-invasively. A robot controller on the other hand should drive the system to perform tasks as desired [9]. For this purpose biomechanicists have developed *computed muscle control* [8] which is extended by de Sapio et. al. [10, 11] into a highly generic approach to control a human shoulder complex in simulation. While these techniques address exactly the challenges of controlling anthropomimetic robots, they lack the measurable performance criteria like robustness and stability for actual robotic systems. Robotic controllers on the other hand lack the scalability for more complex robots. The controller proposed in [12] has proven useful to control the simulation of an anthropomimetic robot arm and will in this work be evaluated for controlling a physical robot based on the control architecture described in [13].

II. The Platform

In this work, a robot is utilized that implements the anthropomimetic idea for a humanoid robot arm with a spherical shoulder joint, a revolute elbow joint and a total of eleven muscles (see Fig. 1). Therefore, the human skeleton, as well as the muscular system is mimicked as close as possible. The artificial muscles (AMs) consist of a DC motor, kite line, and an elastic element, allowing the electric actuator to wind the kite line on the attached spindle and hence either innervate or relax the AM. Therefore, force can only be exerted on the attachment points in one direction, i.e. a muscle can only pull, not push. The elastic element adds the flexibility that is also present in a biological muscle. Even though the technical implementation of the AMs cannot fully capture the muscular contraction dynamics of biological muscles, key features like the elasticity and most importantly the muscle insertion points have been emcompassed.

Both manufacturing techniques, as well as the sensory system were largely improved from the previous anthropomimetic robots, like CRONOS [3] or the ECCE [14]. Especially utilizing 3D printing techniques to fabricate the anthropomorphic structures, makes it now possible to extend previously developed controllers [12] onto real physical robots, as CAD data is available for obtaining reliable kinematic and dynamic system models. Furthermore, this robot features a potentiometer in the elbow for sensing the elbow angle. In this work we cover the control of the revolute elbow joint with two AMs, the Brachialis, and the Triceps as an exemplary configuration to prove the feasibility of the control scheme for robot control. Note that both muscles are implemented as mono-articular muscles here. While this is physiologically correct for the Brachialis, the human Triceps has three heads, of which only the lateral and medial heads are mono-articular. The attachment point chosen for the artificial Triceps is equivalent to the medial head.

III. Modeling Musculoskeletal Robots

In [4, 7] a model of a tendon-driven robot is obtained for the full state-space. This leads to a very complex and also non-linear model for which a controller can be found, provided the system is small enough. However, the process of developing these models and corresponding controllers shows that these controllers lack the scalability for larger systems.

To cope with more complex systems, it is possible to decompose them into a hierarchy of simpler subsystems for which separate control methods can be derived. An anthropomimetic robot can be divided in three subsystems. First a model of the comparably stiff robot components—the skeleton—can be obtained like for any conventional robot (Section III-A), second the AMs are modeled (Section III-B), and last a mapping between the two needs to be found (Section III-C).

A. Skeleton Model

For conventional robots the equation of motion can be expressed in one of two canonical forms [15]. In joint space this is written as follows,

$$\tau = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_G(q) \tag{1}$$

giving a relationship between the joint torques τ and the generalized joint coordinates q, $\dot{q}$, and $\ddot{q}$. Note that in the following, dependencies of q and $\dot{q}$ of the mass matrix H, the matrix of coriolis and centrifugal terms C and the vector of gravity terms τ_G are omitted. A system model in the canonical form can also be found for an anthropomimetic robot, by the well known methods of the Newton-Euler Algorithm or more efficiently using the Composite Rigid Body Algorithm [16]. For a more detailed analysis on how spherical joints can be integrated into the canonical form, refer to [12]. To encompass the effect of Coulomb friction in the canonical form, an additional term τ_{F} [17] has to be added to (1), denoting a constant coulomb friction term τ_c.

$$\tau = H\ddot{q} + C\dot{q} + \tau_G + \tau_{\mathrm{F}} \quad \text{with} \quad \tau_{\mathrm{F}} = \tau_c \cdot \mathrm{sgn}(\dot{q}) \tag{2}$$

B. Muscle Model

The AMs of the anthropomimetic robot consist of a DC motor, kite line and shock cord. A model of an AM could be obtained by combining the standard DC motor model with a model of the gearbox and a linear spring ($F = k \cdot \Delta x$). This led to a continuous time state-space model of the following form,

$$\begin{aligned} \dot{\mathrm{x}} &= A\mathrm{x} + bu_A + g\dot{x} \\ f &= c^T\mathrm{x} \end{aligned} \tag{3}$$

while x is the vector of states, the DC motor voltage u_A is the input and muscle force f is the output of the system. The system disturbance for the low-level muscle control, which is essentially the joint movement, was modeled as a linear coordinate x at the attachment point beyond the elastic element, while $\{A, b, c^T, g\}$ denote the linear system (see also Fig. 4). For a more detailed description of the muscle model, used in this work, please refer to [12].

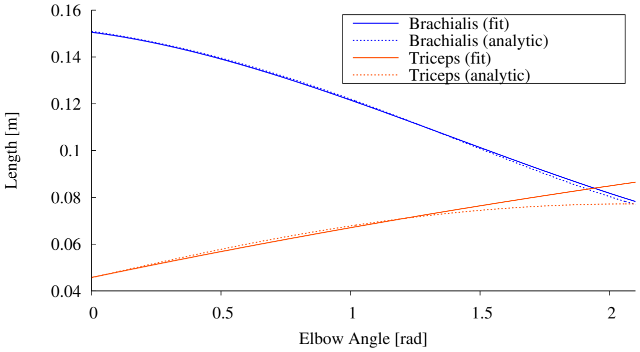

Fig. 2. Muscle lengths for the Brachialis and Triceps are shown, comparing the polynomial fit from samples against the analytic function obtained by assuming straight line muscles.

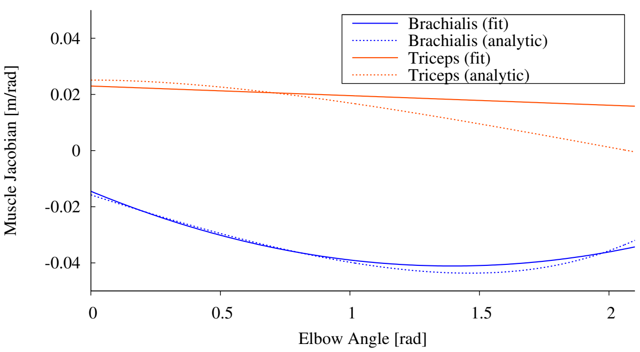

Fig. 3. Muscle Jacobians from differentiating both the polynomial fit and the analytic muscle length functions.

C. Muscle Jacobian

A relationship between the muscle and the joint space can be formulated based on the geometric mapping of muscle lengths l, subject to the joint angles q.

$$l = f(q) \tag{4}$$

By partially differentiating this function with respect to the joint angles the so called muscle Jacobian $L(q)$ is obtained [18]. It gives a relation between the change of the muscle lengths with respect to the change of joint angles at a certain configuration.

$$L(q) = \frac{\delta l}{\delta q} \quad \Rightarrow \quad L(q) \cdot \dot{q} = \dot{l} \tag{5}$$

Using the principle of virtual work, this can be transformed to a relation between the muscle forces f (the negative sign arises from the definition of a positive force when pulling) and the joint torques τ [10].

$$\tau = -L^T(q) \cdot f \tag{6}$$

The muscle Jacobian can be obtained in different ways. Either by directly modeling the moment arms of the muscles [4] or by finding a geometric representation of the muscle lengths and subsequently differentiating it with respect to the joint angles [19].

Finding geometric representations of the muscle lengths is a complex and error prone task, and becomes almost impossible for spherical joints where muscles might also wrap around skeletal structures. Therefore, we propose a third possibility which is to numerically approximate (4) by drawing samples from the system. In the following, a comparison between a numerical approximation and the geometric solution will be drawn for the example of the elbow joint with two muscles. Samples were obtained by utilizing the force control algorithm (see Section IV-A) to maintain a minimum tension of $1\,\mathrm{N}$ and manually moving the joint, covering the work space several times. Throughout the experiment motor positions, joint angles and muscle forces were recorded and subsequently used for the numerical approximation. Samples were generated by moving the elbow joint 4 times and sampling with $50\,\mathrm{Hz}$. By removing samples, where forces were outside of a bound of $\pm 0.2\,\mathrm{N}$ of the reference force, the number of samples was reduced to 1,708. This was necessary, as back-computing the expansion of the elastic elements from the measured forces was too error prone, as a detailed non-linear model was not available. Therefore, the expansion could be neglected. In this case the muscle lengths l can be back-computed from the motor positions φ, utilizing the spindle radius r.

$$l = l_0 + r \cdot \varphi \tag{7}$$

Here, the function could be approximated using a polynomial fit (see Fig. 2). Good fits were achieved with a 2nd order polynomial for the Triceps ($R^2 = 0.99999$) and a 3rd order polynomial for the Brachialis ($R^2 = 0.99991$).

For the geometric solution, the CAD design of the robot, was utilized to determine the two anchor points P_A, P_B of the muscles in the coordinate frame of the upper arm. While P_A is fixed within this frame, P_B is a function of the elbow angle θ

$$P_B = R_Z(\theta) \cdot P_{B0} \tag{8}$$

with R_Z being the rotation matrix around the rotation axis (z) of the elbow and P_{B0} being the anchor point $P_B(0)$ of the unrotated elbow. Assuming straight line muscles, a function of the muscle length could be easily extracted.

$$l(q) = ||P_A - R_Z(q) \cdot P_{B0}|| \tag{9}$$

In Fig. 2, a comparison between an analytically obtained muscle lengths to joint angle relationship and the fitted functions is depicted. This led to a good representation for the Brachialis, even though the anchor points cannot be uniquely determined due to the movement of the kite line inside the guiding eyelets. For this robot the anchor point is able to move inside the plane of the eyelet which has a radius of $3.5\,\mathrm{mm}$. For the Triceps, however, the deviations became significant, as the Triceps cannot be modeled as a straight line muscle. It collides both with the lower as well as the upper arm, depending on the posture. Please note, that it is of course not theoretically impossible to model the behavior of colliding

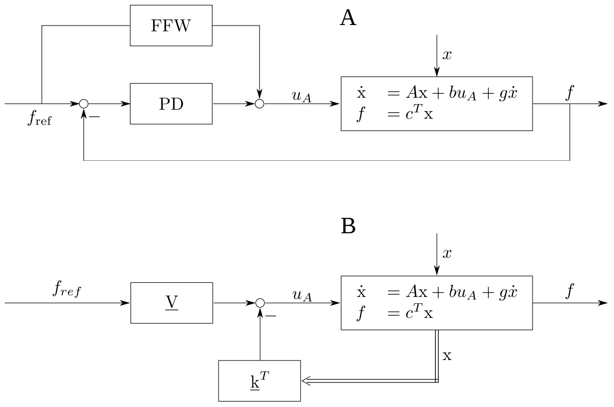
A
FFW
PD
x
u_A
$\dot{\mathrm{x}} = A\mathrm{x} + bu_A + g\dot{x}$
$f = c^T\mathrm{x}$
f
f_{ref}
−
B
f_{ref}
V
u_A
−
x
x
$\underline{\mathrm{k}}^T$

Fig. 4. Muscle Force Controller. A comparison of a **A:** PD force controller with a feed-forward term (FFW) and a **B:** state-space force controller k with a pre-filter V is shown for the linear system $\{A, b, c^T, g\}$ of the muscle model.

muscles analytically, and it has also been done for the simulation model of a human body [20]. However, identifying these parameters for a physical system is error prone, especially when the model will be scaled to larger systems, like the full arm. In this case, an approach like approximating the function through artificial neural networks (ANN) or locally weighted projection regression (LWPR) [21] is needed which has been shown to work well in simulation [12]. In the simple case of a revolute joint, the polynomial fit is fully sufficient. The great advantage of this approach is that the complicated relationships of colliding muscles are captured automatically by the samples from the real robot.

From the polynomial fit, the muscle Jacobian could be easily identified by differentiating. Fig. 3 shows again a comparison between the analytically obtained Jacobian and the polynomial fit. While deviations are comparably small for the Brachialis—with respect to measurement accuracy and the fact that anchor point locations change with the posture—the Triceps deviations are again quite large. It can be seen from the figure that the Triceps could alternatively be modeled by assuming a fixed muscle Jacobian of $\sim 0.02\,\mathrm{m}$, which is essentially the same as the assumption of a fixed moment arm.

IV. Control

A hierarchical control structure (cascade) was developed to control the robot. In a cascade, controllers for the subsystems can be developed independently, provided that dynamics of the inner control loop are at least an order of magnitude faster than the dynamics of the outer control loop [22]. In the following, a controller for the faster inner system—the muscle force control—is synthesized first, and subsequently a controller for the full robot body is developed.

The control approach developed in the following section is distributable in a manner, where fast force control loops can run with a frequency of $500\,\mathrm{Hz}$ on distributed nodes, and the whole body control algorithm runs with a much slower frequency on the central computer. The implementation of this control architecture has been described in detail in [13].

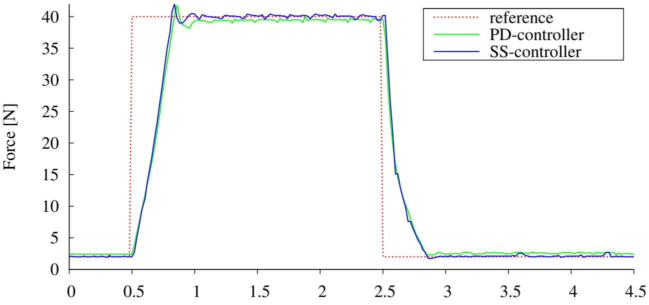

Fig. 5. Force Control Step. Step up and step down from $2\,\mathrm{N}$ to $40\,\mathrm{N}$ for the state-space controller as well as the PD controller.

A. Force Control

Force control for tendon driven systems is generally realized by a P or PD controller, often in conjunction with a feed-forward term (see Fig. 4) [4, 5]. Control parameters can be found by methods from control theory like direct design by root locus, utilizing the Nyquist stability criterion [22] or even by simple trial and error. The linear feed-forward term FFW is determined by calculating the steady state control input

$$\mathrm{FFW} = R_A \cdot \frac{1}{c_\tau} \cdot \frac{r}{g_r \eta_g} \tag{10}$$

where R_A is the anchor resistance and c_τ is the torque constant of the motor, g_r and η_g are the gearbox ratio and energy efficiency, respectively, and r is the radius of the spindle.

We compared the existing approach against a state-space controller (see Fig. 4) that was synthesized using Ackermann's formula [22] for a discrete state-space model. A discrete state-space model could be obtained from the continuous state-space model of a muscle described in Section III-B. As a counter piece to the feed-forward term, a pre-filter V was introduced for compensating the steady state error, by utilizing the state-space system $\{A, b, c^T, g\}$ [22].

$$\begin{bmatrix} M_x \\ M_u \end{bmatrix} = \begin{bmatrix} A & b \\ c^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{11}$$

$$V = M_u + k \cdot M_x \tag{12}$$

The two controllers were compared on a typical muscle of the robotic arm by performing a step up and step down from $2\,\mathrm{N}$ to $40\,\mathrm{N}$ and back (see Fig. 5). The state-space controller followed the force with a RMSD of $0.236\,\mathrm{N}$ and the PD controller with $0.579\,\mathrm{N}$ for the steady state phase at $40\,\mathrm{N}$. It can be seen from the graph in Fig. 5 that this larger error was mainly due to the steady state offset which is $-0.55\,\mathrm{N}$ for the PD and $0.06\,\mathrm{N}$ for state-space controller A similar offset, but in the other direction ($0.44\,\mathrm{N}$ and $0.05\,\mathrm{N}$, respectively) can be observed also at $2\,\mathrm{N}$ leading to the conclusion that this effect cannot be compensated by a linear feed-forward term. Furthermore, the state-space controller also shows a slightly faster rise time. For these reasons, the state-space controller was utilized to control the elbow joint.

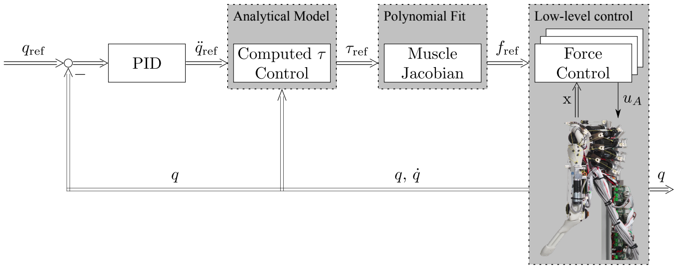

Fig. 6. Joint Space Control. The control scheme uses an analytical model of the skeleton, along with the muscle Jacobian to calculate reference forces for the individual muscle force controllers (see Fig. 4).

B. Joint control

For standard robotic systems there are various control methods based on the canonical form of the skeleton model. Here, the method of computed torque control was adapted [15]. It utilizes (2) to calculate the joint torques τ necessary to produce desired joint accelerations $\ddot{q}_{\text{ref}}$, given the system states q and $\dot{q}$. The reference joint acceleration can be obtained by any control law. In this case a PD controller was chosen.

$$\ddot{q}_{\text{ref}} = P \cdot \Delta q + D \cdot \Delta \dot{q} \tag{13}$$

Even for the simple case of only a single DoF and two antagonistic muscles, solving (6) for the muscle forces f is underdetermined. This problem can be treated by formulating a quadratic optimization problem [23]. In this work, an objective function for the optimization was chosen that is the square of the euclidean distance between the forces. Different objective functions like minimum energy or minimum muscle activation are possible here with different results, especially for multi joint systems. However, in this work only the minimum force criterion was evaluated. The optimization is subject to two constraints. First, the forces are to apply the joint torque acquired by computed torque control, and second, muscles can apply force only in one direction, namely muscle forces have a lower bound. This non-linearity in the system is problematic, especially since slack muscles have to be avoided, as they might get tangled. Therefore the lower bound for the forces f_{min} should be chosen to maintain a certain safety zone from zero, while at the same time reducing the internal forces.

$$\min_{f} ||f||^2 \tag{14}$$

$$\text{subject to} \begin{cases} -L^T(q) \cdot f = \tau \\ f > f_{\text{min}} \end{cases}$$

Therefore, the control input f satisfies the following equation

$$H(P \cdot \Delta q + D \cdot \Delta \dot{q}) + C\dot{q} + \tau_G + \tau_F = -L^T(q) \cdot f \tag{15}$$

which, assuming perfect system models, leads to a closed-loop behavior for the ith joint angle θ_i, due to diagonal $P = \text{diag}\{p_i\}$, and $D = \text{diag}\{d_i\}$.

$$p_i \cdot \Delta\theta_i + d_i \cdot \Delta\dot{\theta}_i = \ddot{\theta}_i. \tag{16}$$

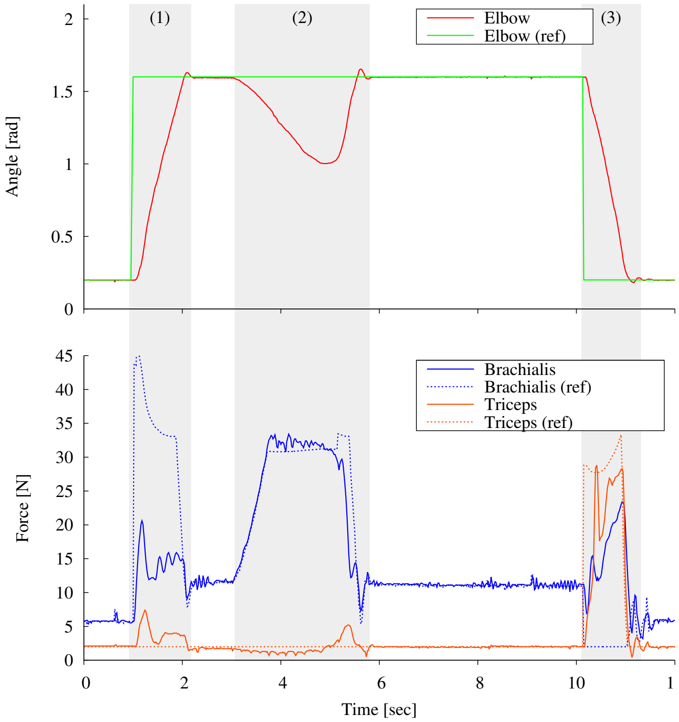

Fig. 7. Elbow Step Response. The elbow angle (top) and the muscle forces (bottom), while (1) performing a step up (2) exerting an external disturbance and (3) performing a step down.

The transfer function of the closed loop $T_f(s)$ can be expressed in the frequency domain for the complex frequency s.

$$T_f(s) = \frac{p_i + d_i \cdot s}{p_i + d_i \cdot s + s^2} \tag{17}$$

Stability for this type of system can be proven using methods from linear system theory [22]. Therefore the controller shows global asymptotic stability, when assuming (1) perfect system models, and (2) quasicontinuous control. In the following section it is shown that the controller behaved in a stable manner, for the derived system models and the discrete distributed implementation.

V. Results

The scalable control scheme shown in Section IV, was implemented for a single revolute joint with two antagonistic muscles. It has to be noted that even though this work only treats a single DoF, it has already been shown to scale to larger systems in simulation [12].

The implementation utilizes the *Robotics Library* (RL)[1] for modeling the dynamics and kinematics of the system and the polynomial fit of the muscle Jacobian (see Section III-C). The optimization problem of calculating muscle forces for given reference accelerations $\ddot{q}_{\text{ref}}$ is solved online, using *QuadProg++*[2] for a minimum force of $f_{\text{min}} = 2\,\text{N}$.

[1] available on sourceforge.net, by Markus Rickert

[2] available on sourceforge.net, by Luca Di Gaspero

In Fig. 7, a step from $0.2\,\mathrm{rad}$ up to $1.6\,\mathrm{rad}$ and a corresponding step down of the elbow angle are shown. We saw accurate tracking of the angle and a rise time of $t_r = 1.00\,\mathrm{s}$ for the step up and $t_r = 0.86\,\mathrm{s}$ for the step down. Between the steps an external disturbance was applied. Here, the muscles flexed to keep the pose steady, however the system was compliant enough to give in to approximately $1\,\mathrm{rad}$, due to the disturbance. As soon as the disturbance was removed, the controller brought the elbow angle back to $1.6\,\mathrm{rad}$.

During the step down it could be seen that the dynamic system model was able to compensate for the non-linear effect of gravity, leading to a rise time that was similar to the rise time of the step up. While stepping down, the role of the muscles was inverted by the optimization, causing the Triceps to pull the forearm down. It is obvious that the force controller was not always able to track the reference forces fast enough to follow. This was due to the constant disturbance of the moving forearm. However, steady state behavior was highly accurate and the overall movements are carried out with sufficient speed. Most importantly, neither of the muscles went slack at any time during the experiment.

VI. Conclusions and Future Works

The scalable joint-space controller for musculoskeletal robots developed in [12] was evaluated on a physical robotic joint, proving feasibility of the overall control strategy. It has been shown that numerical methods can be used to model the muscle kinematics in contrast to an analytical solution assuming straight line muscles. The great advantage of the numerical approximation approach for the muscle kinematics is that it is capable of automatically capturing the main challenges of controlling anthropomimetic robots, i.e. complex joint types, colliding muscles, and multiarticular muscles. Furthermore, the advantages of a state-space force control, in comparison to the previously used PD controller with a feed-forward term were demonstrated.

In the future we would like to extend the controller to multi-joint structures including spherical joints and bi-articular muscles. It has to be noted that intrinsic joint angle sensors for spherical joints are extremely hard to realize and therefore need to be replaced, either by back-computing joint angles from muscle lengths [24] or by an extrinsic sensory system, e.g. stereo-vision based motion tracking. Furthermore, the control scheme will be extended to operational space control. Especially solving the joint, as well as the muscle redundancy in a single local optimization step has been shown to lead to physiologically consistent trajectories [2, 11].

VII. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement no. 231864 - ECCEROBOT

References

[1] A. Albu-Schäffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimbock, S. Wolf, and G. Hirzinger, "Soft robotics," *IEEE Robotics & Automation Magazine*, vol. 15, no. 3, pp. 20–30, 2008.

[2] K. Tahara, S. Arimoto, M. Sekimoto, and Z. W. Luo, "On control of reaching movements for musculo-skeletal redundant arm model," *Applied Bionics and Biomechanics*, vol. 6, no. 1, pp. 11–26, 2009.

[3] O. Holland and R. Knight, "The Anthropomimetic Principle," in *Adaptation in Artificial and Biological Systems*, 2006.

[4] S. C. Jacobsen, H. Ko, E. K. Iversen, and C. C. Davis, "Antagonistic control of a tendon driven manipulator," in *Proc. Conf. IEEE Int Robotics and Automation*, 1989, pp. 1334–1339.

[5] J. K. Salisbury and J. J. Craig, "Articulated Hands - Force Control And Kinematic Issues." *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, Mar. 1982.

[6] S. Hirose and S. Ma, "Coupled Tendon-driven Multijoint Manipulator," in *Proc. IEEE International Conference on Robotics and Automation ICRA 1991*, no. April, 1991, pp. 1268–1275.

[7] V. Potkonjak, B. Svetozarevic, K. Jovanovic, and O. Holland, "Biologically-inspired control of a compliant anthropomimetic robot," in *Proceedings of the 15th IASTED International Conference on Robotics and Applications*, 2010.

[8] D. G. Thelen, F. C. Anderson, and S. L. Delp, "Generating dynamic simulations of movement using computed muscle control," *Journal of Biomechanics*, vol. 36, no. 3, pp. 321–328, 2003.

[9] K. Yamane and Y. Nakamura, "Robot Kinematics and Dynamics for Modeling the Human Body," in *Springer Tracks in Advanced Robotics*, M. Kaneko and Y. Nakamura, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 66, pp. 49–60.

[10] V. De Sapio, J. Warren, O. Khatib, and S. Delp, "Simulating the task-level control of human motion: a methodology and framework for implementation," *The Visual Computer*, vol. 21, no. 5, pp. 289–302, 2005.

[11] V. De Sapio, K. Holzbaur, and O. Khatib, "The control of kinematically constrained shoulder complexes: physiological and humanoid examples," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2006*, 2006, pp. 2952–2959.

[12] M. Jäntsch, C. Schmaler, S. Wittmeier, K. Dalamagkidis, and A. Knoll, "A scalable Joint-Space Controller for Musculoskeletal Robots with Spherical Joints," in *Proc. IEEE International Conference on Robotics and Biomimetics ROBIO 2011*, 2011, pp. 2211–2216.

[13] M. Jäntsch, S. Wittmeier, and A. Knoll, "Distributed control for an anthropomimetic robot," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, 2010, pp. 5466–5471.

[14] H. G. Marques, M. Jäntsch, S. Wittmeier, O. Holland, C. Alessandro, A. Diamond, M. Lungarella, and R. Knight, "ECCE1: The first of a series of anthropomimetic musculoskeletal upper torsos," in *Proc. 10th IEEE-RAS Int Humanoid Robots (Humanoids) Conf*, 2010, pp. 391–396.

[15] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[16] R. Featherstone, "A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 1: Basic Algorithm," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 867–875, 1999.

[17] J. J. Craig, *Introduction to Robotics*, 3rd ed., M. J. Horton, Ed. Pearson, 2005.

[18] V. M. Zatsiorsky, *Kinetics of Human Motion*. Human Kinetics Publishers, 2002.

[19] K. Koganezawa, "Mechanical stiffness control for antagonistically driven joints," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2005*. IEEE, Aug. 2005, pp. 1544–1551.

[20] K. R. S. Holzbaur, W. M. Murray, and S. L. Delp, "A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control," *Annals of Biomedical Engineering*, vol. 33, no. 6, pp. 829–840, Jun. 2005.

[21] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[22] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, T. Hyde, Ed. Prentice Hall, 1997.

[23] V. De Sapio, J. Warren, and O. Khatib, *Predicting reaching postures using a kinematically constrained shoulder model*. Springer Netherlands, 2006, ch. 3, pp. 209–218.

[24] Y. Nakanishi, K. Hongo, I. Mizuuchi, and M. Inaba, "Joint Proprioception Acquisition Strategy Based on Joints-Muscles Topological Maps for Musculoskeletal Humanoids," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2010*, 2010, pp. 1727–1732.