53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Adaptive Neural Network Dynamic Surface Control for Musculoskeletal Robots

Michael Jäntsch*, Steffen Wittmeier*, Konstantinos Dalamagkidis*, Guido Herrmann** and Alois Knoll*

***Abstract*— Musculoskeletal robots are a class of compliant, tendon-driven robots that can be used in robotics applications, as well as in the study of biological motor systems. Unfortunately, there is little progress in controlling such systems. Modern non-linear control approaches are used to overcome the challenges posed by the muscle compliance, the multi-DoF joints, as well as unmodeled dynamic effects such as friction. A controller is derived for a generic model of musculoskeletal robots utilizing a multidimensional form of Dynamic Surface Control (DSC), an extension to backstepping. This controller is extended by an adaptive neural network to compensate for both muscle and joint friction. The developed controllers are evaluated against the state of the art Computed Force Control (CFC), an application of feedback linearization, for a spherical joint which is actuated by five muscles.**

***Keywords*—Musculoskeletal robots, dynamic surface control, adaptive neural networks, compliant actuation, non-linear control, backstepping**

I. Introduction

Within the last decade, focus in robotics research has shifted to more human-friendly robots. When physical human-robot interaction is considered, collisions may not be fully avoidable. In case of a rigid impact, a robot without passive compliance is very likely to damage itself or its surroundings, as active compliance is limited by sensor bandwidth and control frequency [1]. By employing tendon-driven actuation, actuators can be placed with more freedom, to e. g. improve the weight distribution of the robot. Musculoskeletal robots combine this type of actuation with the advantages of passively compliant robots. Prominent examples for these robots are Kenshiro [2] and the ECCEROBOTS [3].

Even though research of the last years has produced extremely impressive robots, the field of controlling such robots has made very little progress, so far. While demonstrations of these robots show usually only Feedforward (FFW) control, the existing feedback control techniques [4], [5] have failed to scale to more complex structures. This is due to the fact that musculoskeletal robots in general exhibit several characteristics that are usually not present in previous tendon-driven systems. These include complex joint types, like spherical joints, difficult to model muscle kinematics and elastic muscles. Passive compliance in the muscles diminishes the control performance of existing control techniques like Computed Force Control (CFC) [6].

In this paper, control techniques are developed, taking inspiration from Na et. al [7] who developed an adaptive neural dynamic surface controller for a single Degree of Freedom (DoF) servo system. We extend a multidimensional approach to backstepping by Oh and Lee [8] with Dynamic Surface Control (DSC) [9] and adaptive neural networks to significantly improve control performance for musculoskeletal robots.

*These authors are with the chair for Robotics and Embedded Systems, Department of Informatics, Technische Universität München, Munich

**This author is with the University of Bristol, United Kingdom

II. Modeling

A generic model for the class of musculoskeletal robots is developed, starting with a model for the skeletal dynamics, the muscle dynamics, and finally the muscle kinematics. Generally, the number of DoF of a robot denotes the number of joints. However, due to the introduction of passive compliance in the muscles, this definition needs to be extended. We call the number of skeletal DoF n and the number of muscles m. Hence the total number of DoF of such a system is $n+m$ with m actuators, yielding an underactuated system.

A. Skeletal Dynamics

For conventional robots, the inverse dynamics relate the joint torques τ with the joint accelerations $\ddot{q}$ for a state of the robot, given by $(q, \dot{q})$. It can be expressed in the so called canonical form:

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

while $H(q)$, the mass matrix, is an $n \times n$ symmetric, positive definite matrix, expressing the robot inertia, $C(q,\dot{q})$ is an $n \times n$ matrix, which accounts for Coriolis and centrifugal effects, and $\tau_G(q)$ is the vector of gravity terms. Even though muscles exhibit significant compliance, the skeleton of musculoskeletal robots can, in most cases, be assumed as rigid links between the joints. Hence, (1) can be used as a general model of the skeletal dynamics. In the presence of spherical joints, however, difficulties arise in representing the joint positions. The three dimensional rotation of a spherical joint can be expressed in several different ways. Due to its lack of singularities, the unit quaternion description of rotation was chosen [10]. To account for the fact, that the dimensionality of a quaternion is higher than the represented DoF, the positional coordinate q is replaced by α, containing the quaternion representation of spherical joints, as well as the angular representation of the other, e. g. revolute, joints. This leads to the following modified joint space dynamic equation:

$$\tau = H(\alpha)\ddot{q} + C(\alpha,\dot{q}) + \tau_G(\alpha) \tag{2}$$

Therefore, a relationship between the derivative of α and the rotational velocities $\dot{q}$ needs to be obtained. The mapping of the derivative of a quaternion and the corresponding

rotational velocities is well known [11]. We define a matrix $A(\alpha)$, as a diagonal matrix, in which each revolute joint is represented by a 1 and each spherical joint is represented by the corresponding mapping, such that [12]:

$$\dot{\alpha} = A(\alpha)\dot{q} \tag{3}$$

B. Muscle Dynamics

A muscle in a musculoskeletal robot with electromagnetic actuators consists of a brushed permanent magnet Direct Current (DC) motor, gearbox, tendon, and an elastic element. The model considered here is a standard model for this type of actuator, which is simplified by neglecting the current as a system state:

$$\ddot{\theta} = -\frac{c_M^2}{R_A J_T}\dot{\theta} - \frac{r_s}{r_G^2 \eta_G J_T} f + \frac{c_M}{R_A J_T r_G} v_A \tag{4}$$

where v_A, R_A and c_M are the armature voltage, resistance and the motor constant, respectively. The actuator velocity is represented by $\dot{\theta}$ and the load torque by the muscle force f times the radius r_s of the spindle that winds up the tendon. Finally, the gearbox is accounted for by a reduction ratio r_G and a factor of efficiency η_G.

This actuator model is extended by a model of the elastic element with a possibly non-linear stress-strain relationship, which is represented by a spring stiffness $K(f)$:

$$\dot{f} = K(f)(\dot{l} + r_s\dot{\theta}) \tag{5}$$

where the change in spring expansion is expressed as the change in muscle length $\dot{l}$ and the change in tendon length, due to turning the spindle.

C. Muscle Kinematics

The muscle kinematics describe the force transmission of the muscles and is hence a geometric representation of how muscles interact with the skeleton. It can be captured by the so called muscle Jacobian which is defined as the partial derivative of the muscle lengths l with respect to the joint angles q. By the principle of virtual work it can be shown that the muscle Jacobian $L(q)$ can be transferred in the force torque domain [13]:

$$L(q) = \frac{\partial l}{\partial q} \quad \longleftrightarrow \quad \tau = -L^T(q) f \tag{6}$$

where the minus sign arises from the definition of a positive force being associated with muscle shortening. Note that any flexibility in the muscles does not affect this purely geometric relation, as it essentially expresses the (negative) matrix of muscle lever arms. In the presence of spherical joints, the pose of the robot is represented by α. Hence $L_\alpha(\alpha)$ is defined as the partial derivative of the muscle lengths with respect to the pose vector as follows:

$$L_\alpha(\alpha) = \frac{\partial l}{\partial \alpha} \tag{7}$$

$A(\alpha)$ can be introduced from (3), yielding:

$$\frac{\partial l}{\partial t} = L_\alpha(\alpha)\frac{\partial \alpha}{\partial t} = L_\alpha(\alpha)A(\alpha)\frac{\partial q}{\partial t} \tag{8}$$

and the muscle Jacobian $L(\alpha)$ can be rewritten as follows:

$$L(\alpha) = \frac{\partial l}{\partial q} = L_\alpha(\alpha)A(\alpha) \tag{9}$$

A generalized model can be defined by introducing (6) into (2) and (5):

$$H(\alpha)\ddot{q} + C(\alpha, \dot{q})\dot{q} + \tau_G(\alpha) = -L(\alpha)^T x_1 \tag{10}$$

$$\dot{x}_1 = K(x_1)\left[L(\alpha)\dot{q} + r_s x_2\right] \tag{11}$$

$$\dot{x}_2 = -\frac{c_M^2}{R_A J_T}x_2 - \frac{r_s}{r_G^2 \eta_G J_T}x_1 + \frac{c_M}{R_A J_T r_G}v_A \tag{12}$$

The system states of (11) and (12) have been replaced by x_1 and x_2 which denote the muscle force f and the actuator velocity $\dot{\theta}$, respectively. Note that the system state of the first equation $x_0 = [\alpha,\ \dot{q}]^T$ is made up of both the pose and the joint velocity. It is obvious that (10)–(12) presents a so called *strict-feedback* system, where the first equation relates the system state x_0 to its derivative and takes the system state of the next equation x_1 as an input. Similarly, the i^{th} system equation ($i \in \{1, \ldots, k\}$, where k is the depth of the system) is a function of x_i and the previous system states $x_0, \ldots, x_{i-1}$ and takes x_{i+1} as an input. A general definition of *strict-feedback* can be found in [14].

III. Computed Force Control

Most tendon-driven controllers utilize feedback linearization, by linearizing (10) and introducing some linear control law. Hence a tendon force needed for a certain movement is computed and is therefore often called Computed Force Control (CFC) or Computed Muscle Control. A trajectory tracking form of this type can be given as follows [6]:

$$-L^T f_d = H\left[\ddot{q}_d + D(\dot{q}_d - \dot{q}) + P\Delta q\right] + C\dot{q} + \tau_G \tag{13}$$

where $\Delta q = q_d - q$ denotes the error in the angle, which is later redefined to account for the utilization of quaternions, and P and D are positive control gains. The redundancy in the muscle space can be resolved in several different ways, where the most common one is to solve a quadratic program of the following form to obtain the muscle forces [6]:

$$\min_{f_d} ||f_d||^2 \qquad \text{subject to} \begin{cases} -L^T(\alpha) f_d = \tau_d \\ f_d \geq f_{\min} \end{cases} \tag{14}$$

There is a drawback in this approach which is that these controllers rely on perfect low-level control. When muscles with passive compliance are considered this is impossible to achieve, as the elastic element leads to slowed system dynamics, i. e. the actuator has to expand the elastic element until the reference force is reached. In larger assemblies this can lead to oscillations or even instability.

IV. Dynamic Surface Control

Apart from feedback linearization, there are several techniques that have been used before in robotics to derive non-linear controllers. For flexible-joint robots, the technique of passivity based control has been utilized extensively [15]. However, this method does not provide a systematic synthetization approach. If the system model can be reformulated to

form a *strict-feedback* system, as in (10)–(12), the method of *backstepping* can be applied [14]. In this method, a controller ϕ_1 for the first system, i. e. equation (10), is found assuming the next state x_1 as virtual control signal. In the case of equation (10) we will adopt a nonlinear controller developed by Oh and Lee [8] for trajectory tracking of flexible joint robots in extension to our musculoskeletal problem:

$$\phi_1 = -L^{T+} \{H [\ddot{q}_d + \Lambda_1(\dot{q}_d - \dot{q})] + C [\dot{q}_d + \Lambda_1 \Delta q] + \tau_G - K_d r\} \quad (15)$$

while L^{T+} denotes the pseudo inverse of the muscle Jacobian, which is associated with the result of the quadratic program in (14). Λ_1 is a diagonal control gain matrix with positive elements. The generalized tracking error r is defined as follows:

$$r = \dot{q} - \dot{q}_d - \Lambda_1 \Delta q \quad (16)$$

In the next step, the dynamics of x_1, i. e. equation (11), need to be controlled so that x_1 converges to ϕ_1 by designing a suitable controller ϕ_2 for the virtual control signal of x_2 in the dynamics of (11). Within the backstepping technique, this requires the calculation of the time derivative of ϕ_1 and the consideration of interaction terms in the combined dynamics of x_0 and x_1 through a stepped Lyapunov analysis. In the next step, the dynamics of x_2 are controlled using equation (12). This again requires time derivatives in particular for ϕ_2. This backstepping from x_0 to x_2 accounts for the interaction dynamics of the controllers but creates an explosion of complexity due to the applied time derivatives. For this purpose Swaroop et. al. [9] developed Dynamic Surface Control (DSC) which utilizes a set of first order low-pass filters in between each level of the control laws to realize stable numeric differentiation. Each of these filters is applied to the signal of the (virtual) controller output ϕ_i and is defined such that:

$$\mu_i \dot{s}_i + s_i = \phi_i \qquad i = \{1, \ldots, k\} \quad (17)$$

where μ_i is the filter constant and s_i the filter output. Thus, the closed loop dynamics for state x_0 also considering the change of coordinate $z_1 = x_1 - s_1$, evaluate to:

$$H\dot{r} + Cr + K_d r = -L^T [z_1 - \mu_1 \dot{s}_1]. \quad (18)$$

This now requires to control the virtual error z_1. Within the next step, the dynamics of the virtual control signal x_1 can be analyzed by rewriting (11):

$$\dot{x}_1 = f_1(\alpha, \dot{q}, x_1) + g_1(x_1) x_2 \quad (19)$$

and introducing it into the derivative of z_1:

$$\dot{z}_1 = \dot{x}_1 - \dot{s}_1 = f_1 + g_1 x_2 - \dot{s}_1 \quad (20)$$

Thus, the virtual control for the state x_2 is:

$$\phi_2 = g_1^{-1} [\dot{s}_1 + Lr - \Lambda_2 z_1 - f_1], \quad (21)$$

where Λ_2 is a diagonal control gain matrix with positive elements. The term Lr will be clarified later. The dynamics of state x_2 are rewritten from (12):

$$\dot{x}_2 = f_2(x_1, x_2) + g_2 v_A \quad (22)$$

and introduced into the dynamics for the second change of coordinate $z_2 = x_2 - s_2$:

$$\dot{z}_2 = \dot{x}_2 - \dot{s}_2 = f_2 + g_2 v_A - \dot{s}_2 \quad (23)$$

For these error dynamics a controller is found:

$$\phi_3 = g_2^{-1} [\dot{s}_2 - g_1 z_1 - \Lambda_3 z_2 - f_2], \quad (24)$$

where Λ_3 is again a diagonal control gain matrix with positive elements and the choice of the term $g_1 z_1$ is explained later. This control law is subsequently to be applied, i. e. $v_A = \phi_3$ (see Fig. 1).

It is evident that all three control laws, the two virtual control laws (15),(21) and the actual control law (24) require the use of the derivatives $\dot{s}_i$. They can now be numerically determined by the following term:

$$\dot{s}_i = \frac{\phi_i - s_i}{\mu_i} = -\frac{e_i}{\mu_i} \quad (25)$$

The method causes an error due to the filter $e_i = s_i - \phi_i$ which is differentiated to obtain the following error dynamics:

$$\dot{e}_i = \dot{s}_i - \dot{\phi}_i = -\frac{e_i}{\mu_i} + \xi_i(\ldots) \quad (26)$$

where ξ_i has been shown to be a bounded continuous function [9], [16], [17] of the system states, their derivatives, and the controller in some bounded set. Without loss of generality, it can be assumed that the controller is only operated within certain bounds and we can define an upper bound to $\dot{e}_i$.

For revolute joints the error in the angle Δq is simply equal to the angular difference. In the case of spherical joints, it needs to be redefined as follows for a single spherical joint:

$$\Delta q = \eta \vec{\epsilon}_d - \eta_d \vec{\epsilon} - S(\vec{\epsilon}_d)\vec{\epsilon} \quad (27)$$

which is zero if and only if the delta rotation is zero, i. e. α matches the demand α_d. The measured and desired rotations are expressed as quaternions, where η is the real and $\vec{\epsilon}$ the three dimensional imaginary part. $S(\cdot)$ is the skew symmetric operator. This definition arises from the stability analysis of quaternion control. In particular, $r = 0$ implies exponential convergence, while $\|r\| < \epsilon_r$ for small positive scalar ϵ_r implies bounded tracking of α with respect to α_d [18].

Considering these prerequisites, the following result shows ultimate bounded stability for the DSC:

Lemma 1: Consider the musculoskeletal robot from (10)–(12) and the DSC from (15),(21) and (24). If the closed loop gains satisfy

$$I_m \mu_1 \leq k_e (c_1 L L^T + I_m)^{-1} \quad (28)$$

$$I_m \mu_2 \leq k_e (c_2 g_1^T g_1 + I_m)^{-1} \quad (29)$$

$$K_d \geq I_n \frac{1}{4c_1} \quad (30)$$

$$\Lambda_2 \geq I_m \frac{1}{4c_2} \quad (31)$$

$$\Lambda_3 \geq 0 \quad (32)$$

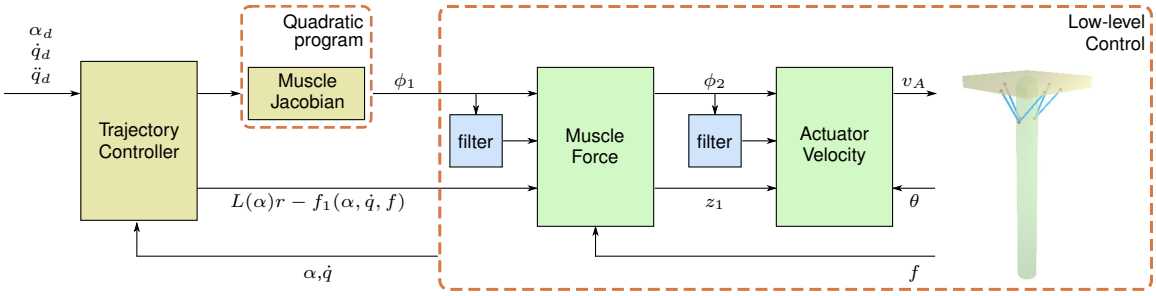

Fig. 1: Block Diagram of Dynamic Surface Control. The controller consists of three levels, starting with the controller for the skeletal dynamics, comprising the trajectory controller and the muscle Jacobian. The other two levels are the muscle force and the actuator velocity controllers. First order filters are added to obtain numeric derivatives of of ϕ_i.

for positive design parameters k_e, c_1 and c_2, then the dynamic surface tracking control achieves ultimate bounded stability in (r, z_1, e_1, z_2, e_2).

Proof: The closed loop dynamics of the first system, evaluate to (18) for which a Lyapunov function candidate is chosen [7]:

$$V_1 = \frac{1}{2} r^T H r + \frac{k_e}{2} e_1^T e_1 \tag{33}$$

where k_e is a positive design parameter. Due to the property of the inertia Matrix being positive definite, it can be seen that $V_1(r, e_1) > 0$ for any $r, e_1 \neq 0$. It can be shown that $[\dot{H} - 2C]$ is skew symmetric [10]. Therefore the Lyapunov function derivative evaluates to the following, after introducing the closed loop dynamics from (18):

$$\begin{aligned}\dot{V}_1 =& \frac{1}{2} r^T \dot{H} r + r^T H \dot{r} + k_e e_1^T \dot{e}_1 \\ =& - r^T K_d r - r^T L^T z_1 + r^T L^T \mu_1 \dot{s}_1 + k_e e_1^T \dot{e}_1 \end{aligned} \tag{34}$$

The dynamics of z_1 are stated in (20). The control law ϕ_2 for the muscle dynamics from (21) his used to obtained the closed loop dynamics by means of $z_2 = x_2 - s_2$:

$$\begin{aligned}\dot{z}_1 &= f_1 + g_1[z_2 + \phi_2 - \mu_2 \dot{s}_2] - \dot{s}_1 \\ &= g_1 z_2 - \Lambda_2 z_1 + L r - g_1 \mu_2 \dot{s}_2 \end{aligned} \tag{35}$$

for which we provide a Lyapunov function candidate:

$$V_2 = V_1 + \frac{1}{2} z_1^T z_1 + \frac{k_e}{2} e_2^T e_2 \tag{36}$$

After differentiation, we can introduce the closed loop dynamics from (35):

$$\begin{aligned}\dot{V}_2 =& \dot{V}_1 + z_1^T \dot{z}_1 + k_e e_2^T \dot{e}_2 \\ =& - r^T K_d r - z_1^T \Lambda_2 z_1 + r^T L^T \mu_1 \dot{s}_1 - z_1^T g_1 \mu_2 \dot{s}_2 \\ &+ k_e e_1^T \dot{e}_1 + k_e e_2^T \dot{e}_2 \end{aligned} \tag{37}$$

Similar to the previous step, the dynamics of z_2 are in (23) for which a controller is found in (24), so that for $v_A = \phi_3$:

$$\dot{z}_2 = -\Lambda_3 z_2 - g_1 z_1 \tag{38}$$

Again, a Lyapunov function candidate is provided:

$$V_3 = V_2 + \frac{1}{2} z_2^T z_2 \tag{39}$$

which is differentiated and merged with (38):

$$\begin{aligned}\dot{V}_3 =& \dot{V}_2 + z_2^T \dot{z}_2 \\ =& - r^T K_d r - z_1^T \Lambda_2 z_1 - z_2 \Lambda_3 z_2 + r^T L^T \mu_1 \dot{s}_1 \\ &- z_1^T g_1 \mu_2 \dot{s}_2 + k_e e_1^T \dot{e}_1 + k_e e_2^T \dot{e}_2 \end{aligned} \tag{40}$$

By introducing the dynamics from (25) and (26) for $\dot{s}_i$ and $\dot{e}_i$ respectively, V_3 can be extended to finally apply Young's inequality [19] which yields an upper bound for $\dot{V}_3$:

$$\begin{aligned}\dot{V}_3 =& - r^T K_d r - z_1^T \Lambda_2 z_1 - z_2^T \Lambda_3 z_2 \\ &- r^T L^T e_1 + z_1^T g_1 e_2 \\ &- \frac{k_e}{\mu_1} e_1^T e_1 - \frac{k_e}{\mu_2} e_2^T e_2 + k_e e_1^T \xi_1(\ldots) + k_e e_2^T \xi_2(\ldots) \\ \leq& - r^T K_d r - z_1^T \Lambda_2 z_1 - z_2^T \Lambda_3 z_2 \\ &+ \frac{r^T r}{4c_1} + c_1 e_1^T L L^T e_1 + \frac{z_1^T z_1}{4c_2} + c_2 e_2^T g_1^T g_1 e_2 \\ &- \frac{k_e}{\mu_1} e_1^T e_1 - \frac{k_e}{\mu_2} e_2^T e_2 + \frac{e_1^T e_1}{4c_3} + \frac{e_2^T e_2}{4c_4} \\ &+ c_3 k_e^2 \xi_1^T(\ldots)\xi_1(\ldots) + c_4 k_e^2 \xi_2^T(\ldots)\xi_2(\ldots) \end{aligned} \tag{41}$$

From the definition of $\xi_i(\ldots)$ in (26) it can be seen that $|\xi_i(\ldots)|$ has an upper bound that is defined as M_i [7]. For simplicity, $c_{3/4}$ are chosen to be $\frac{1}{4}$. Therefore $\dot{V}_3$ implies:

$$\begin{aligned}\dot{V}_3 \leq& - r^T \left[K_d - I_n \frac{1}{4c_1} \right] r - z_1^T \left[\Lambda_2 - I_m \frac{1}{4c_2} \right] z_1 \\ &- z_2^T \Lambda_3 z_2 - e_1^T \left[I_m \frac{k_e}{\mu_1} - c_1 L L^T - I_m \right] e_1 \\ &- e_2^T \left[I_m \frac{k_e}{\mu_2} - c_2 g_1^T g_1 - I_m \right] e_2 \\ &+ \frac{1}{4} k_e^2 M_1^T M_1 + \frac{1}{4} k_e^2 M_2^T M_2 \end{aligned} \tag{42}$$

where $c_{1/2}$ and k_e can be chosen arbitrarily in $\mathbb{R}_{>0}$. Considering now the conditions of (28)–(32), stability of the overall closed loop system is proven (see also [7] and [9]). ■

For very small μ_i the design parameter k_e can be chosen to be as small as necessary, without violating the stability condition of (28)–(32), hence making the last two terms of (42) arbitrarily small. However, this impairs at the same time the numeric stability, as $\dot{s}_i$ becomes sensitive to changes (see (25)). In practice, a trade-off between numeric stability and a

low tracking error has to be found. Hence, three novel control laws have been developed for the subsystems (10)–(12): (i) a trajectory tracking controller, (ii) a novel force controller with compensation for muscle length changes and (iii) an actuator velocity controller (see Fig. 1).

V. Adaptive Control

To prove stability of the previously developed controllers, perfectly known system dynamics were assumed. However, this assumption generally does not hold. One way to overcome this problem, is to introduce adaptive terms into the controller. This technique is applied in the following to compensate for friction. Unlike conventional joint actuated robots, a musculoskeletal robot does not only exhibit joint friction (JF), but also friction in the tendon transmission system, i. e. muscle friction (MF). Adaptive control for friction compensation has been used widely in the field of flexible joint robot control, utilizing a parametric friction model, e. g. [20]. Radial Basis Function Networks (RBFNs), a form of Artificial Neural Networks (ANNs), have also been popular [21].

To find a controller that compensates both the joint, as well as the muscle friction, (10) is rewritten as follows:

$$\begin{aligned} H(\alpha)\ddot{q} + C(\alpha,\dot{q})\dot{q} + \tau_G(\alpha) + \tau_F(\dot{q}) \\ = -L^T(\alpha)\left[x_1 - f_F(x_1)\right] \end{aligned} \tag{43}$$

At this point, two simplifications are made to improve convergence of the adaptive terms: (i) the joint friction is assumed to be only a function of the joint velocity $\dot{q}$ [21] and (ii) the muscle friction is only governed by Coulomb friction, i. e. only a function of the transmitted force x_1. Without loss of generality, let $\tau_F(\dot{q})$ and $f_F(x_1)$ be bounded functions:

$$|\tau_F(\dot{q})| < \underline{\tau}_F \qquad \forall \dot{q} \in \{-\dot{q}_{\min}, \dot{q}_{\max}\} \tag{44}$$

$$|f_F(x_1)| < \underline{f}_F \qquad \forall x_1 \in \{0, f_{\max}\} \tag{45}$$

Similar to the gravity compensation term τ_G, the two friction terms can be integrated into the controller,

$$\begin{aligned} \phi_1 = -L^{T+}\{H(\ddot{q}_d + \Lambda_1(\dot{q}_d - \dot{q})) + C(\dot{q}_d + \Lambda_1\Delta q) \\ + \tau_G - K_d r + \hat{\tau}_F\} + \hat{f}_F \end{aligned} \tag{46}$$

where $\hat{\tau}_F$ and $\hat{f}_F$ denote estimations of the unknown functions. Friction terms can be well approximated by RBFNs which are defined as a vector of RBFs Φ and a vector of weights Θ such that $\phi = \Theta^T\Phi$ [21]. Due to the fact that $\tau_F \in \mathbb{R}^n$ and $f_F \in \mathbb{R}^m$, the two compensators are defined as vectors of RBFNs, where $\phi_{\mathrm{JF}j}$ and $\phi_{\mathrm{MF}j}$ denote the j^{th} entry of each of these vectors, respectively. A general definition for each RBF can be given as follows,

$$\Phi_{ijk}(x_{ij}) = \exp - \frac{(x_{ij} - c_{ik})^T(x_{ij} - c_{ik})}{\sigma_{ik}^2} \tag{47}$$

where $i \in \{\mathrm{JF}, \mathrm{MF}\}$ and $j \in \{1, \ldots, o\}$ with o as the number of RBFNs per friction compensator which is n for the joint friction and m for the muscle friction. Finally $k \in \{1, \ldots, l\}$, where l denotes the number of neurons. The $l \times 1$ vector of neurons Φ_{ij} is pre-multiplied with the $1 \times l$ vector of weights $\hat{\Theta}_{ij}^T$.

$$\hat{\tau}_{Fj} = \hat{\Theta}_{JFj}^T\Phi_{JFj} \qquad \hat{f}_{Fj} = \hat{\Theta}_{MFj}^T\Phi_{MFj} \tag{48}$$

For the proposed control law, the closed loop dynamics are

$$H\dot{r}_1 + Cr + K_d r + \tilde{\tau}_F - L^T\tilde{f}_F = -L^T\left[z_1 - \mu_1\dot{s}_1\right] \tag{49}$$

where $\tilde{\tau}_F = \tau_F - \hat{\tau}_F$ and $\tilde{f}_F = f_F - \hat{f}_F$ represent the respective estimation errors. An adaptation rule for the weights is found, utilizing the method of gradient descent [21],

$$\dot{\hat{\Theta}}_{ij} = -\Gamma_{ij}\Phi_{ij}r_{ij} - \Sigma_i\hat{\Theta}_{ij} \tag{50}$$

where Γ_{ij} is the respective learning factor. Note that the adaptation rule includes the so called σ-Modification [22]. The driving value for the adaptation rule is the error r_i which is defined as follows.

$$r_{JF} = r \qquad r_{MF} = -Lr \tag{51}$$

It has been shown that an ANN of the given form can approximate any continuous function up to a bounded error $\epsilon_{JF} \in \mathbb{R}^n$ and $\epsilon_{MF} \in \mathbb{R}^m$, respectively [23]. Therefore, each friction term can be written as

$$\mathcal{F}_{ij} = \Theta_{ij}^T\Phi_{ij} + \epsilon_{ij} \tag{52}$$

where $\mathcal{F}_i$ denotes the joint friction and the muscle friction, respectively and $|\epsilon_i| \leq \underline{\epsilon}_i$. Therefore $\tilde{\Theta}_{ij} = \Theta_{ij} - \hat{\Theta}_{ij}$ can be defined to denote the error in the weights, leading to the following.

$$\tilde{\mathcal{F}}_{ij} = \Theta_{ij}^T\Phi_{ij} + \epsilon_{ij} - \hat{\Theta}_{ij}^T\Phi_{ij} = \tilde{\Theta}_{ij}^T\Phi_{ij} + \epsilon_{ij} \tag{53}$$

Theorem 1: Consider the stability conditions of (28)–(32) for controlling the musculoskeletal robot from (10)–(12), the modified first virtual control from (46), and the other dynamic surface controllers from (21) and (24). The resulting closed loop is Uniformly Ultimately Bounded (UUB).

Proof: A Lyapunov function is found by adding a term for each of the errors in the weights to (39).

$$\begin{aligned} V = V_3 + \frac{1}{2}\sum_{j=1}^{n}\tilde{\Theta}_{JFj}^T\Gamma_{JFj}^{-1}\tilde{\Theta}_{JFj} \\ + \frac{1}{2}\sum_{j=1}^{m}\tilde{\Theta}_{MFj}^T\Gamma_{MFj}^{-1}\tilde{\Theta}_{MFj} \end{aligned} \tag{54}$$

The derivative of the Lyapunov function is stated as follows:

$$\begin{aligned} \dot{V} = &\dot{V}_3 - r^T\tilde{\tau}_F + r^T L^T\tilde{f}_F \\ &+ \sum_{j=1}^{n}\tilde{\Theta}_{JFj}^T\Gamma_{JFj}^{-1}\dot{\tilde{\Theta}}_{JFj} + \sum_{j=1}^{m}\tilde{\Theta}_{MFj}^T\Gamma_{MFj}^{-1}\dot{\tilde{\Theta}}_{MFj} \end{aligned} \tag{55}$$

Due to the fact that Θ_{ij} is constant, it can be seen from the definition of the error in the weights that $\dot{\tilde{\Theta}}_{ij} = -\dot{\hat{\Theta}}_{ij}$

allowing for the introduction of the adaptation law (50). Again, Young's inequality implies for $\dot{V}$,

$$\begin{aligned}\dot{V} \leq &+ \dot{V}_3 + \underline{\epsilon}_{JF}^T |r| + \underline{\epsilon}_{MF}^T |Lr| \\ &- \sum_{j=1}^{n} \tilde{\Theta}_{JFj}^T \left[\Gamma_{JFj}^{-1} \Sigma_{JF} - \frac{1}{4c_5} I_l \right] \tilde{\Theta}_{JFj} \\ &- \sum_{j=1}^{m} \tilde{\Theta}_{MFj}^T \left[\Gamma_{MFj}^{-1} \Sigma_{MF} - \frac{1}{4c_6} I_l \right] \tilde{\Theta}_{MFj} \\ &+ c_5 \sum_{j=1}^{n} \Theta_{JFj}^T \Sigma_{JF}^2 \Gamma_{JFj}^{-2} \Theta_{JFj} \\ &+ c_6 \sum_{j=1}^{m} \Theta_{MFj}^T \Sigma_{MF}^2 \Gamma_{MFj}^{-2} \Theta_{MFj} \end{aligned} \tag{56}$$

where $c_{5/6}$ are design parameters which can be chosen large enough to retain stability while keeping the set of ultimate boundedness small. Due to the fact that the DSC controller has been shown to be UUB, the same is proven for the adaptive control law, for gains given in (28)–(32). ■

VI. Results

An evaluation of this control scheme was performed, utilizing the simulation of a spherical joint that is spanned by five muscles (see Fig. 2c). The muscles were attached asymmetrically such that muscles (1) and (2) are positioned excentrically, to allow for rotational movements (see Fig. 2b). The simulation model was implemented in MATLAB/Simulink, utilizing the full dynamic model of muscles with linear springs ($K = 5000\,\mathrm{N\,m^{-1}}$) and of the movable link. Actuator parameters were chosen to match a motor-gearbox combination by Maxon motors (RE-25 + GP22HP). The muscle kinematics were simulated by means of computing muscle lever arms for given joint angles, assuming straight line muscles. A friction model for the joint friction (43), as well as friction due to the tendon routing was introduced. The first was implemented by the Stribeck model [21]:

$$\tau_F = \tau_c + (\tau_s - \tau_c) \cdot e^{-|\dot{q}/\omega_s|^{\delta_s}} + \mu_v \cdot \dot{q} \tag{57}$$

where τ_c and τ_s denote the Coulomb and the static friction torques, respectively, μ_v the viscous friction coefficient and ω_s the Stribeck velocity. The latter was kept simple by assuming Coulomb friction which depends only on the transmitted force and the Coulomb friction coefficient μ_c:

$$f_F = \mu_c \cdot f \tag{58}$$

The developed controllers, both adaptive and static, were implemented and compared to CFC. The latter features a central control law (13) and low-level muscle force controllers. Similarly, the DSC control law can be separated into a distributed muscle force controller, consisting of ϕ_2 and ϕ_3, and a central trajectory controller ϕ_1. Only the result of ϕ_1 and a FFW part is needed for computing ϕ_2 which is defined as follows:

$$\phi_{2\mathrm{FFW}} = L(\alpha) r - f_1(\alpha, \dot{q}, f) \tag{59}$$

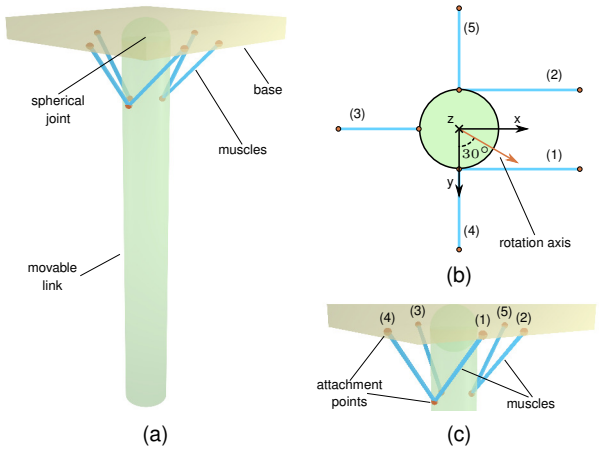

Fig. 2: Simulation Model. (a) rendering of the model, (b) top view, and (c) close-up of the muscle attachment points.

It needs to be evaluated centrally for each of the muscles and communicated to the low-level controllers. This provides for the central high-level and the distributed low-level control. In this experiment a frequency of 200 Hz and 1 kHz is assumed, respectively.

The evaluation was performed by applying a reference trajectory, leading to a movement of the spherical joint around a fixed axis in the X-Y plane. This axis was chosen to apply asymmetric loads to the muscles (see Fig. 2b). Accordingly, a rotation of $45°$ was performed up and back down again, leading to a trajectory of 15 s. This trajectory was initially executed, utilizing CFC and DSC, for friction coefficients which have little effect on the control performance ($\tau_s = 0.01\,\mathrm{N\,m}$, $\tau_c = 0.008\,\mathrm{N\,m}$ and $\mu_c = 0.0$, see Fig. 3a-b). For all experiments, the control parameters were tuned to exhibit similar control effort, i. e. the motor voltage v_A was in the same range. It can be seen from this experiment that the average error in the position was lower for DSC by an order of magnitude (see Fig. 3a-b) which can be given for the CFC and the DSC control law as $0.052\,\mathrm{rad}$ ($\sim 3.0°$) and $0.004\,\mathrm{rad}$ ($\sim 0.2°$), respectively Furthermore, the CFC controller exhibits a considerable steady state offset in the Z-Axis which is the longitudinal axis of the cylinders. This problem could possibly be addressed by increased control gains. However, due to neglected low-level dynamics, this can quickly lead to oscillations.

In a second set of experiments, the friction coefficients were increased to match the characteristics that are found in actual systems ($\tau_s = 0.05\,\mathrm{N\,m}$, $\tau_c = 0.04\,\mathrm{N\,m}$ and $\mu_c = 0.4$). Other parameters of simulation and controller were kept identical. As expected, performing the same experiment with increased friction led to increased trajectory tracking errors for both the CFC, as well as the DSC controller (see Fig. 3c-d). Both exhibit significant steady state offsets. Average position errors for CFC and DSC can be given as $0.080\,\mathrm{rad}$ ($\sim 3.9°$) and $0.069\,\mathrm{rad}$ ($\sim 4.6°$), respectively. Therefore, the adaptive controller, developed in Section V was evaluated

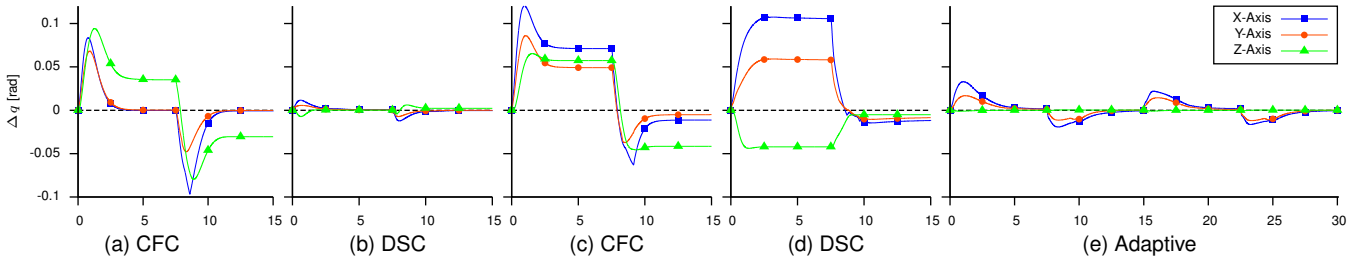

Fig. 3: Controller Performance. The joint velocity errors $\Delta\dot{q}$ and the joint angle error Δq is depicted over the trajectory for the different control scenarios, (a-b) low friction and (c-e) high friction.

for the same system (see Fig. 3e). Parameters for the RBFN compensators were chosen to match friction models, leading to 11 neurons with equidistant RBF centers for the muscle friction and 15 neurons with centers spaced logarithmically for the joint friction. The latter allows for capturing the non-linear behavior of the Stribeck model around zero. The average position error for the second trial (see Fig. 3e) can be given as $0.009\,\text{rad}$ ($\sim 0.5°$).

VII. Conclusions

A generic control framework for the class of musculoskeletal robots was presented, comprising a general model and an improved control law, based on the non-linear control techniques of *backstepping* and DSC. It features an holistic controller which includes the compliance in the muscles, as well as the actuator dynamics. The previously developed CFC, based on feedback linearization, neglected these effects and handled them as a disturbance. The control law developed in this work led to an improved control performance, which was shown in the simulation of a spherical joint with five muscles. In the presence of joint, as well as muscle friction, the control performance was significantly improved by the introduction of adaptive friction compensators, based on RBFNs.

VIII. 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. 288219 - MYOROBOTICS.

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] T. Kozuki, H. Mizoguchi, Y. Asano, M. Osada, T. Shirai, U. Junichi, Y. Nakanishi, K. Okada, and M. Inaba, "Design Methodology for the Thorax and Shoulder of Human Mimetic Musculoskeletal Humanoid Kenshiro -A Thorax structure with Rib like Surface -," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 2012, pp. 3687–3692.

[3] S. Wittmeier, C. Alessandro, N. Bascarevic, K. Dalamagkidis, D. Devereux, A. Diamond, M. Jäntsch, K. Jovanovic, R. Knight, and H. G. Marques, "Towards Anthropomimetic Robotics: Development, Simulation, and Control of a Musculoskeletal Torso," *Artificial Life*, vol. 19, no. 1, pp. 171–193, 2013.

[4] S. C. Jacobsen, H. Ko, E. K. Iversen, and C. C. Davis, "Antagonistic control of a tendon driven manipulator," in *Proc. IEEE International Conference on Robotics and Automation ICRA*, 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] 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.

[7] J. Na, X. Ren, G. Herrmann, and Z. Qiao, "Adaptive neural dynamic surface control for servo systems with unknown dead-zone," *Control Engineering Practice*, vol. 19, no. 11, pp. 1328–1343, Nov. 2011.

[8] J. H. Oh and J. S. Lee, "Control of flexible joint robot system by backstepping design approach," in *Proc. IEEE International Conference on Robotics and Automation ICRA*, vol. 4, Apr. 1997, pp. 3435–3440.

[9] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1893–1899, 2000.

[10] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer New York, 2007.

[11] A. L. Schwab and J. P. Meijaard, "How to draw Euler angles and utilize Euler parameters," in *Proceedings of IDETC/CIE*, 2008.

[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*, Dec. 2011, pp. 2211–2216.

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

[14] H. K. Khalil, *Nonlinear systems*, 3rd ed. Prentice Hall New Jersey, 2002, vol. 3.

[15] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *Internation Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[16] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 195–202, 2005.

[17] S. J. Yoo, J.-B. Park, and Y.-H. Choi, "Adaptive Dynamic Surface Control for Stabilization of Parametric Strict-Feedback Nonlinear Systems With Unknown Time Delays," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2360–2365, 2007.

[18] O.-E. Fjellstad and T. I. Fossen, "Quaternion feedback regulation of underwater vehicles," in *Proc. IEEE Conference on Control Applications*, Aug. 1994, pp. 857 –862.

[19] W. H. Young, "On Classes of Summable Functions and their Fourier Series," *Proceedings of the Royal Society of London. Series A*, vol. 87, no. 594, pp. 225–229, Aug. 1912.

[20] L. Le-Tien and A. Albu-Schäffer, "Adaptive Friction Compensation in Trajectory Tracking Control of DLR Medical Robots with Elastic Joints," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 2012, pp. 1149–1154.

[21] S. S. Ge, T. H. Lee, and S. X. Ren, "Adaptive friction compensation of servo mechanisms," *International Journal of Systems Science*, vol. 32, no. 4, pp. 523–532, Jan. 2001.

[22] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice Hall, 1996.

[23] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," MIT Artificial Intelligence Laboratory, Cambridge, MA, USA, Tech. Rep., 1989.