Reachset Conformance Testing of Human Arms with a Biomechanical Model

Cédric Stark, Aaron Pereira, and Matthias Althoff

Abstract—**Guaranteeing safety in human-robot co-existence often requires a prediction of the volume that could be occupied by the human up to a future time, in order to avoid collisions. Such predictions should be simple and fast for real-time calculation and collision-checking, but account even for unexpected movement. We use a complex biomechanical model to search for extreme human movement, to validate such a prediction. Since the model has a large input space and highly nonlinear dynamics, we use an exploration algorithm based on RRTs to efficiently find the extreme movements. We find that the simple prediction encloses all arm positions found by the exploration algorithm, except where the biomechanical model does not account for collision between body tissue.**

Index Terms—**Conformance checking, human-robot interaction, biomechanical modelling.**

I. Introduction

Robots and humans must increasingly share a workspace in several application areas. Although methods for safe reaction to collision exist, e.g. [1], in many cases non-collision is the only way to ensure safety, especially with high-inertia robots. To ensure a safe working environment one can either separate the working areas of robots and human co-workers, or enable robots to predict and avoid human movement.

Avoiding collisions with people requires a robot to predict the space occupied by their human co-workers. A method such as [2] uses a simple, online prediction of the future human occupancy to ensure that the robot is able to stop before the human can reach it, and thus ensure safety. However, this guarantee of safety relies on the prediction being overapproximative, i.e. accounting for all movement possible for the human. We validate the conservativeness of a low-order human arm occupancy prediction by using a high-order biomechanical human arm model. Movements are simulated with the high-order model and checked whether or not they are included in the occupancy prediction by the low-order model.

A. Human motion prediction

Much work has been done on predicting motion where intention is known. Since natural movement is believed to be near-optimal with regard to cost functions based on dynamics, simulated movements should minimize these functions [3], for example spatial jerk in point-to-point movement [4]. With the development of *Digital Human Models* (DHM), which are simulations of the musculoskeletal structure of the human, more precise predictions can be made, for example taking into account muscle effort [5].

Since human intention cannot be predicted reliably, probabilistic methods have been proposed. Among others, Hidden Markov Models [6] and Markov chains [7] can be used to find probable movements and then calculate the most likely occupancy accordingly [8]. Though these methods work well for most human behaviour, unusual movements like reflex movements, tripping, or grabbing falling objects, may not occur in the training data and may not be accounted for.

An approach to account for unexpected motion is reachability analysis [9], which conservatively predicts the set of states of a system even where dynamics and parameters are uncertain, but the uncertainty is bounded. One drawback of reachability analysis is that it is computationally expensive for nonlinear dynamical systems such as the human body, limiting its online use. One can therefore abstract the dynamics to simpler, linear dynamics, for which reachable sets can be found online. Such a method for the human arm is shown in [10], [11].

The question then arises, whether the regions of human occupancy in space predicted by the simplified dynamics (henceforth *reachable occupancies*) do indeed account for the actual dynamics. In this work we use a DHM to try to find extreme movements of the human arm which are outside of the occupancy predicted by the simplified model.

B. Reachset Conformance Testing

Reachset conformance testing [12] involves a search for traces (behaviors) of a dynamical system which fall outside the predictions of a model which claims to be conservative. To find such a behavior, the system's high-dimensional input and/or parameter spaces need to be efficiently searched. Several approaches exist: Monte Carlo methods, as well as methods from trajectory planning or optimization, such as Ant Colony Optimization or Genetic Algorithms, are used to attempt to falsify properties of hybrid systems or to explore the state space [13], [14].

Simulation of the DHM is time-consuming, so as much input space should be explored in as few simulations as possible. The DHM has high-dimensional inputs, and where deterministic sampling may suffer from the curse of dimensionality (i.e. the size of a grid grows exponentially with dimension), random sampling methods can often excel. The rapidly-exploring random tree (RRT)[15] is promising in this respect, as it guides the search towards previously unsampled areas of the state space and thus explores the space faster.

The authors are with the Department of Informatics, Technische Universität München, 85748 Garching, Germany. cedric.stark@tum.de, aaron.pereira@tum.de, althoff@in.tum.de

For instance, RRTs are used to try and find trajectories of autonomous vehicles which violate set-based predictions [16].

On the other hand, randomness introduces its own problems and random sampling is not always superior to deterministic sampling [17]. Deterministic samples can be taken from a lattice and can also be derived from optimality criteria rewarding "useful" samples [18]. In our case, by applying some additional assumptions to limit the possible inputs, we can deterministically sample a reduced set of inputs to construct a tree which acts as "scaffolding" for the RRT algorithm; the latter will further expand the initial tree.

The DHM also has highly nonlinear dynamics; in [19], the authors linearize the dynamics of a hybrid, nonlinear system at each step to explore the state space. We adapt this approach and augment it with a method of grouping inputs to reduce the dimensionality of the input space dramatically, such that the input space can be explored rapidly.

We state the problem formally in the next section and introduce our human models in Sec. III. In Sec. IV we detail the exploration algorithm, and show the results of our tests in Sec. V.

II. Problem Statement

A prediction of the future occupancy of the human arm can be generated from a low-order kinematic model of the human arm, and maximum joint accelerations, velocities and positions obtained from motion capture data as presented in [11]. There is a risk that movements that are not in the motion capture data set may not be accounted for in the occupancy space obtained, in which case the prediction does not account for all movement of the arm, i.e. is not *overapproximative*, or that the simplification from the complex nonlinear biomechanical system of the human arm to a low-order kinematic model with linear dynamics may neglect certain possible movements.

The aim of this work is to validate the low-order model by using a high-order biomechanical digital human model (DHM) to effectively find extreme movements of the human arm. Validating against a DHM is advantageous as compared to a human, since it does not tire and the initial conditions can be replicated perfectly several times. The problem is not trivial as: 1) the muscle inputs which lead to the extreme positions are not easily determined analytically, 2) there are 50 muscles in the DHM, leading to a large input space and 3) simulation is computationally expensive, so the input space should be explored as quickly as possible. We next describe the arm models used for this work.

III. Human Arm Models

We wish to compare the prediction of occupied space from a low-order, set-based model of the human arm with those of a high-fidelity DHM. Specifically, we wish to show that the predictions of the occupancy from the DHM are always contained in those of the set-based model. This section explains both models: Sec. III-A and III-B describe the dynamics of both models in detail, and Sec. III-C explains how the spatial occupancy is derived from both models for comparison.

A. Set-Based Model

The low-order, set-based prediction [11] is performed as follows. Offline, we obtain motion-capture data of 38 test subjects (see Fig. 1(a)). Reflective markers on the arm are tracked during a range of motions, chosen to cover the entire human workspace as much as possible. Using inverse kinematics, the joint positions of a simple, 4-degree-of-freedom (DOF) model shown in Fig. 2 are determined at each point in time, and from this time series, we find the joint velocity and acceleration (Fig. 1(b,c)). From these, we obtain the linear dynamics of the simple kinematic model as differential inclusions.

To make a prediction during a human-robot collaborative operation, we start from the initial set of states $\mathcal{X}_0$, computed from inverse kinematics applied to the measured position data followed by adding a margin of error to account for sensor uncertainty (Fig. 1(d,e)). We use a Lagrangian technique for reachability analysis [9] implemented in the COntinuous Reachability Analyzer (CORA) [20] to predict the set of joint positions reachable until a time horizon (Fig. 1(f)). This is then converted to a *reachable occupancy* in task space using the method of [21] (Fig. 1(g)). An overapproximative occupancy is obtained, allowing the robot's trajectory planning to avoid these areas. We detail the conversion from joint positions to occupancy in Sec. III-C.

B. Digital Human Model

The high-fidelity human arm model is from [23], shown in Fig. 3. The model consists of 12 rigid bodies, 11 of them bones and 1 being the complete hand. 50 different actuators (muscles) receive an individual excitation between 0 and 1 and exert forces on the rigid bodies. All muscles use a model from Schutte *et al.* [24]. The force exerted by a muscle depends nonlinearly on its length (a function of the kinematics of the arm) and its *activation*. Muscles receive as input an *excitation* $u \in [0, 1]$, which influences the activation $a(u, t)$ through the first order dynamics:

$$a = \int \frac{u - a}{\tau(a, u)} dt, \tag{1}$$

$$\tau(a, u) = \begin{cases} t_{act}(0.5 + 1.5a), & \text{for } u > a, \\ t_{deact}/(0.5 + 1.5a), & \text{for } u \leq a. \end{cases} \tag{2}$$

For the muscles in [24], $t_{act} = 7.667\,ms$ and $t_{deact} = 1.460\,ms$; the resulting activation curve is shown in Fig. 4. This means that muscles which are at rest do not exert their full force immediately, nor do activated muscles stop pulling immediately. Furthermore, muscles and tendons are also elastic, meaning that they exert a passive contractile force when they are stretched beyond their slack length. The force exerted by the muscles is a function of activation, the length of the tendon and length, as well as rate of change of length, of the muscle, and the angle at which the muscle fibres are attached to the tendon (*pennation angle*). Since the focus of this work is not biomechanical modelling, we refer the reader to [24] for further details. The high-order human arm has 7 joint DOFs (shown in Fig. 5):

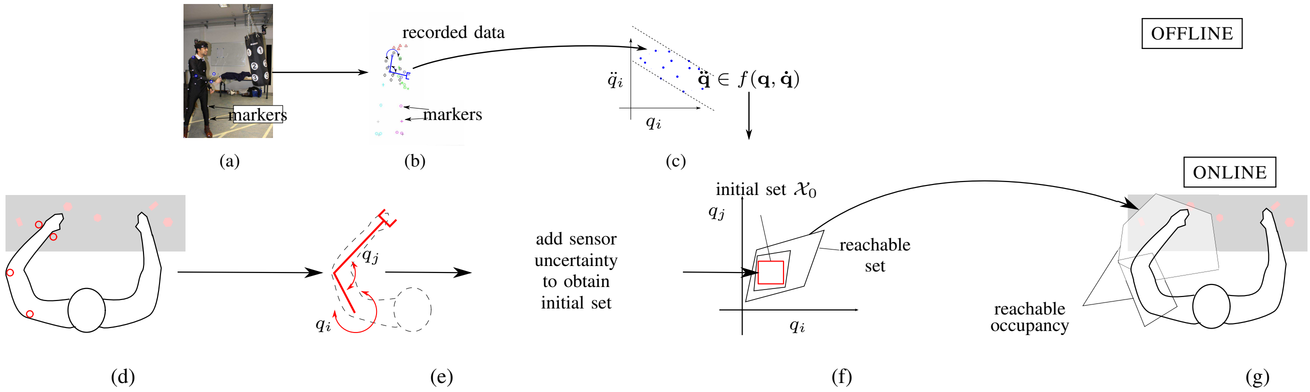

Fig. 1. Method as in [11]: offline, the dynamics are obtained from analysis of human movement and abstraction to a simple kinematic model. Online, the human pose from sensors is fit to the same model, and reachability analysis used to determine the possible reachable occupancies.

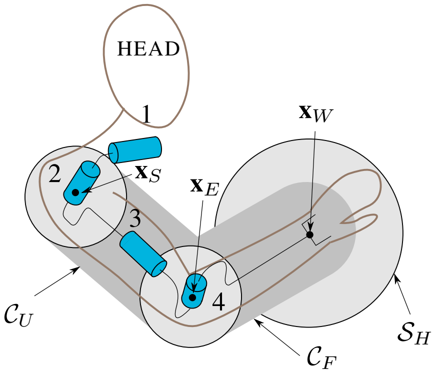

Fig. 2. Low-order human arm model with 4 degrees of freedom (blue capsules). The capsules $\mathcal{C}_U, \mathcal{C}_F$ and sphere $\mathcal{S}_H$, defined at the shoulder ($\mathbf{x}_S$), elbow ($\mathbf{x}_E$) and wrist ($\mathbf{x}_W$) enclose the volume of the arm and are considered to be its spatial occupancy.

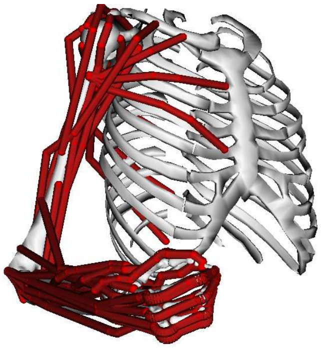

Fig. 3. High-order human arm model[22].

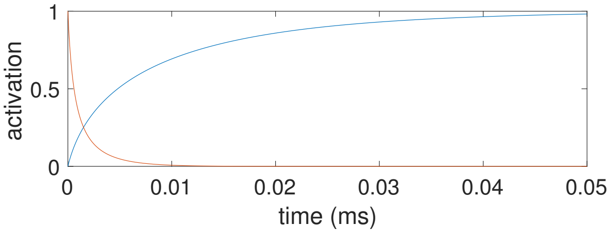

Fig. 4. Activation (blue) and deactivation (orange) profiles of a muscle initially at 0 and 1 activation respectively.

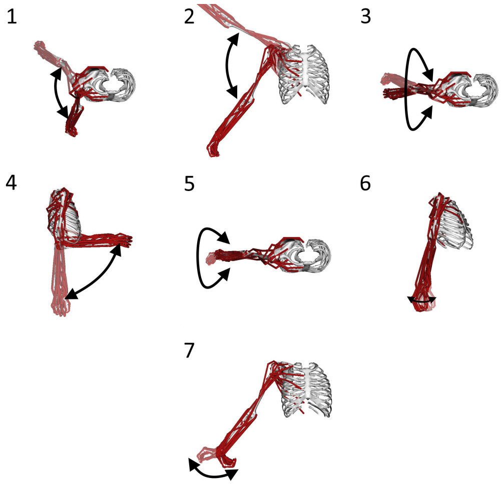

Fig. 5. Joints available in the high-order human arm model [22].

- *elv_angle (1), shoulder_elv (2), shoulder_rot (3)* at the shoulder,
- *elbow_flexion (4)* at the elbow,
- *pro_sup (5)* the pronation/supination of the forearm,
- *deviation (6), flexion (7)* at the wrist.

The rotation of the whole body around a horizontal axis is also a parameter of the model, but we do not change this in the exploration nor is it affected by the muscles, so we do not consider this a DOF.

One drawback of the DHM is that it does not account for collisions between body tissue. That is, positions may be found which are in fact impossible since muscles would intersect. This is shown in Sec. V-A

C. Occupancy

To provide a fair comparison, we define the *occupancy* of the arm and show how to derive it from markers placed on the arm's rigid bodies (Fig. 6). From these markers, we identify the *shoulder* $\mathbf{x}_S$, *elbow* $\mathbf{x}_E$ and *wrist* $\mathbf{x}_W$, and a base coordinate system at the shoulder as in [11].

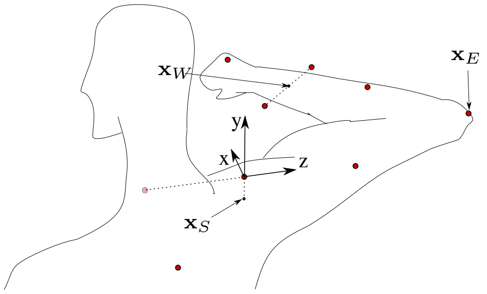

Fig. 6. Markers and local base coordinate system. The base coordinate system is defined by markers on the neck and torso as in [11]; the point $\mathbf{x}_E$ is at a marker on the elbow; the point $\mathbf{x}_W$ is the midpont of two markers and the point $\mathbf{x}_S$ is taken as -40mm in the z-direction from a marker on the shoulder.

We define the terms *Minkowski sum*, *sphere-swept volume*, and *capsule*. The Minkowski sum $\oplus$ of two sets A and B is defined as $A \oplus B = \{a + b \mid a \in A, b \in B\}$. A sphere-swept volume (SSV) is the Minkowski sum of the convex hull (CH) of a set of vertices, and a sphere. A capsule is a special case of an SSV, in which there are only two vertices in the set. We call the set of vertices forming the convex hull the *defining points* of the SSV or capsule. We consider the arm occupancy at any point in time to be overapproximated by a capsule $\mathcal{C}_U$ defined at the shoulder and elbow, a capsule $\mathcal{C}_F$ defined at the wrist and elbow (both of radius $0.1m$), and a sphere $\mathcal{S}_H$ of radius $0.205m$ enclosing the hand, see Fig. 2. Specifically, $\mathcal{C}_U = \text{CH}(\mathbf{x}_S, \mathbf{x}_E) \oplus \mathcal{B}_{0.1}$, $\mathcal{C}_F = \text{CH}(\mathbf{x}_E, \mathbf{x}_W) \oplus \mathcal{B}_{0.1}$, and $\mathcal{S}_H = \mathbf{x}_W \oplus \mathcal{B}_{0.205}$ where $\mathcal{B}_l$ is the closed, zero-centred ball of radius l.

In the low-order model of [11], we obtain the joint angles through inverse kinematics from the base coordinate system and shoulder, elbow and wrist positions found from the markers. Having performed reachability analysis to obtain the final set of joint angles, we can then calculate the sets of positions of the shoulder S, elbow E and wrist W using the method of [21], whereby the occupancy in Cartesian space of points fixed on a kinematic chain can be found overapproximatively, for a set of joint values. The final overapproximative occupancy is then a union of 3 SSVs: $\text{CH}(E, W) \oplus \mathcal{B}_{0.1}$, $\text{CH}(S, E) \oplus \mathcal{B}_{0.1}$ and $W \oplus \mathcal{B}_{0.205}$.

In the high-order model, we find the position of these markers placed on the rigid bodies of the arm, and thereby the shoulder, elbow and wrist positions, at every simulated end position of the arm. We then check if these are inside sets S, E and W found by the low-order model. By the property of convexity, if the defining points of the capsules $\mathcal{C}_U$ and $\mathcal{C}_F$ are inside the respective sets predicted by the low-order model, and the centre of the sphere $\mathcal{S}_H$ is inside W, the entire occupancy is accounted for in the lower-order model. Checking for enclosure is detailed in Sec. V-A.

IV. Exploration algorithm

We detail here the algorithm used to explore the high-dimensional input space of the DHM. Because of infinitely many potential initial positions and excitations, a few simplifications are made, in order to simplify the exploration process:

(a) Muscle excitations are 0 or 1, even though every computer-representable number in between would be possible.

(b) Only the final positions are used for computing the occupancy, ignoring potential extrema in mid-movement.

(c) Muscle excitations are constant until the time horizon t_f, during step 2 of our method (see below).

A. Method

Our method consists of 3 steps:

1) *Reduce input space dimension.* Since the high-order model offers 50 excitable muscles, there are a total of $2^{50} \approx 1.1 \cdot 10^{15}$ possible combinations as input to the model; to simulate them all is infeasible. We therefore cluster the muscles into groups with similar effects. We observed that several muscles act on the same rigid bodies and in the same general direction, e.g. the lateral and medial triceps both extend the elbow and could be grouped together. By grouping similar muscles we obtained 10 groups, in which we actuated the muscles together. This reduced the number of possible input combinations from 2^{50} to 2^{10}, and is performed only once.
2) *Create an initial tree of explored states.* For each new starting state, we perform deterministic sampling based on knowledge of the model dynamics to find states we suspect to be extreme movements, quickly. We divide the time until the time horizon t_f into n equal timesteps. For each input[1] $u \in \{0, 1\}^{10}$, the arm is simulated from the initial position to times $\frac{t_f}{n}, 2\frac{t_f}{n}, ..., t_f$ with a constant input, and a tree of states constructed (see Fig. 7).
3) *Use RRTs, expand the tree.* Starting from the tree found in the previous step, we used RRTs to further explore the state space efficiently and account for cases where the inputs change over the simulation time interval.

We elaborate on each step below.

B. Muscle Clustering

We obtained a quantification of the effects of individual muscles as shown in Alg. 1, where m_{inputs} is the number of inputs, i.e. muscles, $\mathbf{0}$ is a vector of zeros of length m_{inputs}, $\boldsymbol{u}_i$ means that the excitation of the i^{th} muscle is 1, and that of the rest is 0, and the function $simulate(\boldsymbol{s}, \boldsymbol{u}, t)$ returns the state after simulating starting from state $\boldsymbol{s}$ with an input $\boldsymbol{u}$ for a simulated time of t. The state space comprises the 7 joint angles, the muscle activations, fibre lengths and rate of change of length, and the elapsed simulation time.

Algorithm 1 Muscle clustering

```
$\mathbf{a} \leftarrow simulate(\boldsymbol{s}_{init}, \mathbf{0}, 100ms)$
for $i < m_{inputs}$ do
    $\boldsymbol{c}_i \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{u}_i, 100ms) - \mathbf{a}$
    $\hat{\boldsymbol{c}}_i \leftarrow \frac{\boldsymbol{c}_i}{\|\boldsymbol{c}_i\|}$
$perform\ k\text{-}means\ clustering$
```

[1] $\{0, 1\}^n$ is shorthand for $\{0, 1\} \times ... (n \text{ times}) ... \times \{0, 1\}$

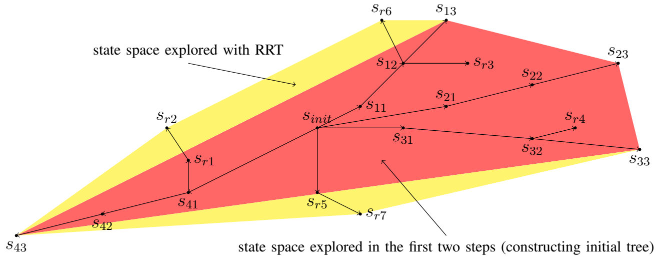

Fig. 7. Exploration process. First an initial tree is constructed by simulating to the time horizon with each possible input. Next, an RRT is used to account for the cases that inputs change during the simulation.s

We then performed k-means clustering on the joint angles $\hat{\boldsymbol{c}}_i$ obtained to find m_{groups} groups of muscles with similar actions, shown in Tab. I (in our case $m_{groups} = 10$). The distance metric is chosen as a weighted Euclidean distance between the joint angles, where the first 4 joints are weighted 1 and the last 3 (pronation/supination of hand and wrist DOFs) are weighted 0.01 since they do not contribute to a change in occupancy using the model from III-C. Exciting muscles in these groups together reduces the number of possible input combinations to 2^{10}.

C. Initial tree construction

Because of the activation dynamics, the highest muscle forces are generated when the excitations are continuously high for several milliseconds (see Fig. 4 and Sec. III-B). The initial tree's purpose is to explore the most extreme cases the fastest, which is why we introduced simplification (c) in Sec. IV. With only 2^{10} inputs, it is now feasible to run the simulation for each input. We construct a starlike tree[2] with the starting state of the arm as the root vertex, and the leaves as the states after t_f, see Fig. 7.

D. Rapidly-exploring Random Trees

Muscle excitations can change during a movement, however. To find the cases which would be ignored by simplification (c), we use RRTs to further explore the input space. Fig. 7 shows the concept. The states $\boldsymbol{s}_{i,j}$ are found by applying the i^{th} combination of excitations as input for j timesteps, where the length of the timestep is $\Delta t = \frac{t_f}{n}$, and (in this particular case), $n = 3$. This is the first and second step of the method generating the volume in red. Combining an RRT with the large initial volume found in the second step should enable us to reach more corner cases by including movements during which the excitations change during the movement.

Compared to most applications of RRTs, we have no goal state – we simply want to explore the space up to time t_f. We therefore adapt the RRT algorithm slightly from [15] to take into account the time we have left to reach the next state, see Alg. 2. Relevant parameters are the number of states η that will be visited to explore the space, and the timestep Δt. The function $find\_closest(tree, \boldsymbol{s})$ finds the state in $tree$ closest to $\boldsymbol{s}$ in terms of Euclidean distance between the joint angles (ignoring the activations, fibre lengths, and their rate of change), whose simulation time is less than t_f.

Algorithm 2 RRT pseudocode

```
1: $tree \leftarrow initialize$
2: for $i < \eta$ do
3:     $\boldsymbol{s}_{goal} \leftarrow random\_value\_in\_state\_space$
4:     $\boldsymbol{s}_{start} \leftarrow find\_closest(tree, \boldsymbol{s}_{goal})$
5:     $\boldsymbol{u}_{best} \leftarrow find\_best\_input(\boldsymbol{s}_{start}, \boldsymbol{s}_{goal})$
6:     $\boldsymbol{s}_{best} \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{u}_{best}, \Delta t)$
7:     $tree \leftarrow add\_to\_tree(tree, \boldsymbol{s}_{start}, \boldsymbol{s}_{best})$
```

Lines 5 and 6 of Alg. 2 attempt to find the state closest to $\boldsymbol{s}_{goal}$ which is reachable from $\boldsymbol{s}_{start}$ in one timestep. Due to the highly nonlinear dynamics, it is infeasible, if not impossible, to determine which input is optimal analytically. We try two methods, one using on random sampling and the other based on linearizing the system, adapted from [19].

E. Random Sampling for RRT Input

We attempt to find the state $\boldsymbol{s}_{best}$ by generating k random inputs $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k$ and finding $\boldsymbol{s}_j \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{u}_j, \Delta t)$ for each of them. We then assign the $\boldsymbol{s}_j$ which minimizes $\|\boldsymbol{s}_j - \boldsymbol{s}_{goal}\|$ to be $\boldsymbol{s}_{best}$. The benefit of this method is that the 10 input groups can be randomly sampled and the corner cases, where the inputs change between samples are found. Alternatively, one can randomly sample all 50 inputs, leading to even more possibilities for different excitations, where not all muscles within one group act together, and potentially finding additional corner cases.

F. RRTs with Linearized Dynamics

Since even the 10-dimensional input space is hard to sample (i.e., k must be chosen very high to adequately sample the space), we adapt the approach of [19] and discretize and linearize the dynamics. To linearize the dynamics, we perform $m_{groups} + 1 = 11$ simulations starting at $\boldsymbol{s}_{start}$ for Δt, firstly with all excitations zero $\boldsymbol{u} = \boldsymbol{0}$, and then with excitations $\boldsymbol{u} = \hat{\boldsymbol{u}}_i, i \in \{1, 2, ...10\}$, where $\hat{\boldsymbol{u}}_i$ means that the excitations of the muscles in the i^{th} *group* are 1, and those of the rest is 0. The approach is shown in Alg. 3.

Algorithm 3 Linearisation procedure

```
$\boldsymbol{\alpha} \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{0}, \Delta t)$
for $i < m_{groups}$ do
    $\boldsymbol{b}_i \leftarrow simulate(\boldsymbol{s}_{start}, \hat{\boldsymbol{u}}_i, \Delta t)$
    $\boldsymbol{\beta}_i \leftarrow \boldsymbol{b}_i - \boldsymbol{\alpha}$
$\boldsymbol{B} \leftarrow [\boldsymbol{\beta}_1^\top \boldsymbol{\beta}_2^\top \cdots \boldsymbol{\beta}_{m_{groups}}^\top]$
```

In this linearized system, we approximate $simulate(\boldsymbol{s}_{start}, \boldsymbol{u}, \Delta t)$ with $\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u}$. Finding the optimal input is minimizing the distance to the goal

[2] a tree is *starlike* if exactly one of its vertices has degree greater than 2.

TABLE I

MUSCLE GROUPS USED FOR GROUPED EXCITATION, WITH THE BODIES THEY ARE CONNECTED TO AND THE GROSS MOVEMENT PERFORMED BY THE MUSCLES IN EACH GROUP

Group	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Muscles	deltoid (anterior, middle, posterior), supraspinatus, infraspinatus	pectoralis major (clavicular, sternal, ribs), coracobrachialis	subscapularis, teres major, latissimus dorsi (thoracic, lumbar, iliac)	teres minor	biceps (short, long), brachialis, brachioradialis	triceps (long, lateral, medial), anconeus	extensor carpi radialis (longus, brevis)	extensor carpi ulnaris, extensor digiti minimi, supinator	Flexor carpi (radialis, ulnaris), palmaris longus, pronator (teres, quadratus), flexor pollicis longus, flexor digitorum (both superficialis & profundus, for digits 2–5)	extensor digitorum communis (digits 2–5), extensor indicis propius, extensor pollicis (longus, brevis)
Rigid bodies	humerus, scapula, clavicle	humerus, scapula, clavicle, thorax	humerus, scapula, thorax	humerus, scapula	humerus, scapula, radius, ulna	humerus, scapula, ulna	humerus, radius, hand	humerus, radius, ulna, hand	humerus, radius, ulna, hand	humerus, radius, ulna, hand
Movement	lift arm	pull arm to thorax	rotate arm inward	rotate arm outward	bend arm	straighten arm	bend hand towards radius	bend hand towards ulna	flex hand	extend hand

$\| \boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{Bu}) \|$, which can be solved as a quadratic programming problem for $\boldsymbol{u}_{best}$:

$$\text{minimize: } (\boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{Bu}))^\top (\boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{Bu})),$$
$$\text{subject to } \boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{1}$$

We finally simulate the RRT with the input $\boldsymbol{u}_{best}$ as in line 6 of Alg. 2.

G. Note on computational complexity

The biomechanical simulation is the most computationally expensive part of our algorithm; the other calculations and write operations are negligible in comparison. The simulation solves dynamical equations numerically with a variable step size, and it is not known *a priori* how many steps are required. We observed, however, that computation time was approximately proportional to the duration we simulate, even on a machine with nondeterministic execution times. We therefore measure computational expense with the *simulation time* $t_{simulated}$, which is the total duration over which the arm has been simulated (i.e., if 20 simulations are run to a time horizon of $50ms$, this is $t_{simulated} = 1sec$).

We look at the complexity with respect to m_{groups}, and also with respect to η. The first step of the 3-step method in Sec. IV-A involves m_{inputs} simulations for a duration of Δt. Computing the initial tree in step 2 requires $2^{m_{groups}}$ simulations for a duration of $n\Delta t$. In step 3, the RRT (linearized) requires $m_{groups}+2$ simulations of a time step Δt per iteration, for η iterations. The total simulation time required by the algorithm is $m_{inputs}\Delta t + 2^{m_{groups}} n\Delta t + \eta(m_{groups} + 2)\Delta t$. Hence the time complexity with respect to m_{groups} is $\mathcal{O}(2^{m_{groups}})$, and with respect to η is $\mathcal{O}(\eta)$.

V. EVALUATION

In this section, we show A) which movements found using our exploration algorithm are accounted for within the predicted reachable occupancies found using [11] in Sec. V-A, and B) we demonstrate the effectiveness of the 3-step approach presented in Sec. IV-A. The simulations were run on an Intel i5-4460, $3.20\,GHz$ with $8\,GB$ of RAM. Simulations were performed in OpenSim 3.3 and the exploration program was written in C++. MATLAB 2016b is used for the quadratic programming. Our visualisation and analysis of volumes uses the Multi-Parametric Toolbox 3 (MPT3) [25].

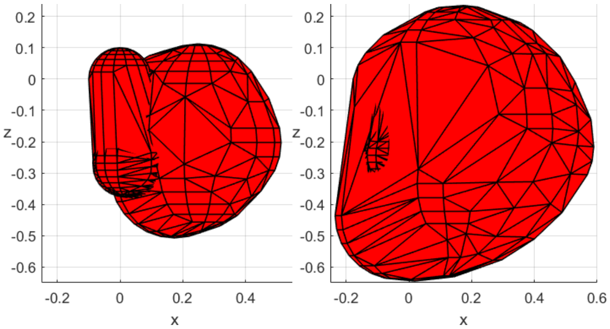

Fig. 8. Growth of the size of the reachable occupancies (axes in m) as calculated using the overapproximative model from [11], intervals $33\,ms$ (left) and $50\,ms$ (right).

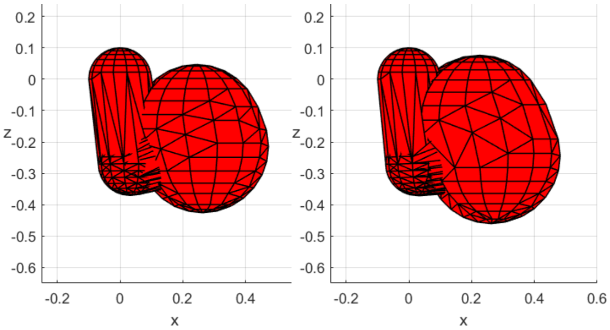

Fig. 9. Growth of the size of the explored volumes (axes in m) calculated using our method, intervals $33\,ms$ (left) and $50\,ms$ (right).

A. Check for Overapproximation

We check for inclusion of the movements found using the DHM in the overapproximative model from [11], for different initial conditions. In the first, the arm is stationary and in the middle of the workspace and muscle activations are zero. In two others, the arm is moving in different directions and

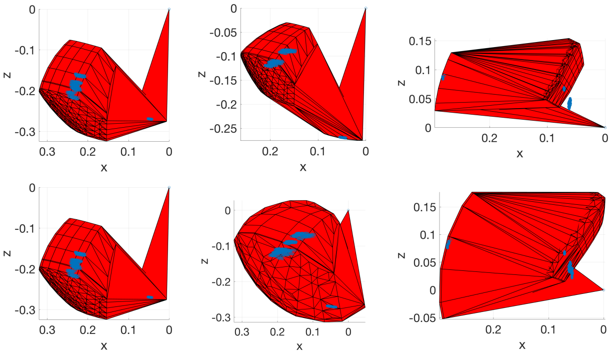

Fig. 10. Shoulder, elbow and wrist points of movements found in the exploration, overlaid on the convex hulls of overapproximative sets S and E, and E and W. The origin is at the shoulder; top row is $t_f = 33ms$ and bottom row is $t_f = 50ms$, for 3 starting states.

muscle activations are nonzero. We check up to two time horizons: $t_f = 33ms$ ($n = 2$) and $t_f = 50ms$ ($n = 3$).

In order to compare the extent of exploration of the state space, we construct the *explored volumes* similarly to the low-order model described in Sec. III-C. We construct the sets S_{sim}, E_{sim} and W_{sim}, which are the convex hulls of all the $\mathbf{x}_S$, $\mathbf{x}_E$ and $\mathbf{x}_W$ found in the simulations. The explored volume is then the union of occupancies $CH(S_{sim}, E_{sim}) \oplus B_{0.1}$, $CH(E_{sim}, W_{sim}) \oplus B_{0.1}$, $W_{sim} \oplus B_{0.205}$. Note that we do not claim this is a set-based prediction: it is simply for visualization in Figs. 8 and 9, and evaluation of the effectiveness of the algorithm in the next section.

We tested a variety of positions; we plot 3 of them in Fig. 10. Plots of the shoulder, elbow and wrist positions found by our algorithm, overlaid on the convex hulls of sets S and E, and E and W, found by our overapproximative method, are shown. In the rightmost position, the simulations do not fall inside the reachable occupancy. On closer inspection, these cases occur when the elbow is flexed further than physically possible, since the biomechanical model does not consider collisions between bodies other than bones. For example, in the rightmost example at 50ms the elbow attains over 175° flexion; the maximum for healthy male subjects is 146° [26].

B. Evaluation of Exploration Approach

We also demonstrate the advantages of the 3-step method presented in Sec. IV. The following algorithms are used for comparison:

1) *random* – we simulate the arm until the time horizon with inputs for each simulation randomly chosen from $\{0, 1\}^{50}$,
2) *enumeration* – only steps 1 and 2 of the method (as described in Sec. IV-B),
3) *RRT naive* – this is step 3, without the previous steps, using the random input method as in Sec. IV-E. We choose $k = 12$.
4) *RRT linearized* – this is step 3, without the previous steps, using the linearization to find the input as in Sec. IV-F.

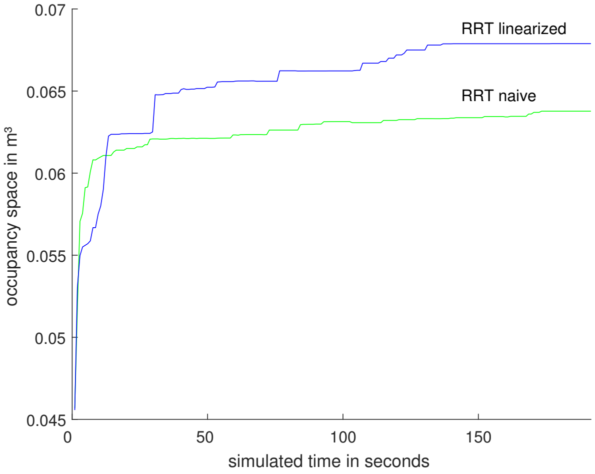

Fig. 11. Comparison of exploration with only RRTs: using random excitations to find the best input (*RRT naive*), and by linearizing the dynamics (*RRT linearized*).

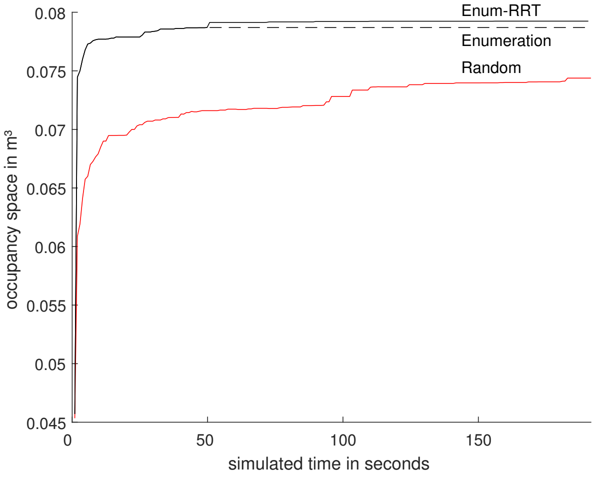

Fig. 12. Comparison of our method (Enum-RRT), and the methods *enumeration* and *random*.

5) *enum-RRT* – the proposed method: *enumeration* followed by *RRT linearized* starting from the obtained initial tree.

For comparison, we let each algorithm simulate an overall simulation time (as defined in Sec. IV-G) of $t_{simulated} = 200$ seconds. We take the number of timesteps to be $n = 3$, and a time horizon of $t_f = 50ms$. In Figs. 11 and 12, the x-axis is $t_{simulated}$, the y-axis is the size of the explored volume reached after x seconds. Fig. 11 compares size of explored volume against the simulation times of the two variations on the RRTs presented in Secs. IV-E and IV-F. The linearization appears to allow the RRT to explore more effectively.

In Fig. 12 we compare *random* excitation with *enumeration* in groups, and the proposed method (i.e. the additional RRT step). Solid lines show the size of the volume during the simulation time; in case the exploration ends before $t_{simulated} = 200s$, dashed lines are used for easier comparability of the volumes' final sizes. Considering explored volume size, the first two steps of the method in Sec. IV-B—building an initial tree from deterministic sampling of the state space—appear to drastically improve the speed of exploration.

Since we set $k = 12$ for *RRT naive*, both it and *RRT linearized* reach 200 seconds of simulated time after 1041 iterations. Interestingly, although the computed volumes are always smaller than the other methods, both *RRT naive* and *RRT linearized* found movements not contained in the computed volumes of *enumeration*, *Enum-RRT* nor *random*.

We found that the most important contribution to fast movements in the biomechanical model is allowing the muscles to reach full activation. Over the timescales that we consider (milliseconds), the activation dynamics is nonnegligible, as shown in Fig 4. During the second step of the exploration, and in methods *random* and *enumeration*, setting the excitations for a muscle group constant until the time horizon means that all muscles reach near-full activation/deactivation and are pulling with close to maximum force or zero force. While RRTs, when used on their own without an initial tree, did discover movements not found in our proposed method, they explored the space much slower than *enumeration*. However, when used in combination with the initial tree, they were able to find corner cases ignored by *enumeration*.

VI. Conclusion

We present a method for validating the conservativeness of an overapproximative, set-based low-order human arm model, using a high-order high-dimensional biomechanical human arm model (DHM). The only positions found using the high-order model which were not included in the occupancy prediction returned by the low-order model, were where the biomechanical model fails to account for collision between bodies of the human arm, which suggests that the overapproximative model does in fact account for all reachable positions.

In order to validate the model we used an efficient algorithm to explore the high-dimensional input space of the DHM quickly and efficiently, based on combining deterministic sampling and RRTs. The exploration method, which performed well on this problem, can be used to validate abstractions of other hybrid and nonlinear models with potentially even higher-dimensional input spaces.

Acknowledgment

The authors gratefully acknowledge financial support by the EC project UnCoVerCPS on grant number 643921.

References

[1] R. Rossi, M. P. Polverini, A. M. Zanchettin, and P. Rocco, "A precollision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 26–31.

[2] A. Pereira and M. Althoff, "Safety control of robots under computed torque control using reachable sets," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 331–338.

[3] S. Albrecht, M. Leibold, and M. Ulbrich, "A bilevel optimization approach to obtain optimal cost functions for human arm movements," *Numerical Algebra, Control and Optimization*, vol. 2, no. 1, pp. 105–127, 2012.

[4] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[5] V. De Sapio, J. Warren, and O. Khatib, "Predicting reaching postures using a kinematically constrained shoulder model," in *Advances in Robot Kinematics*, J. Lenarčič and B. Roth, Eds. Springer, 2006, pp. 209–218.

[6] H. Ding, K. Wijaya, G. Reißig, and O. Stursberg, "Online computation of safety-relevant regions for human robot interaction," in *Proc. 43rd Intl. Symp. Robotics*, 2012.

[7] F. Rohrmüller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using Markov chains for replanning in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2008, pp. 2504–2510.

[8] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2013, pp. 299–306.

[9] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler, "Recent progress in continuous and hybrid reachability analysis," in *Proc. IEEE Conference on Computer Aided Control Systems Design*, 2006, pp. 1582–1587.

[10] M. Ragaglia, A. Zanchettin, and P. Rocco, "Safety-aware trajectory scaling for human-robot collaboration with prediction of human occupancy," in *Proc. Int. Conf. Advanced Robotics*, 2015, pp. 85–90.

[11] A. Pereira and M. Althoff, "Overapproximative human arm occupancy prediction for collision avoidance," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2017, preprint.

[12] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Reachset conformance testing of hybrid automata," in *Proc. Hybrid Systems: Computation and Control*, 2016, pp. 277–286.

[13] Y. Annpureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems." in *Proc. of Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems*, vol. 6605. Springer, 2011, pp. 254–257.

[14] P. Godefroid and S. Khurshid, "Exploring very large state spaces using genetic algorithms," *Int. J. Softw. Tools Technol. Transf.*, vol. 6, no. 2, pp. 117–127, 2004.

[15] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[16] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *Proc. of the American Control Conference*, 2012, pp. 3559–3566.

[17] S. M. LaValle and M. S. Branicky, *On the Relationship between Classical Grid Search and Probabilistic Roadmaps*. Springer, 2004, pp. 59–75.

[18] B. Burns and O. Brock, "Toward optimal configuration space sampling," in *Proceedings of Robotics: Science and Systems*, 2005, pp. 105–112.

[19] A. Bhatia and E. Frazzoli, *Incremental Search Methods for Reachability Analysis of Continuous and Hybrid Systems*. Springer, 2004, pp. 142–156.

[20] M. Althoff, "An introduction to CORA 2015," in *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[21] H. Täubig, B. Bäuml, and U. Frese, "Real-time swept volume and distance computation for self collision detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2011, pp. 1585–1592.

[22] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[23] K. Holzbaur, W. Murray, and S. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Ann. Biomed. Eng.*, vol. 33, no. 6, pp. 829–840, 2005.

[24] L. M. Schutte, "Using musculoskeletal models to explore strategies for improving performance in electrical stimulationinduced leg cycle ergometry," Ph.D. dissertation, Stanford University, 1993.

[25] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. European Control Conference*, 2013, pp. 502–510, http://control.ee.ethz.ch/ mpt.

[26] D. C. Boone and S. P. Azen, "Normal range of motion of joints in male subjects," *J. Bone Joint Surg. Am.*, vol. 61, no. 5, pp. 756–759, 1979.