Calculating Human Reachable Occupancy for Guaranteed Collision-Free Planning

Aaron Pereira[1] and Matthias Althoff[1]

Abstract— **We address the problem of overapproximative short-term prediction of arm movement, for safe human-robot co-existence. A trajectory planner for a robot manipulator which formally verifies non-collision requires fast prediction of human future occupancy, which must be accurate yet still account for all possible human movement. This work presents two approaches to this: a novel, simple approach calculated directly in task space, and another approach using a kinematic model of the human based on the authors' previous work. We compare both approaches in an experimental setup with a robot manipulator. To the best of our knowledge, this is the first implementation and comparison of approaches to formally verified trajectory planning in human-robot co-working. We find that the novel approach has advantages in terms of ease and speed of computation and tightness at short time horizons and is more intuitive to extend to the whole human body; the previous approach offers advantages at longer prediction horizons. We also perform conformance checking to show that both approaches do indeed account for all relevant movement.**

I. Introduction

Ensuring non-collision between humans and moving robots is one approach to guaranteeing safe human-robot interaction [1], especially in the case of high-inertia robots. In the UK manufacturing sector alone between 2011-2015 over 6,800 injuries and 15 fatalities were reported from contact with machinery [2]. Due to the fast and unpredictable nature of human movement, accurate prediction of human occupancy even a few milliseconds in the future is no trivial task, when no knowledge of the human's intention is assumed. This work focusses on human occupancy prediction in the situation of a human working at a workstation with a robot arm nearby. In this case, the human is not ambulatory and human arm movement is the fastest and hence most critical movement for the robot to avoid; we focus therefore on the human arm. Human arms can move fast and movements encompass a wide range of possibilities, hence most predictions used in motion planning are probabilistic. For example, Koppula and Saxena [3] predict high-level activities using information from surrounding objects, while Mainprice and Berenson [4] infer intended motion from early movement using a Gaussian Mixture Model, from which they calculate probabilities that portions of space are occupied and use these in a cost function for a motion planner.

Probabilistic methods, however, cannot account well for scenarios which are unlikely or for which they have not been trained previously. When working at a workstation, humans may perform reflex movements such as recoiling from touching hot or sharp objects, catching a falling object, or swatting a fly. Ding et al. [5] address this limitation: they propose a hybrid stochastic/formal prediction using a Hidden Markov Model to predict occupancy of familiar motions and reachable sets in case of unknown movements. Ragaglia et Al. [6] estimate future occupancy with a kinematic model of the upper body and scale robot motion with distance to the predicted occupancy using assumed limits on joint velocities.

Previous approaches, however, do not guarantee safety formally. The authors present in [7] an algorithm to formally verify robot motion (based on the paradigm of [8]), where the criterion of safety is that the human not be able to touch the robot while it is moving. In contrast to other methods, this approach uses a conservative prediction of the entire set in space that the human may occupy at future times, a *Reachable Occupancy* (RO), to ensure that the robot avoids the human before coming to a stop.

The RO is computed using reachability analysis, a technique to predict all possible states of the system from an initial set of states and some known dynamics, which has been used successfully in verification of automated vehicles [9]. The RO should be a tight overapproximation, i.e. enclose all reachable states, but exclude as many unreachable states as possible. For this we require an accurate model of the system, i.e. of the human body. While biomechanical models can obtain very accurate predictions of motion exploiting knowledge of muscle forces and bone structure [10], such models are hard to use with set-based techniques.

In previous work [11], [12], we propose a *joint space approach* to constructing such an occupancy, using a kinematic model of the arm and a dynamic model built from observed extreme movement. This occupancy encloses relevant arm movements, whereas a model based on the $1.6ms^{-1}$ maximum upper body speed given in the ISO Standard [13] did not. This leaves questions unanswered, namely: 1) how to obtain the occupancy in task space so that it can be checked for intersection against a robot manipulator, 2) whether a kinematic parameterisation is necessary for an tight overapproximation, or if a simpler method can be used, 3) whether such an occupancy prediction is in fact feasible given the real-time constraints of robot motion planning.

The contribution of this work is to present two approaches to prediction and verification of the RO: a novel, simple but accurate *Cartesian space approach* and a *joint space approach* based on previous own work. We demonstrate the conservativeness and tightness of both approaches with reachset conformance checking [14] and their feasibility through experiments.

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

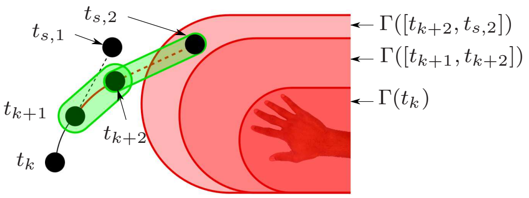

Fig. 1. Verifying safety of a short-term plan. The desired trajectory during time interval $[t_{k+1}, t_{k+2}]$ is found unsafe, as the occupancy of the robot during the subsequent failsafe manoeuvre (green) intersects the RO (red).

The next section describes the problem. Sec. III details the approaches and Sec. IV explains the parameterisation and conformance checking. We compare the approaches in detail in Sec. V-B and show that both methods are conservative, tight, and feasible in a verifiably safe trajectory planner.

II. Problem statement and preliminaries

Our problem is to predict future spatial occupancy of human subjects both 1) conservatively, i.e. accounting for all physically possible human movement, and 2) tightly, i.e. excluding areas which are physically unreachable as much as possible. These requirements are essential in the context of a formally verifiable trajectory planner such as [7], [15]. The concept of a formally verified trajectory planner is that *no movement is executed without being previously verified safe*, see Fig. 1. From the desired trajectory, the robot executes only a short section during time interval $[t_k, t_{k+1}]$, after which a *failsafe manoeuvre* has been planned. The failsafe manoeuvre would bring the robot to a safe state (in our case a stop), at time $t_{s,1}$. We call this a *short-term plan*. During the execution starting at time t_k, the robot plans a failsafe manoeuvre starting at t_{k+2} until $t_{s,2}$, and calculates its occupancy from t_{k+1} until the end of the failsafe manoeuvre starting at $t_{s,2}$ (green capsules), as well as that of the human (red capsules). If they do not intersect, the human cannot reach the robot before the robot is safe, the next short-term plan is said to be *verified*, and the robot proceeds along the desired trajectory at t_{k+1}. Otherwise, the robot executes the failsafe manoeuvre at time t_{k+1} (which had been verified prior to t_k). During the failsafe manoeuvre, the robot can still plan and verify subsequent short-term plans.

Such a trajectory planner requires an overapproximative prediction of human occupancy in time interval $[t_{k+2}, t_{s,2}]$. We call such a prediction a *Reachable Occupancy*:

Definition 1. Reachable Occupancy (RO) *Consider the human body as a dynamical system with an unknown, internal state $\boldsymbol{\xi}(t)$, and dynamics $\dot{\boldsymbol{\xi}} = g(\boldsymbol{\xi}, \boldsymbol{u})$ where $\boldsymbol{u}(t)$ is some input in the set $\mathcal{U}$ of all possible inputs. Let Ξ_0 be the* initial set *of possible body states $\boldsymbol{\xi}(t_0)$ given readings from the sensors at time t_0, allowing for measurement uncertainty. Let $\mathcal{F}(\boldsymbol{\xi}) \subset \mathbb{R}^3$ be the spatial occupancy of the human at a particular state. The system's* reachable set *is: $\Xi([t_i, t_j]) = \{\boldsymbol{\xi}(t_0) + \int_{t_0}^{t} g(\boldsymbol{\xi}, \boldsymbol{u}) dt \mid \boldsymbol{\xi}(t_0) \in \Xi_0, t \in [t_i, t_j], \boldsymbol{u}(t) \in \mathcal{U}\}$. We define the* reachable occupancy *as:*

$$\Gamma([t_i, t_j]) \supseteq \{\mathcal{F}(\boldsymbol{\xi}) | \boldsymbol{\xi} \in \Xi([t_i, t_j])\}.$$

We focus on the human arm: this is the fastest-moving and hence critical body part. Head and torso, while arguably more hazardous during collision, move more slowly and accounting for their movement is less of a technical challenge; we include their prediction in the experiments in Sec. V-B.

We introduce some terminology for the next sections. The Minkowski sum ($\oplus$) is defined over the sets G and H as: $G \oplus H = \{g + h \mid g \in G, h \in H\}$. A *sphere-swept volume* (SSV) is the Minkowski Sum of a polytope [16] and a sphere. A *capsule* is a special case of SSV where the polytope is a line segment, we call the endpoints of the line segment the *defining points* of the capsule. By *task space* we mean the global coordinate system. We next give an overview of the two approaches to prediction before explaining the models in detail and their parameterisation.

III. Approaches for short-term plan verification

We present two approaches for verifying a short-term plan: a *joint space (JS) approach* and a novel *Cartesian space (CS) approach*. Both approaches compute ROs in task space as sets of SSVs to collision-check against the robot occupancy along the short-term plan (SSVs are easy to collision-check) and both approaches use 3 simple models, each accounting for position, velocity and acceleration limits of the human rather than one complex model. The latter would require a hybrid system, and reachability analysis of hybrid systems is complex and not fast enough for our purposes.

The two approaches, however, differ significantly in the verification procedure. The JS approach (Fig. 2, upper part) is first described in [11]. The model we use here has an extra degree of freedom, and we also consider conversion to task space, so we present these novelties in Sec. III-C. The arm position is converted to a set of joint angles, on which we perform 3 reachability analyses over the time horizon of the short-term plan $[t_{k+2}, t_{s,2}]$ using 3 models of joint position, acceleration and velocity limits. The reachable sets are obtained as products of intervals, which we intersect to obtain a reachable set of states satisfying all limits simultaneously. This set is then mapped overapproximatively into task space, and verified against the robot occupancy.

In the CS approach (Fig. 2, lower part), we obtain 3 ROs Γ_{ACC}, Γ_{VEL} and Γ_{POS} directly in task space from 3 models, each accounting for acceleration, velocity or position limits, as unions of SSVs. The exact RO $\Gamma^e([t_{k+2}, t_{s,2}])$ satisfies position, velocity and acceleration limits simultaneously and hence lies in the intersection of these ROs, but we do not compute the intersection, since an intersection of SSVs is not necessarily an SSV. Rather, we verify each occupancy against the robot occupancy. If any RO is verified safe, we know that the exact RO is also safe, hence the short-term plan is verified. Note: the converse is not true, as the robot occupancy may intersect all the models but not the intersection of all models, but this is the trade-off we must make for speed of computation. We next describe how the human geometry is enclosed in capsules. We then present the CS approach and its models in Sec. III-B, and detail the Joint space model in Sec III-C.

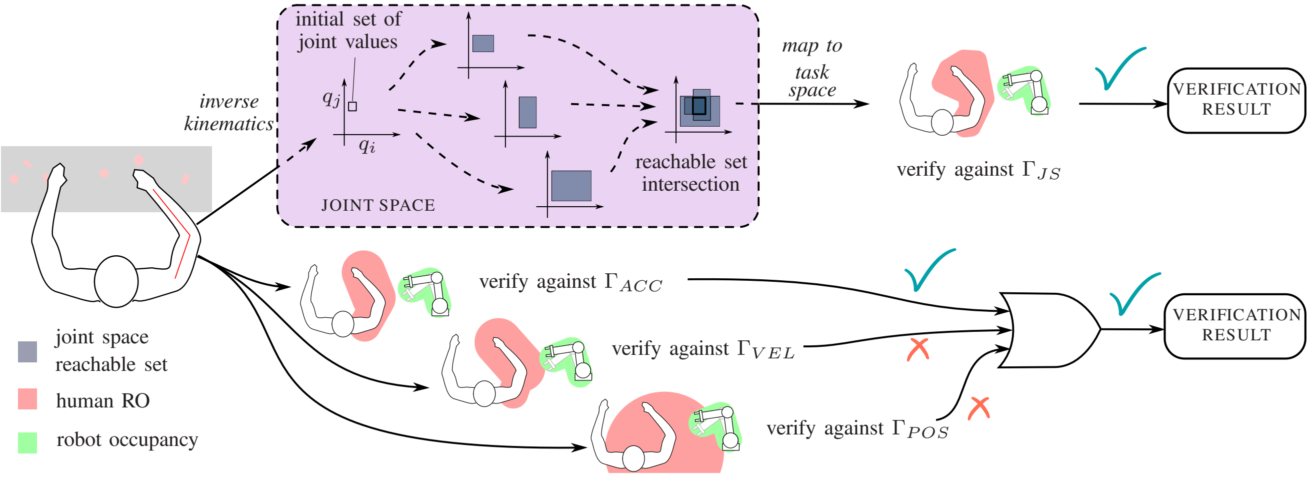
initial set of joint values
q_j
q_i
JOINT SPACE
inverse kinematics
reachable set intersection
map to task space
verify against Γ_{JS}
VERIFICATION RESULT
verify against Γ_{ACC}
verify against Γ_{VEL}
verify against Γ_{POS}
VERIFICATION RESULT
▨ joint space reachable set
🟥 human RO
🟩 robot occupancy

Fig. 2. An illustration of the JS approach (above) and CS approach (below)

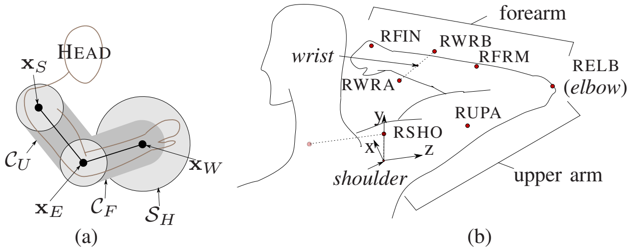

Fig. 3. (a) modelling of the arm as two capsules and a sphere. (b) Markers on the right arm and local base coordinate system for JS approach. The base coordinate system has origin $40mm$ below RSHO and its orientation is with the clavicle coordinate system as in [17].

A. Enclosure of arm

We define two capsules and a sphere which enclose the human arm, see Fig. 3a. The upper arm is enclosed in $\mathcal{C}_U$, with defining points at the shoulder and elbow, and the forearm is enclosed in $\mathcal{C}_F$, defined by the elbow and wrist. We call the positions of shoulder, elbow and wrist $\mathbf{x}_S$, $\mathbf{x}_E$ and $\mathbf{x}_W \in \mathbb{R}^3$ respectively. Both capsules have radius $0.1m$, which is intended to account for arm radius plus clothing. Since wrist movement and forearm rotation are relatively limited and do not greatly affect the occupancy, we account for all hand movement in sphere $\mathcal{S}_H$ centred on the wrist, with radius $0.205m$ – this is the 95th percentile hand length from anthropometric studies [18].

The data to parameterise our models are collected by an infrared motion capture system, where markers are placed on the human body as in Fig. 3b. The *elbow* point is taken at the RELB marker and the *wrist* is the midpoint of RWRA and RWRB. In the JS approach, we must define a base coordinate system for the kinematics. We choose the orientation as that of the clavicle coordinate system as recommended in [17] and the origin at the shoulder, $40mm$ below (in negative y-direction from) the RSHO marker[1] in Fig. 3b. In the CS approach, the ROs are calculated directly in task space, so no base coordinate system for the arm is needed; the *shoulder* is defined $40mm$ vertically below RSHO in task space.

B. Cartesian space approach

1) Terminology: We first introduce some terminology. All norms are Euclidean. Let $B(\mathbf{p};r) = \{\mathbf{x} \mid \|\mathbf{x}-\mathbf{p}\| \leq r\}$ be the closed ball of radius r centred at $\mathbf{p}$. In the subsequent models, we often enclose two balls in a capsule or another ball; we call the operators for this CE and BE respectively. To define CE and BE on the balls $B(\mathbf{p}_1;r_1)$ and $B(\mathbf{p}_2;r_2)$, we first define the following terms:

$$\begin{aligned} i &= \texttt{indmax}(r_1,r_2), & j &= \texttt{indmin}(r_1,r_2) \\ \mathbf{x} &= \mathbf{p}_i - \mathbf{p}_j, & \alpha &= \max(r_i - r_j, \|\mathbf{x}\|) \\ \beta &= \min(r_i - r_j, \|\mathbf{x}\|) & \mathbf{p}_k &= \mathbf{p}_j + \frac{\mathbf{x}}{\|\mathbf{x}\|} \cdot \beta \end{aligned}$$

Operators `indmax` and `indmin` return the indices of the maximum and minimum of their arguments. Let $\mathbf{0} \in \mathbb{R}^3$ be the zero vector. We can then define the operators:

$$\begin{aligned} \mathrm{CE}(B(\mathbf{p}_1;r_1), B(\mathbf{p}_2;r_2)) &:= \overline{\mathbf{p}_i,\mathbf{p}_k} \oplus B(\mathbf{0};r_i), \\ \mathrm{BE}(B(\mathbf{p}_1;r_1), B(\mathbf{p}_2;r_2)) &:= B\Big(\frac{\mathbf{p}_i+\mathbf{p}_k}{2}; \frac{r_i+r_j+\alpha}{2}\Big), \end{aligned}$$

where $\overline{\mathbf{p}_i,\mathbf{p}_k}$ denotes the line segment between $\mathbf{p}_i$ and $\mathbf{p}_k$. The subscripts *S, E, W, U, F* and *H* in the following sections refer to *shoulder, elbow, wrist, upper arm, forearm* and *hand*.

2) Model using Acceleration Limits: This model accounts for movement using a second order model. We consider that a point $\mathbf{y}$ with initial speed $\dot{\mathbf{y}}(0)$ and position $\mathbf{y}(0)$ can accelerate at maximally $a_{\mathbf{y},\max}$ in any direction, i.e. $\|\ddot{\mathbf{y}}\| \leq a_{\mathbf{y},\max}$. We show how to obtain this parameter in Sec. IV. The position of $\mathbf{y}$ after time t is:

$$\mathbf{y}(t) = \mathbf{y}(0) + \dot{\mathbf{y}}(0) \cdot t + \int_0^t \int_0^{\tau'} \ddot{\mathbf{y}}(\tau) d\tau d\tau',$$

which can be bounded by:

$$\mathcal{R}_{\mathbf{y}}(t) = B(\mathbf{y}(0);\delta y) \oplus B(\dot{\mathbf{y}}(0)\cdot t; \delta\dot{y}\cdot t) \oplus B(\mathbf{0}; \frac{a_{\max}}{2}\cdot t^2), \tag{1}$$

[1] Marker names beginning with R are on the right arm and L on the left; we use R in this description.

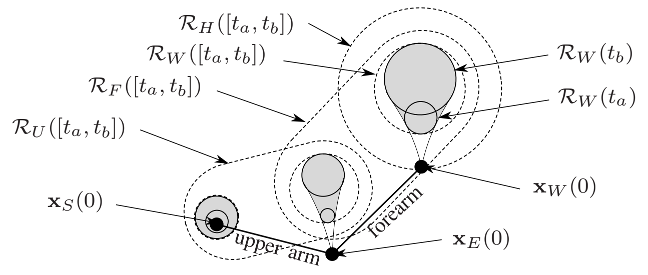

Fig. 4. Occupancy Γ_{ACC} of the human arm for time interval $[t_a, t_b]$ using the acceleration model.

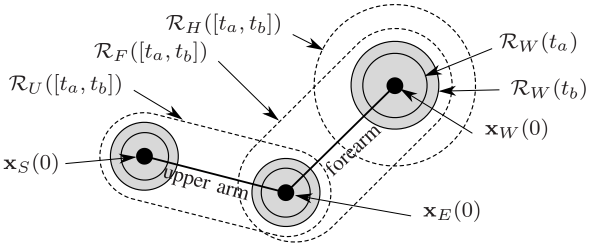

Fig. 5. Occupancy Γ_{VEL} of the human arm for time interval $[t_a, t_b]$ using the velocity model.

where δy and $\delta\dot{y} \in \mathbb{R}$ are the maximum position and velocity measurement uncertainties. $\mathcal{R}_{\mathbf{y}}(t)$ is the RO of $\mathbf{y}$ at time t. The RO over a time interval, $\mathcal{R}_{\mathbf{y}}([t_a, t_b])$, is enclosed in $\text{BE}(\mathcal{R}_{\mathbf{y}}(t_a), \mathcal{R}_{\mathbf{y}}(t_b))$. The proof that the occupancy of the interval $\cup_{t=t_a}^{t_b} \mathcal{R}_{\mathbf{y}}(t)$ is enclosed follows from [19, Prop. V.2].

We calculate the RO of shoulder, elbow and wrist, $\mathcal{R}_S([t_a, t_b])$, $\mathcal{R}_E([t_a, t_b])$ and $\mathcal{R}_W([t_a, t_b])$, in this way. Fig. 4 illustrates the calculation of the reachable occupancy. In the remaining derivation, we omit time dependency for clarity. The occupancy of the forearm $\mathcal{R}_F$ and of the upper arm $\mathcal{R}_U$ are capsules enclosing $\mathcal{R}_W$ and $\mathcal{R}_E$, and $\mathcal{R}_E$ and $\mathcal{R}_S$, respectively; $\mathcal{R}_F$, $\mathcal{R}_U$ and $\mathcal{R}_W$ are extended by the arm thickness and hand length as discussed in Sec. III-A.

$$\begin{aligned}\mathcal{R}_U &= \text{CE}(\mathcal{R}_S, \mathcal{R}_E \oplus B(\mathbf{0}; 0.1)),\\ \mathcal{R}_F &= \text{CE}(\mathcal{R}_E, \mathcal{R}_W \oplus B(\mathbf{0}; 0.1)),\\ \mathcal{R}_H &= \mathcal{R}_W \oplus B(\mathbf{0}, 0.205).\end{aligned} \tag{2}$$

Γ_{ACC} is the union of the capsules enclosing the forearm and upper arm and the sphere enclosing the hand:

$$\Gamma_{ACC} = \mathcal{R}_F \cup \mathcal{R}_U \cup \mathcal{R}_H. \tag{3}$$

3) Model using Velocity Limits: The acceleration model does not take velocity limits into account, hence we present a second Cartesian space model, using only the position $\mathbf{y}(0)$ and the velocity bound $\|\dot{\mathbf{y}}\| \leq v_{\mathbf{y},\max}$, where $v_{\mathbf{y},\max}$ is also found in Sec. IV. The equation of motion can be written as:

$$\mathbf{y}(t) = \mathbf{y}(0) + \int_0^t \dot{\mathbf{y}}(\tau) d\tau,$$

from which we obtain:

$$\mathcal{R}_{\mathbf{y}}(t) = B(\mathbf{y}(0); \delta y + v_{\mathbf{y},\max} \cdot t), \tag{4}$$

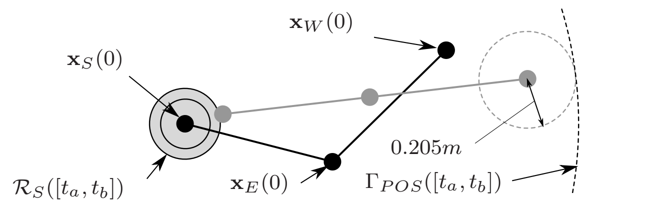

Fig. 6. Occupancy Γ_{POS} of the human arm for time interval $[t_a, t_b]$ using the position model.

As can be seen from Fig. 5, the reachable occupancy at the future time t_b strictly encloses reachable occupancies at previous times, hence $\Gamma_{VEL}([t_a, t_b]) = \Gamma_{VEL}(t_b)$. After calculating the RO of shoulder, wrist and elbow from (4), Γ_{VEL} is then found also from (2) and (3), only substituting Γ_{VEL} for Γ_{ACC} in the latter.

4) Model using position limits: The final Cartesian space model (Fig. 6) uses the fact that the arm has a constant length. The RO cannot extend from the shoulder more than the distance from shoulder to elbow plus elbow to wrist, plus 0.205 accounting for the hand (unless parts of the arm detach, in which case there are bigger problems). The movement of the shoulder can be accounted for using the velocity model described previously. Hence:

$$\begin{aligned}\Gamma_{POS}([t_a, t_b]) = B(\mathbf{x}_S(0); v_{S,\max} \cdot t + \|\mathbf{x}_S(0) - \mathbf{x}_E(0)\|\\ + \|\mathbf{x}_E(0) - \mathbf{x}_W(0)\| + \delta y + 0.205)\end{aligned}$$

C. Joint space Approach

Here we detail the JS approach. We first describe the kinematics, and explain how singularities in the workspace of the kinematic model are avoided. We then detail the dynamic model and how to convert the joint space reachable set into the reachable occupancy Γ_{JS} in Cartesian space.

1) Kinematics: 4-DOF models are widely used to model arm movement, e.g. for determining arm workspace [20] or predicting position during movements [21]. In [6] the authors use a 4-DOF arm model with a unicycle model of human torso movement to estimate human occupancy at a future time. In [11] the authors identify difficulties in a 4-DOF overapproximative model which arise from the need to avoid singularities in the parameterisation, and propose a 3-DOF model to alleviate these; while simpler, a 3-DOF model suffers from large overapproximation at low prediction horizons. We address the difficulty of workspace singularities here as well as the heretofore unsolved problem of accounting for a moving base coordinate system. We simplify the arm to a 4-DOF kinematic chain, see Fig. 7 and call the position of the i^{th} joint q_i.

As mentioned in Sec. III-A, our parameterisation has its base coordinate system at the shoulder. The shoulder spherical joint is represented by three revolute joints in series. The axis of the third joint lies on the upper arm; the other two are chosen such that joint 1 and joint 2 are orthogonal, and joint 2 and joint 3 are orthogonal, see Fig. 7. However, if the upper arm moves such that it aligns with the axis of joint

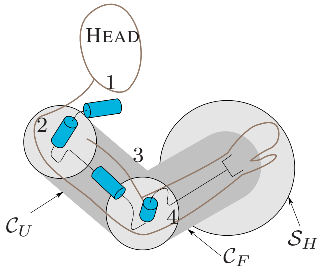

Fig. 7. 4-DOF model of arm

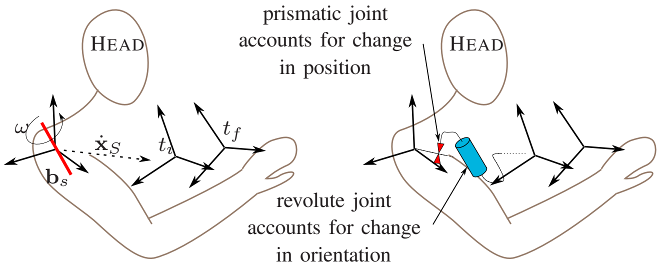

Fig. 8. Accounting for movement of the shoulder coordinate system with a prismatic and revolute joint. Not shown: accounting for rotational and translational accelerations, which are added as a ball to the ROs.

1, the inverse kinematics is singular. As discussed in [11], we avoid this by defining the joint 1 axis along the clavicle.

2) Elbow singularity and mechanical stop: When the elbow is completely extended or contracted, the arm is at a singularity and the value of q_3 is undefined; in the vicinity, the inverse kinematics used to determine $\dot{q}_3$ and $\ddot{q}_3$ in Sec. IV are erroneous. Since we cannot obtain a reliable measurement, as in [12] we omit readings of $\dot{q}_3$ and $\ddot{q}_3$ in this region from calculation of the extreme values, and assume they are the same as elsewhere. Conformance checking in Sec. V-A validates this assumption.

Furthermore, the elbow encounters a mechanical stop at full extension. This leads to very high values of acceleration which are not representative of accelerations elsewhere, hence we also exclude values of $\ddot{q}_4$ in the region where $\|q_4\| > \frac{\pi}{2} - \epsilon$, since $q_4 = -\frac{\pi}{2}$ is full extension and $q_4 = \frac{\pi}{2}$ is full contraction. We take $\epsilon = 0.5rad$.

3) Joint dynamic model: We calculate the reachable sets in joint space, using the parameters $\mathbf{q}_{\min}, \mathbf{q}_{\max}, \dot{\mathbf{q}}_{\min}, \dot{\mathbf{q}}_{\max}, \ddot{\mathbf{q}}_{\min}$ and $\ddot{\mathbf{q}}_{\max}$; these are the minimum and maximum joint positions, velocities and accelerations as obtained in Sec. IV. As shown in Fig. 2, we compute reachable sets from 3 models and intersect them; the intersection of sets $\mathcal{R}_{\mathbf{q}}([t_a, t_b]) = \mathcal{R}_{\mathbf{q}}^{(0)}([t_a, t_b]) \cap \mathcal{R}_{\mathbf{q}}^{(1)}([t_a, t_b]) \cap \mathcal{R}_{\mathbf{q}}^{(2)}([t_a, t_b])$ from three simple models, each accounting for joint position, velocity and acceleration limits yields a tighter overapproximation than any one model alone, and is quicker to compute than a complex model accounting for all limits simultaneously. The models are the following:

1) a position model: $\mathcal{R}_{\mathbf{q}}^{(0)} = [\mathbf{q}_{\min}, \mathbf{q}_{\max}]$
2) a velocity model: $\mathcal{R}_{\mathbf{q}}^{(1)}(t) = \mathcal{Q}_0 \oplus [\dot{\mathbf{q}}_{\min} t \;,\; \dot{\mathbf{q}}_{\max} t]$
3) an acceleration model:

$$\mathcal{R}_{\mathbf{q}}^{(2)}(t) = \mathcal{Q}_0 \oplus \dot{\mathcal{Q}}_0 t \oplus \left[\ddot{\mathbf{q}}_{\min}\frac{t^2}{2} \;,\; \ddot{\mathbf{q}}_{\max}\frac{t^2}{2}\right]$$

where $\mathcal{Q}_0$ and $\dot{\mathcal{Q}}_0$ are the sets of initial joint position and velocities. The sets obtained are Cartesian products of intervals (hyperrectangles).

To obtain the reachable set of the time interval $[t_a, t_b]$, the acceleration model must compute the convex hull of $\mathcal{R}_{\mathbf{q}}^{(2)}(t_a)$ and $\mathcal{R}_{\mathbf{q}}^{(2)}(t_b)$ (proof follows from [19, Prop. V.2]), which is again enclosed in a product of intervals; in the velocity model $\mathcal{R}_{\mathbf{q}}^{(1)}([t_a, t_b]) = \mathcal{R}_{\mathbf{q}}^{(1)}(t_b)$, and $\mathcal{R}_{\mathbf{q}}^{(0)}$ is time independent.

4) Conversion to RO: We use the method from [22] to convert a product of intervals in joint space to a sphere-swept volume (SSV). This method has been also used in [6] and [11] to define the occupancy of a human arm modelled as a kinematic chain. Due to space limitations we do not describe the method here but refer the reader to [22].

5) Accounting for moving shoulder coordinate system: The shoulder coordinate system, in which the RO is first obtained, is not fixed: it can rotate and translate as the human torso moves. The change in the shoulder coordinate system T_0^S can be modelled as 1) the measured values of translational velocity $\dot{\mathbf{x}}_S$ and angular velocity ω around axis $\mathbf{b}_S$, taken as constant plus 2) an error term to account for the fact that $\dot{\mathbf{x}}_S$, ω and $\mathbf{b}_S$ vary. To account for the first, we consider the reachable occupancy obtained previously in the Sec. III-C.4 to be at the end of a kinematic chain consisting of a prismatic joint fixed in task space, followed by a revolute joint, the joint axes of which are the directions of $\dot{\mathbf{x}}_S$ and $\mathbf{b}_S$ respectively (see Fig. 8, right), and the ranges of motion in the time interval $[t_i, t_f]$ are $[\|\dot{\mathbf{x}}_S t_i\|, \|\dot{\mathbf{x}}_S t_f\|]$ and $[\omega t_i, \omega t_f]$. This is analogous to extending the kinematic chain by two joints, so we use the method from [22] again to find the RO Γ_{JS} in task space.

The error made by the assumption that the translational and angular velocities of T_0^S are constant is accounted for in a ball $B(\mathbf{0}, (a_{S,\max} + u_{\max} r_{\max})\frac{t_f^2}{2})$, which is (Minkowski-) added to the SSVs. $u_{\max}$ is the maximum angular acceleration of T_0^S which we find from the data in Sec. IV, and $r_{\max}$ is the maximum distance from the axis of rotation – for the upper arm occupancy, the length of the upper arm, and for the forearm and hand occupancy, the length of the entire arm. Note: accounting for the movement of the shoulder coordinate system is an unavoidable step in the JS approach – a RO must be in the task space in order for collision checking with the robot. A more accurate kinematic model of shoulder and torso movement could possibly increase accuracy, but could further increase complexity: the complexity of conversion from joint space to a RO from [22] is exponential in degrees of freedom. The CS approach foregoes this requirement altogether by computing the ROs directly in the task space. Below, we show how the models are parameterised with extreme movement data.

IV. Parameterisation and conformance checking

Our models must be parameterised such that they account for all physically possible movement which can be observed in a HRC scenario. Extreme movement data was collected from a range of subjects performing movements designed to simulate unexpected or reflex motions in a factory setting. The 38 subjects ranged from 18 to 49 years, 12 were female and 26 male, and the median amount of sport done was 3 hours a week. The movements performed were punching and sweeping the arm sideways, both as fast as possible. Motion was captured using a 10-camera Vicon tracking system at $120Hz$; times where markers were incorrectly tracked were identified by simple distance checks between markers, and omitted from the parameterisation.

From the time series of marker positions we obtain filtered positions, velocities and accelerations with a Kalman filter. Error covariance was taken as a diagonal matrix of $0.0005m$ and process covariance was taken as a diagonal matrix of zeros for the position and velocity parts of the state and $1000ms^{-2}$ for the acceleration part; these parameters eliminated noise without attenuating velocities and accelerations.

For the CS approach, we obtained the velocity and acceleration parameters for the shoulder, elbow and wrist ($v_{S,\max}$, $v_{E,\max}$, $v_{W,\max}$, $a_{S,\max}$, $a_{E,\max}$ and $a_{W,\max}$) from their maxima over the whole dataset. The procedure for obtaining joint velocity and acceleration limits is described in Appendix. 1.

We perform reachset conformance checking [14] of the approach with the parameters obtained above, to test whether the predictions are indeed overapproximative and evaluate the RO volumes. We compare the two approaches with each other and with a prediction based on the ISO standard. Our test data is from a publicly available dataset from the Graphics Lab at Carnegie Mellon University[2]. The first 5 datasets are similar to movements found at a factory workstation and the latter 3 test the prediction limits.

Our datasets are:

1) subject 62: construction work/random (25 motions);
2) subject 70: carrying a suitcase; (10 motions)
3) subject 80: selected everyday motions (44 motions);
4) subject 82: jumping, pushing, banging (10 motions);
5) subject 76: swatting at bug (1 motion);
6) subject 94, Indian dance (16 motions);
7) subject 102, basketball (32 motions);
8) subject 124, sport-related motions (13 motions)

We predict the ROs of both arms over time intervals $[t_i, t_f] = [8.3, 16.7]$; $[25.0, 33.3]$; $[41.7, 50.0]$ and $[58.3, 66.7]$ at every timestep, and check that all markers at time t_f are contained within the ROs[3]. Measurement uncertainties in position were taken as $\pm 0.002m$ and $\pm 0.01rad$, and in velocity, $0.02m/s$ and $0.1rad/s$. Tab. I shows the number of movements predicted entirely accurately by the approaches; for the CS approach, a movement is considered accounted for, only if the arm is accounted for by each of Γ_{ACC}, Γ_{VEL} and Γ_{POS}. To compare our approaches with the standard assumptions on upper-body motion, we parameterise the Cartesian velocity model from Sec. III-B.3 with the maximum speed of the upper body from ISO Standard 13855 [13], $1.6m/s$, and check whether the arm is contained in this occupancy. As shown in Tab. I in the last column, this does not account for all movements from the first 5 datasets and performs even worse on the final 3.

TABLE I

Conformance Checking for 8 datasets. Bold text: not all movements in dataset accounted for

Dataset	JS approach	CS approach	ISO approach
1 (25 motions)	25	25	25
2 (10 motions)	10	10	**9**
3 (44 motions)	44	44	**35**
4 (10 motions)	10	10	**7**
5 (1 motion)	1	1	**0**
6 (16 motions)	16	16	**4**
7 (32 motions)	**3**	**25**	**0**
8 (13 motions)	**8**	**0**	**0**

TABLE II

Volume comparison over a sample of data (m^3)

Prediction interval (ms)	Γ_{JS}	Γ_{ACC}	Γ_{VEL}	Γ_{POS}	Γ_{ISO}
$[8.3, 16.7]$	0.08	0.08	0.44	2.05	0.09
$[25.0, 33.3]$	0.24	0.21	1.51	2.39	0.11
$[41.7, 50.0]$	0.69	0.64	3.87	2.76	0.15
$[58.3, 66.7]$	1.88	2.17	7.98	3.17	0.19

V. Evaluation and discussion

We compare the volumes of the predicted occupancies, and describe and discuss the implementation in a HRC scenario.

A. Volume comparison

Tab. II shows average RO volumes over a sample of the data from the previous section. We use the same time intervals as the conformance checking. Using our parameters, Γ_{ACC} is tighter than Γ_{VEL} and Γ_{POS}. However, this does not mean the latter are redundant: cases where Γ_{VEL} is verified safe and Γ_{ACC} is not verified safe may arise during high-speed movements, since Γ_{ACC} does not consider that the arm cannot accelerate beyond velocity limits and may include volume which cannot be reached. Also, at times greater than $50ms$, both Γ_{VEL} and Γ_{ACC} become extremely large and extend far from the shoulder. Since Γ_{POS} incorporates the constraint of the arm's fixed length, this could help verify the trajectory at larger time horizons, for example during temporary sensor occlusion.

At large time horizons, it can be seen that Γ_{JS} starts to perform better, since it inherently accounts for both the constant size of the body parts as well as maximum joint

[2] Available mocap.cs.cmu.edu, accessed 11.08.15.

[3] since the evaluation data was captured at 60 and $120Hz$, t_f corresponds to a whole number of timesteps.

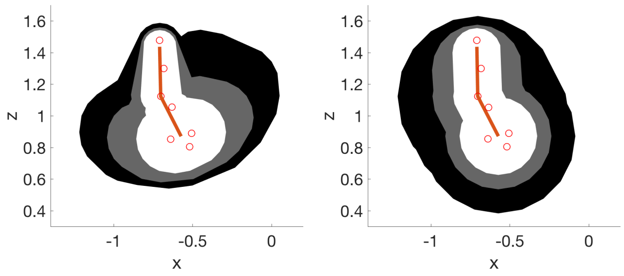

Fig. 9. Human arm with the shoulder at the top and hand at the bottom, underlaid with projections of reachable occupancies $\Gamma([8.3, 16.7])$ in white, $\Gamma([25.0, 33.3])$ in grey and $\Gamma([41.7, 50.0])$ in black. Left: Γ_{JS}, right: Γ_{ACC}.

speeds, accelerations and positions. However, at $t_f \leq 50ms$, the volume of Γ_{JS} is comparable with Γ_{ACC} – in our tests even slightly larger. We found that accounting for the moving base of the joint space RO as in Sec. III-C.5 had a significant effect on the volume, increasing it at times by up to 40% from that of the RO prior to this step. A projection of the ROs Γ_{JS} and Γ_{ACC} is shown in Fig. 9.

B. Implementation and comparison of methods

We implemented our predictions on a setup with a Schunk PWA4 Arm controlled by a Speedgoat Target Machine running real time Simulink 2015b. The robot performs arbitrary movements and the human is sensed using a 6-Camera Vicon Vero system at $250Hz$. We consider only the upper body. In both CS and JS approaches, we accounted for the head and torso in a capsule and predicted its occupancy with the Cartesian acceleration model (Sec. III-B.2) using the maximum acceleration of the shoulder $a_{S,\max}$. The robot is controlled in interpolation position mode at $500Hz$, i.e., the controller sends a demanded position every $2ms$, which the joint motor controllers interpolate at $1000Hz$. Collisions are detected using the GJK algorithm [23] in the joint-space approach and a simple capsule-capsule collision detection algorithm in the CS approach. Measurement uncertainties were the same as in Sec. IV Sensor latency is taken as $10ms$ [4]. Since this work focusses on human occupancy prediction, we do not describe the enclosure of the robot occupancy (this can also be done as in [22], but we use a simpler approach involving capsules) or the generation of the stopping trajectory, but mention that the latter respects given acceleration and jerk limits. The video attachment illustrates trials from both approaches; a still is shown in Fig. 10.

Computation times are shown in Tab. III. Both were fast, but the CS approach was better: generation and collision-checking of complex SSVs in the JS approach take more time than the efficient capsule-capsule collision-checking in the CS approach. The inverse kinematics and accounting for movement of the shoulder coordinate system also slow calculation.

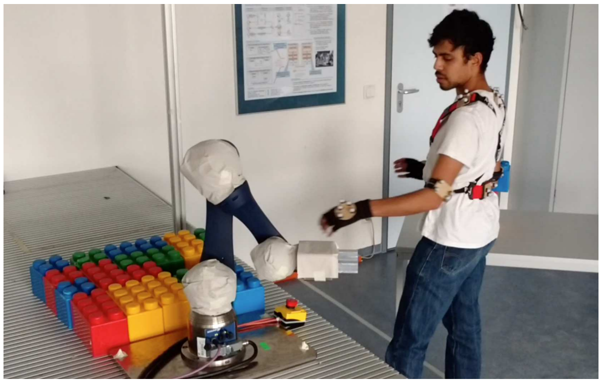

Fig. 10. Human wearing reflective markers working alongside robot. The robot is performing a stop as the human is reaching into its workspace. Not shown: ceiling mounted motion-tracking system.

TABLE III

EXECUTION TIME ON REAL-TIME MACHINE AVERAGED OVER $720s$ OF OPERATION (μs)

Approach used	JS approach	CS approach
Plan and send joint positions	352	352
Compute ROs and verify	4	124
Total	356	476

We found that the parameterisation of the human models affects performance. As we used extreme movements in parameterisation, the robot behaviour was more cautious; furthermore, in order to reduce stopping times (and thus the prediction horizon, which is $t_{s,2}$ plus sensor latency), we allowed large joint jerks, which led to jerky movement. Allowing a less strict parameterisation, e.g. using the ISO standard speed of $1.6m/s$, could improve robot performance, at the expense of not accounting for some human movement.

The implementation of the CS approach was also far more straightforward than the JS approach. The inverse kinematics require reliable and accurate sensing, which is currently not available for a factory environment. The CS approach, on the other hand, also works with more basic sensors: replacing the Vicon system with a Xbox Kinect 2 sensor, we could not perform inverse kinematics nor get accurate velocity measurements to calculate Γ_{ACC}, but we could calculate Γ_{VEL} and Γ_{POS}. The robot works verifiably safely, though performance is reduced because of the lack of the more accurate model Γ_{ACC} as well as the higher latencies. It is well worth noting that the CS approach is highly parallelisable and also immediately extendable to other body parts, since one only needs to define capsules enclosing body parts, not a kinematic model.

Finally, we note that if the verification procedure is computationally expensive, the JS approach may be better since it has only one verification calculation, whereas if the inverse kinematics and collision-checking of SSVs is expensive, the CS approach presents advantages. Because of the property of overapproximation, if a trajectory is verified safe using

[4] The computation time of the motion capture software is given as 2.8ms (www.vicon.com/products/software/tracker, retrieved 14.2.15) so this is a conservative estimate.

any overapproximative model, it can be considered to be safe. ROs using different models, some with more expensive calculations, can therefore be calculated in parallel and verification performed in parallel using only those models that are calculated in time.

VI. Conclusion

We show how to generate reachable occupancies which account for all human movement in an accurate and fast manner. To this end, we present and compare two approaches: a joint space approach which generates one complex occupancy, and a Cartesian space approach which generates three quick and simple occupancies. We show with real experiments, for the first time, how a trajectory planner can incorporate fast but accurate prediction of *all* human motion, to generate robot motion that is *guaranteed* to be safe.

Appendix: Obtaining joint space parameters

We show how we obtain joint positions, velocities and accelerations from the Cartesian positions, velocities and accelerations. We first find the joint angles $\mathbf{q}$ from the shoulder coordinate system and elbow and wrist positions $\mathbf{x}_E$ and $\mathbf{x}_W$ as in Fig. 3, from inverse kinematics. We express $[\mathbf{x}_E^\top, \mathbf{x}_W^\top]^\top$ as $\mathbf{x}_{EW}$ for brevity. From $\mathbf{q}$, we compute the Jacobians $J_E(\mathbf{q})$ and $J_W(\mathbf{q})$ at elbow and wrist, concatenating them to $J_{EW}(\mathbf{q}) = [J_E(\mathbf{q})^\top, J_W(\mathbf{q})^\top]^\top$, hence:

$$\dot{\mathbf{x}}_{EW} = J_{EW}(\mathbf{q}) \cdot \dot{\mathbf{q}}. \tag{5}$$

The joint vector $\mathbf{q}$ has dimension 4×1 and the vector $\mathbf{x}_{EW}$ has dimension 6×1, so to obtain the joint velocity we calculate $J_{EW}^{+}(\mathbf{q})$, the pseudoinverse of $J_{EW}(\mathbf{q})$. We therefore premultiply (5) by $J_{EW}^{+}(\mathbf{q})$ and obtain the joint velocities from $\dot{\mathbf{q}} = J_{EW}^{+}(\mathbf{q}) \cdot \dot{\mathbf{x}}_{EW}$, when the arm is not at a singularity. To obtain the accelerations, we differentiate (5), rearrange, and premultiply by $J_{EW}^{+}(\mathbf{q})$:

$$\begin{aligned} \ddot{\mathbf{x}}_{EW} &= J_{EW}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{J}_{EW}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} &= J_{EW}^{+}(\mathbf{q}) \cdot \left(\ddot{\mathbf{x}}_{EW} - \dot{J}_{EW}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \dot{\mathbf{q}}\right). \end{aligned}$$

Acknowledgment

The authors are grateful to Andrea Giusti who suggested determining joint accelerations directly from Cartesian accelerations, to Dario Beckert, Florian Böck and Natalie Reppekus for work on the implementation, and to Jonas Schmidtler, Thomas Illa and Asuman Sezgin, who captured the motion data. The comparison data used in this project was obtained from `mocap.cs.cmu.edu`; this database was created with funding from NSF EIA-0196217. This research received funding from the Marie Curie Actions of the EU's 7th Framework Programme, REA grant number 608022. The authors gratefully acknowledge financial support by the EC project UnCoVerCPS, grant number 643921.

References

[1] A. E. Kirchner, J. Gea Fernandez, P. Kampmann, M. Schröer, J. H. Metzen, and F. Kirchner, *Formal Modeling and Verification of Cyber-Physical Systems*. Springer, 2015, ch. Intuitive Interaction with Robots – Technical Approaches and Challenges, pp. 224–248.

[2] "Reported injuries to workers in Great Britain 2011/12 - 2014/15," Health and Safety Executive UK, Tech. Rep., 2011-15. [Online]. Available: hse.gov.uk/statistics/tables/ridaggr.xlsx

[3] H. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, 2015.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. IEEE-RSJ Int. Conf. Intell. Robots and Systems*, 2005, pp. 2210–2215.

[9] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Dissertation, Technische Universität München, 2010.

[10] 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.

[11] A. Pereira and M. Althoff, "Overapproximative arm occupancy prediction for human-robot co-existence built from archetypal movements," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 1394–1401.

[12] ——, "Overapproximative human arm occupancy prediction for collision avoidance," *IEEE Transactions on Automation Science and Engineering*, vol. preprint, pp. 1–14, 2017.

[13] "Safety of machinery - positioning of safeguards with respect to the approach speeds of parts of the human body," International Organization for Standardization, ISO Standard 13855:2010, 2010.

[14] 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.

[15] M. J. Zeestraten, A. Pereira, M. Althoff, and S. Calinon, "Online motion synthesis with minimal intervention control and formal safety guarantees," in *Proc. IEEE Sys., Man. Cybern.*, 2016, pp. 2116–2121.

[16] B. Grünbaum, *Convex Polytopes*, 2nd ed., S. Axler, F. W. Gehring, and K. A. Ribet, Eds. Springer, 2003.

[17] G. Wu, F. C. van der Helm, H. D. Veeger, M. Makhsous, P. V. Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. W. Werner, and B. Buchholz, "ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion-part ii: shoulder, elbow, wrist and hand," *J. Biomech.*, vol. 38, no. 5, pp. 981 – 992, 2005.

[18] S. Pheasant and C. M. Haslegrave, *Bodyspace: Anthropometry, Ergonomics and the Design of Work*. Taylor & Francis CRC Press, 2006, ch. Hands and Handles, pp. 143–160.

[19] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, 2014.

[20] N. Klopčar and J. Lenarčič, "Kinematic model for determination of human arm reachable workspace," *Meccanica*, vol. 40, no. 2, pp. 203–219, 2005.

[21] 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.

[22] 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.

[23] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, 1988.