Safety Control of Robots under Computed Torque Control Using Reachable Sets

Aaron Pereira and Matthias Althoff

***Abstract*— A failsafe control strategy is presented for online safety certification of robot movements in a collaborative workspace with humans. This approach plans, predicts and uses formal guarantees on reachable sets of a robot arm and a human obstacle to verify the safety and feasibility of a trajectory in real time. The robots considered are serial link robots under Computed Torque schemes of control. We drastically reduce the computation time of our novel verification procedure through precomputation of non-linear terms and use of interval arithmetic, as well as representation of reachable sets by zonotopes, which scale easily to high dimensions and are easy to convert between joint space and Cartesian space. The approach is implemented in a simulation, to show that real time is computationally within reach.**

I. Introduction

Control of robots in workspaces shared with humans is increasingly topical as the presence of robots in manufacturing grows. Humans and robots must often work together to take advantage of the intelligence and manipulation skills of the human and the superior power, speed and repeatability of the robot. Guaranteeing human safety while obtaining maximum efficiency and minimum downtime is no trivial task, and in the absence of a failsafe strategy to comply with stringent international standards, factories often require expensive or bulky solutions such as protective cages and light curtains.

ISO10218-1 specifies the current regulations for the behaviour of machines which share a collaborative workspace with humans and stipulates: *"The robot shall stop when a human is in the collaborative workspace... the robot may resume automatic operation when the human leaves the collaborative workspace. Alternatively, the robot may decelerate, resulting in a category 2 stop... fault of the safety-rated monitored stop function shall result in a category 0 stop."* [1] (Categories of stop are defined in Def. 4). Power and load must also be reduced to within heuristic values estimated to minimise damage to humans working with heavy machinery [2]. Even so, it is not formally guaranteed that these values will be enough to prevent injury.

Previous work on guaranteeing safety in human-robot co-working has focussed mainly on making the robots themselves inherently harmless. The DLR's Light-Weight Robot (LWR) uses compliance control [3], however, even without this, the damage to the human on impact is very low during blunt unconstrained impact, according to various injury scales [4]. The cases of constrained collision or sharp impact, even for the LWR, still present problems and each application needs certification. Furthermore, in applications where industrial robots with greater power and consequently greater inertia are required, one cannot necessarily rely on the inherent compliance of the robot.

Where strategies for collision avoidance between robots and humans have been considered, they have either been statistical, non-robust or rely on assumptions that would be unrealistic in the application considered. For example, [5] computes trajectories avoiding moving obstacles bounded by a maximum speed with the concept of dynamic envelopes. However, the maximum speed of the obstacles is in the order of 0.01ms^{-1}, which is too slow to be relevant to human motion. Examples of strategies without formal guarantees are [6] and [7], in which the authors demonstrate a reactive collision avoidance method where a counteracting force proportional to relative distance and relative speed of the obstacle is used. Similarly, [8] propose and realise an approach where speed of the robot is reduced according to the minimum distance and relative velocity between the human and 1-D links.

Offline simulation with human models is used in [9] to determine movement and configurations of the robot that are not only safe, but ergonomic for the human. Offline simulation reduces time-expensive online calculation, especially when using complex models of obstacles. Modelling human arm motion in relation to human-robot interaction is performed using Hidden Markov Models by Ding et al. in [10], [11]. However, simulation and statistical analysis have the inherent drawback that they cannot account for all possible eventualities in finite time; indeed, reachable sets are proposed by the authors of [11] to account for unknown motions.

Reachable sets have been used successfully to guarantee safe trajectories for autonomous vehicles [12] and power networks [13], among other applications. The reachable set of a system is the set of all possible states of the system after a certain time, given an initial set and some known but uncertain dynamics, and allows for sensor error and noise, and disturbances. Its applicability to any hybrid dynamic system (i.e. mixed discrete and continuous dynamics) makes it a promising choice for robot safety applications.

This paper introduces an algorithm for planning and verifying safe paths and stopping trajectories for serial link robot arms which share workspaces with humans, as well as novel techniques for verifying efficiently whether nonlinear constraints are met. This control strategy formally guarantees non-contact between a moving robot part and a human in the workspace, given knowledge of the robot parameters and environment. It is able to comply with the relevant

The authors are with the Department of Computer Science, Technische Universität München, 85748 Garching, Germany. Corresponding email: aaron.pereira@tum.de

ISO standards [1], [2] and offers a low cost and low-space alternative to existing safety measures such as cages.

The remainder of the paper is structured as follows: presented in Sec. II are the aim and overview of the approach, which can be summarised as *Planning, Prediction and Verification*. Computed Torque control of robots is briefly discussed in Sec. III. Sec. IV recapitulates the theory of reachability analysis and describes the adaption to human-robot safety. Sec. V explains how a safe path is *planned* in our Safety Control strategy and in Sec. VI, it is shown how a safe path is *verified*. The results of a simulation are presented in Sec. VII, and finally, conclusions are drawn in Sec. VIII.

II. Problem Statement and Approach

Our goal is a self-certifying Safety Control for a robot that plans, predicts and verifies safety of partial trajectories to the desired goal, only allowing a trajectory if it is certain that it is collision free as suggested in [14]. We certify the trajectory piecewise as shown in Fig. 1. The calculations during the time interval $[t_k, t_{k+1}]$ plan, predict and verify the safe trajectories of the system for the next time interval $[t_{k+1}, t_{k+2}]$, starting from the state measured at t_k. During $[t_k, t_{k+1}]$, the robot follows the safe path that has been planned and verified during the previous timestep as presented in Fig. 2. If, for any reason, the alternative paths for $[t_{k+1}, t_{k+2}]$ cannot be verified, (e.g. the verification calculations take longer than the allotted time step, or no safe paths exist), the system mode will default to the safe path previously verified during timestep $[t_{k-1}, t_k]$. For a path to be verified as "safe", two criteria must be met:

- the possible range of desired torques, velocities and joint positions, allowing for all possible error, are within the prescribed limits $\mathcal{T}_{acceptable}$, $\mathcal{Q}_{limits}$ and $\mathcal{V}_{limits}$ respectively.
- the human is not be able to touch the robot while it is moving., i.e. the reachable set of the human $\mathcal{R}_{human}$ and the reachable set of the robot $\mathcal{R}_{robot}$ during the path to be verified must not intersect as long as the robot is *not* stationary.

To formalise the above criteria, we introduce reachable sets:

Definition 1 (Reachable Set)**.** *Given a system with state $x(t)$, input $u(t)$ and dynamics $\dot{x}(t) = f(x(t), u(t))$, where t is time, and the possible initial states $x(0)$ and inputs $u(t)$ are bounded by sets, $x(0) \in \mathcal{X}_0, u(t) \in \mathcal{U}(t)$. The reachable set at $t = r$ is:*

$$\mathcal{R}(r) = \left\{ x(r) = \int_0^r f(x(t), u(t))dt \;\middle|\; x(0) \in \mathcal{X}_0, u([0,r]) \in \mathcal{U}([0,r]) \right\}$$

Further, the reachable set of a time interval is:

$$\mathcal{R}([0,r]) = \bigcup_{t \in [0,r]} \mathcal{R}(t)$$

Where $q \in \mathcal{Q} \subset \mathbb{R}^n$, $\dot{q} \in \mathcal{V} \subset \mathbb{R}^n$ are the robot joint positions and velocities respectively, we define the state of the system as:

$$x = [q^\top, \dot{q}^\top]^\top \in \mathcal{X} \subset \mathbb{R}^{2n}$$

We further introduce the following mappings on the reachable set: map_{CS}, mapping from the state space to the union of points in the Cartesian space occupied by the robot, and $proj_{JL}$, $proj_{VL}$ and map_{TL}, being the projections from state space to joint space, joint velocity space and the mapping to torque space respectively[1]:

Definition 2 (Mappings)**.**

$$\begin{aligned} \mathcal{Y} &= map_{CS}(x) : \mathcal{X} \to \mathcal{P}(\mathbb{R}^3) \\ q &= proj_{JL}(x) : \mathcal{X} \to \mathcal{Q} \\ \dot{q} &= proj_{VL}(x) : \mathcal{X} \to \mathcal{V} \\ \tau &= map_{TL}(x, u) : \mathcal{X} \times \mathcal{U} \to \mathcal{T} \subset \mathbb{R}^n \end{aligned}$$

$\mathcal{P}(\mathbb{R}^3)$ is the power set of $\mathbb{R}^3$.

Where $\ddot{q}_{des}$ is the desired acceleration (see Sec. III), $\mathcal{R}(t)_{robot}$ is the reachable set of the robot at time t and $\mathcal{V}_{stationary}$ is the set of joint velocities within which the robot is considered stationary, the robot is considered stationary at a time r when:

$$\ddot{q}_{des} = 0 \quad \wedge \quad proj_{VL}(\mathcal{R}(r)_{robot}) \subseteq \mathcal{V}_{stationary}$$

Definition 3 (Valid Trajectory)**.** *Where $\mathcal{R}(t)_{human}$ and $\mathcal{R}(t)_{robot}$ are the reachable sets of the human and robot respectively, a trajectory is called a valid trajectory if the following should hold:*

$$\begin{aligned} \{t \in \mathbb{R}^+ \mid \neg(\ddot{q}_{des} = 0 \wedge proj_{VL}(\mathcal{R}(t)_{robot}) \subseteq \mathcal{V}_{stationary})\} : \\ map_{CS}(\mathcal{R}(t)_{robot}) \cap map_{CS}(\mathcal{R}(t)_{human}) = \emptyset \\ \wedge \; proj_{JL}(\mathcal{R}(t)_{robot}) \subseteq \mathcal{Q}_{limits} \\ \wedge \; proj_{VL}(\mathcal{R}(t)_{robot}) \subseteq \mathcal{V}_{limits} \\ \wedge \; map_{TL}(\mathcal{R}(t)_{robot}, \mathcal{U}(t)) \subseteq \mathcal{T}_{acceptable} \end{aligned}$$

The valid trajectory stops the robot before its reachable set intersects that of the human, while keeping joint positions, velocities and torques within limits.

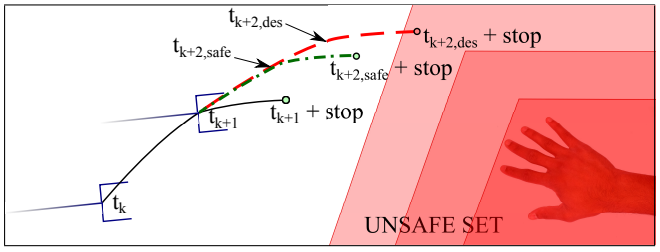

Fig. 1. Illustration of the control strategy. Solid line is the current, fixed trajectory, verified during timestep $[t_{k-1}, t_k]$. Dashed line is the desired trajectory, which is unsafe, and dot-dashed line is a replanned safe trajectory.

We introduce 3 kinds of stops:

[1] when these mappings are used as set-based mappings, sets are replaced by their corresponding power sets.

Definition 4 (CATEGORIES OF STOPS)**.** *The definition of categories of stops is taken from [15] and is as follows:*

- *Category 0 stop: removal of all power to the actuators.*
- *Category 1 stop: deceleration to stationary state effected by actuators; subsequent removal of power from actuators.*
- *Category 2 stop: no removal of power from robot; deceleration to stationary state effected by actuators.*

Our controlled stops are category 2 stops with constant deceleration, because initiating category 0 or category 1 stops wastes operator time and power in an environment where humans continually move in and out of the collaborative workspace. Category 2 stops are also believed to be more intuitive to the human as a smooth trajectory can be planned.

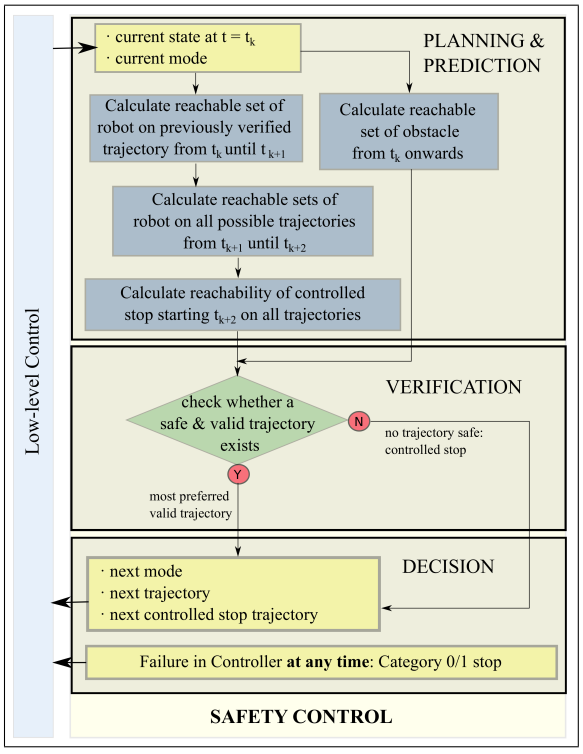

Fig. 2. Outline of Safety Control Strategy

III. COMPUTED TORQUE CONTROL OF ROBOTS

In this paper, serial link robots with joint space Computed Torque Control as described in [16] are considered. Computed Torque Control describes a range of closed-loop robot control schemes in which the control input is the desired acceleration. The known equation of motion of a serial-link robot is:

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

where $q, \dot{q}$ are the vectors of joint positions and joint velocities respectively. $H(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, $F(\dot{q})$ are the torques due to friction and $g(q)$ are the torques due to gravity. We further introduce the feedback error $e = q_{des} - q$, where q_{des} is the desired joint position; $\dot{e}$ and $\int e\,\mathrm{d}t$ are defined similarly. A controller using PID Computed Torque Control takes as input ν, which is the desired acceleration along with the feedback errors:

$$\nu = \ddot{q}_{des} + K_p e + K_v \dot{e} + K_i \int e\,\mathrm{d}t$$

K_i, K_p and K_v are the integral, position and velocity gains respectively, and are of size $n \times n$ where n is the length of q. The computed values are indicated by an overbar [16]:

$$\tau = \overline{H(q)}\nu + \overline{C(q,\dot{q})}\dot{q} + \overline{F(\dot{q})} + \overline{g(q)}$$

We define the set-based sum and product:

Definition 5 (SET-BASED SUM AND PRODUCT)**.** *Set based sum (Minkowski Sum) and product are defined on $A, B \subseteq \mathbb{R}^n$ as follows:*

$$A \oplus B := \{a + b \,|\, a \in A, b \in B\}$$
$$A \otimes B := \{ab \,|\, a \in A, b \in B\}$$

The difference between the calculated matrices and the physical robot matrices may be due, among other things, to uncertainty in measurement of mass parameters and uncertainty when modelling friction. The computed torque may be expressed as equation (1) including an interval of torques due to these modelling errors τ_{ME}, which may be determined experimentally and verified, for example, as in [17].

$$\tau \in H(q)\nu + C(q,\dot{q})\dot{q} + F(\dot{q}) + g(q) \oplus \tau_{ME} \tag{2}$$

Equate (1) and (2): the Coriolis, friction and gravity values cancel, and leave:

$$\begin{aligned} H(q)\ddot{q} &\in H(q)\nu \oplus \tau_{ME} \\ &= H(q)\left(\ddot{q}_{des} + K_p e + K_v \dot{e} + K_i \int e\,\mathrm{d}t\right) \oplus \tau_{ME} \end{aligned} \tag{3}$$

$H(q)$ is positive definite and Hermitian [16], hence $H^{-1}(q)$ exists [18]; premultiplication of (3) yields:

$$\ddot{q} = \ddot{q}_{des} + K_p e + K_v \dot{e} + K_i \int e\,\mathrm{d}t \oplus H^{-1}(q) \otimes \tau_{ME}$$

We calculate $H^{-1}(\mathcal{Q}_{limits})$, the union over all $q \in \mathcal{Q}_{limits}$ of the inverse H matrix. Thus the governing equation of the system of the robot under Computed Torque control is:

$$\begin{aligned} \ddot{q} &\in \ddot{q}_{des} + K_p e + K_v \dot{e} + K_i \int e\,\mathrm{d}t \oplus H^{-1}(\mathcal{Q}_{limits}) \otimes \tau_{ME} \\ &= \ddot{q}_{des} + K_p q_{des} + K_v \dot{q}_{des} + K_i \int q_{des}\,\mathrm{d}t \\ &\quad - K_p q - K_v \dot{q} - K_i \int q\,\mathrm{d}t \oplus H^{-1}(\mathcal{Q}_{limits}) \otimes \tau_{ME} \end{aligned} \tag{4}$$

IV. PREDICTION

The prediction aspect of the algorithm applies techniques for computing reachable sets of linear differential equations to serial link robots. The robot under Computed Torque Control is described by a linear differential inclusion of the form:

$$\dot{x} \in Ax \oplus \mathcal{U}(t) \tag{5}$$

To compute the reachable set over a certain time period, the calculations are performed iteratively from an initial set, and enclosed in an overapproximative hull. These techniques are described in [19].

The reachable sets are overapproximated by zonotopes, a way to represent convex volumes in n dimensions. Zonotopes

are convenient to work with due to their ease of computation of the required operations while calculating the sets of the linear system and when translating from the joint space to the Cartesian space.

Definition 6 (ZONOTOPE)**.** *A zonotope is defined as:*

$$\mathcal{Z} := \left\{ x \in \mathbb{R}^n \;\middle|\; x = c + \sum_{i=1}^{p} \beta_i g^{(i)}, \beta_i \in [-1, 1] \right\}$$

A zonotope can be represented as a centre position vector $c \in \mathbb{R}^n$, and p generator vectors $g^{(i)} \in \mathbb{R}^n$.

Zonotope addition and matrix-zonotope multiplication are described in [19, eq. 5].

A. Computation of Reachable Sets for Computed Torque Control

Converting (4) to the canonical form for the computation of reachable sets (5) results in:

$$\underbrace{\begin{bmatrix} q \\ \dot{q} \\ \ddot{q} \end{bmatrix}}_{\dot{x}} \in \underbrace{\begin{bmatrix} \mathbf{0_{2n,n}} & \mathbf{I_{2n}} \\ -K_i & -K_p \quad -K_v \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \int q \\ q \\ \dot{q} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \mathbf{0_{2n\,,1}} \\ \ddot{q}_{des} + K_i \int q_{des} \mathrm{d}t + K_p q_{des} + K_v \dot{q}_{des} \end{bmatrix}}_{\bar{u}} \oplus \underbrace{\begin{bmatrix} \mathbf{0_{2n\,,1}} \\ \delta\mathcal{U} \end{bmatrix}}_{\mathcal{U}}$$

$$\delta\mathcal{U} = H^{-1}(\mathcal{Q}_{limits}) \otimes \tau_{ME}$$

Here, the input vector has been separated into the constant input $\bar{u}$ and the uncertain input $\mathcal{U}$ (modelling uncertainties). For PD control the state is just the lower two thirds of the PID state: $[q^\top, \dot{q}^\top]^\top$ and A and u are adjusted accordingly. $\mathbf{I_i}$ is the identity matrix of size i, $\mathbf{0_{i\,,j}}$ is a matrix of zeros of size i, j .

B. Translation to Cartesian Space

In order to check intersection, both robot and human reachable sets are mapped to Cartesian space.

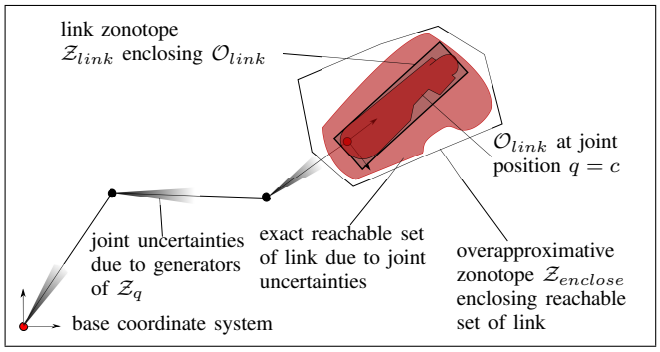

Fig. 3. Uncertainties in joint positions lead to uncertainty in the position of the link, which can be overapproximated by a zonotope.

First, we obtain $\mathcal{Z}_q = proj_{JS}(\mathcal{Z}_x) \subset \mathcal{Q}$, where $\mathcal{Z}_x$ is the zonotope enclosing some reachable set of the robot. The forward kinematic function $F(q) : \mathbb{R}^n \to SE(3)$ gives the transformation of the base coordinate system to the coordinate system of link k from joint angles q. $SE(3)$ is the special Euclidean group of translations and rotations in $\mathbb{R}^3$. An element of $SE(3)$ represented in homogeneous coordinates by a transformation matrix:

$$F(q) = \begin{bmatrix} R(q) & T(q) \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix}$$

$R(q) \in SO(3)$ is a 3×3 matrix representing the rotation ($SO(3)$ is the group of rotations in $\mathbb{R}^3$) and $T(q) \in \mathbb{R}^3$ the translation of the coordinate system to be transformed. The entries of the matrix $F(q)$ are highly non-linear, therefore to obtain some overapproximative matrix of intervals $\mathcal{M}$, where $\mathcal{M} \supseteq \{F(q) \in \mathbb{R}^{4\times4} | q \in \mathcal{Z}_q\}$, a linearisation and overapproximation procedure is necessary.

We refer to the centre of zonotope $\mathcal{Z}_q$, $c \in \mathcal{Q}$ as the *joint position* and the generators $g^{(j)}, 1 \leq j \leq p$ as the *joint generators*. All expressions refer to the k^{th} link of n links.

We linearise $F(q)$ using a Taylor expansion about the joint position c with a Lagrangian Remainder:

$$F(q) \in F(c) + \left.\frac{\partial F}{\partial q}\right|_c (q - c) \oplus \underbrace{\left\{ (q-c)^\top \left.\frac{\partial^2 F}{\partial q^2}\right|_{q^*} (q-c) \;\middle|\; q^* = \alpha c + (1-\alpha) q \,,\, \alpha \in [0,1] \right\}}_{\text{higher order terms approximated by Lagrange remainder } \mathcal{L}(c,q)} \tag{6}$$

Here, $\left.\frac{\partial F}{\partial q}\right|_c$ and $\left.\frac{\partial^2 F}{\partial q^2}\right|_{q^*}$ are the tensors of partial derivatives and partial second derivatives of the change in transformation matrix with respect to joint angles (not to be confused with the Robot Jacobian $J(q)$). They are evaluated at c and q^* respectively, where q^* is defined as in (6), and this is a consequence of the mean value theorem [20]. Offline, we overapproximate $\mathcal{L}(c, q)$ over all of $\mathcal{Q}$ and obtain the fourth order tensor of intervals:

$$H = \left\{ \left.\frac{\partial^2 F}{\partial q^2}\right|_q \;\middle|\; q \in \mathcal{Q}_{limits} \right\}$$

Finally, using (6) and where $g^{(j)}$ are the *joint generators* as defined above, we define:

$$\mathcal{M} = \underbrace{F(c)}_{\text{constant matrix}} \underbrace{\bigoplus_{j=1}^{p} \left\{ \left.\frac{\partial F}{\partial q}\right|_c g_j \oplus g_j^\top \otimes H \otimes g_j \;\middle|\; g_j = \beta g^{(j)}, \beta \in [-1, 1] \right\}}_{\text{matrix of intervals}}$$

So that $q \in \mathcal{Z}_q \implies F(q) \in \mathcal{M}$.

We have precomputed the zonotope $\mathcal{Z}_{link} \subset \mathbb{R}^3$, which encloses $\mathcal{O}_{link}$, the union of all the points occupied by the

link in the link coordinate system, see Fig. 3. We wish to apply the uncertain transformation $\mathcal{M}$ to $\mathcal{Z}_{link}$ to obtain the zonotope in Cartesian space that encloses the link in its position. We split $\mathcal{M}$ into a constant matrix M and a matrix of symmetric intervals $\mathcal{S}, \mathcal{S}_{ij} = [-s_{ij}, s_{ij}]$, So $\mathcal{M} = M \oplus \mathcal{S}$. Then, as a matrix multiplied by a zonotope is a zonotope and a matrix of symmetric intervals multiplied by a zonotope can be overapproximated with another zonotope [19], we find $\mathcal{Z}_{enclose}$:

$$\mathcal{M} \otimes \mathcal{Z}_{link} \subseteq M \otimes \mathcal{Z}_{link} \oplus \mathcal{S} \otimes \mathcal{Z}_{link} \subseteq: \mathcal{Z}_{enclose}$$

$\mathcal{Z}_{enclose}$ is calculated independently using $\mathcal{M}$ for each link and an overapproximation of the entire robot in $\mathbb{R}^3$, $map_{CS}(\mathcal{Z}_x)$, is built from n zonotopes.

C. Reachable Set of the Human

Having proven that reachable sets can be computed for a robot under Computed Torque Control, it remains to compute the reachable set of the human obstacle to check for intersection. This expands upon the idea in [11] that the reachable sets of human motion be used to supplement a Hidden Markov Model where unexpected motion is detected. Our method, however, involves computing the reachable set of all possible motion with no probabilistic element in the algorithm. In contrast to the robot, human parameters are more uncertain[2] as are measurements e.g. from a camera or from a light curtain. Furthermore, the input is completely uncertain, so the entire set of accelerations possible for the human must be taken. Consequently, the reachable set of the human expands rapidly. The challenge is to implement the formal verification fast enough so that the robot will not be constantly stopping, or worse still, be unable to move at all.

Light curtains are a popular and relatively inexpensive way to detect the presence of obstacles in robot workspaces. In the simplest case, the human is detected by a collection of light curtains, where $n^{(i)}$ and $d^{(i)}$ are the unit normal and distance to the origin of the i^{th} of l light curtains, and t_r is the maximum combined time of the light curtain response including the time for the safety relay to activate, which is typically between 10-65 ms depending on model and resolution[3]. v_{max} is the maximum speed of the human. As the position of the human along the light curtain is unknown, the reachable set of the human is the union of all half-spaces on the non-robot sides of the planes defined by the light curtains, offset by the maximum penetration of the human through the light curtain towards the robot. $\mathcal{R}_{human}(t)$ is computed as:

$$\bigcup_{i=1}^{l} \left\{ x \in \mathbb{R}^3 \;\middle|\; n^{(i)\top} x \geq d^{(i)} - v_{max}(t_r + t) \right\} \qquad (7)$$

Maximum upper limb movements are typically overapproximated by the Hand Speed Constant 1.6 ms^{-1} [2], however, if the strategy is to be ISO compliant, v_{max} should be set at the speed of a walking human from [2], 2.0 ms^{-1}.

As the state x is simply the Cartesian coordinates of the reachable set, A in (5) becomes a 3×3 matrix of zeros:

$$\dot{x} = [-v_{max} n\,, 0]$$

Where a more complete model of the human is available, for example through camera systems such as the SafetyEYE[4], the human is considered in terms of links and joints with degrees of freedom, limits of joint values, speed and acceleration. The human reachable set can then be represented as a series of zonotopes corresponding to the links on a simplified, overapproximative model of a human. Describing such a complex model of the human and the mathematical techniques behind it in detail is not within the scope of this paper.

V. Planning

In order to avoid collision when an unsafe scenario is predicted, either the planned spatial path or the velocity must be modified [21]. For example, Zanchettin and Rocco [8] describe a strategy whereby the path is consistent and the speed is reduced in relation to the human's speed and proximity to the robot (sensed by a camera).

Here, the Safety Control works in parallel with the low-level robot controller, as shown in Fig. 2. The cycle time of the Safety Control (Major Timestep) is larger than that of the low-level control (Minor Timestep), in order to have adequate time for planning, prediction and verification, but must be small enough so that the system remains stable and that the error from overapproximation remains small. The trajectory planner in the Safety Control has at least two modes, stopping and following precomputed trajectory, but may include others such as a reduced or increased speed trajectory, or a trajectory with a different spatial path.

If the Safety Control fails, the low-level control executes a category 0 or 1 stop. This is shown in Fig. 2 and complies with ISO 10218-1 [1]

A. Trajectory Planning

When the robot is in controlled stop mode, the Safety Control computes a master trajectory from the current state to the goal state. As this trajectory has six constraints – the speed, position and acceleration at both ends – it may take the form of a polynomial in time of degree 5 or greater (i.e. the desired position of the i^{th} joint $q_{i,des}(t)$ is of the form $\sum_{k=0}^{n} a_{i,k} t^k$ where $a_{i,k}$ are chosen to satisfy the six constraints and $n \geq 5$); other functions for master trajectories are also possible.

A reduced speed or increased speed trajectory, or a new spatial trajectory, may be generated whereby the speed is reduced but the spatial path of the master trajectory is preserved. Following one Major Timestep at the chosen trajectory, a controlled stop is planned. The Major Timestep plus the controlled stop constitute the candidate safe path, as explained in Sec. II.

[2]consider variance in length, position and orientation of human limbs, joint axes etc. as well as different accelerations and maximum velocities.

[3]e.g. pilz.com, techniconiec.com, cedes-sa.com, recovered 26.09.14

B. Controlled Stop

The controlled stop is a category 2 stop with constant deceleration as in Def. 4. For a variety of stopping trajectories, the path of the robot is predicted and it is checked whether $map_{CS}(\mathcal{R}(t)_{robot}) \cap map_{CS}(\mathcal{R}(t)_{human}) = \emptyset$ at every timestep up until and including the first timestep at which the robot is stationary as described in Sec. II; such a trajectory is a *safe* braking trajectory. The safe trajectory with the longest braking time (and consequently the smallest desired acceleration) is selected, as this is most likely not to exceed the maximum torques. This is illustrated in Fig. 4.

It is then checked whether the latter three conditions of a Valid Trajectory as defined in Def. 3 are fulfilled, i.e. that the joint positions and velocities are within limits and the torques within acceptable bounds. If not, as in the bottom example of Fig. 4, then the motors cannot provide sufficient torque to execute this trajectory and a failure signal is given. If positions, velocities and torques are within limits and the reachable sets of the robot do not intersect with those of the human for both the controlled stop and the entire trajectory before it, as in the top example in Fig. 4, then this is a valid stopping trajectory; it is saved as the new safe path.

braking time	0.052	0.056	0.060	0.064	0.068
stops in time?	1	1	(1)	0	0
torques within limits?	0	1	1	1	1

braking time	0.024	0.028	0.032	0.036	0.040
stops in time?	1	(1)	0	0	0
torques within limits?	0	0	0	0	1

Fig. 4. Finding braking trajectory: select the trajectory with slowest possible stopping time (ringed value). Then check demanded torques; in the top example they are within limits, but not in the lower example.

Due to the conditions imposed, i.e. a constant deceleration and the preservation of the instantaneous velocity direction, it may be that a valid controlled stopping trajectory exists, which is not found. Furthermore, the overapproximation in the torque introduced by the partitioning of the state space and precomputation of the inertia matrix and remainder vector may indicate that the torque limits are breached when in fact the demanded torques are within limits. Currently, the computation times of a non-linear system make it infeasible to compute optimum stopping trajectories and verify formally online in time, without these limits imposed.

VI. Joint Position, Velocity and Torque Limits

To verify that joint torques do not exceed motor torques requires a dynamic model of the robot which can be checked at each step. The non-linearity of the dynamic model and the complexity of the equations of the inertia, Coriolis, friction and gravity matrices mean that the translation of the reachable set of the input into the torque space is not possible in real time.

For this reason, the state space is partitioned into a set of c cells $\mathcal{C}_i$, which are comprised of intervals $[x_{k,min}, x_{k,max}]$ in each dimension k of the state space. $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \, \forall \, i \neq j \in \{1, \ldots, c\}$ and $\bigcup_{i=1}^{c} \mathcal{C}_i = \mathcal{Q}_{limits} \times \mathcal{V}_{limits} \subset \mathcal{X}$. For simplicity, denote $proj_{JL}(\mathcal{C}_i)$ by $[q_i]$, and similarly $proj_{VL}(\mathcal{C}_i)$ by $[\dot{q}_i]$. Using interval arithmetic as described in [22] one can overapproximate (1) to obtain the inertia interval matrix $H([q_i])$ and remainder interval vector $rem([q_i], [\dot{q}_i])$, which are evaluated offline for each cell $\mathcal{C}_i$:

$$rem([q_i], [\dot{q}_i]) = C([q_i], [\dot{q}_i])[\dot{q}_i] \oplus F([\dot{q}_i])[\dot{q}_i] + g([q_i])$$

$$[\tau_{min}, \tau_{max}] = H([q_i])\ddot{q} \oplus rem([q_i], [\dot{q}_i])$$

S, the union of cells $\mathcal{C}_i$ of the state space which intersect with the reachable set, is found online and the hulls of all relevant inertia matrices and remainder vectors are taken:

$$[\tau_{min}, \tau_{max}] = \bigcup_{i \in S} H([q_i])[\ddot{q}_{min}\,,\, \ddot{q}_{max}] \oplus \bigcup_{i \in S} rem([q_i], [\dot{q}_i])$$

Here, the union of matrices and vectors of intervals are defined element-wise: $[A \cup B]_{ij} = A_{ij} \cup B_{ij}$.

This gives an interval vector which is an overapproximation of the torques in the reachable set produced by an acceleration (also an interval vector). This can be compared with the maximum torque to check whether limits are exceeded.

The detected cells are also used to check whether the joint positions and velocities are within limits. If $\mathcal{R}(t)_{robot} \not\subseteq \bigcup_{i=1}^{c} \mathcal{C}_i$, the joint positions and velocities are outside the limits $\mathcal{Q}_{limits}$ and $\mathcal{V}_{limits}$. In this case, the trajectory being verified is invalid.

VII. Implementation Simulation with a Light Curtain and PUMA560

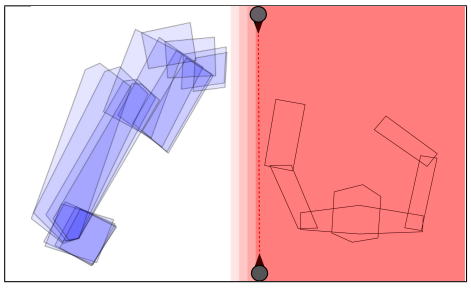

Fig. 5. Setup of the simulation. On the left are the reachable sets of the robot, on the right are those of the human. Here, the robot stops in time and the sets do not intersect while the robot is moving.

We present a simulation involving a Unimation PUMA 560 Robot and a Techniconiec SLC 430 Type 4 light curtain[5] to sense human movement, illustrated in Fig. 5. The PUMA 560 is chosen due to its well-known and available dynamic and kinematic model [23]. This model uses the Robotic Toolbox [24] and the Interval Laboratory [25] as well as the Continuous Reachability Analyser[6]. The robot controller is assumed to use position and velocity (PD) Computed Torque Control; the master trajectory is a quintic polynomial in time

[5]http://techniconiec.com/techguides/Tech%20Schmersal%20Light%20Curtains.pdf, recovered 26.09.14

[6]available: www6.in.tum.de/Main/SoftwareCORA

as described in Sec. V, with acceleration set to zero at both ends. There are four possible trajectories: *stop*, *follow desired trajectory*, *reduce speed* and *restore original speed*. The linear system has 12 dimensions and the simulation is run on MATLAB R2014b on a 2.8GHz Intel Core i7 processor with 16GB 1600 MHz DDR3 RAM running Mac OSX 10.9.4.

The parameters used for this are a Minor Cycle time of 4ms (a typical cycle time of modern robot controllers, e.g. Stäubli CS8C Controller). The dynamic and kinematic models are taken from [23]. Joint limits are taken from the Unimation PUMA 560 manual and speed and torque limits are estimated at:

$$\dot{q_i}\ (rad/sec) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right],\ \tau\ (Nm) \in \begin{bmatrix} -90 & 90 \\ -100 & 100 \\ -80 & 80 \\ -20 & 20 \\ -20 & 20 \\ -20 & 20 \end{bmatrix}$$

The uncertainty of the input is based on the modelling errors as described in Sec. III; the interval τ_{ME} is taken to be 0.1% of the interval of maximum torques given above. The gains K_p and K_v are 5 and 60 respectively, for each joint. The state, when resampled at each new Major Timestep, is a random state from within the reachable set. The human is detected by one light curtain. In equation (7), n is the x-axis unit vector and d is 0.95m, t_r is taken as 50ms and v_{max} is 2.0 ms^{-1}. $\mathcal{V}_{stationary}$ is $\pm$0.05 rad/sec on each joint.

The state space is partitioned into 11025 cells for pre-computation of the H matrices and rem vectors for the torque, friction terms are neglected. This took approximately 10 hours of computing time on the same computer as that on which the simulation is run. The simulation itself consisted of 8 pairs of trajectories, where the start position of the first was the goal position of the second and vice-versa. Four simulations were run: A and B had Major Cycle time 100ms whereas C and D had Major Cycle time 80ms. A and C verified only two trajectories (*follow desired trajectory*, *stop*) whereas B and D also used *reduce speed* and *restore original speed*. For B and D, not all trajectories were computed every timestep, e.g. *reduced speed* would only be computed if *follow desired trajectory* could not be verified safe; this reduced computation time as parallelisation was not implemented. The results are in Table I.

A wider option of trajectories in B and D appear to increase their effectiveness at following the desired trajectory although computation times have greater mean and standard deviation than A and C. If different paths were verified in parallel, the computing time need not necessarily increase with more alternative paths as long as more processors are available. A decrease of 20% in the Major Timestep appears to make a significant change to the effectiveness of the safety control. Because the Safety Control reacts quicker, human motion can be detected earlier and the reachable sets of the human during verification are smaller in C and D than A and B. Consequently, aversive action such as stopping or reducing speed need not be taken as often, and the safe trajectory can follow the desired trajectory.

TABLE I
RESULTS OF THE SIMULATION

	A	B	C	D
Average increase in total trajectory duration (trajectories successfully completed in all simulations A – D only)	64.2%	25.5%	57.9%	22.3%
Mean computation time per Major timestep (ms)	400	530	373	501
Standard deviation computation time per Major timestep (ms)	104	145	102	135
Successfully completed master trajectories (16 attempted)	11	14	12	16

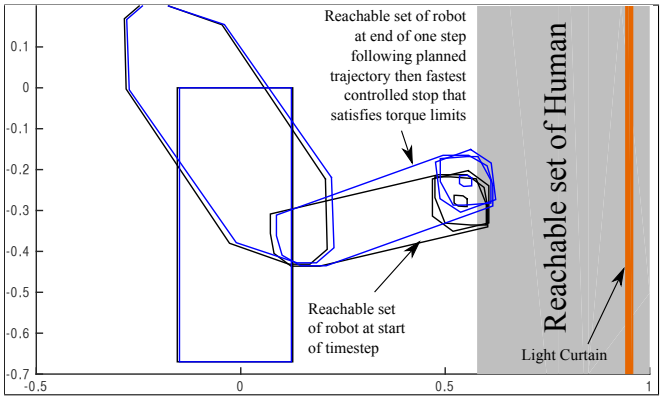

Fig. 6. Blue: Reachable set of human and of robot at start of timestep and after following intended trajectory then executing fastest controlled stop that satisfies torque limits. The robot cannot stop in time. Scale in metres.

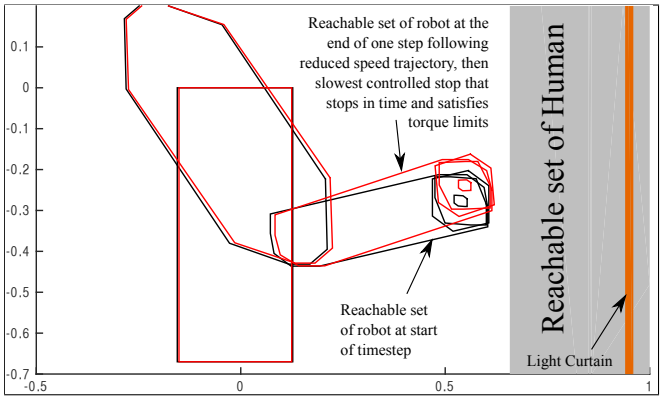

Fig. 7. Reachable sets of Human and of Robot at start of timestep and after following a safe trajectory, stopping in time. Scale in metres.

Figure 6 illustrates how the human reachable set intersects with the robot reachable set if the *follow desired trajectory* is taken; this can be avoided by following *reduce speed*, as shown in Figure 7. The video attachment to this paper demonstrates how the different simulations A, B, C and D cope with the same trajectory.

VIII. Conclusions

This paper presents a workable method for an ISO Standards-compliant safety control strategy to safeguard humans in a collaborative workspace. This algorithm is easy to implement in parallel to the existing low-level robot controller and may be used in situations where humans frequently cross into the workspace of industrial robots such as assembly lines in food production and car production to save time and space on the production line. Other benefits include introducing flexibility in the production line, as robots can be moved without restructuring the whole factory, as well as upholding standards of worker safety.

Due to the formal methods used in this approach, the algorithm automatically guarantees each movement as safe and therefore, in contrast to other safety control schemes, "certifies" its own safety automatically online.

Currently, although the offline computation of the inertia matrices and remainders has brought computation time down to typically within about 10 times the Major Cycle time, this is not yet enough to run the control online. Parallel processing, improved processors and optimisation of the code is expected to bring times within real-time. The recognition and computation of the reachable set of the human promises ample opportunity for further development. Lastly, the applicability of the algorithm described depends on the availability of an accurate dynamic model; ways to adjust for parametric uncertainty must be investigated for applicability in industrial settings.

Acknowledgement

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement number 608022.

References

[1] "Robots and robotic devices - safety requirements for industrial robots - part 1: Robots," ISO Standard 10218-1:2011, International Organization for Standardization, Geneva, Switzerland, 2011.

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

[3] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[4] A. Albu-Schäffer, S. Haddadin, C. Ott, T. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[5] R. Vatcha and J. Xiao, "Detection of robustly collision-free trajectories in unpredictable environments in real-time," *Autonomous Robots*, vol. 37, no. 1, pp. 81–96, 2014.

[6] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 338–345, 2012.

[7] M. P. Polverini, A. M. Zanchettin, and P. Rocco, "Collision avoidance in human-robot interaction based on kinetostatic safety field," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4136–4141, 2014.

[8] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1131–1136, 2013.

[9] C. Thomas, F. Busch, B. Kuhlenkoetter, and J. Deuse, "Ensuring human safety with offline simulation and real-time workspace surveillance to develope a hybrid robot assistance system for welding of assemblies," in *Enabling Manufacturing Competitiveness and Economic Sustainability* (H. A. ElMaraghy, ed.), book section 76, pp. 464–470, Springer Berlin Heidelberg, 2012.

[10] H. Ding, G. Reißig, K. Wijaya, D. Bortot, K. Bengler, and O. Stursberg, "Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5875–5880, 2011.

[11] H. Ding, K. Wijaya, G. Reißig, and O. Stursberg, "Online computation of safety-relevant regions for human robot interaction," in *Proc. of 43rd International Symposium on Robotics*, pp. 29–31, 2012.

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

[13] M. Althoff, "Formal and compositional analysis of power systems using reachable sets," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2270–2280, 2014.

[14] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2210–2215, 2005.

[15] "Safety of machinery - electrical equipment of machines," IEC Standard 60204-1:2006, International Organization for Standardization, Geneva, Switzerland, 2006.

[16] W. Chung, L.-C. Fu, and S.-H. Hsu, "Motion control," in *Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), book section 6, pp. 133–159, Springer, 2008.

[17] 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*, pp. 3559–3566, 2012.

[18] D. S. Bernstein, *Matrix Mathematics: theory, fact and formulas*. New Jersey: Princeton University Press, 2 ed., 2009.

[19] M. Althoff and B. H. Krogh, "Reachability analysis of nonlinear differential-algebraic systems," *IEEE Transactions on Automatic Control*, vol. 59, pp. 371–383, October 2013.

[20] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *International Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.

[21] N. Pedrocchi, F. Vicentini, M. Matteo, and L. Molinari, "Safe human-robot cooperation in an industrial environment," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–13, 2013.

[22] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer, 2001.

[23] P. I. Corke and B. Armstrong-Hélouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1608–1613, 1994.

[24] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

[25] S. Rump, "INTLAB - INTerval LABoratory," in *Developments in Reliable Computing* (T. Csendes, ed.), pp. 77–104, Dordrecht: Kluwer Academic Publishers, 1999. http://www.ti3.tuhh.de/rump/.