Online Motion Synthesis with Minimal Intervention Control and Formal Safety Guarantees

Martijn J.A. Zeestraten[1,4], Aaron Pereira[2,4], Matthias Althoff[2], Sylvain Calinon[3,1]

***Abstract*—We present a framework for online coordinated obstacle avoidance with formal safety guarantees. Such a formally verified trajectory planner can be used in shared human-robot workspaces to guarantee safety. The obstacle avoidance is based on estimation of the human occupancy on two different time scales. A long-term plan is created based on a probabilistic task representation, learned by demonstration, and an estimate of the human occupancy to be avoided. Using an additional overapproximative, short-term prediction of human motion we guarantee that the robot can always account for sudden or reflex movements. We demonstrate our two-level obstacle avoidance in simulation. The results show that our method reduces the number of safety stops one would encounter when using only the formal safety verification, and synthesizes alternative movement plans that preserves the coordination observed in the original demonstrations.**

I. Introduction

Programming by Demonstration (PbD) allows humans to transfer skills to robots using only a few demonstrations [1]. Small and medium enterprises that require flexible automation can benefit from such a technique, as it allows fast reprogramming without advanced programming expertise. Operating robots in such environments, however, requires safety guarantees. In previous work we showed how a robot trajectory can be formally verified online to guarantee safety of humans separated from the robot by a light curtain [2]; the robot will execute a previously verified controlled stop if the proposed trajectory is unsafe. Although the safe stops provide a formal safety guarantee, they might be avoided if the robot is allowed to deviate from its original trajectory.

Tracking a single trajectory is unnecessarily restrictive when deviations from the trajectory do not interfere with task performance. Instead, we rely on a control strategy based on the *minimal intervention principle* stating "Deviations from the average trajectory are only corrected when they interfere with task performance." [3]. In previous work [4], [5], we showed that task-space representations with a probabilistic form can be used to define task performance. This probabilistic form effectively represents the task as a distribution over trajectories instead of a single trajectory. The encoded variance of, and correlation among the state variables facilitate synthesis of alternative movements that maintain the original movement characteristics, also referred to as *coordination* patterns [6].

In this work we employ the previously proposed probabilistic task representation in [5] to generate alternative trajectories to avoid unnecessary stops caused by the formal safety verification. These alternative trajectories are obtained by intentionally deviating the robot from the average trajectory, by introducing via-points in the objective function.

Avoiding dynamic obstacles such as humans in trajectory planning gives rise to several approaches. Distance metrics are used to scale robot velocity while preserving the spatial path so that the robot slows down as the human gets closer [7]. Dynamic Road Maps (DRMs) [8] precompute a graph of configurations, and paths are quickly planned by graph traversal of non-colliding nodes. Mainprice and Berenson [9] predict human motion to compute occupancy probabilities in task space, used by a trajectory optimiser to plan a non-colliding path.

Exploiting task representations to quickly generate alternative movement plans can lead to high-performing and versatile algorithms. The Dynamic Movement Primitives (DMP) framework has multiple extensions that involve obstacle avoidance. Park et al. [10] use a dynamic potential field that considers the dynamics of the obstacle to achieve smoother avoidance behavior compared to *static* potential fields. Krug and Dimitrov use the DMP framework in a Model Predictive Control (MPC) approach [11], where obstacles are avoided by including them as a constraint in an optimization problem. However, since DMP does not encode correlation among state variables, coordinated recovery of disturbances that interfere with the task performance is not possible.

Khansari and Billard [12] propose a local obstacle avoidance technique for autonomous dynamical systems (DS), which modulates the DS such that it avoids dynamic obstacles, while conserving the stability properties of the original DS. In contrast to DMP, this approach can recover from avoidance in a coordinated way, but the complexity of the motions that can be encoded is limited.

Although the methods described above are all effective, none of them provides a formal guarantee on safety. Two main ways to guarantee safety in shared workspaces exist [13]: ensuring non-contact, and limiting force and power within contacts. We focus on the former, though our approach may be adapted to the latter. Piecewise planning which formally guarantees noncollision is proposed by Petti and Fraichard for mobile robots [14] and adapted to serial-link robots in [2].

Our goal is to control a robot to complete a previously

[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy. martijn.zeestraten@iit.it, sylvain.calinon@iit.it

[2] Department of Informatics, Technische Universität München, 85748 Garching, Germany. aaron.pereira@tum.de, althoff@in.tum.de

[3] Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland.

[4] These authors contributed equally to this work.

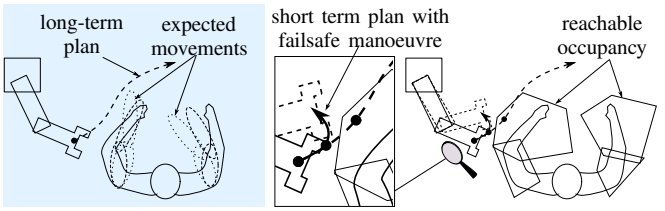

Fig. 1: (a) Long-term planning around expected human movement; (b) short-term failsafe manoeuvre in case of unexpected motion.

demonstrated task while *formally guaranteeing* that the trajectory used to complete the task is safe at all times. We define the terms *Reachable Occupancy* and *Expected Occupancy* of the human and *Safe Trajectory* of the robot:

Definition 1 (Expected and Reachable Occupancy)**.** *Let $\mathcal{Y}(t, \tilde{u}) \subset \mathbb{R}^3$ be the subset of Cartesian space which the human occupies at time t, following a trajectory $\tilde{u}$. The reachable occupancy at time t is $\Gamma^r(t) = \{\mathcal{Y}(t, \tilde{u}) | \tilde{u} \in \tilde{\mathcal{U}}\}$ where $\tilde{\mathcal{U}}$ is the set of all possible trajectories. Expected occupancy $\Gamma^e(t) = \{\mathcal{Y}(t, \tilde{u}) | \tilde{u} \in \tilde{\mathcal{U}}^e\} \subseteq \Gamma^r(t)$ contains the most likely trajectories, $\tilde{\mathcal{U}}$.*

Definition 2 (Safe Trajectory)**.** *We define a robot trajectory as a function $\boldsymbol{\xi} : [0, \infty) \rightarrow \mathbb{R}^n$, where $\boldsymbol{\xi}(t) \in \mathbb{R}^n$ is the state of the robot at time t. Let $A(\boldsymbol{\xi}) \subset \mathbb{R}^3$ be the spatial occupancy of the robot. We call a robot trajectory $\boldsymbol{x}$ safe when $\forall t > 0 : A(\boldsymbol{\xi}(t)) \cap \Gamma^r(t) = \emptyset \vee \dot{\boldsymbol{\xi}}(t) = \mathbf{0}$ holds.*

In other words, if the robot is not stationary, the robot must not collide with the human. Such a measure of safety may be used to comply with workplace standards in areas where robots work with humans [15]. In this first work we only consider avoiding collision between the robot end-effector and the human arm — one of the fastest moving parts of the human body. Methods for quickly calculating occupied volumes of kinematic chains over a range of joint positions (e.g., [16]) can be used to extend the proposed method to collision-checking for the entire robot. The method used to model the human arm (see Sec. III-A) can be straightforwardly extended to the entire body.

The novelty of the proposed approach lies in the combination of long-term and short-term plans. As humans move fast and unpredictably, the reachable occupancy grows fast and would unnecessarily restrict movement if we were to plan the trajectory around it. Hence we take a two-step approach: A long term plan (Fig. 1a) is generated from the probabilistic task encoding, by avoiding expected human motion. A short-term plan (Fig. 1b) consists of the first part of the long-term plan, followed by a failsafe stopping manoeuvre, which has been verified safe. If the next part of the long-term plan is not verified safe, the failsafe trajectory is executed. Both plans are continuously updated.

The paper is structured as follows: in the next section we describe the proposed approach. The methods are detailed in Sec. III. We present the results of a simulation in Sec. IV, and discuss and conclude in Sec. V.

II. Approach

Algorithm 1 Formally verified obstacle avoidance

1: ▷ Initialization:
2: $\boldsymbol{u}_{1:N_p}, \tilde{\boldsymbol{\xi}}_{1:N_p} \leftarrow$ predictControl($\boldsymbol{\xi}_0$, $\emptyset$)
3: $\bar{\boldsymbol{u}}_{0:\infty} \leftarrow \mathbf{0}$
4: $k \leftarrow 0$
5: **while** not at end of motion **do**
6: $\quad \boldsymbol{\gamma}_k \leftarrow$ getHumanState()
7: $\quad$ ▷ Verify control for next iteration:
8: $\quad \bar{\boldsymbol{u}}_{k+1:\infty} \leftarrow$ verify($\bar{\boldsymbol{u}}_{k:\infty}, \boldsymbol{u}_{k+1}, \boldsymbol{\gamma}_k, \boldsymbol{\xi}_k$)
9: $\quad$ ▷ Prediction:
10: $\quad \boldsymbol{p} \leftarrow$ generateViaPoints($\boldsymbol{\gamma}_k$, $\tilde{\boldsymbol{\xi}}_{k+(1:N_p)}$)
11: $\quad \boldsymbol{u}_{k+(2:(N_p+1))}, \tilde{\boldsymbol{\xi}}_{k+(2:(N_p+1))} \leftarrow$ predictControl($\tilde{\boldsymbol{\xi}}_{k+1}, \boldsymbol{p}$)
12: $\quad$ ▷ Apply command verified previous timestep:
13: $\quad \boldsymbol{\xi}_{k+1} \leftarrow$ applyControl($\bar{\boldsymbol{u}}_k$)
14: $\quad k \leftarrow k + 1$

Algorithm 1 explains our approach. $\boldsymbol{\xi}_k$ is the state of the system and $\boldsymbol{\gamma}_k$ is the state of the human at discrete time k, or a sequence of time steps $k_1 : k_2$. Estimated states are indicated by $\tilde{\boldsymbol{\xi}}$. Verified control inputs $\boldsymbol{u}_k$ are indicated by $\bar{\boldsymbol{u}}_k$. $\boldsymbol{p}$ is the vector of via-points that are generated to avoid obstacles. N_p is the prediction horizon.

At each time step k our method:

- verifies safety of the new short-term plan $\boldsymbol{u}_{k+1:\infty}$ (input $\boldsymbol{u}_{k+1}$ plus stop) using an overapproximative model of human behaviour (line 8) and updates the control input,
- generates inputs $\boldsymbol{u}_{k+(2:(N_p+1))}$ [1] taking into account predicted occupancy of the human (lines 10-11),
- executes the control command $\bar{\boldsymbol{u}}_k$, and observes the state of the system to be used in the computations of the next time step. (line 13).

These three steps may be executed in parallel since their inputs and outputs are independent. The verification of the subsequent time step provides a formal safety guarantee on a short horizon because no control command is executed without being previously verified. In case the verifier fails to verify $\boldsymbol{u}_{k+1}$, the previously verified failsafe manoeuver $\bar{\boldsymbol{u}}_{k+1:\infty}$ guarantees a safe stop. To guide the trajectory around the human and hence minimise the chance that a trajectory will be determined to be unsafe, we also check if the *expected* human occupancy collides with the long term motion plan $\tilde{\boldsymbol{\xi}}_{k:(1+N_p)}$. If collision is detected, we generate a via-point $\boldsymbol{p}$ that is taken into account in the control input prediction (lines 10-11). Details on human occupancy and verification of its intersection by the robot are given in Secs. III-A and III-B, respectively.

The robot motion is encoded in a Hidden Semi-Markov Model (HSMM), providing a probabilistic movement representation. This representation allows reproduction in which

[1] Here, a scalar added to a vector index e.g. $k + (2 : (N_p + 1))$ is taken to mean $(k + 2) : (k + N_p + 1)$

TABLE I: Acceleration parameters used in both models, ms^{-2}.

	Overapproximative ($\Gamma^r(t)$)	Expected $\Gamma^e(t)$
Hand	189	10
Elbow	163	5

deviations from the desired path are only corrected when they interfere with task performance, as demonstrated in previous work [4], [5]. Secs. III-C and III-D describe the learning and reproduction of the task in detail.

III. Methods

A. Human Arm Expected and Reachable Occupancies

The human arm occupancy $\Gamma(t)$ must be quick to calculate, easy to collision-check, and conservative for the case of the formal verification. In [17], a kinematic model of the human arm is used to create a set of enclosing swept volumes, though complexity grows exponentially with the number of degrees of freedom of the kinematic model, increasing collision-checking time.

In our prediction, we consider the arm as two rigid bodies: the *upper arm* from shoulder (S) to elbow (E) and the *forearm* from elbow (E) to hand (H). Reachable occupancies of points S, E and H are calculated from the sensor data of position and velocity found online (in our case, infrared motion capture of markers on shoulder, elbow and wrist), and a set of accelerations determined offline. Both expected and (overapproximative) reachable occupancies are based on the same model; the former uses empirically chosen expected accelerations whereas the latter uses maximum accelerations from analysis of motion capture data. 38 subjects aged 18–49 performed punching, sideways and upward-sweeping motions as fast as possible to capture maximum forward, lateral and vertical accelerations, see Tab. I.

We show now how the arm occupancy is calculated. Let $B(r) = \{\boldsymbol{z} \in \mathbb{R}^3 | \, \|\boldsymbol{z}\|_2 \leq r\}$. We call the set in $\mathbb{R}^3$ in which a point $\boldsymbol{y} \in \mathbb{R}^3$ after time t may be located the *reachable set* $\mathcal{R}_{\boldsymbol{y}}(t)$, which is:

$$\mathcal{R}_y(t) = \boldsymbol{y}(0) \oplus B(\delta y) \oplus (\dot{\boldsymbol{y}}(0) \oplus B(\delta \dot{y})) \cdot t \oplus B(\frac{a_{\max}}{2} \cdot t^2),$$

where $\boldsymbol{y}(0)$ and $\dot{\boldsymbol{y}}(0)$ are the initial position and speed of the point, δy and $\delta \dot{y} \in \mathbb{R}$ are maximum measurement uncertainties of position and velocity respectively, $a_{\max} \in \mathbb{R}$ is the maximum acceleration magnitude and $\oplus$ is the Minkowski sum defined over sets G and H as $G \oplus H = \{g+h \,|\, g \in G, h \in H\}$. We enclose the reachable set of time interval $[t_a, t_b]$, denoted by $\mathcal{R}_y([t_a, t_b])$, in a sphere enclosing $\mathcal{R}_y(t_a)$ and $\mathcal{R}_y(t_b)$. We omit the proof that the sphere encloses the reachable set of the interval for brevity. The occupancy of the forearm $\mathcal{R}_F$ and upper arm $\mathcal{R}_U$ are capsules enclosing the convex hull (CH) of $\mathcal{R}_H$ and $\mathcal{R}_E$, and of $\mathcal{R}_E$ and $\mathcal{R}_S$, respectively. $\mathcal{R}_F$ and $\mathcal{R}_U$ are extended by the maximum length of a human hand, $0.205m$ (from [18]) and the estimated maximum radius of the upper arm with clothes, $0.08m$, respectively. $\mathcal{R}_U$ and $\mathcal{R}_F$ are capsules, which are fast to collision-check as in [19], see Fig. 2.

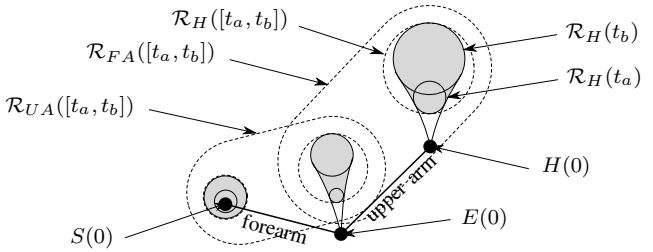

Fig. 2: Occupancy of the human arm for time interval $[t_a, t_b]$. For the hand, the reachable set of the interval encloses the reachable sets from t_a to t_b.

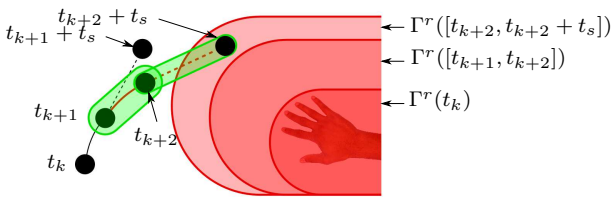

Fig. 3: Verifying safety of a manoeuvre. The desired trajectory during time interval $[t_{k+1}, t_{k+2}]$ is determined to be unsafe, as the subsequent safe stop intersects the reachable occupancy. Hence, the safe stop verified in the previous timestep is executed.

$$\begin{aligned}
\mathcal{R}_F([t_a, t_b]) &= \text{CH}(\mathcal{R}_H([t_a, t_b]), \mathcal{R}_E([t_a, t_b])) \oplus B(0.205), \\
\mathcal{R}_U([t_a, t_b]) &= \text{CH}(\mathcal{R}_S([t_a, t_b]), \mathcal{R}_E([t_a, t_b])) \oplus B(0.08), \\
\Gamma([t_a, t_b]) &= \mathcal{R}_F([t_a, t_b]) \cup \mathcal{R}_U([t_a, t_b]).
\end{aligned} \tag{1}$$

To generate the via points in line 10 of Alg. 1, we check whether human and robot intersect at specific times in the future. If so, we find the plane on which both forearm and upper arm in $\Gamma^e([t_a, t_b])$ lie. We then move any intersecting points normal to this plane in the direction from which the end-effector approaches, until they lie just outside the boundary of the expected occupancy.

By abstracting the human body to a set of rigid links enclosed in capsules, this method can be extended to the whole body of one or more humans, where the complexity grows linearly with the number of capsules.

B. Formal Verification of Robot Trajectory

We now consider verifying the short-term plans, shown in Fig. 3. During time interval $[t_k, t_{k+1}]$, we verify whether the trajectory for $[t_{k+1}, t_{k+2}]$ and a subsequent controlled stop of duration t_s will intersect the reachable occupancies in their respective time intervals. If no intersection occurs, the control input $\boldsymbol{u}_{k+1}$ at t_{k+1} is *verified*. Otherwise, the control input $\bar{\boldsymbol{u}}_{k+1}$ — part of the failsafe manoeuvre verified in the previous time step — is applied as shown in Fig. 3. In our implementation, input is the acceleration of the robot, which is constant over a time step and during a controlled stop.

Proposition 1. *The centre of a sphere of radius r under constant acceleration $\boldsymbol{a}$ has position $\boldsymbol{s}(\tau)$ at time τ. Then*

the Minkowski sum of line segment L from $\boldsymbol{s}(0)$ to $\boldsymbol{s}(t)$ and a sphere of radius $\frac{|\boldsymbol{a}|}{8}t^2 + r$ bounds the set $\{\boldsymbol{s}(\tau)|\ 0 \leq \tau \leq t\}$.

Proof. Let $\boldsymbol{v}$ be the velocity at time 0. Then:

$$\boldsymbol{s}(\tau) = \boldsymbol{s}(0) + \boldsymbol{v}\tau + \frac{\boldsymbol{a}}{2}\tau^2. \tag{2}$$

Where $0 \leq \tau \leq t$, the point

$$\boldsymbol{s}^*(\tau) = \boldsymbol{s}(0)\frac{t-\tau}{t} + \boldsymbol{s}(t)\frac{\tau}{t} \tag{3}$$

lies on L. Substituting $\boldsymbol{s}(t) = \boldsymbol{s}(0)+\boldsymbol{v}t+\frac{\boldsymbol{a}}{2}t^2$ into (3), subtracting (2) and rearranging, we obtain $\boldsymbol{s}^*(\tau) - \boldsymbol{s}(\tau) = \frac{\boldsymbol{a}}{2}\tau(t-\tau)$. The magnitude of this is a maximum at $\tau = \frac{t}{2}$, hence the distance from $\boldsymbol{s}(\tau)$ to the point $\boldsymbol{s}^*(\tau)$ on L is no greater than $\frac{|\boldsymbol{a}|}{8}t^2$, to which we add r to account for the moving sphere's radius. □

C. Probabilistic Movement Encoding

The robot is programmed by providing N demonstrations of the skill to transfer. Each demonstration consists of T_n data points $\boldsymbol{\xi_t} = [\boldsymbol{x}_t^\top, \dot{\boldsymbol{x}}_t^\top]^\top$, where $\boldsymbol{x}_t$ and $\dot{\boldsymbol{x}}_t$ are the robot position and velocity in task space, $n \in \{1, \dots, N\}$ and $t \in \{1, \dots, T_n\}$.

The demonstrated data are encoded in a Hidden Semi-Markov Model (HSMM) [20], an extension of the Hidden Markov Model (HMM) in which the state self-transistion probabilities a_{ii} are set to zero, and state duration is explicitly modeled as a probability distribution. In this work we model this duration using a Gaussian $\mathcal{N}(\mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}})$ defined by a duration mean $\mu_i^{\mathcal{D}}$ and variance $\Sigma_i^{\mathcal{D}}$. Each hidden state is represented by a single multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ encoding the local movement dynamics with the mean and covariance defined as

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_{i,\boldsymbol{x}} \\ \boldsymbol{\mu}_{i,\dot{\boldsymbol{x}}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_{i,\boldsymbol{x}\boldsymbol{x}} & \boldsymbol{\Sigma}_{i,\boldsymbol{x}\dot{\boldsymbol{x}}} \\ \boldsymbol{\Sigma}_{i,\dot{\boldsymbol{x}}\boldsymbol{x}} & \boldsymbol{\Sigma}_{i,\dot{\boldsymbol{x}}\dot{\boldsymbol{x}}} \end{bmatrix},$$

respectively. Summarizing, an HSMM with K states is defined by the parameters $\{a_{i,j}, \Pi_i, \mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i,j}^K$, with Π_i the state priors. These parameters are estimated from the demonstration data by Expectation Maximization (EM) using an efficient Forward-Backward algorithm [21].

D. Control Prediction

The control commands are computed using linear unconstrained Model Predictive Control (MPC) [22]. Similarly to our previous work [5], we consider a quadratic cost function in both system state and control input, namely

$$J = \sum_{r=k+1}^{k+N_p+1} \big(\hat{\boldsymbol{\xi}}_r - \boldsymbol{\xi}_r\big)^\top \boldsymbol{Q}_r \big(\hat{\boldsymbol{\xi}}_r - \boldsymbol{\xi}_r\big) \ + \sum_{r=k+1}^{k+N_p} \boldsymbol{u}_r^\top \boldsymbol{R}_r\, \boldsymbol{u}_r,$$

and a discrete linear system $\boldsymbol{\xi}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}_k + \boldsymbol{B}\boldsymbol{u}_k$.

The task specifying parameters of the cost function are generated from the probabilistic model of the movement and the newly introduced via-points $\boldsymbol{p}_h$:

$$\hat{\boldsymbol{\xi}}_r = \begin{cases} \boldsymbol{p}_h, & \text{if } r = h \\ \boldsymbol{\mu}_{s_r}, & \text{otherwise} \end{cases} \qquad \boldsymbol{Q}_r = \begin{cases} \begin{bmatrix} c_x \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & c_{\dot{x}} \boldsymbol{I} \end{bmatrix}, & \text{if } r = h \\ (\boldsymbol{\Sigma}_{s_r})^{-1}, & \text{otherwise} \end{cases}$$

where s_r indicates the index of the activated state at time r, and h indicates a time step on the control horizon. The list of indices $[s_{k+2}, \dots, s_{k+N_p+2}]$ is generated from the HSMM transition model using a procedure detailed below. The importance of the via-point is controlled through the tracking costs c_x and $c_{\dot{x}}$ representing the tracking cost on the desired position and velocity of the via-point.

The cost is optimized using the standard Riccati equations for a tracking problem (see e.g. [23]),

$$\begin{aligned} \boldsymbol{P}_r &= \boldsymbol{Q}_r - \boldsymbol{A}^\top (\boldsymbol{P}_{r+1}\boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{P}_{r+1}\boldsymbol{B} + \boldsymbol{R})^{-1} \\ &\qquad \boldsymbol{B}^\top \boldsymbol{P}_{r+1} - \boldsymbol{P}_{r+1})\boldsymbol{A} \\ \boldsymbol{d}_r &= (\boldsymbol{A}^\top - \boldsymbol{A}^\top \boldsymbol{P}_{r+1}\boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{P}_{r+1}\boldsymbol{B} + \boldsymbol{R})^{-1}\boldsymbol{B}^\top) \\ &\qquad (\boldsymbol{P}_{r+1}(\boldsymbol{A}\hat{\boldsymbol{\xi}}_r - \hat{\boldsymbol{\xi}}_{r+1}) + \boldsymbol{d}_{r+1}), \end{aligned}$$

using initial conditions $\boldsymbol{P}_{N_p+1} = \boldsymbol{Q}_{N_p+1}$ and $\boldsymbol{d}_{N_p+1} = \boldsymbol{0}$. $\boldsymbol{P}_r$ and $\boldsymbol{d}_r$ are then used to compute the control inputs $\boldsymbol{u}_{k+(2:(N_p+1))}$ and the state predictions $\tilde{\boldsymbol{\xi}}_{k+(2:N_p+1)}$, using forward integration

$$\tilde{\boldsymbol{\xi}}_{r+1} = \boldsymbol{A}\tilde{\boldsymbol{\xi}}_r + \boldsymbol{B}\boldsymbol{u}_r, \qquad \boldsymbol{u}_r = \boldsymbol{K}_r(\hat{\boldsymbol{\xi}}_r - \tilde{\boldsymbol{\xi}}_r) + \boldsymbol{f}_r$$

with feedback gain and feedforward terms defined as

$$\begin{aligned} \boldsymbol{K}_r &= (\boldsymbol{B}^\top \boldsymbol{P}_r \boldsymbol{B} + \boldsymbol{R})^{-1} \boldsymbol{B}^\top \boldsymbol{P}_r \boldsymbol{A}, \\ \boldsymbol{f}_r &= -(\boldsymbol{B}^\top \boldsymbol{P}_r \boldsymbol{B} + \boldsymbol{R})^{-1} \boldsymbol{B}^\top (\boldsymbol{P}_r(\boldsymbol{A}\hat{\boldsymbol{\xi}}_r - \hat{\boldsymbol{\xi}}_r) + \boldsymbol{d}_r). \end{aligned}$$

The construction of the objective function used in our approach is based on a state sequence $\boldsymbol{s} = \{s_1, \dots, s_{N_p}\}$, that is regenerated at each time step of the reproduction. This process relies on the forward variable $\alpha_{i,k}$ of the HSMM. It defines the probability of being in state i at time step k given the observation $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_k\}$, i.e. $\mathcal{P}(i|\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_k)$, and is recursively computed with (see e.g. [24])

$$\alpha_{i,k} = \Pi_i \mathcal{N}_{k,i}^{\mathcal{D}} \prod_{r=1}^{k} \mathcal{N}_{r,i} + \sum_{j=1}^{K} \sum_{d=1}^{k-1} \alpha_{j,t-d}\, a_{j,i}\, \mathcal{N}_{d,i}^{\mathcal{D}} \prod_{r=k-d+1}^{k} \mathcal{N}_{r,i},$$

when k is smaller than the time history d_{max}, otherwise

$$\alpha_{i,k} = \sum_{d=1}^{d_{max}} \sum_{j=1}^{K} \alpha_{j,k-d}\ a_{j,i}\ \mathcal{N}_{d,i}^{\mathcal{D}} \prod_{r=k-d+1}^{k} \mathcal{N}_{r,i},$$

with $\mathcal{N}_{r,i} = \mathcal{N}\big(\boldsymbol{\xi}_r|\ \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\big)$ and $\mathcal{N}_{d,i}^{\mathcal{D}} = \mathcal{N}(d|\mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}})$.

At each time step the forward variable is used for two purposes. First, to keep track of the probability $\mathcal{P}(i|\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_k)$. Here, $\alpha_{i,k}$ is computed while taking into account the current system state $\boldsymbol{\xi}_k$. This process is initialized with the priors, i.e. $\alpha_{i,0} = \Pi_i$. Second, N_p predictions are computed to create the state sequence prediction $\boldsymbol{s}$ with

$$s_r = \underset{i \in \{1, \cdots, K\}}{\arg\max}\ \alpha_{i,r},\ \forall\ r \in \{k+1, k+2, \cdots, k+N_p\}.$$

When computing the predictions, we assume that the probability of $\boldsymbol{\xi}_r$ being in a given state is 1 for all states, i.e. $\mathcal{N}_{r,i} = 1 \forall i$

When a safety stop is active — because the action $\boldsymbol{u}_k$ was not verified to be safe — one can either continue or pause

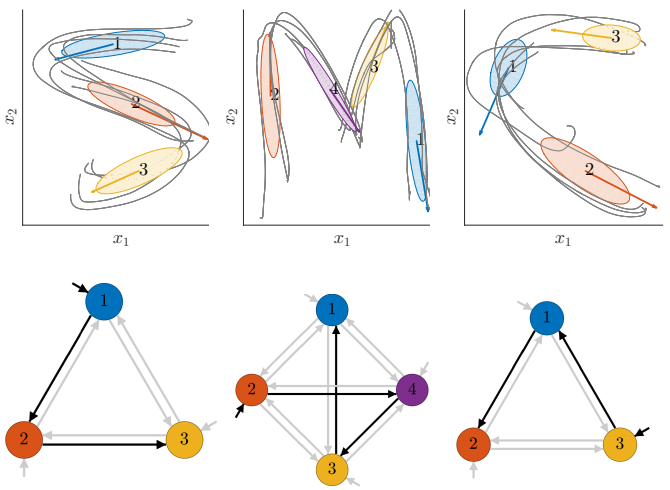

Fig. 4: Visualization of the HSMM models used to verify the proposed approach. Each column represents the model of one letter. *Top row:* 2D representation of the 3D-letter models used, gray lines indicate (partial) demonstrations, each coloured ellipsoid represents a Gaussian (boundary is one standard deviation, colours correspond to transition in the second row), arrows originating from the centers represent the mean velocity encoded by that Gaussian. *Bottom row:* The state transition duration model. Interconnecting arrows represent between state transition probability, outer arrows indicate state prior probability. The colour intensity of the arrows indicate magnitude of the probability.

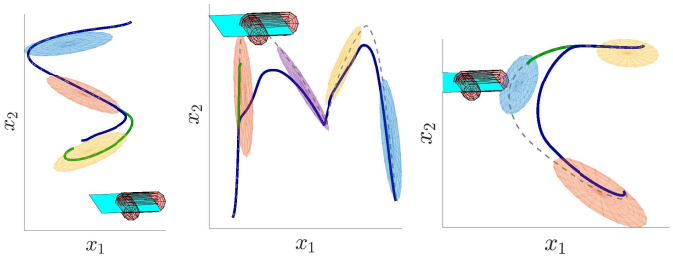

Fig. 5: Movement reproduction results for each of the letter models. Each trajectory represents a different test case: C0 (grey-dotted), C1 (green) and C2 (blue). The location of the human arm, and the plane used to generate the via-points are visualized by the red capsules intersected with the cyan plane. The 3D ellipsoids represent the same Gaussians as presented in Fig. 4, with corresponding colours.

the update of the forward variable. In the former case, the temporal behavior encoded in the HSMM is respected; The predicted state sequence will evolve, thereby omitting part of the trajectory to ensure that the system reaches the final state in time. In the latter case, such re-planning will not occur, and the system will continue the trajectory once the robot resumes normal operation. We choose the latter to ensure reproduction of the original movement.

TABLE II: Number of safety stops observed for different experimental conditions: no interference (C0), formal verification (C1), formal verification & replanning (C3).

	S	M	C
C1	0	128	18
C2	0	0	0

IV. Evaluation

We evaluate the approach on a 3D data set containing free-air drawing motions of the letters S, M and C captured, filtered and scaled from a Leap Motion Controller.[2] The demonstration data consists of a combination of full and partial demonstrations: S:(1,6), M:(1,6) C:(3,3), (full, partial). The movement dynamics are encoded in K Gaussians forming an HSMM, where K is set empirically. The EM procedure is repeatedly initialized using the K-means algorithm while keeping the model with highest log-likelihood. Fig. 4 shows $2D$ projections of the models obtained. In the reproductions the trajectory is performed close to a moving arm. The arm movement is from the publicly available CMU Graphics Lab Motion Capture Database[3].

Each letter is reproduced under three different conditions: C0) Normal reproduction without considering the human, C1) Reproduction with avoidance based on the formal safety verification, C2) Reproduction with avoidance based on the formal safety verification with additional obstacle avoidance. We empirically set the prediction horizon to $N_p = 50$, the cost matrix $\boldsymbol{R} = 1e{-}8\,\boldsymbol{I}$ and the via-point strength $c_x = 2e-3$, $c_{\dot{x}} = 0$. The time step is $16.7ms$; potential via-points were calculated at time steps 10 and 20 of the prediction horizon. A reproduction is terminated when the HSMM reaches the final state and the observed velocity is zero.

The resulting trajectories are visualized in Fig. 5 and information on the number of safety stops is summarized in Tab. II. A clear difference between the different conditions can be seen for the letters M and C. Although the via point introduced in C2 deforms the letters, the curvature remains consistent with the shape of the original trajectory, because the motion synthesis respects the encoded movement coordination. The reproduction of C1 fails since it is stopped by the verifier, and no safe control command was found before terminal conditions of the experiments were met.

The reproduction of the letter S shows that C2 does not always outperform C1: during reproduction, C2 seemed to be too cautious in planning via-points to avoid the human, while the formal verification did not require to stop once.

Tab. III displays mean, average, maximum and minimum computation times required for the verification and the prediction step. These values correspond to a simulation of the experiment on a computer with a 2.5GHz i7 processor and 16GB RAM using MATLAB. In our implementation the

[2] `https://www.leapmotion.com/`, retrieved April 15, 2016

[3] Subject 80, movement 69 "Painting" `http://mocap.cs.cmu.edu`, retrieved August 1, 2015

TABLE III: Overview of computation times $[ms]$ (n=1299)

	μ_t	σ_t	t_{max}	t_{min}
Verification	0.19	0.08	1.02	0.15
Prediction/Optimization	5.50	0.38	11.60	5.28

prediction and verification were performed sequentially, but they can be performed in parallel as previously discussed in Sec. II. There are no iterative optimisation or collision-checking steps in the method, hence computation time is deterministic and suitable for real-time application.

The correctness of this approach depends on the human models used being conservative. We account for reflex movements by determining the dynamic parameters of the human from data of unrestricted human movements executed as fast as possible. Such movements may happen in industrial scenarios, for example when a worker touches something hot or sharp, or deliberately tries to 'trick' the robot.

The effectiveness of this approach depends on the capability of the sensors. The sensor data in our experiment is updated with $60Hz$, however, industrial safety-certified camera systems[4] have lower frame rates and higher latency.

V. Conclusion & Future Work

We presented an approach that combines probabilistic task representations with formal methods to achieve formally safe coordinated obstacle avoidance. In both scenarios C1 and C2, safety is guaranteed, but where the trajectory is replanned through via points, no unnecessary stopping occurs. The formal verifier provides a closed form alternative to the usage of hard contraints to guarantee obstacle avoidance. This reduces the computation required for trajectory optimization, while guaranteeing avoidance of the robot and maintaining coordination patterns found in the demonstration data.

In future work we plan to improve the proposed method by improved selection of the via-points and their strength parameters ($c_{\boldsymbol{x}}$ and $c_{\dot{x}}$), and verify the approach in a real robot-human scenario. Additionally, we may consider a different safety criterion to Def. 2, such as limiting impact energy specified in [25]. To the authors' best knowledge, this work is the first to combine the ease of imitation learning with formal safety guarantees for the human.

ACKNOWLEDGEMENT

The authors are grateful to Jonas Schmidtler, Asuman Sezgin and Thomas Illa who set up and captured the motion data. The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's 7th Framework Programme FP7/2007-2013/ under REA grant agreement number 608022. The comparison data used in this project was obtained from `mocap.cs.cmu.edu`; this database was created with funding from NSF EIA-0196217. The authors also gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921.

[4]e.g. Safe camera system SafetyEYE, Pilz GmbH & Co. KG

References

[1] A. G. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 1371–1394, Springer, 2008.

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

[3] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nat. Neurosci.*, vol. 5, pp. 1226–1235, 2002.

[4] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. IEEE Intl Conf. Robotics and Automation*, pp. 3339–3344, 2014.

[5] M. J. A. Zeestraten, S. Calinon, and D. G. Caldwell, "Variable duration movement encoding with minimal intervention control," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 497 – 503, 2016.

[6] M. T. Turvey, "Coordination," *Am. Psychol.*, vol. 45, no. 45, pp. 938–953, 1990.

[7] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, 2016.

[8] M. Kallman and M. Mataric, "Motion planning using dynamic roadmaps," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4399–4404, 2004.

[9] 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*, pp. 299–306, 2013.

[10] D. Park, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *IEEE Int. Conf. on Humanoid Robots*, pp. 91–98, 2008.

[11] R. Krug and D. Dimitrov, "Model predictive motion control based on generalized dynamical movement primitives," *J. Intell. & Robotic Syst.*, vol. 77, no. 1, pp. 17–35, 2015.

[12] S. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

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

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

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

[16] 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*, pp. 1585–1592, 2011.

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

[18] S. Pheasant and C. M. Haslegrave, *Bodyspace: Anthropometry, Ergonomics and the Design of Work*, ch. 6, pp. 143–160. Taylor & Francis, 2006.

[19] C. Ericson, *Real-Time Collision Detection*. Morgan Kaufmann, 2004.

[20] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, pp. 215–243, 2010.

[21] S.-Z. Yu and H. Kobayashi, "Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden Markov model," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1947–1951, 2006.

[22] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for linear and hybrid systems*. 2015. available: mpc.berkeley.edu/mpc-course-material.

[23] A. E. Bryson, *Dynamic optimization*. Addison Wesley Longman, 1999.

[24] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[25] "Robots and robotic devices – collaborative robots," ISO/TS Standard 15066:2016, International Organization for Standardization, Geneva, Switzerland, 2016.