



# Trajectory optimisation in collaborative robotics based on simulations and genetic algorithms

Andrea Maria Zanchettin<sup>1</sup> · Costanza Messeri<sup>1</sup> · Davide Cristantielli<sup>1</sup> · Paolo Rocco<sup>1</sup>

Received: 1 May 2021 / Accepted: 4 May 2022 / Published online: 30 May 2022  
© The Author(s) 2022

## Abstract

The separation distance between humans and robots in manufacturing shop-floors has been progressively reduced, thanks to the modern safety functionalities available in robot controllers. However, the activation of these safety criteria usually stops the production or reduces the productivity of machines and robots. With the aim of improving this situation, this paper presents a real-time trajectory optimisation method for collaborative robots. The robot trajectory is parameterised at instruction level, i.e. through the parameters characterizing the robot motion instruction. A genetic algorithm randomly modifies the parameters of the nominal trajectory of the robot, thus obtaining new sets of candidate trajectories. Each trajectory is simulated on a digital twin of the collaborative workspace, which allows to reproduce and simulate the robot motion, and to represent the volume of the work-cell occupied by the human operator. A lexicographic optimization is used to evaluate online the optimal robot trajectory that simultaneously minimizes the risk of collision with the human operator and the trajectory traversal time. The method is validated in an industrial scenario involving the ABB YuMi dual-arm robot for a small parts assembly task.

**Keywords** Collaborative robotics · Intelligent and flexible manufacturing · Trajectory optimization · Genetic algorithm · Collision avoidance

## 1 Introduction

Collaborative robots are endowed with safety functionalities allowing them to safely share the workspace with human workers. ISO Technical Specification 15,066 (ISO TC184/SC2 2013) introduces two criteria to enforce safety in human-robot coexistence: Speed and Separation Monitoring (SSM) and Power and Force Limiting (PFL). The former prescribes how to maintain a minimum safety distance between the robot and the human operator, so as to ensure

cooperative human-robot coexistence while avoiding harmful contacts (Hamid et al. 2017).

The latter allows the robot to work even in close proximity with a reduced energy, so that a possible impact, which will eventually stop its motion, would not cause any injury to the human. No matter which safety criterion is implemented by the robot controller, the possibility of executing trajectories that minimise the probability of triggering a safety countermeasure is of paramount importance to guarantee the highest level of productivity. In fact, the two safety criteria force the robot to slow down or to eventually stop along the path, thus reducing its efficiency. For this reason, researchers are looking for effective models to describe the human behaviour in the shop-floor with the ultimate goal of using this information to plan optimal trajectories for the robot.

The problem of adapting the robot trajectories to dynamic environments is a longstanding one, (Brock and Khatib 2000). In the field of Human-Robot Collaboration (HRC), researchers are trying to include prediction of human motion in the trajectory generation or adaptation, (Balan and Bone 2006; Ragaglia et al. 2015; Koppula and Saxena 2016;

---

✉ Costanza Messeri  
costanza.messeri@polimi.it

Andrea Maria Zanchettin  
andreamaria.zanchettin@polimi.it

Davide Cristantielli  
davide.cristantielli@mail.polimi.it

Paolo Rocco  
paolo.rocco@polimi.it

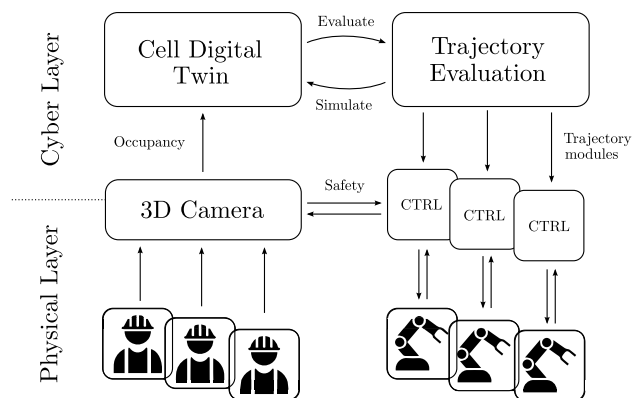
<sup>1</sup> Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Lasota et al. 2017; Zanchettin and Rocco 2017; Zanchettin et al. 2019).

When it comes to use models of human motion in trajectory planning, several examples can be reported. For example (Ziebart et al. 2009) used a navigation strategy to avoid pedestrians. Ding et al. (2011) adopted a Mixed Integer Linear Programming (MILP) algorithm to avoid collisions. The work in Mainprice et al. (2016) minimises the intersection of robot and human workspaces. Pellegrinelli et al. (2016) uses a probabilistic representation of human occupancy to select a trajectory within an offline generated set. The work by Pereira et al. (2018) adopts reachable sets gathered from human archetypal movements and uses them online in safety-oriented strategies such as stopping the robot or performing a local avoidance manoeuvre. By interleaving online planning and execution, the work in Unhelkar et al. (2018) proposed an adaptation strategy for robot trajectories to avoid humans and minimise the stopping time of the robot. Similarly to (Lasota et al. 2017; Wang et al. 2018) adopts time series to predict human motion and develops a local avoidance strategy. Collected occupancy patterns of the human, which are typically represented using voxels (Antão et al. 2019), are used in Zhao et al. (2018) to generate a trajectory that possibly avoids the region previously occupied by the human. The prediction method in Ragaglia et al. (2015) has been used in Ragaglia et al. (2018) within a Quadratic Programming (QP) problem to locally deform a given trajectory, and in Zanchettin et al. (2019) for a local obstacle avoidance strategy. In Casalino et al. (2019) a path deformation strategy minimising the intersection between human and robot spaces is proposed. Finally, in Park et al. motion prediction is used to locally detour the robot trajectory, (Park et al. 2019).

This paper proposes a strategy for the adaptation of the robot motion profiles based on an occupancy model of the human fellow operator. Collecting long-term occupancy data of the human operator, the proposed algorithm evaluates the *optimal trajectory* for the robot that simultaneously minimises the probability of collisions and the duration of the motion profile. The optimisation is based on a *digital twin*, (Tao et al. 2018; Arne Bilberg and Ali Ahmad Malik 2019), of the robot motion controller as well as of its safety strategy. In this way the optimization strategy can take into account how the robot behaves during its nominal operations, and how its speed is reduced in close proximity of the human. The general architecture of the developed strategy is shown in Fig. 1.

The remainder of the paper is organised as follows. Section 2 reports a review of state of the art methods that are relevant for the problem handled in this work. Section 3 formalises the problem of generating minimum-time trajectory allowing the robot to reach a certain goal configuration from a starting one, while guaranteeing minimum intervention



**Fig. 1** Highlights of the method: the algorithm generates feasible trajectories for the robots that simultaneously minimise the probability of collisions with the human operator and the traversal time. The risk of collisions is evaluated based on a probabilistic representation of the space occupied in the long-term by the operator. As the optimisation algorithm runs continuously, the robots automatically adapt to operator

of safety functionalities. Section 4 details the optimisation algorithm introduced to optimally solve the trajectory planning problem. Section 5 describes the software architecture that implements the method and the layout of the experimental setup adopted for the verification. Finally, the outcome of the experimental campaign is discussed in Sect. 6.

## 2 Survey on existing methods and comparison

Prediction of human occupancy volumes tends to be overconservative, (Pereira and Althoff 2018), especially for long-time predictions. It follows that strict avoidance constraints might limit the performance of the robot. In this work, the SSM criterion is relaxed according to the occupancy probability (refer to Sect. 3.2 for more details), allowing the robot to achieve better performance. Several works are related to the problem handled in this paper. In the following, detailed comparisons between the proposed solution and state of the art methods is given. Table 1 at the end of this Section summarises the analysis.

In Balan and Bone (2006), a simple model-based prediction of the human motion is used in the robot controller to search for collision-free paths by moving the end-effector along a set of pre-defined search directions while balancing between the attraction to the goal and repulsion from the human.

Ding et al. (2011) adopted a Hidden Markov Model (HMM) for the prediction of human reaching motion to be used in a MILP strategy for motion planning. Based on a long-term prediction, the trajectory generation is

**Table 1** Summary of the comparison with existing methods

| Paper                       | Human prediction |               |                |               | Trajectory optimisation |        |
|-----------------------------|------------------|---------------|----------------|---------------|-------------------------|--------|
|                             | Model-based      | Intended goal | Occupied space | Reachable set | Local                   | Global |
| Balan and Bone (2006)       | ✓                |               | ✓              |               | ✓                       |        |
| Ding et al. (2011)          |                  | ✓             |                |               |                         | ✓      |
| Mainprice et al. (2016)     |                  | ✓             |                |               | ✓                       |        |
| Pellegrinelli et al. (2016) |                  |               | ✓              |               | Offline                 |        |
| Pereira and Althoff (2018)  |                  |               |                | ✓             | ✓                       |        |
| Unhelkar et al. (2018)      |                  | ✓             |                |               | ✓                       |        |
| Wang et al. (2018)          | ✓                |               |                |               | ✓                       |        |
| Zhao et al. (2018)          |                  |               | ✓              |               |                         | ✓      |
| Zanchettin et al. (2019)    |                  |               |                | ✓             | ✓                       |        |
| Casalino et al. (2019)      |                  | ✓             |                |               | Path                    |        |
| Park et al. (2019)          |                  | ✓             |                |               | ✓                       |        |
| <b>This paper</b>           |                  |               | ✓              |               |                         | ✓      |

handled in a robust manner, allowing the robot to avoid collisions within a specified confidence level. The work in Pellegrinelli et al. (2016) also presents a long-term occupancy representation of the human worker based on probability grids. A pre-trained set of trajectories is made available to the robot at run time. The planner decides online which trajectory to execute within the finite set in order to minimise the variability in the execution time, knowing that the robot might have to slow down or stop due to the proximity of the operator. Though being highly correlated to the problem addressed in this paper, no adaptation capabilities are reported. In Mainprice et al. (2016) a prediction of human occupancy in terms of swept volumes is used by a motion planner to generate robot trajectories minimising the penetration of the robot in the space possibly occupied by the human. The behaviour of the robot is obtained by interleaving planning and execution during the motion. In Pereira and Althoff (2018), the authors gathered data from a motion capturing campaign to predict any possible future spatial occupancy of human arm movements. This model of the human arm occupancy is used to anticipate a safety countermeasure of the robot. The work detailed in Unhelkar et al. (2018) adopts the multiple-predictor method in Lasota et al. (2017) to feed a robotic planner that interleaves online planning and execution. Prediction based on time-series has been also used in Wang et al. (2018) together with a local optimal detour strategy. Zhao et al. (2018) constructed a map of the space previously occupied by human which is then used to feed standard trajectory optimisation solvers. In Zanchettin et al. (2019), the robot is informed of the short-term prediction of human occupancy gathered using the method in Ragaglia et al. (2015) and implements a SSM criterion. A dodging trajectory is executed before the robot velocity has to be reduced due to the proximity of the operator. A

path planning system that adapts during its execution has been presented in Casalino et al. (2019). The system incrementally learns the occupancy of the human arm while reaching a certain goal and adapts the path of the robot to optimally handle the trade-off between the length of the path and the possible interference with the human. The intended goal of the human operator has been used in Park et al. (2019) to estimate the corresponding reaching motion and its velocity which is used to locally detour the trajectory of the robot in the vicinity of the estimated occupied volume.

Most of the reported works adopt a short-term prediction of the human motion, either based on the intended goal to be reached, or using reachable sets within a predefined prediction horizon. Consistently, the trajectory generation problem is typically handled locally (i.e. based on a reactive strategy that deforms a pre-planned trajectory or path) or globally, but focusing only on short-term prediction. When it comes to consider the long-term occupancy prediction, the problem of generating a suitable trajectory for the robot is mainly handled offline, (Pellegrinelli et al. 2016), or based on the minimisation of multi-objective cost functions, (Zhao et al. 2018). In particular, the work in Kalakrishnan et al. (2011) uses the Stochastic Trajectory Optimisation for Motion Planning (STOMP) library to minimise both the acceleration and the number of voxels occupied by the robot that are also likely to be occupied by the human.

This paper addresses the problem of generating a trajectory that globally minimises the risk of collision with the operator, based on long-term occupied volume. The trajectory generation module embeds a model of the SSM functionality the robot adopts to reduce its speed in case of close proximity with the operator and tries consistently to avoid the intervention of the safety functionality which will inevitably reduce the productivity of the robot.

### 3 Trajectory optimisation method

The method developed in this work allows the robot to plan and adapt its trajectories based on the occupancy data of the human operator. An optimisation algorithm is developed with the aim of simultaneously minimising the risk of collisions and the traversal time of the trajectory. In particular, the trajectory is obtained considering that, according to the SSM criterion, the closer the manipulator is to the operator, the slower its motion must be. A trade-off between shorter cycle times and vicinity of the operator (hence shorter paths) is achieved by the optimisation algorithm that accounts for the safety of the operator as a constraint on the velocity of the robot. The presence of the operator in the work-cell is accounted for in terms of an occupancy probability grid, hence the safety constraint is handled in a probabilistic manner and the collision avoidance criterion is relaxed according to the occupancy probability, as will be carefully described in Sect. 3.2.

#### 3.1 Parameterisation of robot trajectories

There are several available ways to describe a trajectory: path and timing law, interpolating polynomials, splines, etc. (Siciliano et al. 2010). In order to be compatible with most of the robot controllers, in this paper the problem is solved directly at *instruction level*, i.e. the trajectory is parameterised by means of motion instructions. The clear benefit of this approach is that the outcome of the trajectory generation algorithm can be directly executed by the robot controller. Regardless of the particular syntax that depends on the robot manufacturer, a motion instruction usually looks like to following one:

```
MoveL ToPoint Speed Zone;
```

where *MoveL* is a motion instruction requiring the robot to move linearly (along a straight line) from its current position to the specified position, *ToPoint* specifies the position and orientation that the robot shall reach at the end of the motion, *Speed* specifies the maximum speed of the robot along the path, while *Zone* specifies how much the path can cut corners when close to *ToPoint*.

In the light of the discussion above, we assume the trajectory planner to be parameterised through the following set of parameters:

$$\theta = \left[ \underbrace{m_1, \mathbf{p}_1, v_{0,1}, R_1, \dots, m_n, \mathbf{p}_n, v_{n-1,n}, R_n}_{\text{intermediate points}}, \underbrace{m_{n+1}, v_{n,n+1}}_{\text{end point}} \right]^T \in \Theta$$

where  $m_i$  is the interpolation type (Cartesian or joint space, Siciliano et al. 2010),  $\mathbf{p}_1, \dots, \mathbf{p}_n$  indicate the intermediate

via points,  $v_{i,i+1}$  stands for the maximum velocity along the segment connecting two consecutive targets,  $R_i$  contains the blending radii to be applied when close to via point  $\mathbf{p}_i$ . The first four parameters are the typical parameters of a motion instruction. Finally, symbols  $\mathbf{p}_0$  and  $\mathbf{p}_{n+1}$  denote the first and the last target points for the robot trajectory which are not subject to optimisation. Given the set of parameters  $\theta$ , we assume that the trajectory planner is capable of evaluating the joint position of the robot  $\mathbf{q}$  at a given time instant  $t$  as  $\mathbf{q}_t = \mathbf{q}(t, \theta)$ , together with its time derivatives  $\dot{\mathbf{q}}_t = d\mathbf{q}_t/dt$  and  $\ddot{\mathbf{q}}_t = d^2\mathbf{q}_t/dt^2$ . Moreover, given the traversing velocity along the path, which is also computed by the planner, it is relatively easy to evaluate the traversal time  $T = T(\theta)$ .

#### 3.2 Strategy for speed and separation monitoring

As described previously, the proposed optimisation methods aims at adapting the parameters of the robot trajectory based on the occupancy of the human fellow operator. The optimisation procedure also takes into account the SSM criterion, as will be described in the following.

Consider an obstacle occupying the cell (i.e. a volumetric unit) centred in  $\mathbf{r}$  and assume that the SSM constraint can be specified as follows (see Ragaglia et al. (2015) for an explicit example of the derivation of such a constraint and the definition of the related variables):

$$\mathbf{E}(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t \leq \mathbf{f}(\mathbf{q}_t, \mathbf{r}). \quad (1)$$

Basically, as explained in Ragaglia et al. (2015), the SSM criterion is applied between each point along the links of the robot and the obstacle located in  $\mathbf{r}$ . By elaborating such constraints, sufficient conditions can be expressed in terms of the joint velocities  $\dot{\mathbf{q}}_t$  as in eq. (1), where  $\mathbf{E}(\mathbf{q}_t, \mathbf{r})$  is a matrix and  $\mathbf{f}(\mathbf{q}_t, \mathbf{r})$  is a vector of suitable dimensions.

The occupancy is however non deterministic and can be regarded as a Bernoulli distributed stochastic variable:

$$B_r \in \{0, 1\}, B_r \sim \mathcal{B}(\mathcal{P}_r)$$

where  $\mathcal{P}_r$  indicates the occupancy probability of the cell centred in  $\mathbf{r}$ . The occupancy probability  $\mathcal{P}_r$  is here considered as an almost-stationary quantity, as it will be used to represent the long-term occupancy pattern of the human operator. The constraint can be then rewritten as follows:

$$B_r \mathbf{E}(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t \leq \mathbf{f}(\mathbf{q}_t, \mathbf{r}) \quad (2)$$

Notice that whenever  $B_r = 1$ , Eq. (2) coincides with Eq. (1), whilst when  $B_r = 0$ , the constraint is automatically satisfied, being  $\mathbf{f}(\mathbf{q}_t, \mathbf{r}) \geq 0$ , (Ragaglia et al. 2015). In order to get rid of the stochasticity of variable  $B_r$ , the left hand side of Eq. (2) is replaced by its expected value:

$$\mathbb{E}[B_r E(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t] \leq f(\mathbf{q}_t, \mathbf{r}) \tag{3}$$

which allow us to introduce the following deterministic constraint:

$$\mathbb{E}[B_r] E(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t = \mathcal{P}_r E(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t \leq f(\mathbf{q}_t, \mathbf{r}) \tag{4}$$

From a geometric perspective, the deterministic constraint in Eq. (4) allows for a higher velocity of the robot  $\dot{\mathbf{q}}_t$ , with respect to the one satisfying the original constraint, if the corresponding cell centred in  $\mathbf{r}$  has a low probability of being occupied. Finally, notice that at time  $t$  the position  $\mathbf{q}_t$  and the velocity  $\dot{\mathbf{q}}_t$  of the robot depend uniquely on the trajectory parameters  $\theta$ . For the development of the method it is then convenient to explicitly highlight this dependency and rewrite the constraint in Eq. (4) as follows:

$$\mathcal{P}_r a(t, \theta, \mathbf{r}) \leq b(t, \theta, \mathbf{r}) \tag{5}$$

where  $a(t, \theta, \mathbf{r})$  and  $b(t, \theta, \mathbf{r})$  stand for  $E(\mathbf{q}_t, \mathbf{r}) \dot{\mathbf{q}}_t$  and  $f(\mathbf{q}_t, \mathbf{r})$ , respectively.

The SSM criterion prescribes that the robot reduce its velocity according to the closeness of the human. Since the information about the human position is only available in a statistical manner by means of a probabilistic occupancy grid, the prescription of the SSM criterion is rendered in a probabilistic way. In particular, the robot is allowed to traverse an area which is possibly occupied by the human with a speed which is in turn proportional to the probability of such area being not occupied. The constraints in Eq. (4) clearly express this property: the higher the occupancy probability  $\mathcal{P}_r$  of the cell centred in  $\mathbf{r}$ , the lower the velocity of the robot  $\dot{\mathbf{q}}_t$  when passing in the vicinity of  $\mathbf{r}$ . Therefore, it should be further clarified that, during the actual execution of whatever collaborative task, the SSM criterion is always active to limit the robot speed according to the distance from the human, thus enforcing the safety of the cooperation. However, during the evaluation of the proposed trajectory optimisation algorithm, since the information related to the actual human occupancy can only be stochastically estimated, the SSM criterion is evaluated according to Eq. (3).

### 3.3 Long-term occupancy model of the human

In the following a recursive law to update the occupancy probability  $\mathcal{P}_r$  is developed. For the long-term prediction purposes of this work, it is reasonable to assume that the long-term occupancy probability is a stationary distribution. Given  $N$  occupancy samples, the Maximum Likelihood Estimator (MLE) at discrete time  $t$  of the parameter  $\mathcal{P}_r = \mathbb{E}[B_r]$  of a Bernoulli process is:

$$\hat{\mathcal{P}}_r^{t,N} = \frac{1}{N} \sum_{i=0}^{N-1} B_r^{t-i}$$

where  $B_r^t \in \{0, 1\}$  represents the value of the Bernoulli process at discrete time  $t$ . In turn, at discrete time  $t + 1$ , i.e. when a new sample is available, we have:

$$\begin{aligned} \hat{\mathcal{P}}_r^{t+1,N+1} &= \frac{1}{N+1} \sum_{i=-1}^{N-1} B_r^{t-i} \\ &= \frac{1}{N+1} (N \hat{\mathcal{P}}_r^{t,N} + B_r^{t+1}) \end{aligned} \tag{6}$$

Therefore, an Infinite Impulse Response (IIR) or Exponentially Weighted Moving Average (EWMA) filter can be adopted as a recursive MLE of  $\mathcal{P}_r$ , meaning that, whenever a new sample is available, the occupancy probability of the cell centred in  $\mathbf{r}$  can be updated as follows:

$$\hat{\mathcal{P}}_r^{t+1} = \alpha \hat{\mathcal{P}}_r^t + (1 - \alpha) B_r^{t+1} \tag{7}$$

where  $0 < \alpha \ll 1$  is a tunable parameter. Particularly small values of  $\alpha$  guarantee a severe low-pass behaviour of the filter, and are thus suitable to represent the long-term occupancy volume of the human operator.

### 3.4 Constrained trajectory optimisation

We are now in position to introduce the minimum time trajectory generation problem to be solved. Assume that  $n$  intermediate waypoints have to be used to generate the corresponding trajectory, then the optimisation of the parameters  $\theta_n$  can be handled in terms of lexicographic optimisation as follows:

$$n, \theta_n \in \Theta_n \left\{ \max_{t, \mathbf{r}} [0, \hat{\mathcal{P}}_r a(t, \theta_n, \mathbf{r}) - b(t, \theta_n, \mathbf{r})], T(\theta_n) \right\} \tag{8}$$

where the cost function includes the two prioritised objectives: the former, with higher priority, accounts for the minimisation of the risk, which in turn is expressed in terms of the worst case slack variable  $s(t, \theta_n, \mathbf{r}) = \hat{\mathcal{P}}_r a(t, \theta_n, \mathbf{r}) - b(t, \theta_n, \mathbf{r})$  associated to the obstacle avoidance constraints in Eq. (5), while the latter accounts for the traversing time. In the following, the higher priority objective in Eq. (8) will be also referred to as ‘penalty’, while the other one will be also indicated as ‘cost’.

The reason why a lexicographic optimisation has been adopted is that completely eliminating the risk of collisions is practically impossible. In realistic scenarios, especially when parts of the workspace are shared between the human and the robot, it is impossible to guarantee the existence of a collision-free path for the robot. In particular, in scenarios

where the target point of the robot can be also occupied by the human, the absence of a collision-free path is obvious. Nevertheless, one would still want to minimise the risk of collisions between the human and the robot. To this end, the optimisation algorithm in Eq. (8) first tries to minimise the risk of collisions, which is proportional to the worst-case slack variable  $s$ . Then, for the same amount of risk, the algorithm prefers trajectories with a shorter traversal time  $T$ .

## 4 Trajectory optimisation algorithm

The algorithmic functionalities adopted to solve the optimisation problem in Eq. (8) deserve a particular attention. The optimisation problem, in fact, is highly nonlinear, non-convex and non-smooth since some parameters of the robot trajectory are taken from discrete sets. For example, the interpolation type  $m_i$  can be represented in terms of a boolean variable (0 indicates interpolation in the joint space, while 1 indicates interpolation in the Cartesian space). Differently, the maximum velocity along the path  $v_{i,i+1}$  (expressed in mm/s) as well as the blending radius  $R_i$  (expressed in mm) are both represented by continuous values defined within the specific robot admissible ranges.

In turn, the position (and orientation) of each waypoint  $\mathbf{p}_i$  lies in  $SE(3)$ , which is not a Euclidean space. In general, solving-partly combinatorial optimisations results in a  $\mathcal{NP}$ -hard problem. Moreover, the relationship between the parameter space  $\Theta$  and the cost function  $T(\theta)$ , i.e. the model of the trajectory planner, is far from being smooth or at least differentiable.

In the light of the above, the classical gradient-based optimization methods are unsuitable to search an optimal solution for the optimisation problem in Eq. (8), while genetic algorithms (GA) can be efficiently used to solve these optimization problems (Nia et al. 2009). Indeed, GA are a family of a gradient-free metaheuristic models whose easiness, accuracy and adaptable topology allows them to efficiently find the global minimum or maximum of non-linear optimization problems (Nia et al. 2009). Hence, similarly to the work in Abo-Hammour et al. (2011), we here adopt a gradient-free metaheuristic optimisation method (i.e. a GA) to solve the optimization problem in Eq. (8). In the following, the main phases of the GA are detailed.

### 4.1 Initialization

The first step of the proposed genetic algorithm is represented by the initialization phase. At this stage, the starting nominal trajectory is designed and its fitness in terms of risk of collision and traversal time  $T$  is evaluated (see Eq. (8)). Then,  $N$  clones of this trajectory are created to produce the initial population of candidate trajectories.

### 4.2 Reproduction

This phase consists in the selection of the individual (trajectory) of the population which the modification (i.e. mutation) will be applied to. To perform reproduction, a random integer value is sampled from a uniform distribution defined over the interval  $(1, N)$ , where  $N$  is the population size.

### 4.3 Modifications of trajectories

In the following, the adopted strategy to modify a given trajectory is detailed.

*Cartesian or joint space interpolation* The trajectory between two consecutive waypoints can be defined and interpolated in the joint space or in the operational (or Cartesian) space. The present binary operator applies a modification on a certain segment of a trajectory by simply swapping the interpolation type.

*Waypoint position, speed and blending adjustment*

The trajectory between two consecutive waypoints is defined in terms of a path and a velocity profile. The velocity profile, in turn, is usually specified in terms of the maximum cruise velocity along the path. The velocity is randomly modified by exploiting a normal distribution:

$$\mathcal{N}\left(\frac{v_j + v_{j+1}}{2}, \sigma_v^2\right)$$

To ensure that the velocity value remains limited within an arbitrary minimum robot speed,  $v_{min}$ , and the prescribed maximum allowed speed for the considered robot,  $v_{max}$ , the following check is applied:

$$\min \left\{ \max \left\{ \mathcal{N}\left(\frac{v_j + v_{j+1}}{2}, \sigma_v^2\right), v_{min} \right\}, v_{max} \right\}$$

Moreover, a geometric parameter is usually introduced to specify the blending radius of the parabolic blend. Another operator applies a random modification to the blending radius according to a normal distribution  $\mathcal{N}(0, \sigma_b^2)$  (an example is given in Fig. 2).

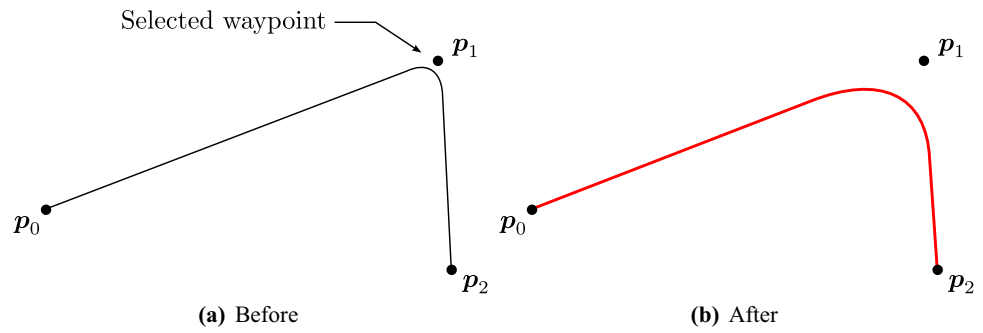
Another possibility is to modify the position of a waypoint. In this case a waypoint is randomly selected and its position<sup>1</sup> is modified by applying a random displacement generated by the normal distribution  $\mathcal{N}(\mathbf{0}, \sigma_p^2 \mathbf{I}_3)$ .

*Waypoint insertion and removal*

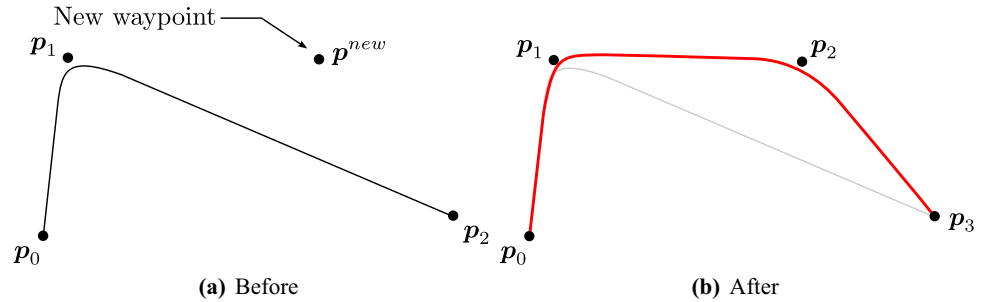
Another important operator allows the algorithm to insert or remove waypoints. In case of insertion of a new waypoint, the algorithm first randomly selects two consecutive and existing waypoints  $\mathbf{p}_j$  and  $\mathbf{p}_{j+1}$ . The position of the new

<sup>1</sup> The orientational part is handled in a similar way. Details are omitted.

**Fig. 2** Trajectory before and after the adjustment (increase) of a blending radius



**Fig. 3** Trajectory before and after the insertion of a new waypoint



waypoint is generated according to the following normal distribution:

$$\mathcal{N}\left(\frac{p_j + p_{j+1}}{2}, \sigma_p^2 I_3\right)$$

Figure 3 reports an example of a modification produced by this operator.

Notice that the variances  $(\sigma_v^2, \sigma_p^2, \sigma_b^2)$  of the aforementioned normal distributions used to modify the trajectories are all tunable parameters.

#### 4.4 Policy for selection of the genetic operator

At each iteration of the genetic algorithm only one of the above-mentioned mutation operators is applied. Let  $m_k^{(i)}$ ,  $i = 1, \dots, 7$  denote the  $i$ -th mutation operator used at iteration  $k$ . Even though, typically, the selection of mutation operator is performed according to a randomised procedure, in this work we propose to exploit a knowledge-based criterion to improve the effectiveness of the mutation phase. The ultimate goal of this approach is to increase the frequency of selection of those mutation operators that turned out to produce more efficient trajectories with respect to the optimisation in Eq. (8) as well as improving the convergence rate of the genetic algorithm.

In order to evaluate the efficiency of mutation  $m_k^{(i)}$ , the difference in terms of fitness between the offspring trajectory and its parent is computed over non-overlapping windows containing  $\bar{k}$  iterations,  $T_w$ . Hence, if the fitness of the offspring

trajectory improves its parent's one, mutation  $m_i$  is considered efficient. Otherwise, it is considered inefficient.

Based on the above, the probability  $P$  of selecting mutation  $m^{(i)}$  in the next time window is computed as follows:

$$P(m^{(i)}) = \frac{v_{m^{(i)}}}{\tau_{m^{(i)}}}$$

where  $v_{m^{(i)}}$  is the number of trajectories where mutation  $m_i$  was efficient within  $T_w$  and  $\tau_{m^{(i)}}$  is the number of iterations where  $m_i$  was applied. Eventually, the probability of the mutations are normalized to one.

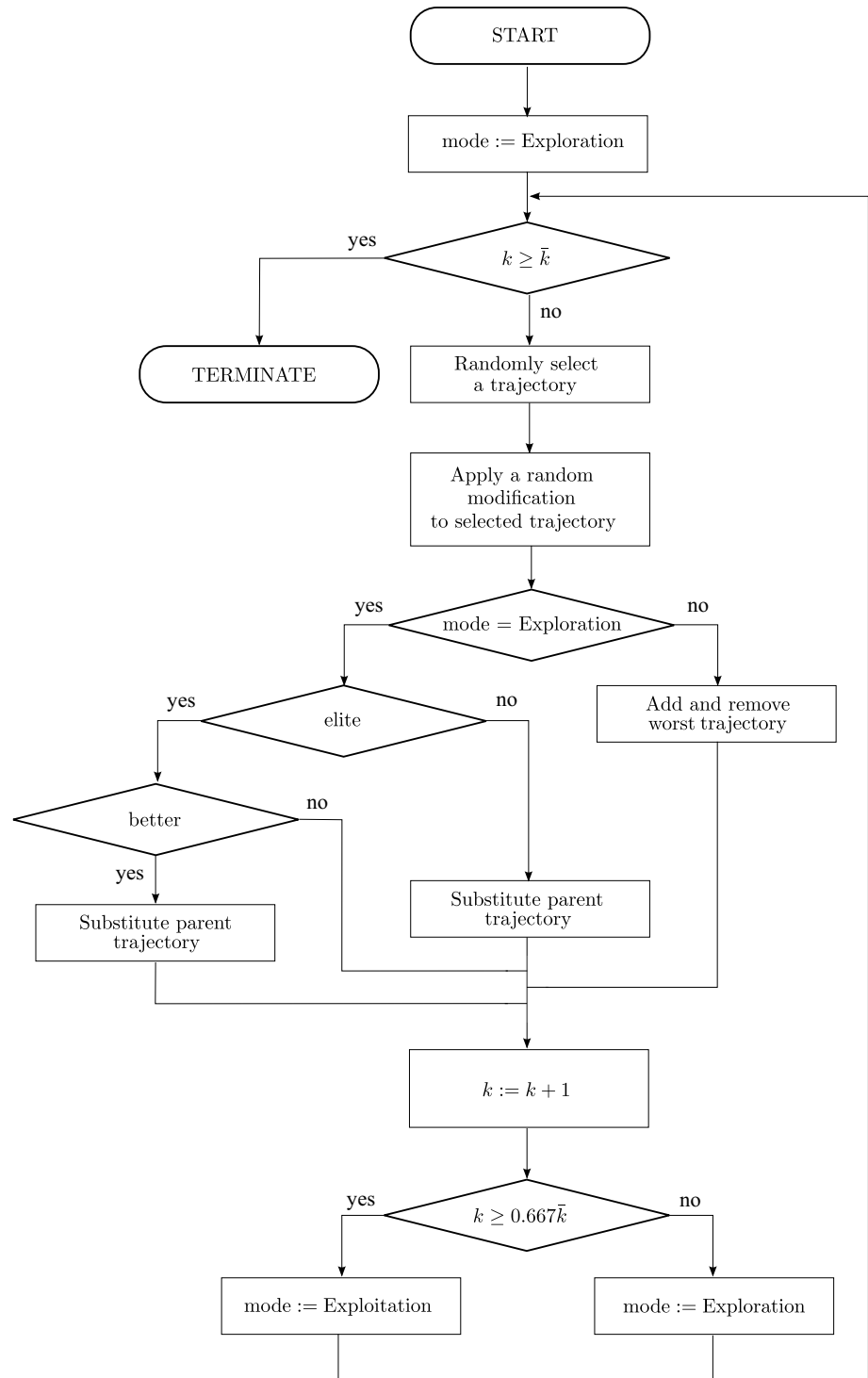
Notice that the dynamic variation of mutation probabilities might entail an excessive decrease for low-fitness mutation, thus preventing their selection. To solve this problem, a lower bound  $p_{min}$  is set for each mutation operator:

$$P(m^{(i)}) = \max \left\{ p_{min}, \frac{v_{m^{(i)}}}{\tau_{m^{(i)}}} \right\}$$

#### 4.5 Exploration vs. exploitation

Due to the complexity of the problem at hand and its non-convex structure, it is beneficial to first explore the space of parameters before attempting to find the (possibly local) optimum. The optimisation procedure is therefore divided into two subsequent parts: exploration and exploitation.

**Fig. 4** Flow chart of the optimisation procedure

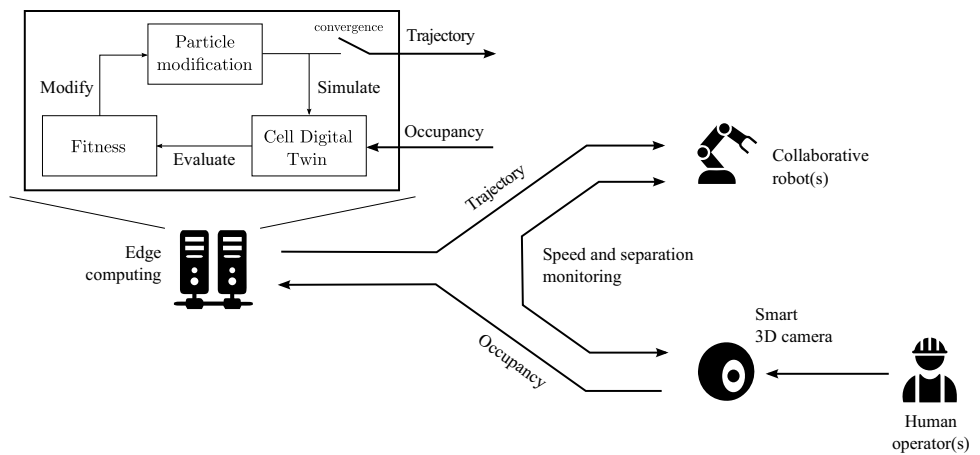


The exploration tries to diversify as much as possible the population of trajectories, thus exploring the space of parameters. When a new trajectory is obtained, it is added to the population by replacing its parent trajectory. A small percentage of best individual trajectories (elite) is however preserved in this procedure (substitution occurs only if an improvement in the fitness is registered). The exploitation

part, in turn, tries to improve as much as possible the overall fitness of the population. After the application of an operator, the new trajectory replaces the worst one in the population. The overall optimisation algorithm is further detailed in the flow chart of Fig. 4.



**Fig. 5** Architecture of the system: the smart 3D camera produces the occupancy grid which is sent to the edge computing device. Such a device elaborates the grid and produces the occupancy probability which is used by the optimisation algorithm to produce an optimal trajectory for the robot. The trajectory is returned to the robot, in the form of an executable text file, via FTP connection



## 5 Use-case and implementation

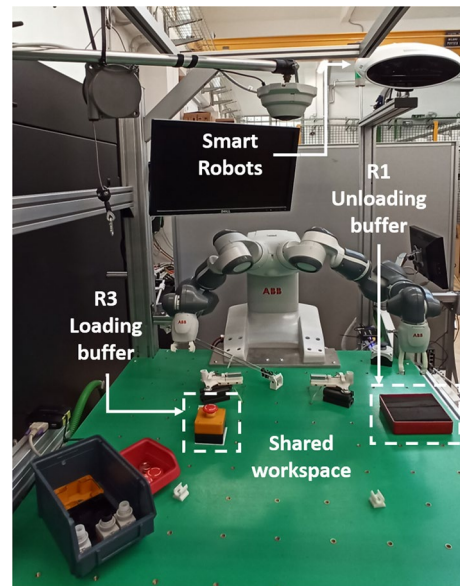
As introduced in Sect. 3, the objective of the work detailed in this paper is to derive an optimal trajectory for a collaborative robot, based on the space occupied by the human operator. The occupancy of each cell in each discrete time instant is returned by a smart 3D camera commercialised by SMART ROBOTS.<sup>2</sup>

When invoked, the optimisation algorithm described in the previous Section is fed with the most updated occupancy probability of each cell. Its execution runs on a dedicated server (INTEL i7-3770 4-Core 3.4 GHz, 16 GB RAM). The fitness of a trajectory is evaluated based on a digital twin of the work-cell containing a custom replica of the robot controller (running in the ABB RobotStudio simulation environment), together with the long-term occupancy model recursively updated using Eq. (7). The simulations run on another dedicated server (INTEL i7-960 4-Core 2.67 GHz 8 GB RAM). The two PCs constitute an edge computing service that communicates with the shop-floor.

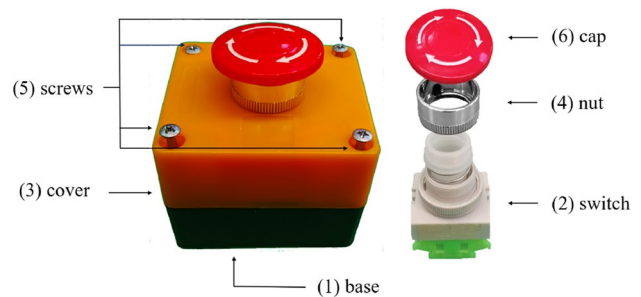
The simulation of each trajectory on the Digital Twin requires on average 2.5 s. Clearly, this time depends on the trajectory duration and on the computational capacity of the server for the digital twin.

Conversely, the evaluation of the fitness of each simulated trajectory on the main server requires on average 0.64 s, thanks to a multithread computing process.

The SMART ROBOTS device is in dialog with the server and implements the SSM functionality described in Ragaglia et al. (2015). The trajectory obtained by the optimisation algorithm is then translated into the proprietary code of the robot manufacturer and saved in a text file. This file is finally sent to the robot via FTP connection. The overall architecture of the system is sketched in Fig. 5.



**Fig. 6** Layout of the experimental setup



**Fig. 7** Emergency stop button assembled by the human operator

For the verification, a collaborative assembly station has been setup, as illustrated in Fig. 6, where the ABB IRB14000“YuMi” robot and the human operator perform a set of activities to assemble an emergency stop button

<sup>2</sup> Smart Robots web site: <http://smartrobots.it/>.



**Fig. 8** An operator assembling the button in the shared workspace

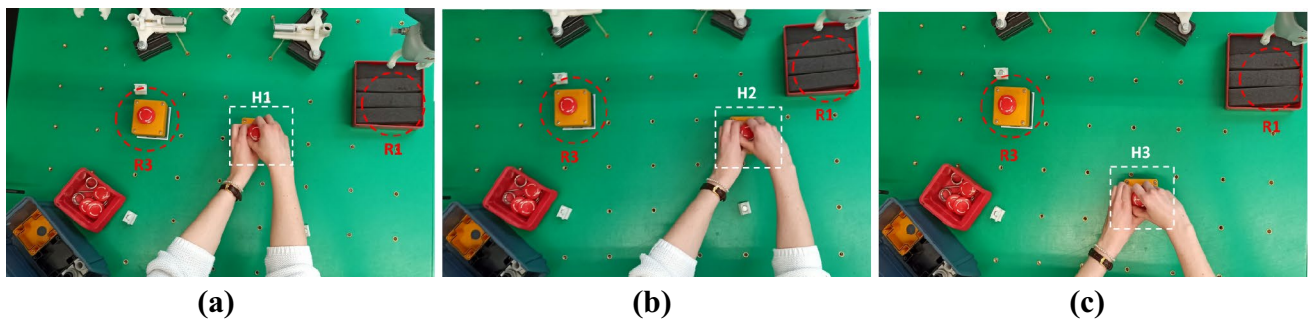
(see Fig. 7). The station is always supervised by the SMART ROBOTS device, see again Fig. 6. The main parameters of the use-case are reported in Table 2.

The proposed task exemplifies a typical assembly activity performed in an industrial scenario by a human operator and a cobot. In this specific framework the human and the robot execute simultaneously different sets of actions during which

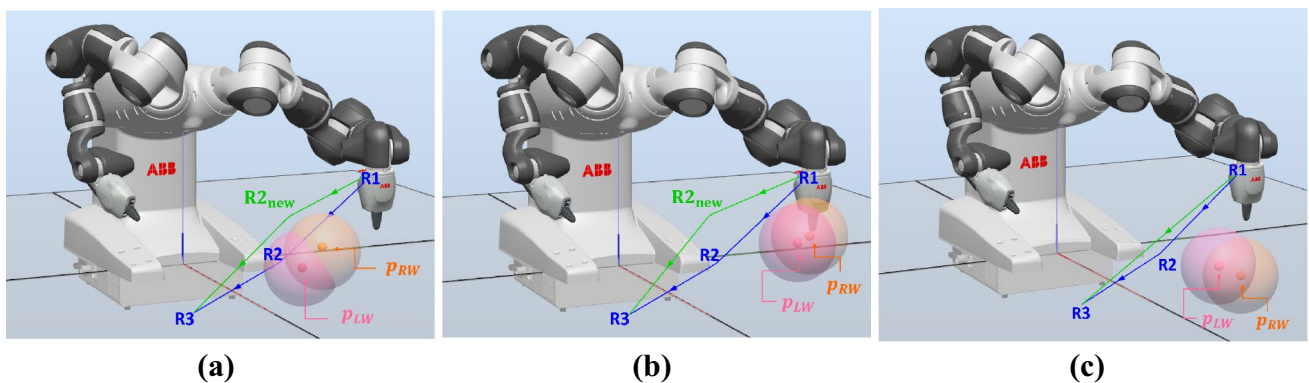
their working regions may overlap. The human activity consists in assembling all the components of the emergency stop button illustrated in Fig. 7. As can be noticed, the button is composed by several small parts that require high cognitive skill and accuracy to be assembled. The assembly is performed by the human (see Fig. 8) in a prescribed location of the shared workspace and requires a relatively large amount of time. During this phase, the human occupancy can be considered quasi-static. Once the human operator has completed the assembly of one unit, he/she has to place the button in the R3 loading buffer. After that, he/she can start a new assembly cycle.

As mentioned previously, the interaction is synchronous. Hence, for each button completed by the human operator, the robot has to perform a motion trajectory to reach location R3, pick the just completed button, and unload it in buffer R1 (see Fig. 6). More specifically, the robot nominal trajectory (i.e. the initial robot motion trajectory provided to the genetic algorithm) consists in a linear motion from position R1 to R3 passing through a waypoint (R2) that is located approximately in an intermediate position between the segment connecting R1 and R3.

As it is apparent from this description, positions R1 and R3 are assumed to be constrained (fixed) target locations



**Fig. 9** Different regions of the shared workspace are occupied by the human operator during the assembly: **a** H1; **b** H2; **c** H3



**Fig. 10** Robot nominal trajectory (blue) and optimized trajectory (green) according to the volume occupied by the human during the assembly: **a** H1; **b** H2; **c** H3.  $p_{LW}$  and  $p_{RW}$  represent the position of the centre of the human left wrist and right wrist, respectively. The

occupancy associated with the human hands in assembly position H1, H2 and H3 is depicted through two spheres centered in  $p_{LW}$  and  $p_{RW}$ , respectively

**Table 2** Parameters of the use-case

| Parameter             | Description                      | Value                         |
|-----------------------|----------------------------------|-------------------------------|
| $N$                   | Population size                  | 20                            |
| $E$                   | Elite size                       | 20% of $N$                    |
| $\bar{k}$             | Maximum number of iterations     | 100                           |
| $dt$                  | Trajectory sampling time         | 100 ms                        |
| $L \times W \times H$ | Cell size                        | $1.2 \times 1.2 \times 1.2$ m |
| $M$                   | Number of cells                  | $20 \times 20 \times 20$      |
| $\Delta$              | Cell resolution                  | $6 \times 6 \times 6$ cm      |
| $\alpha$              | IIR filter parameter             | 0.005                         |
| $f_s$                 | Sampling frequency of IIR filter | $\sim 25$ Hz                  |

for the task. Differently, the human operator is allowed to perform the assembly in one of the following three regions:

- 1 H1, as reported in Fig. 9a and in Fig. 10a: in this case, the human operator, during the assembly, occupies a volume of the workspace (H1) slightly intersected by the nominal trajectory of the robot;
- 2 H2, as reported in Fig. 9b and in Fig. 10b: in this case, the human operator does the assembly by occupying a region of the workspace (H2) located in closed proximity to the robot unloading position (R1).
- 3 H3, as reported in Fig. 9c and in Fig. 10c: in this case, the operator, during the execution of the task, occupies a volume (H3) that is located far away from the manipulator nominal trajectory.

## 6 Results and discussion

The purpose of the validation framework is to analyse how the robot adapts its motion trajectory from R1 to R3 according to the volume occupied by the human during the assembly, by optimizing Eq. (8).

To test the effectiveness of the trajectory optimisation algorithm, each volunteer (1 male and 1 female) was asked to perform three different use-cases. In the following, the volunteers will be referred to as ‘operator 1’ and ‘operator 2’, respectively.

The use-cases performed by the subjects were characterized by the same setup, same assembly task and by the same robot initial (nominal) path. However, each use-case was associated with a different human occupancy. Hence, in use-case *UC-H1* the operator performed the assembly task in H1, in *UC-H2* he/she worked in H2 and in *UC-H3* he/she did the assembly in H3. Each volunteer performed the use-case with his/her natural timing and posture.

To evaluate the effectiveness of the adaptation of the robot motion trajectory to the different work-cell human occupancies, the following performance metrics have been considered:

- the distance between the operator assembly position and the robot end-effector during its motion along the optimal trajectory with respect to the nominal one;
- the trend of the average penalty and cost of the population of trajectories over the iterations performed by the algorithm;
- the relative average time saved during the execution of the optimised trajectory with respect to the nominal one, when the speed of the robot is scaled according to the vicinity to the human operator.

### 6.1 Results

Figure 10 shows the approximate volumetric occupancies of the operator’s hands during the execution of assembly task as well as the position of the centres of his/her left and right wrists with respect to the robot base frame. This figure also illustrates the robot nominal path (blue) and the path resulting by the optimisation procedure (green) for each use-case.

Hereafter, a more detailed comparison between the performance metrics achieved for each use-case is proposed.

*UC-H1*: Fig. 11 shows the voxel-based occupancy grid associated with H1: the darker the voxel, the higher the human occupancy probability of the work-cell. The voxels traversed by the nominal robot trajectory (Fig. 11a) and by the optimised trajectory (Fig. 11b) have been coloured according to a stoplight-coloured scale where green indicates a negligible probability of collision with the human and red denotes a high risk of collision.

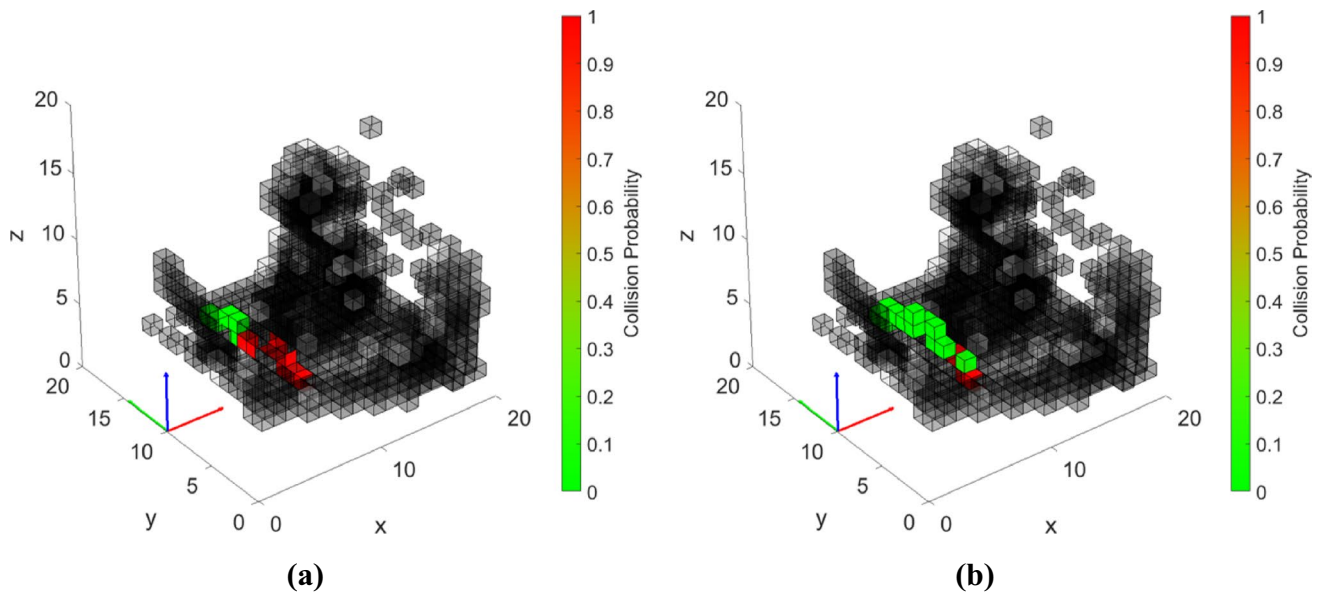
Figure 14 reports the distance between the centre of the human assembly position (H1) and the robot end-effector during its motion along the nominal trajectory (blue), the optimal trajectory for operator 1 (red) and for operator 2 (green).

Eventually, Fig. 17 illustrates the trend of the average penalty and cost of the population of trajectories over the iterations of the genetic algorithm for operator 1 (Fig. 17a and b) and for operator 2 (Fig. 17c and d).

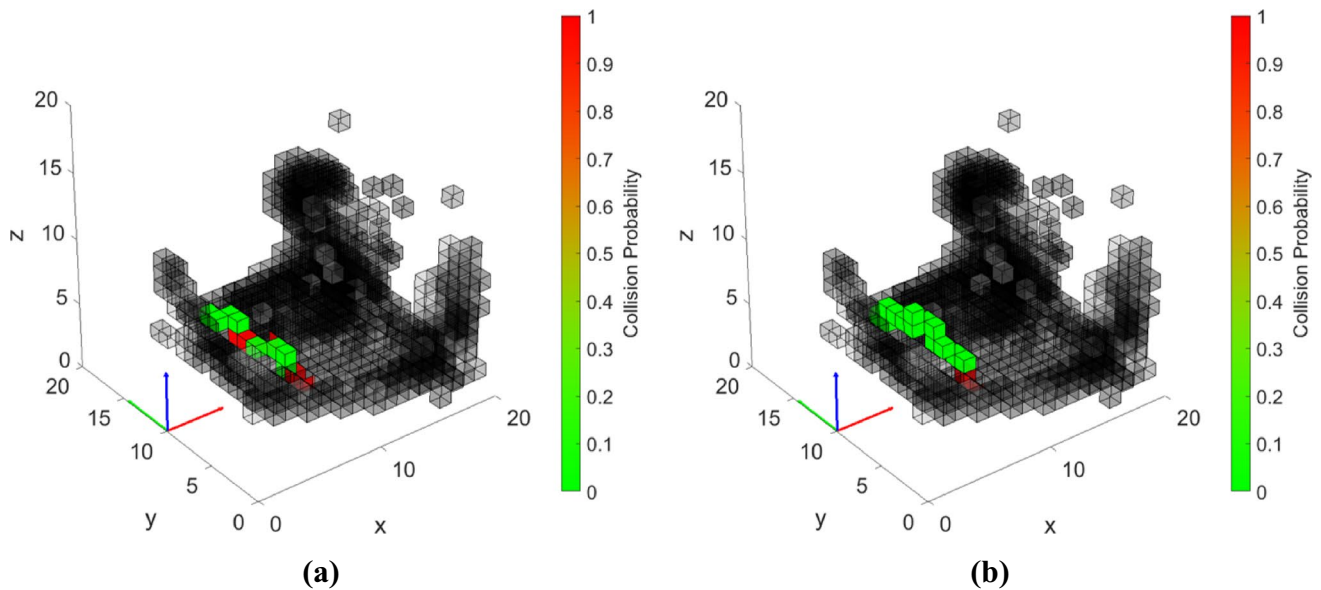
*UC-H2*: Figs. 12, 15 and 18 report the same content of Figs. 11, 14 and 17, respectively, when the human performs the assembly task in region H2.

*UC-H3*: Figs. 13, 16 and 19 illustrate the same results depicted in Figs. 11, 14 and 17, respectively, when the human occupies region H3 during the assembly.

In Table 3 the average time saved by the optimised trajectory with respect to the nominal one are reported



**Fig. 11** Probabilistic occupancy grid and probability of collision associated with robot trajectories when the human operator works in H1: **a** nominal trajectory; **b** optimized trajectory

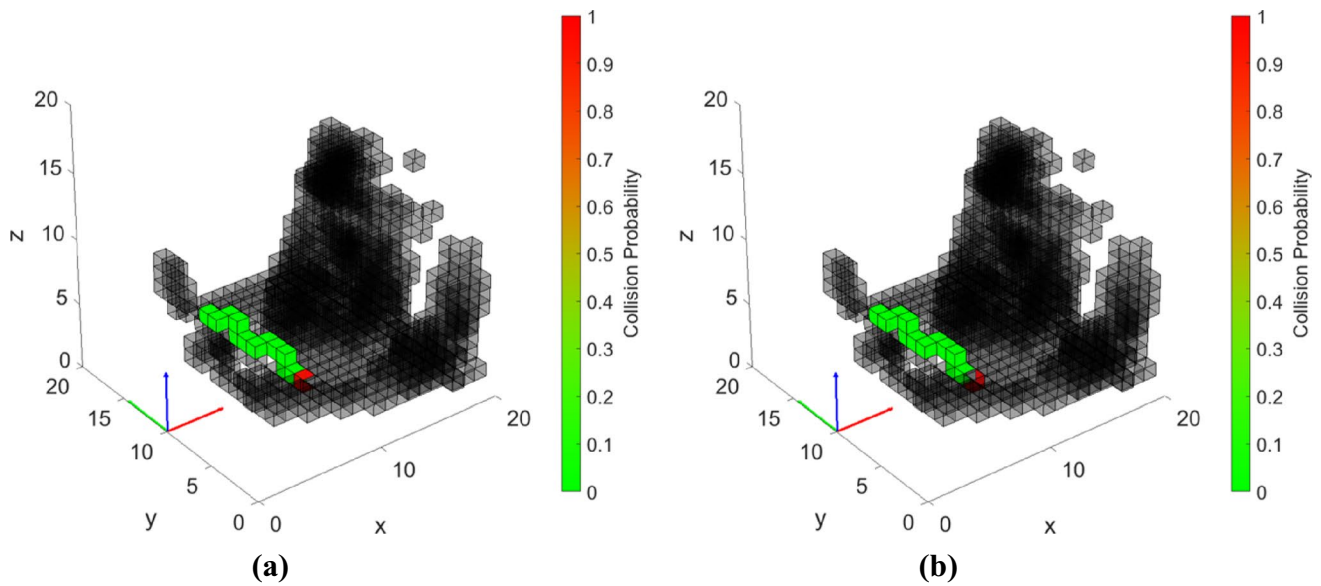


**Fig. 12** Probabilistic occupancy grid and probability of collision associated with robot trajectories when the human operator works in H2: **a** nominal trajectory; **b** optimized trajectory

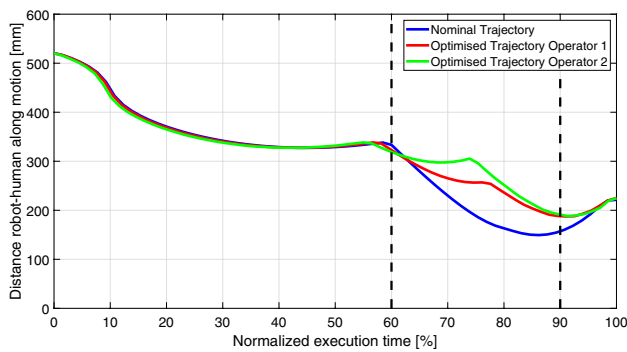
for each use-case. Notice that these outcomes have been obtained taking into consideration the effective execution of the trajectories during the task, when the robot speed scaling might be triggered according to the proximity to the operator.

## 6.2 Discussion

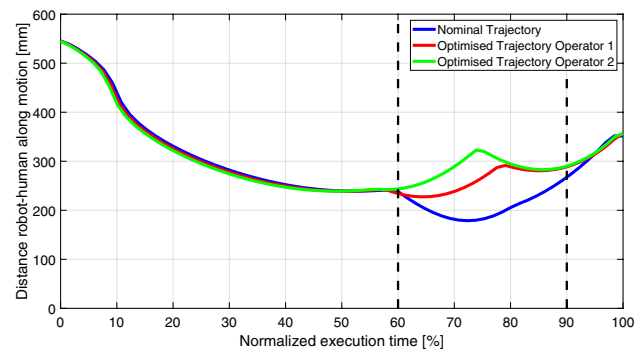
By inspecting Fig. 11 it is possible to notice, even from a qualitative perspective, the effectiveness of the proposed optimisation method. Indeed, the optimized trajectory is compliant with the different human occupancies associated with *UC-H1*, *UC-H2*, *UC-H3* and adapts to them. More specifically, by observing Fig. 11a it is apparent that, since the region occupied by the human partially intersects the robot



**Fig. 13** Probabilistic occupancy grid and probability of collision associated with robot trajectories when the human operator works in H3: **a** nominal trajectory; **b** optimized trajectory



**Fig. 14** Distance between the human assembly position H1 and the robot end-effector during its motion along the nominal trajectory (blue), optimized trajectory for operator 1 (red), optimized trajectory for operator 2 (green). The critical region where the robot moves close to the human assembly position is delimited by two dashed lines



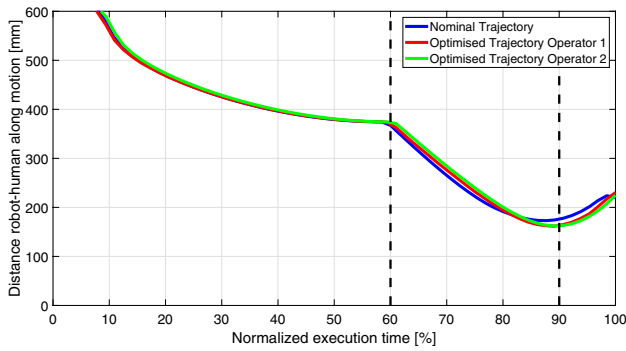
**Fig. 15** Distance between the human assembly position H2 and the robot end-effector during its motion along the nominal trajectory (blue), optimized trajectory for operator 1 (red), optimized trajectory for operator 2 (green). The critical region where the robot moves close to the human assembly position is delimited by two dashed lines

nominal trajectory, the genetic algorithm favours the modification of the position of waypoint R2, by both adjusting its height and applying a lateral displacement. A similar behavior can be noticed in Fig. 11b, even though this time, since the critical region is closer to position R1 (fixed), the genetic algorithm applies only a variation of the height of R2, so as to allow the robot arm to quickly move far away from the human. Differently, as illustrated in Fig. 11c, since the robot nominal trajectory is sufficiently far from the human, waypoint R2 is removed so as to favour the reduction of the trajectory traversal time.

For what concerns the minimization of the risk of collision, the goodness of the proposed optimisation is demonstrated by both Figs. 11, 12 and 13 and by Figs. 14, 15 and

16. Indeed, from the first set of figures just mentioned, it is possible to observe that in *UC-H1* and *UC-H2* the optimized trajectory (Figs. 11b and 13b) shows a greater number of voxels with low collision probability than the ones associated with the corresponding nominal trajectory (Figs. 11a and 13a). Differently, in *UC-H3*, since the nominal trajectory shows a low risk of collision with the human, the optimized one turns out to be almost identical to the nominal in terms of collision probability.

These outcomes are confirmed also by Figs. 14–15, from which it is evident that, when the distance between the robot and the human becomes rather small (black dashed lines), the proposed algorithm tries to optimize the actual trajectory by increasing this distance. Clearly, the opposite occurs in

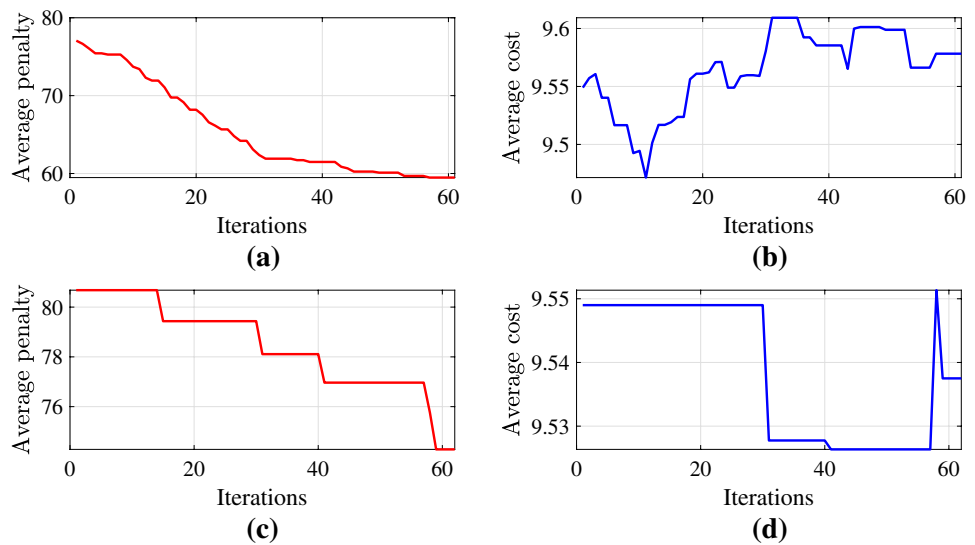


**Fig. 16** Distance between the human assembly position H3 and the robot end-effector during its motion along the nominal trajectory (blue), optimized trajectory for operator 1 (red), optimized trajectory for operator 2 (green). The critical region where the robot moves close to the human assembly position is delimited by two dashed lines

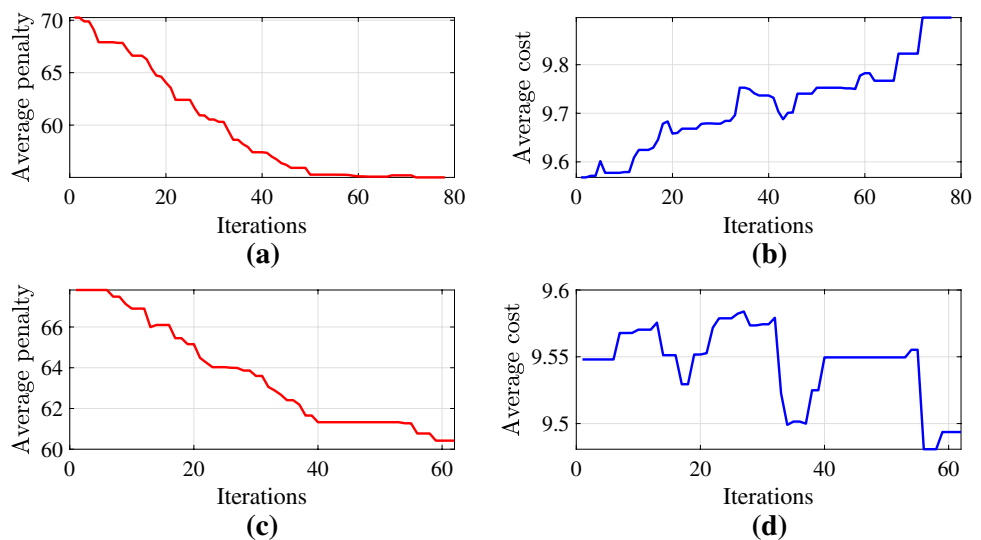
*UC-H3* (Fig. 16) since the human is working sufficiently far from the robot workspace, hence the risk of collision is minimal.

The increase of the distance from the human highlighted for the optimized trajectories is in line with our expectation. In fact, since the optimisation in Eq. (8) takes into consideration that, the closer the robot is to the human, the lower will be its speed (SSM criterion), the intervention of the robot speed scaling is assumed to occur with a low frequency for the optimized trajectories. Actually, when these are applied by the robot during the execution of the assembly task, the intervention of the SSM is significantly reduced. Indeed, as shown in Table 3, especially for what concerns *UC-H1* and *UC-H2*, the SSM activates rarely for the optimized trajectories. This

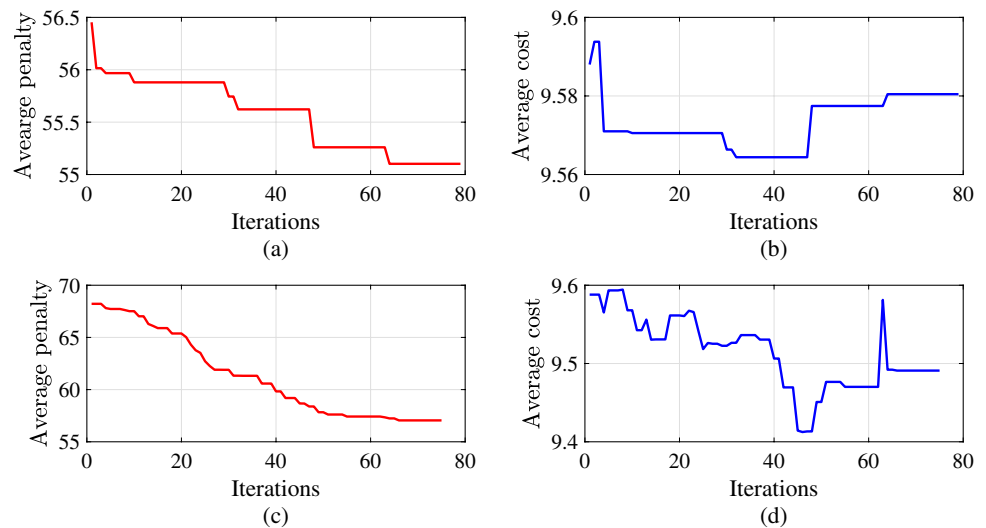
**Fig. 17** Trend of the average penalty and cost of the population of trajectories associated with *UC-H1*. **a** and **b** Refer to the task involving operator 1, while **c** and **d** refer to operator 2



**Fig. 18** Trend of the average penalty and cost of the population of trajectories associated with *UC-H2*. **a** and **b** Refer to the task involving operator 1, while **c** and **d** refer to operator 2



**Fig. 19** Trend of the average penalty and cost of the population of trajectories associated with *UC-H3*. **a** and **b** Refer to the task involving operator 1, while **c** and **d** refer to operator 2



**Table 3** Relative average time saved during the execution of the optimized trajectory with respect to the nominal one

|              | Average time saved |
|--------------|--------------------|
| <i>UC-H1</i> | 65%                |
| <i>UC-H2</i> | 56%                |
| <i>UC-H3</i> | 9%                 |

Notice that during the execution of the trajectories the velocity scaling might be triggered, thus affecting the trajectory traversal time

entails an average reduction of the cycle time by more than 55 % with respect to the corresponding nominal trajectory.

Eventually, the trend of penalty and cost illustrated in Figs. 17, 18 and 19 demonstrates the good convergence properties of the genetic algorithm: the average penalty of the population of trajectories at the end of the iterations is significantly reduced with respect to the corresponding initial value. Notice that this outcome is achieved without considerably penalizing the cost (traversal time) that, in fact, over the iterations, shows limited oscillations around the corresponding initial value.

## 7 Conclusions

In this work a trajectory optimisation method suitable for collaborative robots has been proposed. The goal was to enable the cobot to adapt online its motion trajectory based on long-term occupancy data collected for the human operator that works in close proximity to the cobot. The proposed optimisation method is based on a genetic algorithm that aims at simultaneously minimising the risk of collisions

during the task execution and the traversal time of the trajectory. The optimized robot trajectory is obtained considering that, according to the Speed and Separation Monitoring (SSM) criterion, the robot is prescribed to reduce its speed according to the proximity to the operator, thus potentially penalizing the productivity of the co-working team. The presence of the human operator inside the work-cell is expressed in terms of probabilistic occupancy grid, thus the collision avoidance criterion is relaxed according to the human occupancy probability.

The proposed method was tested on a realistic assembly task that demonstrated its effectiveness: the robot was able to adapt its motion trajectory according to the specific human occupancy, by increasing the separation distance from the human and significantly reducing the intervention of the safety countermeasures, thus improving the productivity.

**Funding** Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abo-Hammour, Z.S., Alsmadi, O.M.K., Bataineh, S.I., Al-Omari, M.A., Affach, N.: Continuous genetic algorithms for collision-free cartesian path planning of robot manipulators. *Int. J. Adv. Robot. Syst.* **8**(6), 74 (2011)
- Antão, L., Reis, J., Gonçalves, G.: Voxel-based space monitoring in human-robot collaboration environments. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 552–559. (2019)
- Balan, L., Bone, G.M.: Real-time 3D collision avoidance method for safe human and robot coexistence. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 276–282. (2006)
- Bilberg, A., Malik, A.A.: Digital twin driven human-robot collaborative assembly. *CIRP Ann.* **68**(1), 499–502 (2019)
- Brock, O., Khatib, O.: Elastic strips: a framework for integrated planning and execution. In: *Experimental Robotics VI*. pp. 329–338. Springer (2000)
- Casalino, A., Bazzi, D., Zanchettin, A.M., and Rocco, P.: Optimal proactive path planning for collaborative robots in industrial contexts. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6540–6546. (2019)
- Ding, H., Reißig, G., Wijaya, K., Bortot, D., Bengler, K., Stursberg, O.: Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction. In: 2011 IEEE International Conference on Robotics and Automation, pp. 5875–5880. (2011)
- Hamid, O.H., Smith, N.L., and Barzanji, A.: Automation, per se, is not job elimination: How artificial intelligence forwards cooperative human-machine coexistence. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), pp. 899–904. (2017)
- ISO TC184/SC2. ISO/TS 15066 Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation, (2013)
- Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., and Schaal, S.: Stomp: stochastic trajectory optimization for motion planning. In: 2011 IEEE International Conference on Robotics and Automation, pp. 4569–4574. (2011)
- Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 14–29 (2016)
- Lasota, P.A., Shah, J.A.: A multiple-predictor approach to human motion prediction. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2300–2307. IEEE (2017)
- Mainprice, J., Hayne, R., Berenson, D.: Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. *IEEE Trans. Rob.* **32**(4), 897–908 (2016)
- Nia, M.B., Alipouri, Y.: Speeding up the genetic algorithm convergence using sequential mutation and circular gene methods. In: 2009 Ninth International Conference on Intelligent Systems Design and Applications, pp. 31–36. (2009)
- Park, J.S., Park, C., Manocha, D.: I-planner: intention-aware motion planning using learning-based human motion prediction. *Int. J. Rob. Res.* **38**(1), 23–39 (2019)
- Pellegrinelli, S., Moro, F.L., Pedrocchi, N., Tosatti, L.M., Tolio, T.: A probabilistic approach to workspace sharing for human-robot cooperation in assembly tasks. *CIRP Ann.* **65**(1), 57–60 (2016)
- Pereira, A., Althoff, M.: Overapproximative human arm occupancy prediction for collision avoidance. *IEEE Trans. Autom. Sci. Eng.* **15**(2), 818–831 (2018)
- Ragaglia, M., Zanchettin, A.M., Rocco, P.: Safety-aware trajectory scaling for human-robot collaboration with prediction of human occupancy. In: 2015 International Conference on Advanced Robotics (ICAR), pp. 85–90. IEEE, (2015)
- Ragaglia, M., Zanchettin, A.M., Rocco, P.: Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. *Mechatronics* **55**, 267–281 (2018)
- Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modeling, Planning and Control*. Springer Science & Business Media (2010)
- Tao, F., Zhang, H., Liu, A., Nee, A.Y.C.: Digital twin in industry: state-of-the-art. *IEEE Trans. Ind. Inform.* **15**(4), 2405–2415 (2018)
- Unhelkar, V.V., Lasota, P.A., Tyroller, Q., Buhai, R.-D., Marceau, L., Deml, B., Shah, J.A.: Human-aware robotic assistant for collaborative assembly: integrating human motion prediction with planning in time. *IEEE Rob. Autom. Lett.* **3**(3), 2394–2401 (2018)
- Wang, Y., Sheng, Y., Wang, J., Zhang, W.: Optimal collision-free robot trajectory generation based on time series prediction of human motion. *IEEE Rob. Autom. Lett.* **3**(1), 226–233 (2018)
- Zanchettin, A.M., Rocco, P.: Probabilistic inference of human arm reaching target for effective human-robot collaboration. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6595–6600. IEEE (2017)
- Zanchettin, A.M., Casalino, A., Piroddi, L., Rocco, P.: Prediction of human activity patterns for human-robot collaborative assembly tasks. *IEEE Trans. Ind. Inform.* **15**(7), 3934–3942 (2019)
- Zanchettin, A.M., Rocco, P., Chiappa, S., Rossi, R.: Towards an optimal avoidance strategy for collaborative robots. *Rob. Comput.-Integrat. Manuf.* **59**, 47–55 (2019)
- Zhao, X., Pan, J.: Considering human behavior in motion planning for smooth human-robot collaboration in close proximity. In: 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 985–990. (2018)
- Ziebart, B.D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J.A., Hebert, M., Dey, A.K., and Srinivasa, S.: Planning-based prediction for pedestrians. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3931–3936. (2009)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Andrea Maria Zanchettin** received the M.Sc. degree in computer science engineering and the Ph.D. degree in information technology from the Politecnico di Milano, Milan, Italy, in 2008 and 2012, respectively. In Spring 2010, he spent a research stay at the Department of Automatic Control (Reglerteknik), Lund University. From January 2012 to February 2014, he was a temporary Research Assistant and from March 2014 to September 2016, a fixed-term Assistant Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, where he is currently an Associate Professor. He has coauthored around 100 papers in automatic control and intelligent human-robot interaction. He has been a member of the IEEE Robotics and Automation Society since 2009. In September 2014, he was a recipient of the Young Author Best Paper Award sponsored by the Italian Chapter of the IEEE Robotics and Automation Society (I-RAS). In 2019, he has been elected as the Chair of I-RAS.





**Costanza Messeri** received the M.Sc. degree in Automation and Control Engineering and the Ph.D degree in Information Technology from Politecnico di Milano, Milan, Italy, in 2018 and 2022, respectively. From June 2018 to October 2018 she conducted research on a project related to the development of a holographic interface to program an industrial manipulator. From November 2018 she joined the Mechatronics and Robotics Lab (MeRLIn) at Politecnico di Milano, conducting research on

several aspects of collaborative robotics. Her research interests concern the optimization of the human-robot interaction, encompassing the estimation of the worker's cognitive stress and physical fatigue, the optimization of the robot strategic behaviour and the use of mixed-reality to facilitate the interaction.



**Davide Cristantielli** achieved the B.Sc. degree in computer science and automation engineering at the Politecnico di Bari, Bari, Italy and the M.Sc. degree in automation engineering at the Politecnico di Milano, Milan, Italy, in 2017 and 2021, respectively. His academic research has focused on optimization of the collaborative robots' motion and on integration of vision devices into robotics systems. He has the role of software developer in the R&D dept for industrial automation companies since 2021.



**Paolo Rocco** is a Full Professor in automatic control and robotics with the Politecnico di Milano, Milan, Italy. He is also a Co-Founder of Smart Robots, a spin-off company of Politecnico di Milano. He has authored about 200 papers in the areas of robotics, motion control, and mechatronics. His current research interests include industrial robotics, with a particular focus on safe and productive human-robot interaction. Dr. Rocco has served in various positions in the editorial boards

of journals and conferences. At present, he serves as a Senior Editor for the IEEE Transactions on Automation Science and Engineering and as an Associate Editor for the IFAC journal Mechatronics. He serves in the Board of Directors of euRobotics, the association of all stakeholders in robotics in Europe, in the Board of Directors of I-RIM, the Italian Institute of Robotics and Intelligent Machines, and in the Board of Governors of Politecnico di Milano. He is also responsible for projects in MADE, the Industry 4.0 Competence Center in Milan. He has been in charge of several research projects with industrial partners and public bodies.