

Optimization-based Trajectory Prediction Enhanced with Goal Evaluation for Omnidirectional Mobile Robots

Wei Luo^a and Peter Eberhard^b

*Institute of Engineering and Computational Mechanics,
University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany*

Keywords: Goal Intention Evaluation, Monte-Carlo Sampling, Optimization, Trajectory Prediction, Complementary Progress Constraint, Mobile Robot.

Abstract: In this paper, an optimization-based trajectory prediction enhanced with goal evaluation for omnidirectional mobile robots is proposed. The proposed approach tries to predict the mobile platform's trajectory based on its previous positions. A two-stage strategy is introduced. At the first stage, the likely goal of the robot in the scenario is evaluated based on an improved Bayesian framework, which also predicts the possible waypoints in a discrete roadmap based on Monte-Carlo sampling in the future. Then, based on the predicted waypoints, an optimization problem is formulated based on the complementary progress constraints, the system dynamics, and the model constraints. After solving the proposed optimization problem, a more reasonable predicted trajectory can be generated. At the end, an experimental scenario is set up, and it is verified with the experimental data, whether the trajectories can be predicted well.

1 INTRODUCTION

Nowadays, robots are widely applied in different applications, such as logistic transport, industrial production (Qian et al., 2017), and disaster relief (Su et al., 2015). In most cases, the robots in the field are organized decentralized. Furthermore, in an environment with humans, the human beings' potential actions cannot be known to robots. Therefore, one of the critical capabilities for these robots is the trajectory prediction of the other robots or human beings in the same working environment. Of course, such a prediction assumes a reasonable 'predictable' behavior and cannot consider sudden changes in the intention. One can increase the cooperation efficiency with the predicted trajectory. For example, if the potential trajectory can be forecasted well, the navigation method can consider this in the own motion planning and avoid likely collisions. The collision probability will be reduced once the trajectory of the other agents can be predicted in advance. This ability is also interesting in the application of autonomous systems, e.g., autonomous driving systems. The trajectory prediction of the other vehicles and also the pedestrians on

the street can help autonomous cars to make a reasonable decision and generate safer paths (Li et al., 2019).

Let us describe the investigated scenario. A flying quadcopter is observing a scene with obstacles and one or several robots on the ground and is recording current position data for future use. The quadcopter wants to approach one of the moving mobile robots on the ground and for the purpose has to complete and permanently update its own trajectory such that at contact time the position and velocity of the quadcopter and mobile ground robot agree. However, this trajectory planning is not part of this paper. In this paper it is considered, how the quadcopter predicts the unknown but most likely path of the mobile ground robot just based on information about its past motion. Of course, this has to assume that the mobile ground robot behaviour in a 'reasonable' way and has a certain intention which should be predicted. Naturally, e.g., a pure random path would not follow an intention and then no prediction would be possible. When the quadcopter collects more and more information about the mobile ground robots past motion and where it approaches him, the prediction will become more and more precise. Note that the quadcopter can neither know the intention future path of the mobile ground robot nor can it influence its mo-

^a  <https://orcid.org/0000-0003-4016-765X>

^b  <https://orcid.org/0000-0003-1809-4407>

tion or trajectory planning. The trajectory planning for the mobile ground robot computed on the quadcopter only serves the purpose to make the best, most likely prediction. The scenario is very similar to the one described in (Best and Fitch, 2015), difference will be commented later. Technically, we introduce a two-stage trajectory prediction strategy for omnidirectional mobile robots, only utilizing the previous observable robot positions in a known environment. The proposed approach tries to identify the movement intention of the observed mobile robot, and predicts its potential trajectory within the reasonable time allocation, see Fig. 1.

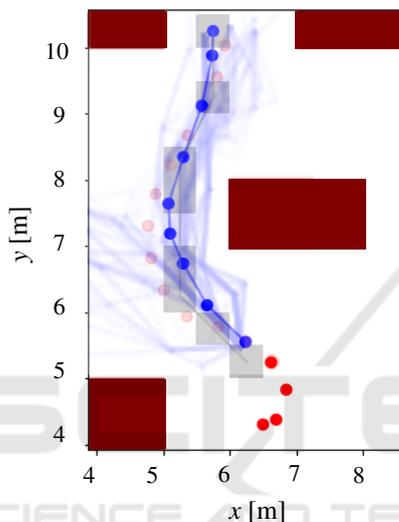


Figure 1: Trajectory prediction based on the previous robot positions marked with red circles in a known environment. The grey squares are the predicted path waypoints, which are based on the sampling results from the shaded traces. The final predicted most likely trajectory is connected by blue circles.

The proposed strategy consists of an improved goal evaluation-based Bayesian framework and an optimization-based trajectory generator. At the first stage, the proposed framework evaluates the likely goal of the observed robot in the known environment and creates a cursory path, which is only composed of several key waypoints that have a relatively high inferential probability. At the second stage, given the estimated most likely path waypoints, an optimization-based framework is formulated based on the complementary progress constraints (CPC), which handles the progress over the predicted waypoints, and the model constraints that are observed in the past positions. Then, the optimization framework is applied for generating a reasonable trajectory prediction for the omnidirectional mobile robot.

Our main contributions for the proposed two-stage trajectory prediction strategy are:

- An improved goal evaluation-based Bayesian path waypoint predictor is introduced, which uses only the previous robot positions, and guesses the likely motion. A momentum parameter is utilized, which significantly improves the efficiency of the trajectory sampling. Furthermore, a motion tendency-based goal intention probability function is applied for evaluating the robot’s potential goal according to its recent positions.
- Instead of using a velocity distribution to predict the trajectory of the observed robot along the linear sample paths in the map graph in (Best and Fitch, 2015), here an optimization-based solution is proposed, to guess an optimized path with a reasonable time-allocation passing through the estimated waypoints under the observable robot constraints based on CPC.

The paper is organized as follows. Section 2 gives an overview of the state-of-the-art in related works. In Section 3, the proposed two-stage approach is described. Finally, the experimental results are illustrated in Section 4, and conclusions are presented in Section 5.

2 RELATED WORK

Trajectory Model-based Method. Trajectory model-based approaches utilize some pre-defined model assumptions to assist the prediction of the possible poses of the observed agent in the future. In (Acuna et al., 2018), the observed agent’s trajectory was assumed as being polynomial, and the observer only needs to find a fitting parameter set for the polynomial trajectory based on the previous trace of the observed agent. In (Schöller et al., 2020), the research showed that even a constant velocity model could make a good prediction of the pedestrian motion compared with state-of-the-art approaches.

Neural Network-based Method. With the rapid development of neural network technology, several researchers utilized neural networks to predict the sequential trajectory of the observed agent. One of the typical network structures, recurrent neural networks (RNN), exhibit the ability to handle time series problems with data-driven techniques. The long short-term memory (LSTM), which is a variation of RNN, was utilized to predict the trajectory of vehicles on the street in (Dai et al., 2019). Some approaches used generative models to predict the trajectory. In (Gupta et al., 2018), the generative adversarial networks (GANs) were first successfully applied

to predict a course of the pedestrian with a recurrent sequence-to-sequence model.

Goal-Conditional-based Method. In some recent works, instead of directly predicting the trajectory of the observed agent, the procedure of the prediction is divided into two or multiple stages. At the first stage, the goal of the observed agent will be evaluated, and then the trajectory will be predicted based on the past data and the evaluated goal. In (Best and Fitch, 2015), a Bayesian mathematical formulation is used to estimate the agent’s intention, and the resulting probability distribution was used to generate the trajectory in the future. In (Dendorfer et al., 2020) the goal condition was combined with the GANs, the proposed method showed a better performance than the typical generative models. However, the physical limitations of the observed agent are ignored, and the predicted trajectory is lack of time information in these works.

3 APPROACH

The task of this paper is to predict the future trajectory of the omnidirectional robot on the 2D ground plane. Note that a trajectory in this work is defined as a path combined with a corresponding time allocation. The past trajectory is available from measurements, and will be utilized as input of the proposed algorithm of this work. At each time step t_i , the proposed algorithm will predict a sequence of the omnidirectional robot positions marked with $\mathbf{X}_{i+1:i+k} := \{\mathbf{x}_{i+j} = (x_{i+j}, y_{i+j}) \in \mathbb{R}^2 | j = 1, \dots, k\} \in \mathcal{X}$ for the next time points t_{i+1}, \dots, t_{i+k} given the past observation set $\mathbf{X}_{1:i}$, where \mathcal{X} denotes the continuous space of the whole scenario.

3.1 Goal Evaluation-based Bayesian Path Waypoint Prediction

At the first stage of the proposed approach, based on the already stored trajectory, the potential goal of the observed robot is evaluated based on the goal intention evaluation, and the future potential path is estimated based on the Bayesian framework. The process of this stage is shown in Algorithm 1.

3.1.1 Goal Intention Evaluation

The proposed method in this work focuses on predicting a likely trajectory for the robot with a certain destination in the known environment. Although the exact goal of the observed robot cannot be known in advance, we assume that the set of all potential goal

Algorithm 1: Bayesian Goal Evaluation-based Path Waypoint Prediction Approach.

```

1: // pre-computation
2: /* generate roadmap based on  $k$ -PRM* */
3:  $DB(\cdot, \cdot), \hat{\mathcal{X}} \leftarrow k\text{-PRM}^*(\mathcal{X}, n_{\text{nodes}})$ 
4: // main while loop
5: while mission in process do
6:   /* get the current robot position from sensor
   and find the closed vertex in roadmap  $\hat{\mathcal{X}}$  */
7:    $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$ 
8:   /* goal intention update*/
9:    $\Pr(\theta_\eta | \mathbf{X}_{1:i}) \leftarrow \text{Eq. 1}$ 
10:  /* trajectory waypoint prediction */
11:  the counting map  $\mathbf{M}_c(\theta_\eta, \bar{\mathbf{x}}, j) \in \mathbb{R}^{n_t \times c_x \times c_y \times k}$ 
12:  for each  $\theta_\eta \in \Theta$  do
13:     $N_\eta \leftarrow N \times \Pr(\theta_\eta | \mathbf{X}_{1:i})$ 
14:    for each  $n \in N_\eta$  do
15:       $\hat{\mathbf{X}}_{i+1:i+k} \leftarrow \text{MC-sampling}(\hat{\mathbf{x}}_i, \theta_\eta)$ 
16:    end for
17:    for each  $j \in [1, k]$  do
18:      /* sum-pooling */
19:       $\bar{\mathbf{x}}_j \leftarrow \text{Pooling}(\hat{\mathbf{x}}_j \in \hat{\mathbf{X}}_{i+1:i+k})$ 
20:       $\mathbf{M}_c(\bar{\mathbf{x}}, j) = \mathbf{M}_c(\bar{\mathbf{x}}, j) + 1$ 
21:    end for
22:  end for
23:  /* waypoint generation*/
24:  for each  $\theta_\eta \in \Theta$  do
25:    for each  $j \in [1, k]$  do
26:       $\mathbf{x}_j^{\text{pred}} \leftarrow \text{argmax}_{\bar{\mathbf{x}}} \mathbf{M}(\theta_{\eta, \text{max}}, \bar{\mathbf{x}}, j)$ 
27:    end for
28:  end for
29: end while

```

regions $\Theta := \{\theta_\eta | \eta \in [1, \dots, n_t]\}$ is feasible and finite, where the number of goal regions is defined as n_t . For instance, a potential goal could be the exit of the scenario, the working station, or the shelves in logistics warehouses, etc.

In this work, we use the probability distribution to describe the intention of the mobile robot to each goal region. Then, a motion tendency-based goal intention probability function is introduced, which only takes the l latest robot positions to evaluate the robot’s motion intention. In each step, given the new coming observation \mathbf{x}_i , the goal intention can be estimated by

$$\Pr(\theta_\eta | \mathbf{X}_{1:i}) \propto \prod_{j=i-l}^i (\exp(f_d(\mathbf{x}_{j-1}, \theta_\eta) - f_d(\mathbf{x}_j, \theta_\eta)) \cdot (f_r(\mathbf{x}_{j-1}, \mathbf{x}_j, \theta_\eta) + 1)), \quad (1)$$

where the function f_d indicates the shortest path distance between two given positions, and the function f_r describes the cosine of the angle between three given

positions in 2D, which is defined as

$$f_r(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \frac{(\mathbf{b} - \mathbf{a}) \cdot (\mathbf{c} - \mathbf{b})}{\|\mathbf{b} - \mathbf{a}\| \|\mathbf{c} - \mathbf{b}\|}. \quad (2)$$

First, the proposed goal intention function evaluates the path distance changes to every evaluating goal region θ_η observing the l latest robot positions. The target could be deemed to be the most likely goal of the robot, if its path distances to the robot in the last l positions are most reduced. Additionally, the goal region which is located behind the direction of motion should have a lower probability of being selected. Hence, a component with the cosine of the angle between the previous robot motion direction and the potential motion direction from the robot's current position to a candidate goal region is multiplied.

3.1.2 Discrete Roadmap

Instead of working on the continuous 2D space, which has infinite possible states to describe the mobile robot, we utilize a discrete roadmap \mathcal{X} at the first stage to represent the robot's position and the potential path between the its current position and the goal region. The discrete roadmap is a graph data structure, which is composed of several randomly distributed nodes and the edges that represent collision-free paths between nodes. The total number of nodes n_{nodes} on the roadmap should be specified balancing the computational time and the coverage rate by the user. The paths through the roadmap are utilized to approximate the potential route of the observed robot. Besides, based on the nodes and the edges on the roadmap, the shortest path between two arbitrary nodes is determined by the A* algorithm. In this work, the k -nearest optimal probabilistic roadmap (k -PRM*) is utilized to create an offline roadmap (Karaman and Frazzoli, 2011).

$$\begin{aligned} r_{\text{PRM}} &= (\sqrt{6}(A_{\text{free-space}}/\pi)^{0.5} + 1) \\ &\quad (\log(n_{\text{nodes}})/n_{\text{nodes}})^{0.5}, \\ k_{\text{PRM}} &= 2e \log(n_{\text{nodes}}), \end{aligned} \quad (3)$$

where the free area of the scenario is marked with $A_{\text{free-space}}$. To improve the real-time computing performance, the shortest paths between nodes and their corresponding path distances will be calculated offline and stored in the database $DB(\mathbf{a}, \mathbf{b})$, where the nodes \mathbf{a} and \mathbf{b} are two arbitrary nodes on the roadmap.

3.1.3 Improved Probabilistic Dynamics Model

To determine the next position \mathbf{x}_{i+1} based on its probability distribution, a probabilistic dynamic model is introduced given the previous position \mathbf{x}_i and the goal

region. Instead of only considering the path distance as in (Best and Fitch, 2015), the probabilistic dynamic model in this work introduces a new parameter β to demonstrate the effect of the linear momentum on the probability distribution. The improved probabilistic dynamics model is defined as

$$\begin{aligned} \Pr(\mathbf{x}_{i+1} | \mathbf{x}_i, \theta_\eta) &\propto \\ &\exp(-\alpha(f_d(\mathbf{x}_i, \mathbf{x}_{i+1}) + f_d(\mathbf{x}_{i+1}, \theta_\eta) - f_d(\mathbf{x}_i, \theta_\eta)))\beta. \end{aligned} \quad (4)$$

Note that it is unnecessary to estimate the probability for every node on the roadmap, and one only needs to consider the nodes in the area $\tilde{\mathcal{X}}_i \in \mathcal{X}$ that can be reached within next time step, see Fig. 2.

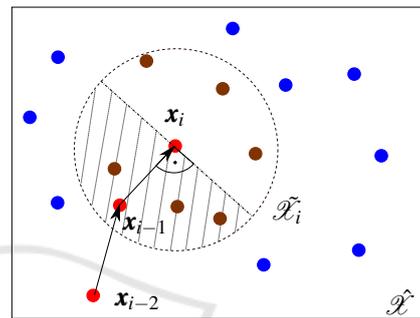


Figure 2: Illustration for the probabilistic dynamic model. The dotted circle illustrates the candidate area $\tilde{\mathcal{X}}_i$, and the nodes, that the robot can arrive within next time step, are marked with brown circles. On the contrary, nodes which are out of range, are marked with blue circles.

In Eq. 4, the parameter α is non-negative and needs to be specified by the user. When $\alpha \rightarrow 0$, the probability for each node in $\tilde{\mathcal{X}}_i$ will be almost identical when the effect of parameter β is ignored. On the contrary, if $\alpha \rightarrow +\infty$, the assumption is that the robot always takes the shortest path to the goal position. However, the choice of the parameter α could be tricky since it is challenging to balance the exploration and the exploitation when only considering the distance relationship with parameter α , especially for the scenarios with multiple goal regions. Therefore, the linear momentum parameter β is introduced

$$\beta = \max\{1e-6, f_r(\mathbf{x}_{i-1}, \mathbf{x}_i, \tilde{\mathbf{x}}_{i+1})\|\mathbf{x}_i - \mathbf{x}_{i-1}\|\}. \quad (5)$$

In Eq. 5, if a candidate $\tilde{\mathbf{x}}_{i+1}$ in next step has a remarkable moving direction difference compared with the last observed motion, the result of the momentum parameter will be set close to zero, which dominates the candidate's probability of being chosen. For instance, given the past course $\mathbf{X}_{i-2:i}$ marked with red circles in Fig. 2, the candidates, which are located in the shadow region, are less likely to be chosen as the potential path waypoint of the mobile robot, based on the proposed probabilistic dynamics model in Eq. 4.

3.1.4 Bayesian Path Waypoint Prediction

Based on the improved probabilistic dynamic model in Eq. 4 and the goal intention evaluation from Eq. 1, one can estimate the next possible position $\hat{\mathbf{x}}_{i+1} \in \hat{\mathcal{X}}$ considering a candidate goal region through

$$\Pr(\mathbf{x}_{i+1}|\mathbf{X}_{1:i}, \theta_\eta) = \Pr(\mathbf{x}_{i+1}|\mathbf{x}_i, \theta_\eta) \times \Pr(\theta_\eta|\mathbf{X}_{1:i}). \quad (6)$$

Intuitively, one can further, based on Eq. 6, recursively estimate the position of the robot \mathbf{x}_{i+j} in the coming time horizon

$$\Pr(\mathbf{x}_{i+j+1}|\mathbf{X}_{1:i}, \theta_\eta) = \sum_{\mathbf{x}_{i+j} \in \hat{\mathbf{X}}_{i+j}} [\Pr(\mathbf{x}_{i+j+1}|\mathbf{x}_{i+j}, \theta_\eta) \times \Pr(\mathbf{x}_{i+j}|\mathbf{X}_{1:i}, \theta_\eta)]. \quad (7)$$

However, as mention in (Best and Fitch, 2015), the analytical evaluation of Eq. 7 is difficult due to the branching factor of the roadmap. Therefore, the trajectory waypoints will be estimated through the Monte-Carlo sampling approach.

Based on the evaluated goal intention probability distribution in Eq. 1, N_η trajectories will be sampled from current position \mathbf{x}_i to the goal region θ_η . In each sampling, the next possible position node $\hat{\mathbf{x}}_{i+j}$ in the region $\hat{\mathbf{X}}_{i+j-1}$ is chosen based on the probability given the improved probabilistic dynamic model in Eq. 4.

Rather than greedily choosing the most sampled nodes at each prediction time step, a sum-pooling procedure in this work is utilized to generate the final predicted waypoints. Each visited node in $\hat{\mathbf{X}}_{i+1:i+j}$ at the time step j will be pooled and converted into a grid graph $\hat{\mathcal{X}} \in \mathbb{R}^{c_x \times c_y}$, where the parameters c_x and c_y are the number of grids in x -/ y -coordinate of $\hat{\mathcal{X}}$, respectively. Then, for each goal region, the visiting times of each grid on the graph will be summed into a counting map \mathbf{M}_c at each prediction time step, which indicates the grid's frequency of being visited. The sum-pooling process sacrifices the prediction accuracy to reduce the distribution unbalance of the generated nodes on the roadmap $\hat{\mathcal{X}}$ and eliminate the prediction of too short paths, especially when a relatively small α in Eq. 4 is chosen. At the end, for each goal region, the most visited grid to the goal region at each prediction time step will be recorded and formulated as the predicted waypoint of the observed robot as $\mathbf{X}_{i+1:i+k}^{\text{pred}} := \{\mathbf{x}_{i+j}^{\text{pred}} \in \mathbb{R}^2 | j \in [1, \dots, k], \theta_\eta\}$.

3.2 Optimization-based Trajectory Prediction

The proposed method in Section 3.1 can effectively provide a rough path based on the goal intention eval-

uation and Monte-Carlo sampling approach. However, the predicted path is discontinuous and only composed of several key waypoints. Besides, the definition of the available area $\hat{\mathbf{X}}_i$ is based on the parameters r_{PRM} and k_{PRM} in Eq. 3. These two parameters concern the probabilistic completeness of the generated roadmap, and the physical limitations of the observed robot are neglected. Therefore, the time step mentioned in last section cannot provide a precious time allocation of the predicted paths. At the second stage, instead of using a velocity model to estimate the time allocation along the predicted line segments in (Best and Fitch, 2015), an optimization-based trajectory prediction in this work is proposed to predict a more reasonable trajectory in the future based on the previously indicated path waypoints.

The proposed optimization formulation will determine an optimized trajectory that fulfils several reasonable constraints. First, the robot's dynamics function and some observable physical limitations need to be satisfied. For instance, the observed maximal absolute velocity and the acceleration can be estimated given the robot's previous trajectory, and they are utilized to bound the predicted state of the mobile robot in the optimization problem. Besides, since the robot has its certain motion intention, the robot is unlikely to linger about the scenario. Therefore, the estimated total travel time and the optimized trajectory distance should be minimized as possible. Furthermore, the predicted trajectory should pass through the previously estimated path waypoints in sequence. To that end, the complementary progress constraints (CPC) are introduced in the proposed optimization problem, which will be detailed in the next section.

3.2.1 Complementary Progress Constraints

At this stage, an optimized trajectory with a fixed time interval will be generated to present the most likely trajectory of the observed omnidirectional mobile robots in the future. The time interval dt is defined as t_N/n_p , where the number of the new generated optimized trajectory nodes is marked as n_p , and t_N is the total travel time of the optimized trajectory. To handle the predicted path waypoints, a progress variable set $\mathbf{\Lambda} := \{\lambda_p \in \mathbb{R}^{n_w} | p = [1, \dots, n_p]\}$ is introduced to indicate whether the optimized trajectory passes through the desired waypoints in a sequence (Foehn and Scaramuzza, 2020). Here, n_w is the number of path waypoints estimated from last section, which satisfies $n_p > n_w$ obviously. Due to the sum-pooling procedure in last section, the number of the waypoints n_w meets $n_w \leq k$, since the predicted waypoint at different time step j may stay at the same grid on the counting map \mathbf{M}_c .

The progress variable λ_p^w in Λ indicates the relationship between the w -th waypoint $\mathbf{x}_w^{\text{pred}}$ and the optimized trajectory node $\check{\mathbf{x}}_p$ at the time step p . If $\check{\mathbf{x}}_p$ has passed through the given the waypoint $\mathbf{x}_w^{\text{pred}}$, the progress variable λ_p^w will become zero; otherwise it will keep its inertial value. To ensure the optimized trajectory passes through the predicted path waypoints in order, the following condition should be fulfilled

$$\lambda_p^w \leq \lambda_p^{w+1}, \quad \forall w \in [1, n_w - 1] \text{ and } p \in [1, n_p], \quad (8)$$

which ensures the optimized $\check{\mathbf{x}}_p$ passes the waypoint $\mathbf{x}_w^{\text{pred}}$ earlier than the next waypoint $\mathbf{x}_{w+1}^{\text{pred}}$. Besides, each element λ_p^w is initialized as one at the beginning of the optimization, and it has the following basic characters further:

$$\begin{cases} 0 \leq \lambda_p^w \leq 1 \\ \lambda_1^w = 1 \\ \lambda_{n_p}^w = 0 \end{cases} \quad \forall w \in [1, n_w] \text{ and } p \in [1, n_p]. \quad (9)$$

Instead of introducing a new progress change parameter in (Foehn and Scaramuzza, 2020) to handle the state switch of the progress variable that may increase the burden of solving the optimization problem, the complementary progress constraints (CPC) in this work are formulated as

$$\begin{aligned} & f_{\text{prog}}(\mathbf{x}_w^{\text{pred}}, \check{\mathbf{x}}_p, \Lambda) \\ & = \underbrace{[(\lambda_p^w - \lambda_{p+1}^w)]}_{P_1} \underbrace{[f_i(\mathbf{x}_w^{\text{pred}}, \check{\mathbf{x}}_p) - v_p^w]}_{P_2} \stackrel{!}{=} 0, \end{aligned} \quad (10)$$

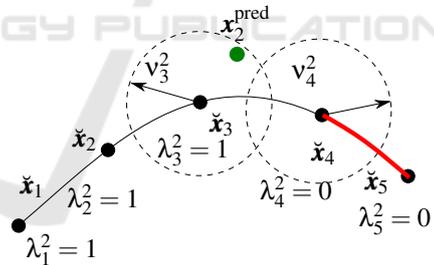
where the function f_i estimates the Euclidean distance between two given positions. Furthermore, it is not a wise strategy to force the optimized trajectory $\check{\mathbf{X}}$ passing all the predicted waypoints $\mathbf{X}_{i+1:i+k}^{\text{pred}}$ exactly. On the one hand, the predicted waypoints may have some outliers, which will strongly impact the result of the optimized trajectory. On the other hand, the accuracy of the predicted waypoints is limited by the grid size (c_x, c_y) from last stage. Therefore, in Eq. 10, a relaxation parameter v_p^w is introduced, which satisfies

$$0 \leq v_p^w \leq d_{\text{tolerance}}, \quad \forall p \in [1, n_p - 1], \quad (11)$$

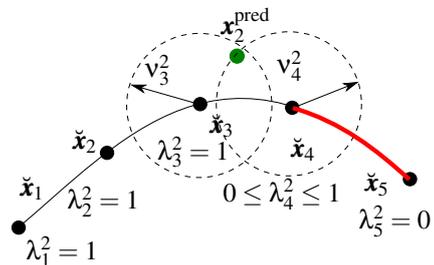
where $d_{\text{tolerance}}$ indicates the maximum acceptable offset to the predicted waypoint. The complementary progress constraint defined in Eq. 10 consists of two components, P_1 and P_2 . If an optimized trajectory waypoint $\check{\mathbf{x}}_p$ gets closed enough to one of the predicted waypoint $\mathbf{x}_w^{\text{pred}}$, which satisfies the area defined in Eq. 11, the component P_2 will become zero. In this case, the next progress variable λ_{p+1}^w for the same waypoint can be reduced to zero to meet the

constraint definitions in Eq. 9. Otherwise, the component P_1 should always stay zero to meet the complementary constraint in Eq. 10.

Ideally, a predicted path waypoint should attract only one optimized trajectory waypoint. For instance, in Fig. 3a, once an optimized trajectory waypoint $\check{\mathbf{x}}_3$ fulfills the tolerance condition to the predicted path waypoint $\mathbf{x}_2^{\text{pred}}$, the rest progress variables $\lambda_{4:n_p}^2$ should go to zeros. However, the constraint defined in Eq. 10 alone cannot prevent a non-optimal trajectory waypoint distribution, see Fig. 3b. Since the position of the $\check{\mathbf{x}}_4$ fulfills all constraints in Eqs. 10-11, the progress variable λ_4^2 has not been restrained at all. In this case, a predicted path waypoint could attract more than one optimized trajectory waypoints, which may result in an unbalance waypoint distribution. Furthermore, the unbalance waypoint distribution may cause an inappropriate estimation of the total travel time. Since the proposed optimized trajectory has a fixed time interval, the total travel time is depended on the maximum track length between every two adjacent optimized trajectory waypoints and the physical limitations of the observed robot. If a non-ideal waypoint distribution occurs, an unexpected long trajectory track may be predicted, which results in an unnecessary long total travel time. To prevent the non-ideal distribution, the sum of all process variables will be minimized in the proposed optimization formulation.



(a) An ideal waypoint distribution.



(b) A non-ideal waypoint distribution.

Figure 3: Illustration for different waypoint distributions with the progress variables. Under the non-ideal waypoint distribution, the trajectory track between $\check{\mathbf{x}}_4$ and $\check{\mathbf{x}}_5$ in Fig. 3b is longer than the one in Fig. 3a, which may lead to errors in the total travel time estimation.

3.2.2 Optimization Formulation

To implement the optimization formulation for the trajectory prediction, the state of the mobile robot and its dynamics will be defined first. Note that, in this work an omnidirectional mobile robot is utilized as the observed target; however, the proposed approach also can be applied to other robots, for which one should just specify the appropriate state definition and dynamic limitations accordingly.

The state of the omnidirectional mobile robot is described as $\check{\mathbf{x}} := [\check{x}, \check{y}, \check{\dot{x}}, \check{\dot{y}}]^T$, and the control input for the robot is assumed to be $\check{\mathbf{u}} := [F, \phi]^T$, where F is the unknown applied control force on the omnidirectional mobile robot, and the angle between the applied force and the x -axis of the global system is defined as ϕ . Although we cannot know the robot's exact mass, we still can assume that the input force is mass normalized, which is proportional to the robot acceleration. Therefore, the dynamic of the omnidirectional mobile robot can be described as

$$f_{\text{dyn}} = [\check{\dot{x}}, \check{\dot{y}}, F \cos(\phi), F \sin(\phi)]^T. \quad (12)$$

The full optimization state set of the optimization problem \mathbf{X}^{opt} consists of the robot states $\check{\mathbf{x}}_p$ and the control inputs $\check{\mathbf{u}}_p$ of the robot, the progress parameter λ_p^w and the relaxation parameter v_p^w at every time step p . Besides, the total travel time t_N is also introduced as one of the optimization states.

The cost function of this optimization problem is composed of three components. First, the total travel time should be short under some observable physical limitations. Then, the total traveled trajectory distance is to be minimized, which makes optimizer prefer a non-aggressive trajectory given the same total travel time. The third component is the sum of all progress variables, which prevents the non-ideal waypoint distribution. Based on the introduction above, the optimization problem is formulated as

$$\begin{aligned} \min_{\mathbf{X}^{\text{opt}}} & \gamma_1 t_N + \gamma_2 \sum_{l=1}^{n_p-1} (\|\check{x}_{l+1} - \check{x}_l, \check{y}_{l+1} - \check{y}_l\|_2^2) \\ & + \gamma_3 \sum_{p=1}^{n_p} \sum_{w=1}^{n_w} \lambda_p^w \\ \text{s.t.} & \\ & dt = t_N/n_p, \\ & \check{\mathbf{x}}_1 = \mathbf{x}_i \\ & \check{\mathbf{x}}_{p+1} = \check{\mathbf{x}}_p + dt f_{\text{RK4}}(\check{\mathbf{x}}_p, \check{\mathbf{u}}_p), \forall p \in [1, n_p - 1] \\ & \mathbf{x}_{\min} \leq \check{\mathbf{x}}_p \leq \mathbf{x}_{\max}, \forall p \in [1, n_p] \\ & \mathbf{u}_{\min} \leq \check{\mathbf{u}}_p \leq \mathbf{u}_{\max}, \forall p \in [1, n_p - 1] \\ & \text{and further constraints based on Eqs. 8 – 11,} \end{aligned} \quad (13)$$

where f_{RK4} is the 4th-order Runge-Kutta approximation of the system dynamic from Eq. 12, and the parameters $\gamma_{1/2/3}$ determine the weights of the total travel time, the trajectory length and the sum of the progress variable set, respectively. Furthermore, the constraints of the control input $\mathbf{u}_{\max, \min}$, and the system state $\mathbf{x}_{\max, \min}$ can be determined relying on the previous robot positions. By implementing of the optimization problem, CasADi (Andersson et al., 2019) is utilized with the solver IPOPT (Wächter and Biegler, 2005).

4 EXPERIMENTAL VALIDATION

To verify the performance of the proposed two-stage approach in this work, a scenario is set up in the simulation platform Gazebo, where the blocks indicate the obstacles which the omnidirectional mobile robot will avoid, see Fig. 4. In the simulation experiment, total of 500 nodes are randomly generated to create a k -PRM* roadmap. Among them, 469 nodes present the possible positions of the omnidirectional mobile robot, and the rest nodes are randomly distributed in the goal regions. So, there are 14539 path connections in total, and the path between each node and its travel distance are estimated offline. The whole offline procedure is processed on a machine with Intel i9 CPU, and the calculation time is less than 5.7 seconds.

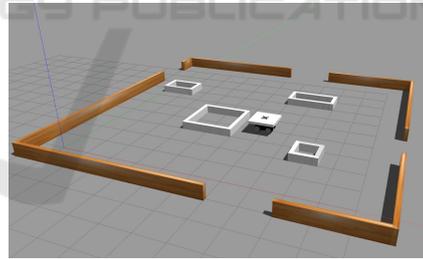


Figure 4: An experimental simulated scenario in Gazebo.

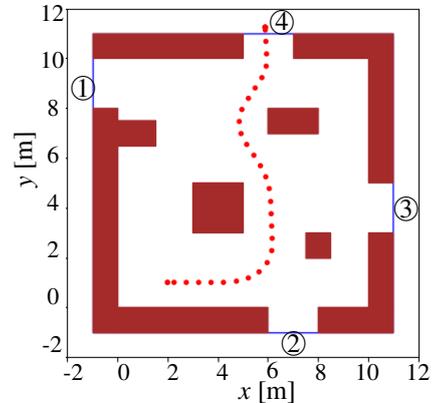


Figure 5: Previously unknown path of the mobile robot.

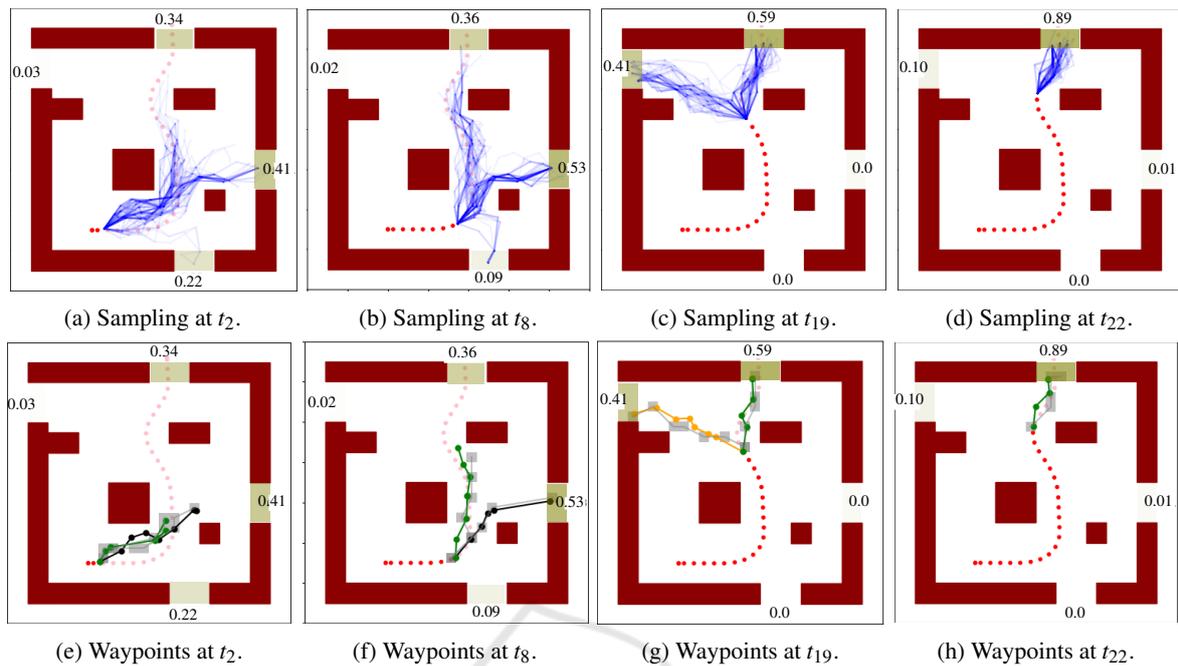


Figure 6: Path waypoint prediction at stage one. Figures 6a- 6d illustrate all predicted paths for the next $k = 8$ prediction steps at the given simulation time step. The corresponded path waypoint estimations are shown in Fig. 6e- 6h.

4.1 Path Waypoint Prediction

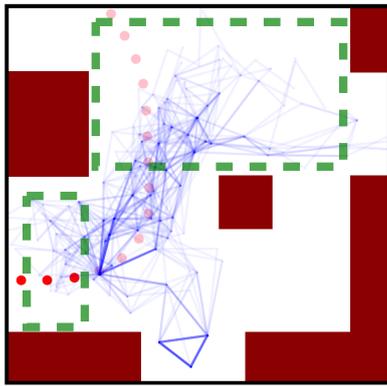
In the experiment, the omnidirectional mobile robot goes to the goal region ④ while avoiding the obstacles in the scenario. The ground truth trajectory of the mobile robot is marked with red circles, which are sampled by 1 Hz, as illustrated in Fig. 5.

In each step, the newly measured pose of the robot will be taken, and the potential path waypoints in the future will be evaluated based on Algorithm 1. In this experiment, 8 time steps ($k = 8$) in the future will be estimated, and in each iteration, $N = 200$ samplings will be processed. The calculation time for the path waypoint prediction in each iteration requires 0.04 seconds on average using the Numba library (Lam et al., 2015).

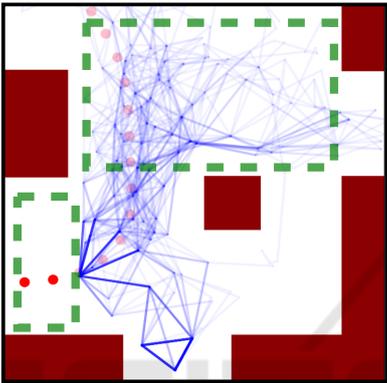
In Fig. 6, the predicted path waypoints at four different simulation time steps are illustrated. Figures 6a-6d show the sampled paths in the next eight time steps in the future based on the Monte-Carlo sampling method. By utilizing the proposed probabilistic dynamics model, the sampling efficiency is improved significantly. In Fig. 7, the proposed method is compared with the model in (Best and Fitch, 2015) under same sampling conditions ($N = 200$, $\alpha = 1.5$). The proposed model has more concentration on sampling the nodes in front of motion direction due to the linear momentum parameter in Eq. 5, instead of wasting the sampling with the nodes behind the current motion tendency, especially on the

areas marked in Fig. 7. Based on the sampling results, the predicted path waypoints at given simulation time steps are illustrated in Figs. 6e-6h, respectively. Note that only the predicted paths to the goal region with a goal intention over 30% will be drawn. The finally predicted path waypoints after the sum-pooling procedure are illustrated with grey squares.

During the experiment, the evaluated goal intention changes for all four goal regions are illustrated in Fig. 8. The proposed goal intention model is compared with the model in (Best and Fitch, 2015) with two different parameter α setups. As expected, the estimated intentions based on the model in (Best and Fitch, 2015) are highly depended on the choice of parameter α . If the parameter α is set to a large value, the evaluated goal intention will increase or decrease drastically. On the contrary, the model will become unresponsive given a small value of α . The proposed goal intention evaluation function provides a relatively stable performance and it can response to the robot moving tendency fleetly. For instance, between the simulation steps t_{18} and t_{20} , the robot may tend to move in the upper-left direction, see Figs. 6c and 6g. In theory, given previous robot positions, both region ① and region ④ should have a similar goal intention to the observed robot during this time. However, compared with the model from (Best and Fitch, 2015), only the proposed model shows a significant response to the potential change of robot's intention.

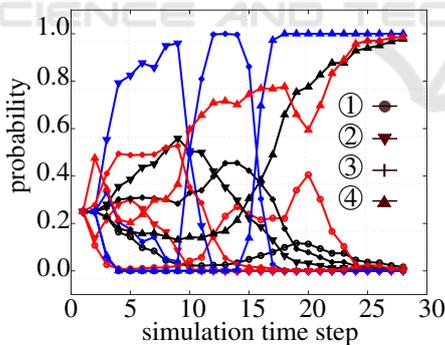


(a) Model in (Best and Fitch, 2015).



(b) Proposed model.

Figure 7: Sampling efficiency comparison.


 Figure 8: Evaluated goal intention changes during the experiment. The result of the proposed goal intention function in red is compared with the intention inference model in (Best and Fitch, 2015) with $\alpha = 0.5/5.0$, which are marked with black and blue, respectively.

4.2 Trajectory Estimation

Once the path waypoints are predicted, the guessed future trajectory of the omnidirectional mobile robot will be estimated by solving the proposed optimization approach. Each trajectory estimation can be accomplished within 0.24 seconds. To quantitatively

verifying the performance, only the predicted trajectories to the goal region ④ are taken in the comparison. In Fig. 9, the predicted trajectories at four different simulation time steps are presented. Although, the future ground-truth trajectory marked with pink circles is not yet known, the proposed method can estimate the potential trajectory based on the sampling results, which does not stay far away from the ground-truth trajectory. Besides, compared with the predicted path waypoints, the predicted trajectories are smooth and continuous, which are more reasonable for the mobile robots.

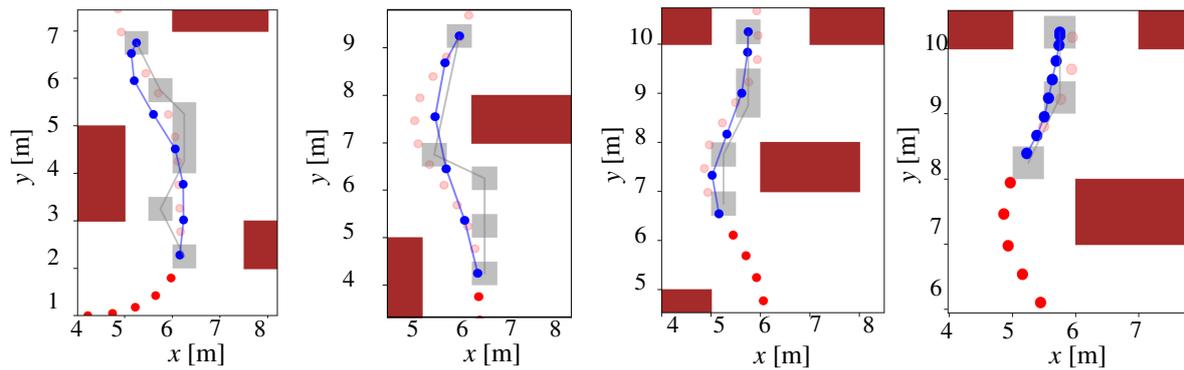
A further advantage of solving the proposed optimization problem is that one obtains not only the predicted trajectory in the future, but also the total travel time t_N of the predicted trajectory. Compared with the vague time step definition for the predicted path waypoints, the optimized trajectory has a fixed time interval between each predicted robot positions. Therefore, every optimized trajectory waypoint has its own estimated arrival time, which is essential for the applications, such as navigation planning and the collision avoidance. In Fig. 10, the predicted trajectories at different simulation time steps (t_{10} , t_{14} , t_{19} and t_{23}) are compared with the ground truth trajectory of the mobile robot. The results on both coordinates show a good prediction, and the maximal error along the prediction time horizon is less than 0.5 m. Considering the grid size of the counting map and the optimization constraint $d_{\text{tolerance}}$ that both are set to 0.5 m, the predicted results are acceptable.

5 CONCLUSION

In this paper, a novel two-stage strategy is proposed for predicting the potential trajectory of an omnidirectional mobile robot given its past trajectory. The effectiveness and efficiency of the proposed strategy are verified in the simulation experiment. The results show that the proposed method can identify goal intentions of the observed robot based on its latest positions, and the errors of the finally predicted trajectory and its allocated time stay in the acceptable range. In future work, one may consider applying the proposed algorithm to hardware experiments in more scenarios.

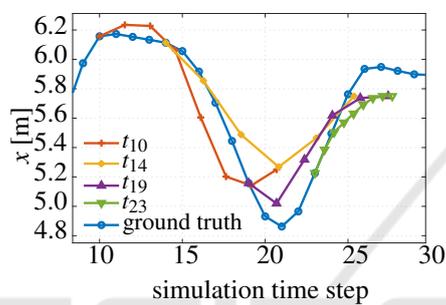
ACKNOWLEDGEMENTS

This research is funded by the German Research Foundation (DFG) under Germany's Excellence Strategy - EXC 2075 - 390740016, project PN4-4 "Theoretical Guarantees for Predictive Control in

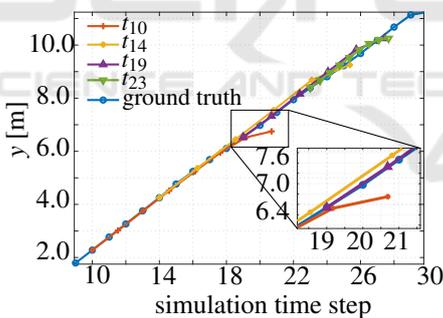


(a) Predicted trajectory at t_{10} . (b) Predicted trajectory at t_{14} . (c) Predicted trajectory at t_{19} . (d) Predicted trajectory at t_{23} .

Figure 9: Results of the optimization-based trajectory prediction. The grey squares are the predicted path waypoints. The final predicted trajectory is marked with blue circles.



(a) Predicted results on the x axis.



(b) Predicted results on the y axis.

Figure 10: The optimized trajectory prediction results at the iteration time step $t_{10/14/19/23}$.

Adaptive Multi-Agent Scenarios”. Also, this research benefited from the support by the China Scholarship Council (CSC, No. 201808080061) for Wei Luo.

REFERENCES

Acuna, R., Zhang, D., and Willert, V. (2018). Vision-based uav landing on a moving platform in gps denied environments using motion prediction. In *Latin American Robotic Symposium, Brazilian Symposium on Robotics (SBR) and Workshop on Robotics in Education (WRE)*, pages 515–521, Joao Pessoa, Brazil.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

Best, G. and Fitch, R. (2015). Bayesian intention inference for trajectory prediction with an unknown goal destination. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5817–5823, Hamburg, Germany.

Dai, S., Li, L., and Li, Z. (2019). Modeling vehicle interactions via modified lstm models for trajectory prediction. *IEEE Access*, 7:38287–38296.

Dendorfer, P., Ošep, A., and Leal-Taixé, L. (2020). GoalGAN: Multimodal trajectory prediction based on goal position estimation. In *Asian Conference on Computer Vision (ACCV)*, pages 1–17, virtual conference.

Foehn, P. and Scaramuzza, D. (2020). CPC: Complementary progress constraints for time-optimal quadrotor trajectories. In *Robotics: Science and Systems*, pages 1–12, virtual conference.

Gupta, A., Johnson, J., Li, F.-F., Savarese, S., and Alahi, A. (2018). Social GAN: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, Salt Lake City, USA.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894.

Lam, S. K., Pitrou, A., and Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, New York, USA.

Li, J., Ma, H., and Tomizuka, M. (2019). Conditional generative neural system for probabilistic trajectory prediction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6150–6156, Macao, China.

Qian, J., Zi, B., Wang, D., Ma, Y., and Zhang, D. (2017). The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. *Sensors*, 17(9):1–15.

Schöller, C., Aravantinos, V., Lay, F., and Knoll, A. (2020).

What the constant velocity model can teach us about pedestrian motion prediction. *Robotics and Automation Letters (RA-L)*, 5(2):1696–1703.

Su, X., Zhang, M., and Bai, Q. (2015). A wireless mobile robots deployment approach for maximising the coverage of important locations in disaster rescues. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 2, pages 17–20, Singapore.

Wächter, A. and Biegler, L. T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

