

*Intent inference in shared-control
teleoperation system in consideration of
user behavior*

**Liangliang Wang, Qiang Li, James Lam,
Zheng Wang & Zhengyou Zhang**

Complex & Intelligent Systems

ISSN 2199-4536

Complex Intell. Syst.

DOI 10.1007/s40747-021-00533-4



Your article is published under the Creative Commons Attribution license which allows users to read, copy, distribute and make derivative works, as long as the author of the original work is cited. You may self-archive this article on your own website, an institutional repository or funder's repository and make it publicly available immediately.



Intent inference in shared-control teleoperation system in consideration of user behavior

Liangliang Wang¹ · Qiang Li² · James Lam¹ · Zheng Wang³ · Zhengyou Zhang⁴

Received: 15 March 2021 / Accepted: 19 August 2021
© The Author(s) 2021

Abstract

In shared-control teleoperation, rather than directly executing a user's input, a robot system assists the user via part of autonomy to reduce user's workload and improve efficiency. Effective assistance is challenging task as it requires correctly inferring the user intent, including predicting the user goal from all possible candidates as well as inferring the user preferred movement in the next step. In this paper, we present a probabilistic formulation for inferring the user intent by taking consideration of user behavior. In our approach, the user behavior is learned from demonstrations, which is then incorporated in goal prediction and path planning. Using maximum entropy principle, two goal prediction methods are tailored according to the similarity metrics between user's short-term movements and the learned user behavior. We have validated the proposed approaches with a user study—examining the performance of our goal prediction methods in approaching tasks in multiple goals scenario. The results show that our approaches perform well in user goal prediction and are able to respond quickly to dynamic changing of the user's goals. Comparison analysis shows that the proposed approaches outperform the existing methods especially in scenarios with goal ambiguity.

Keywords Shared-control teleoperation · User behavior · Learning from demonstration · Intent inference

Introduction

Teleoperation is an effective way for robotic system to implement complex tasks in the unstructured environment [1]. In direct teleoperation system, the real-time input command from the user via an interface is simply mapped to the actuation signal for remote robotic manipulator control in task space. While direct teleoperation systems anticipate exploiting the user capabilities and intelligence to interact with the remote environment, it can result in heavy workload for the user due to the intense concentration on monitoring of remote motion details. For example, consider most

repetitive and tedious point to point space motions, such as approaching to target object and moving to the desired location in “pick and place” task. Direct teleoperation is also limited by the inadequacies of the interface, noise of input, and delay in communication, leading to inefficient and poor task performance, or even failure.

Shared-control teleoperation systems blend user input and robotic autonomy, thus alleviating user burden and improving the efficiency of the executing tasks. Here, shared-control system should tackle two fundamental problems for effective assisting: (1) intent inference, that is predicting the user goal as well as planning the path that the user would like to take towards the predicted goal, and (2) formalizing the arbitration policy to blend user input and the prediction. The two problems are successive steps in shared-control. The correct intent inference is the prerequisite to take effective assistance. Many approaches have been investigated to tackle the intent inference problem. But few of the existing work considered the user behavior in goal prediction and it remains controversial whether the predicted path to the goal is that the user prefers. For example, in grasping task, the user has specific decision and preference on how to approach the objects. As shown in Fig. 1a, to grasp the

✉ Qiang Li
qli@techfak.uni-bielefeld.de

¹ Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, China

² Center for Cognitive Interaction Technology (CITEC), Bielefeld University, Bielefeld, Germany

³ Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China

⁴ Robotics X Lab, Shenzhen, Tencent, China

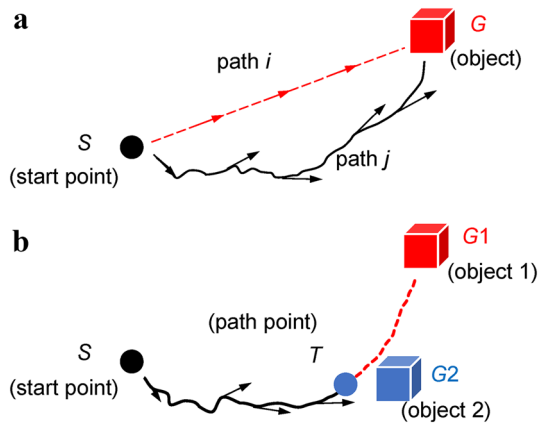


Fig. 1 User behavior in the approaching task. **a** Different paths approaching to the goals, path j is the preferred one but path i is the shortest one. **b** User behavior affects the goal prediction

object G , the user may prefer path j than i , though path i looks more optimized—shorter and less time-consuming. Assistance policy giving predicted path that the user does not prefer might backfire on the overall performance and result in laborious, noisy, and time-consuming movements. User behavior may also affect the prediction of the goal as the scenario illustrated in Fig. 1b. The object G_2 is considered as the intended object by mistake before passing by it. Actually, the object G_1 is the targeted one. Therefore, considering the user behavior in intent inference is nontrivial and will contribute to efficient assistance.

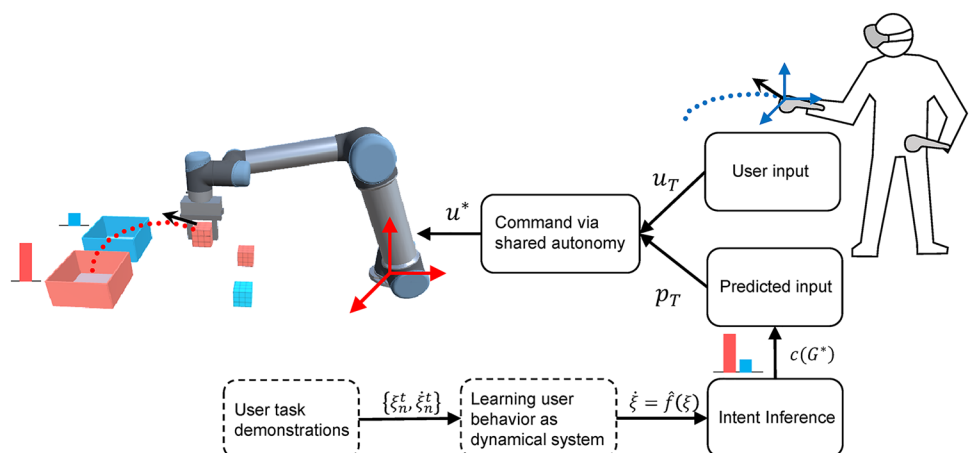
In this work, we focus on tackling the intent inference problem in shared-control teleoperation. Particularly, the user behavior was investigated and considered in goal prediction and planning the path to the goal. We took a primitive motion—point to point teleoperation task as an example

in this study. In the point to point teleoperation task, we assumed that users preferred to adopt the motion paths following their behaviors. Our approach modeled the user behavior and generated the action paths via learning from demonstration (LfD). A simulation system (Fig. 3a) was developed to demonstrate the approaching task in virtual reality. The recorded point to point motions were modeled as nonlinear autonomous dynamical systems (DS) with a single attractor at the goal. The learned DS based user behavior model was then used to compute the similarity metrics between the short-term path of the user and the user behavior. At each time step of teleoperation, maximum entropy principle was employed to model a probability distribution over the candidate goals based on the similarity metrics. A smooth action path to the most likely goal was generated based on the DS based user behavior model. Figure 2 illustrates our system schema. Experimental results indicated that our approaches perform well in goal prediction by taking into account of the user behavior. Especially, our methods could quickly respond to goal changes or perturbations.

The contributions of this work include:

- We considered the user behavior in intent inference under shared-control framework and developed a simulated teleoperation system to sample and model the user's motion behavior.
- For goal prediction, we proposed two metrics—path distance and directionality deviation to measure the similarity of the user executing path to the modeled user behavior. Under the proposed similarity metrics, two short-term memory based goal prediction approaches—short-term path distance (SPD) and short-term directionality deviation (SDD) were respectively proposed using maximum entropy principle.

Fig. 2 System schema. Core components include user behavior learning from demonstrations and user intent inference for shared autonomy assistance



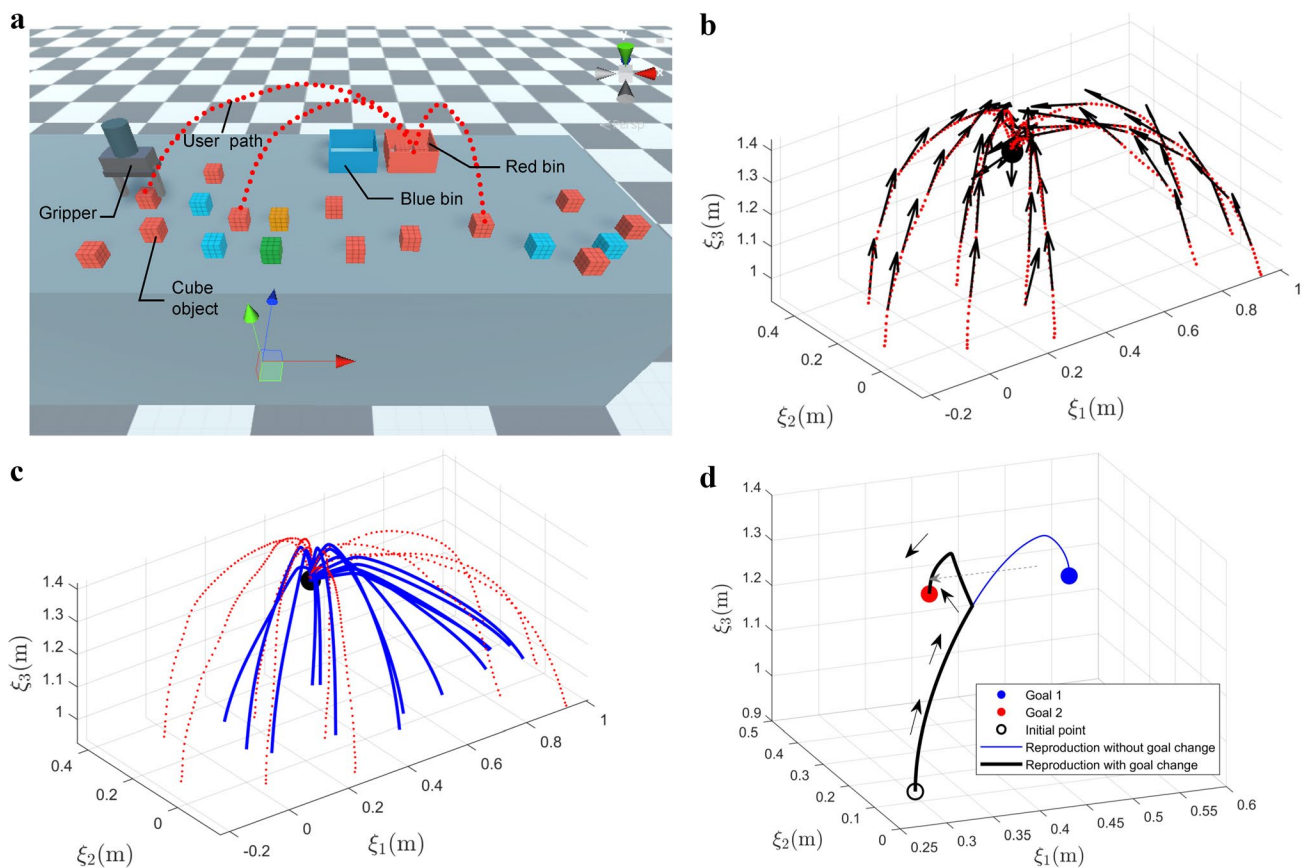


Fig. 3 Learning user behavior from demonstrations. **a** Virtual scene for pick and place task simulations. **b** User demonstrated paths in approaching task. **c** Illustration of generated paths by the learned

model given different starting points. **d** Adaptivity of the model to a goal change during the approaching motion

Related work

The first case involved assistive teleoperation was presented by Goertz [2] in the task of turning cranks. Since then, a great variety of methods have been proposed to introduce autonomy in teleoperation system [3]. For example, Aigner and McCarragher proposed to blend user input with a potential field to avoid obstacles [4]. Gopinath et al. developed an interactive interface allowing user to optimize the arbitration parameters for assistance during teleoperation [5]. A predict-then-blend framework for shared control teleoperation was discussed by Dragan and Srinivasa [6], providing a unifying view of assistance. The predict-then-blend framework were also used in some other studies [7, 8].

Goal prediction problem, which referred to the first task in intent inference, has been widely investigated under various human–robot interaction scenarios. Examples include goal prediction of a shared control teleoperation system considering eye gaze behaviors [9], short-term goal prediction using Bayesian reasoning [10], and task types estimation given the 2D mouse inputs [11]. Many methods have been proposed for addressing the goal prediction problem by leveraging

the machine learning tools, such as Hidden Markov model [12, 13] and Bayesian inference [14]. Recently, Rakita et al. developed a shared control-based bimanual manipulation system. They implemented a sequence-to-sequence recurrent neural network architecture to infer the class of bimanual action, which can be considered as a goal prediction problem with four candidate goals [15]. Li et al. examined three classical machine learning models (neural network, support vector machine, and Bayesian network) and determined the best one for inferring manipulation intent based on the grasping configuration [16]. The well-established principle of maximum entropy was also widely adopted for goal prediction, which was usually relevant to the historical observations of user input [6, 17]. In this work, we also used the principle of maximum entropy for goal prediction. However, unlike most of previous methods using the whole historical data of user input, we exploited the short-term observations for goal prediction in order to enhance the efficiency and adapt to dynamic changes in task environment.

Planning a path that the user prefers to reach the predicted goal is another task in intent inference. Most motion planners, such as probabilistic roadmaps (PRMs) [18, 19]

and rapidly-exploring random trees (RRTs) [20, 21], were effective for planning the path towards the predicted goal in shared autonomy. Alternative approaches were known to use trajectory optimizer to acquire the intended path [22, 23]. These methods could provide optimal solutions in path planning but they might not be the motion strategy that the user prefers to the goal. This phenomenon was depicted by Dragan and Srinivasa [6] in their user study. LfD framework provides an alternative to reproduce the motions from the demonstrated samples. Promising results have been achieved in recent years. Pastor et al. used dynamical movement primitives (DMPs) [24] to learn the pick and place operation and a water-serving task [25]. Calinon et al. proposed to represent the demonstrated motions using Gaussian mixture model (GMM) and reproduce the motion by Gaussian mixture regression (GMR) [26]. Havoutis et al. introduced a task-parameterized Hidden semi-Markov model (TP-HSMM) to learn task representations and generate robot motion from a few demonstrations of ROV teleoperation tasks [27]. With the algorithm, the operator's work was more efficient and the ROV could autonomously execute the learned tasks even communication is lost. Most recently, Zhang et al. built a virtual reality teleoperation system to implement high-quality demonstrations and proposed a deep neural network architecture for learning the tasks from RGBD images and robot arm pose data [28]. This work developed a simulated teleoperation system for capturing user data and LfD approach was employed to learn user behavior model. We gained insight from multiple works within the LfD framework using dynamical systems (DS) [29, 30] to represent motion. Particularly, an approach known as stable estimator of dynamical systems (SEDS) was introduced in our implementation. The SEDS algorithm learned the parameters of a time-independent dynamical system under stability constraints ensuring global asymptotic stability [31].

Materials and methods

User behavior modeling via learning from demonstrations

With the paradigm of LfD, the user behavior can be learned and generated from observations of task demonstrations. We here briefly present the modeling of the motions in approaching task as autonomous dynamical systems (DS) with a single attractor at the target.

Let $\xi \in \mathbb{R}^d$ be the state-space variable that can be used to denote the observation of the user demonstration (i.e., the position of the user hand in Cartesian space). Given N demonstrations of the approaching task where the state

vector and its velocities are recorded at system time steps, $\{\xi_n^t, \dot{\xi}_n^t\}, \forall t \in [0, T_n], n \in [1, N], T_n$ represents total number of data points in the n -th demonstration. These instances are assumed to be generated by a latent function that can be formulated by a first-order autonomous ordinary differential equation (ODE)

$$\dot{\xi} = f(\xi; \theta) + \varepsilon \quad (1)$$

where $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a nonlinear continuous and continuously differentiable function with a single equilibrium point at the attractor $\dot{\xi}^* = f(\xi^*; \theta) = 0$, f is parameterized by θ and ε denotes white Gaussian noise. The optimal values of θ can be obtained from the set of demonstrations using different statistical approaches, such as Gaussian processes regression (GPR), locally weighted projection regression (LWPR) and Gaussian Mixture Regression (GMR). We employed the widely used method GMR in this work to estimate f by encoding demonstrated paths through a mixture of K Gaussian components, which can be determined based on the Bayesian information criterion (BIC) [30, 31]. Using mixture modeling, each recorded point $[\xi_n^t, \dot{\xi}_n^t]$ in the paths is assumed to be generated from a joint probability distribution $P(\xi_n^t, \dot{\xi}_n^t)$

$$P(\xi_n^t, \dot{\xi}_n^t; \theta) = \sum_{k=1}^K P(k)P(\xi_n^t, \dot{\xi}_n^t | k) \quad (2)$$

where $P(k) = \pi^k$ is the prior, satisfying the constraint $\sum_{k=1}^K \pi^k = 1$, and $P(\xi_n^t, \dot{\xi}_n^t | k)$ is the conditional probability density function

$$P(\xi_n^t, \dot{\xi}_n^t | k) = \frac{1}{\sqrt{(2\pi)^{2d} |\Sigma^k|}} e^{-\frac{1}{2}([\xi_n^t, \dot{\xi}_n^t] - \mu^k)^T (\Sigma^k)^{-1} ([\xi_n^t, \dot{\xi}_n^t] - \mu^k)} \quad (3)$$

with the mean $\mu^k \in \mathbb{R}^{2d}$ and the covariance matrices $\Sigma^k \in \mathbb{R}^{2d \times 2d}$ of the k -th Gaussian. $\theta = \{\pi^k, \mu^k, \Sigma^k\}_{k=1}^K$ is a set of complete parameters will be required to estimate f .

Taking the mean estimate of conditional density $E(P(\dot{\xi} | \xi))$ yields the estimate of the latent function:

$$\dot{\xi} = \hat{f}(\xi) = \sum_{k=1}^K \lambda^k(\xi) (A^k \xi + b^k) \quad (4)$$

where $\lambda^k(\xi) = \frac{\pi^k P(\xi | k)}{\sum_{i=1}^K \pi^i P(\xi | i)}$, $A^k = \sum_{\xi \in \xi} (\Sigma^k)^{-1}$, $b^k = \mu_{\xi}^k - A^k \mu_{\xi}^k$.

To ensure that the estimate of $\hat{f}(\xi)$ is globally asymptotically stable at the target, the stable estimator of dynamical system (SEDS) approach [31] can be used to compute optimal values of θ by maximizing the likelihood of the demonstrated instances

$$\min_{\theta} J(\theta) = -\frac{1}{\sum_{n=1}^N T_n} \sum_{n=1}^N \sum_{t=0}^{T_n} \log P(\xi_n^t, \dot{\xi}_n^t; \theta) \quad (5)$$

under strict constraints: (1) $b^k = -A^k \xi^*$, (2) $A^k + (A^k)^T < 0$, $\forall k = 1, \dots, K$.

The estimated model characterizes the user’s decision on the path taking to reach a goal from any point in the task space. Next, we explain how this basic user behavior model is exploited to infer user intent.

User intention inference and assistance

The problem of inferring the user intent can be dealt with two successive steps: (1) predict user’s goal given available data; (2) plan a user preferred motion to the predicted goal.

Goal prediction

One of the critical cues for goal prediction is where the user came from. Prediction can be memory-based by taking into account the historical path of the user inputs. Motivated by previous work about goal prediction based on historical path of the user, we here present short-term memory based goal prediction methods to adapt to the dynamic changes and reduce the high computational cost due to large size of memory data. Rather than using all the historical observations of the user input, our short-term memory based method only considers the latest N observations of the user’s input. Let U_0 denote the observation of user’s input at the starting motion step, U_T denote the current observation of user input at motion step T , $\zeta_{U_{T-(N-1)} \rightarrow U_T}$ denote the short-term sequential observations of the user inputs from $U_{T-(N-1)}$ until U_T , and Ω denote a set of approachable goals in the task environment. The problem of predicting the intended goal G^* given $\zeta_{U_{T-(N-1)} \rightarrow U_T}$, can be formulated as

$$G^* = \arg \max_{G \in \Omega} P(G | \zeta_{U_{T-(N-1)} \rightarrow U_T}) \quad (6)$$

We implement an intuitive idea to predict the user goal: a goal is more likely the intended one, if the user is taking the short-term movements $\zeta_{U_{T-(N-1)} \rightarrow U_T}$ that is more similar to the way the user prefers to behave towards that goal. That is, we assume that the user prefers the path towards a goal following similar dynamics of the behavior in the demonstrations. Here, two metrics are proposed to be used for describing the similarity $C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})$ on goals $G \in \Omega$ given $\zeta_{U_{T-(N-1)} \rightarrow U_T}$: (1) path distance; (2) directionality deviation.

Path distance

The user behavior model $\hat{f}(\xi)$ can be utilized to predict a path according to user preference from a point to any approachable goals in the task environment. Let $\zeta_{U_{T-(N-1)} \rightarrow G}^*$ denote the predicted path from motion step $U_{T-(N-1)}$ to G and $\xi_{U_T \rightarrow G}^*$ denote the predicted path from current motion step U_T to G . We then concatenate the short-term user path $\zeta_{U_{T-(N-1)} \rightarrow U_T}$ with the path $\xi_{U_T \rightarrow G}^*$ into a complete path to G through U_T , which is denoted as $\zeta_{U_{T-(N-1)} \rightarrow U_T \rightarrow G}$. Next, the path distance indicating the similarity $C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})$ between $\zeta_{U_{T-(N-1)} \rightarrow U_T \rightarrow G}$ and $\zeta_{U_{T-(N-1)} \rightarrow G}^*$ is defined as

$$C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T}) = D(\zeta_{U_{T-(N-1)} \rightarrow U_T \rightarrow G}, \zeta_{U_{T-(N-1)} \rightarrow G}^*) \quad (7)$$

where $D(\zeta_{U_{T-(N-1)} \rightarrow U_T \rightarrow G}, \zeta_{U_{T-(N-1)} \rightarrow G}^*)$ is the dynamic time warping distance (DTWD) between path $\zeta_{U_{T-(N-1)} \rightarrow U_T \rightarrow G}$ and $\zeta_{U_{T-(N-1)} \rightarrow G}^*$. The path distance metric measures how close the observed user path behaves to the path that the user prefers to a goal. Under this metric, the method predicts the goal with smallest path distance.

Directionality deviation

The directionality deviation metric explores the similarity of the motion directions between the observed user movements and that the user prefers. Given the data of latest N motion steps, it is considered that the goal is likely not the intended one with larger cumulative directionality deviation between the observed user movement velocity and the preferred one to the goal. At each motion step, the velocity following the user behavior can be estimated by model $\hat{f}(\xi)$. The similarity $C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})$ indicated by the directionality deviation metric then is defined as

$$C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T}) = \sum_{t=T-(N-1)}^T \left| 1 - \frac{\hat{f}(\xi_t)^T \dot{\xi}_t}{\|\hat{f}(\xi_t)\| \|\dot{\xi}_t\|} \right| \quad (8)$$

Using the maximum entropy principle (MEP), the optimal prediction G^* can be deduced by the similarity $C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})$:

$$G^* = \arg \max_{G \in \Omega} P(G | \zeta_{U_{T-(N-1)} \rightarrow U_T}) = \arg \max_{G \in \Omega} \frac{e^{-\beta C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})}}{\sum_{g \in \Omega} e^{-\beta C_g(\zeta_{U_{T-(N-1)} \rightarrow U_T})}} \quad (9)$$

where β is an adjust parameter. The optimal value of β can be derived as $\beta_{\text{opt}} = -\frac{\ln \varepsilon}{C_{G,\min}}$, where ε is a small positive constant (such as $\varepsilon = 0.01$) [32] and $C_{G,\min} = \min_{G \in \Omega} C_G(\zeta_{U_{T-(N-1)} \rightarrow U_T})$. Both the two metrics of the similarity are adopted in goal prediction, where we name our goal prediction method short-term path distance (SPD) based on path distance metric and short-term directionality deviation (SDD) based on directionality deviation metric.

Evaluating confidence

Assistance with wrong goal prediction might backfire on the overall performance. To provide effective assistance, we should take into account the confidence in the goal prediction, $c(G^*)$. Here, we measure it by the difference between the probability of the predicted goal and that of the next most probable candidate [6]

$$c(G^*) = P(G^* | \zeta_{U_{T-(N-1)} \rightarrow U_T}) - \max_{G \in \Omega \setminus G^*} P(G | \zeta_{U_{T-(N-1)} \rightarrow U_T}) \quad (10)$$

Motion planning based on user behavior

Based on the user behavior model learned from demonstrations, we can easily plan a user preferred motion at every motion step once the goal has been predicted. At motion step T , given the predicted goal G^* located relatively at P_{G^*} , the motion in next step p_T can be obtained by the user behavior model converging to G^*

$$p_T = \hat{f}(\xi_T - P_{G^*}) \quad (11)$$

Assistance under shared-control framework

To formulate assistance, a blending function can be implemented under shared-control paradigm [6, 8]

$$u^* = (1 - \alpha) \cdot u_T + \alpha \cdot p_T \quad (12)$$

where u_T denotes the user input, p_T denotes the planned motion to the predicted goal, u^* is the shared-control command sent to the robot, and $\alpha \in [0, 1]$ is a blending factor which decides how much autonomy control is blended in the task. In the implementation, α is a piecewise linear function of the confidence in the goal prediction. The switching threshold of confidence $c(G^*)$ is set to σ_{\min} below which assistance is not active, i.e., $\alpha = 0$, while α achieves to the maximum (α_{\max}) and keep a constant if confidence $c(G^*)$ larger than σ_{\max} .

Experimental results

In this section, we conducted a user study in approaching scenario, and aimed at analyzing the prediction methods on the teleoperation data. The experiments were performed with an in-house developed simulated teleoperation system. This simulation system was implemented by game engine Unity3D. The HTC VIVE headset was used for immersive displays while the controllers with buttons were configured for tracking user's hand pose as input and controlling the gripper.

With the simulation system, user approaching demonstrations were firstly sampled. Based on the recorded data (3D position, 50 Hz), user behavior model was learned in MATLAB using SEDS [31]. The learned user behavior model was then incorporated in the implementation of the proposed SPD and SDD methods.

User behavior learned from demonstrations

In the simulation system, the user demonstrated the approaching task as shown in Fig. 3a. The task was to pick up small cubes located at different positions on the table using the gripper, transporting and placing them into a bin. The paths of gripper motion from different cube locations to the bins are recorded.

Figure 3b illustrated the 10 demonstrated paths (dotted red line) obtained from a single user. The direction of motion was indicated by arrows. The user behavior model learned via SEDS [31] was proved to be able to generate paths (thick blue line) following the same dynamics given new starting positions, as showed in Fig. 3c. The learned DS model was also robust if the goal was changed during the task execution. Figure 3d illustrated a case that the model could adapt to the new goal. In the case, the original path (thin blue line) was planned to Goal 1 while a new path is used if the intended goal is changed to Goal 2.

Goal prediction study

We evaluated the proposed goal prediction methods—SPD and SDD, and compared them with Amnesic and Memory-based prediction [6]. The goal prediction was computed during the teleoperation without assistance. For goal prediction, the amnesic and memory-based approaches tackled with the historical information in different way and they both ignored the user behavior.

Amnesic prediction

The amnesic approach predicts the intended goal based on the distance to the goal and the closest goal gets highest probability. This approach only considers the current observation but neglects all the historical information.

Memory-based prediction

The memory-based approach takes into account all the historical observations of the trajectory for predicting the most likely goal. Based on the principle of maximum entropy, the predicted goal g^* is given as

$$g^* = \arg \max \frac{e^{(-c_g(\zeta_{s \rightarrow x}) - c_g(\zeta_{x \rightarrow g}^*))}}{e^{-c_g(\zeta_{s \rightarrow g}^*)}} P(g) \tag{13}$$

where s is the starting point, x is the current point, and g is the goal, $c_g(\zeta_{s \rightarrow x})$ is the cost of history trajectory between s and x , $c_g(\zeta_{x \rightarrow g}^*)$ is the cost of the optimal trajectory between x and g , $c_g(\zeta_{s \rightarrow g}^*)$ is the cost of the optimal trajectory between s and g , and $P(g)$ is the prior of g .

We compared the performance of these four approaches in two scenarios: (i) no change of goal, where the user maintained a single goal from start to end, and (ii) change of goal, where the user changed the intended goal during the course of approaching. The study involved two goals. The cubes were located at different positions and required to be picked up and placed into 2 potential bins, i.e., blue bin (Goal 1) and red bin (Goal 2), as shown in Fig. 4. For each cube, the user was instructed to (i) pick up the cube and place into the red bin and (ii) perform an additional trial in which the targeted bin was changed from the red bin to the blue one during the approaching. The time stamp of the goal change was recorded via a button press. Starting from the initial

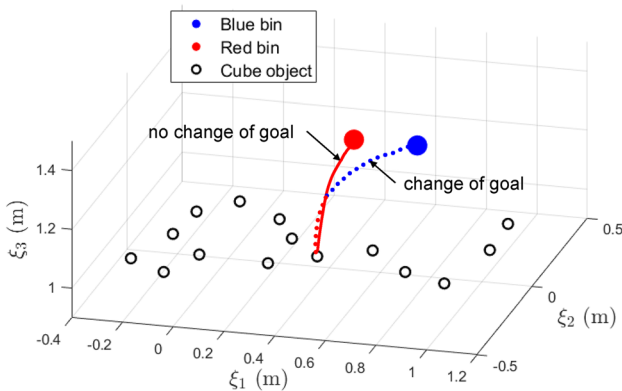


Fig. 4 Schematic diagram of the user task—picking cubes into two bins

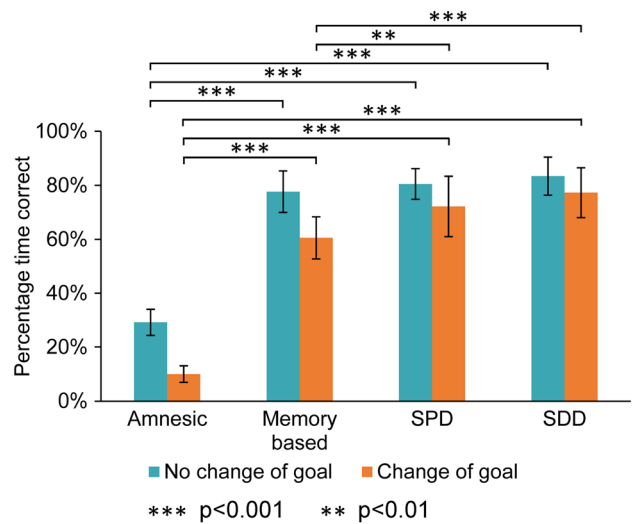


Fig. 5 Performance of the four methods in goal prediction

position of the task, all four approaches were used for predicting the goal at each motion step and the predicted results were recorded.

We assessed the performance of the four methods by analyzing the percentage of time that the predictions were correct. Figure 5 shows the performance of the four prediction methods in the approaching tasks. Over all, SDD outperformed other approaches for the tasks with and without goal change. According to the statistical analysis, the amnesic method performed the worst compared to the other three methods ($p < 0.001$), even in most of time it outputs a wrong prediction. Both the proposed SPD and SDD methods performed slightly better than the memory-based method, but it was not significant. However, it was found that the performance of SPD and SDD methods were not enormously affected by the change of user’s intended goal during task execution. Compared to the memory-based method, the short-term memory based methods SPD ($p < 0.01$) and SDD

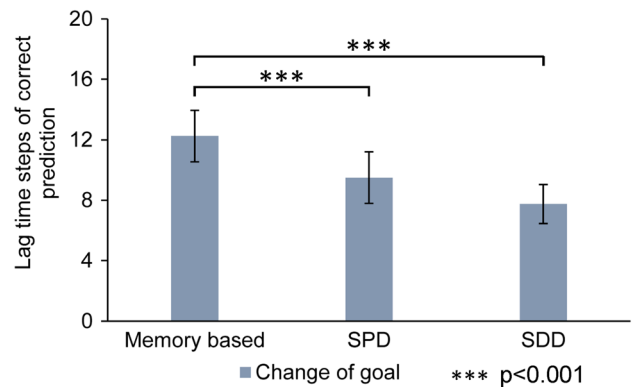


Fig. 6 Lag time steps before providing correct prediction after goal change

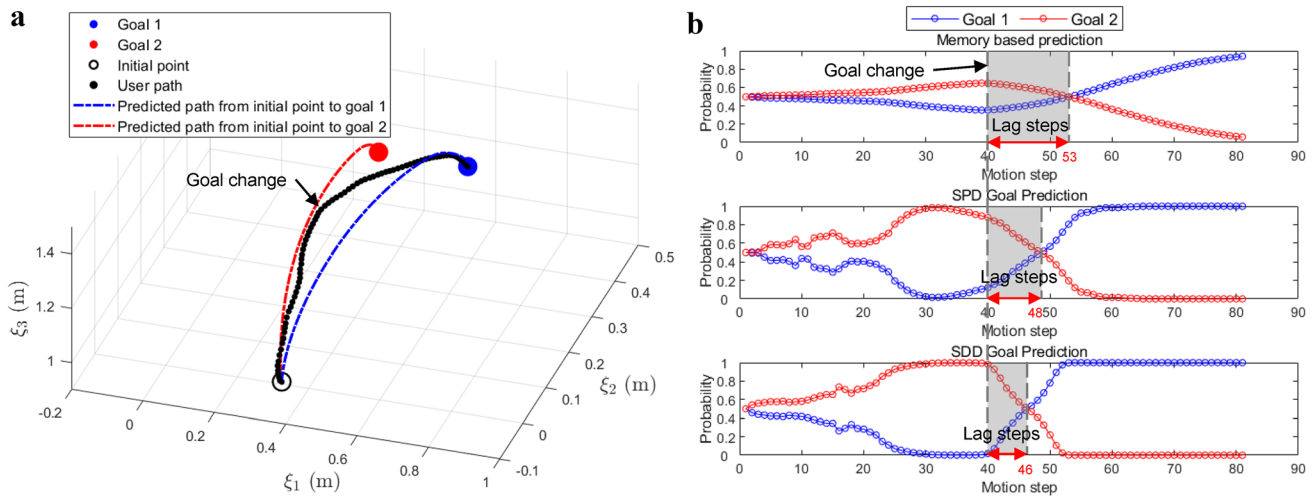


Fig. 7 A concrete example of goal prediction in the scenario with goal change. **a** At 40th motion step, user changed the intended goal from Goal 2 to Goal 1. **b** Prediction results during the user approaching motion

($p < 0.001$) were more robust in responding to the dynamic goal change. Further analysis confirmed that the predictions from SPD and SDD were faster than memory-based method ($p < 0.001$) if the intended goal was changed. The averaged lag time of correct prediction was less than 10 steps as shown in Fig. 6. Figure 7 presents a concrete example of the goal prediction in the scenario with goal change. As the user changed the intended goal at the 40th motion step, SDD predicted the goal change after 6 motion steps, SPD took 8 motion steps for correctly predicting the goal change, while Memory-based method performed the worst, lagging 13 motion steps. This is because the memory-based method used the whole historic trajectory observations for goal prediction while SPD and SDD only used short-term trajectory data. In the scenario that the intended goal was changed, the Memory-based method needs more steps for predicting of the new goal, thus causing the lag response.

Exploratory cases

We showed that our goal prediction method could work in more challenging scenarios, which were firstly presented by Dragan and Srinivasa [6], as illustrated in Fig. 8a and c. With theoretic proof and experimental trials, Dragan and Srinivasa found the limitations of the Memory-based prediction method—the prediction was incorrectly biased towards to the further goal when the goals were collinear with the start point or towards to the rightmost goal. Compared to the Memory-based prediction method, we considered the user behavior in our prediction approaches (SPD and SDD). We tested the memory-based, SPD, and SDD methods on the scenario that two goals were collinear with the start point, as shown in Fig. 8a. The results in Fig. 8b indicated that

the Memory-based method was biased to the farther goal (Goal 2) during the approaching motion, while our methods SPD and SDD were able to predict the goal correctly. We also conducted an exploratory experiment on another scenario with unintended goal on the right of the intended one as shown in Fig. 8c. According to the prediction results in Fig. 8d, during the path heading to intended goal (Goal 1), the Memory-based method always predicted the intended goal to be Goal 2 before the user passing by it. The SPD and SDD methods were also robust to this scenario and provided correct predictions in the whole path thanking to the consideration of the user behaviors that learned from demonstration.

Conclusion and future work

In this work, we presented a mathematical formulation for intent inference in shared-control teleoperation. Particularly, we considered the user behavior in the intent inference problem. A simulated teleoperation system was developed to sample and learn user's approaching behavior. Using the learned behavior model, user-preferred path could be generated with any given goal. We proposed SPD and SDD methods to tackle with the goal prediction problem. User studies were conducted to examine the efficiency of goal prediction on approaching task. Results showed that our approaches were able to achieve sound performance of goal prediction even in dynamic environment, e.g., intended goal changed during the approaching task. Exploratory experiments were also conducted in some special scenarios, where the existing memory-based approach tended to fail. Our methods were able to adapt to those special scenarios because a better

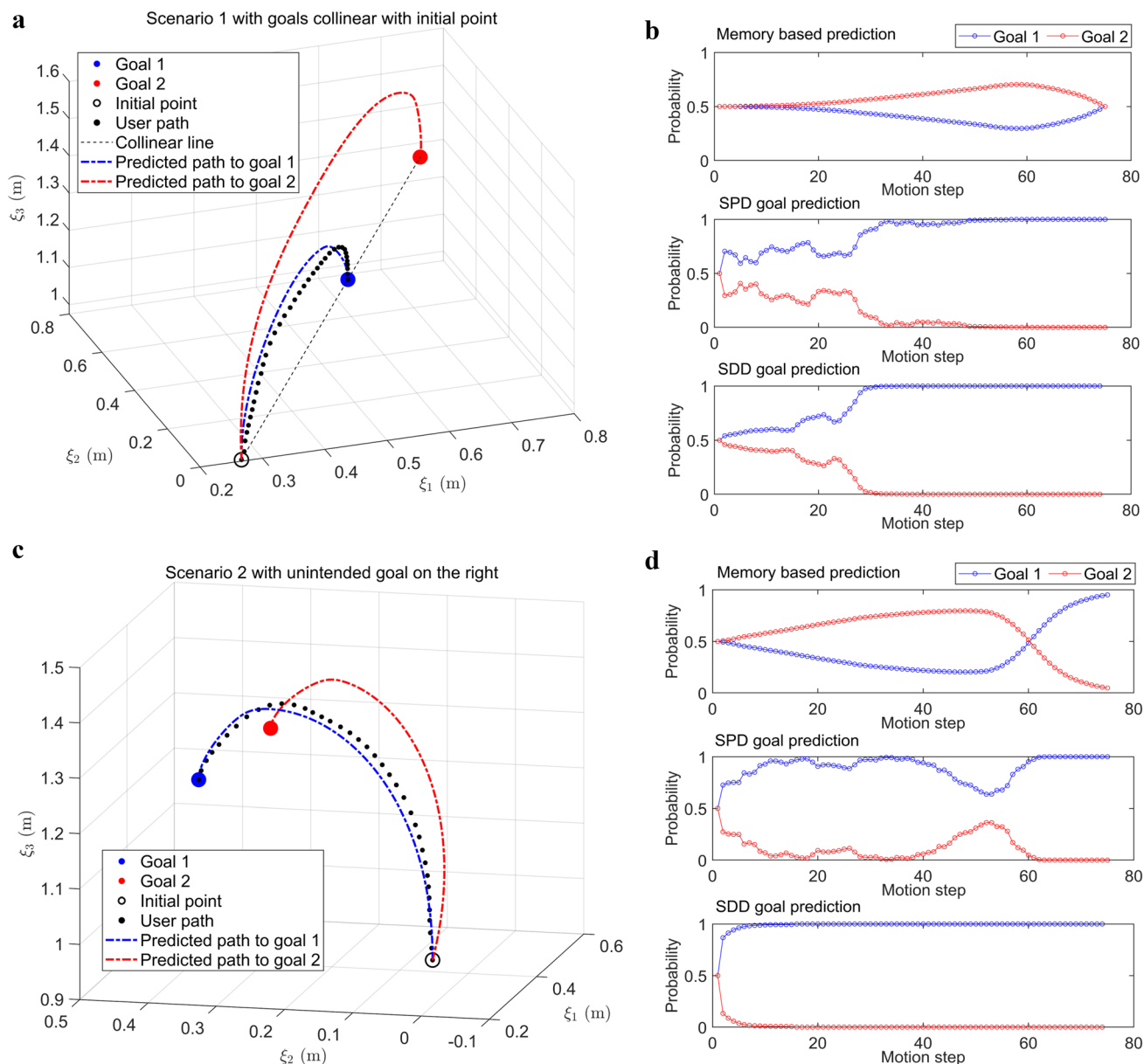


Fig. 8 Goal prediction in two exploratory cases. **a** Scenario 1 with goals collinear with the initial point: the user performed approaching motion to Goal 1. **b** Prediction results during the user approaching motion to Goal 1 in scenario 1. **c** Scenario 2 with unintended goal

on the right: the user performed approaching motion to Goal 1. **d** Prediction results during the user approaching motion to Goal 1 in scenario 2

model of user behavior was learned. Overall, the proposed system was able to provide efficient prediction in consideration of user behavior.

In teleoperation task, the user behavior is one of the critical factors which should be taken into account. In the presented work, the results have shown that the blended input based on the intent inference could directly affect the user’s behavior during task execution. Current work exploited a learned static user behavior model. In future work, the user’s real time behavior will be taken into account to modify the

offline learned model so as to obtain robust intent inference. In addition, in the current method, we model the user behavior in a simple scenario, e.g., there are no obstacles in the working space. In the scenario where the robot is deployed, it is quite possible that there are obstacles. Therefore, how the presence of obstacles affects the intent inference with current algorithm is an open question that could be addressed in the future studies. Other interesting research points following the current work can also be: (1) studying the feasibility to obtain a generalized behavior model among

the different users, (2) considering more goal variables in intent inference, e.g., including the orientation, shape as the goal variables.

Acknowledgements This work is supported in part by the “DEXMAN” project (Project number: 410916101) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Lian-gliang Wang and Qiang Li would like to thank for the support of Ten-cent Robotics X Lab in preparing the work.

Author contributions LW and QL conceived the methodology, the experiment, and validated the results. LW wrote the software. LW, QL, and ZZ contributed to write the manuscript. James Lam and Zheng Wang oversaw the work. All authors read and approved the final manuscript.

Funding This work is supported in part by the “DEXMAN” project (Project number: 410916101) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

Availability of data and material Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

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

- Niemeyer G, Preusche C, Stramigioli S, Lee D (2016) Telerobotics. In: Siciliano B, Khatib O (eds) Springer handbook of robotics, 2nd edn. Springer, Berlin, pp 1085–1108
- Goertz R (1963) Manipulators used for handling radioactive materials. In: Bennett EM (ed) Human factors in technology. McGraw-Hill, New York
- Lipton JI, Fay AJ, Rus D (2018) Baxter's homunculus: virtual reality spaces for teleoperation in manufacturing. *IEEE Robot Autom Lett* 3(1):179–186. <https://doi.org/10.1109/LRA.2017.2737046>
- Aigner P, McCarragher B (1997) Human integration into robot control utilising potential fields. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ROBOT.1997.620053>
- Gopinath D, Jain S, Argall BD (2017) Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robot Autom Lett* 2(1):247–254. <https://doi.org/10.1109/LRA.2016.2593928>
- Dragan AD, Srinivasa SS (2013) A policy-blending formalism for shared control. *Int J Robot Res* 32(7):790–805. <https://doi.org/10.1177/0278364913490324>
- Xi B, Wang S, Ye X, Cai Y, Wang R (2019) A robotic shared control teleoperation method based on learning from demonstrations. *Int J Adv Robot Syst* 16(4):1–13. <https://doi.org/10.1177/1729881419857428>
- Jain S, Argall B (2018) Recursive bayesian human intent recognition in shared-control robotics. *Proc IEEE/RSJ Int Conf Intell Robot Syst.* <https://doi.org/10.1109/IROS.2018.8593766>
- Henny A, Srinivasa SS (2016) Predicting user intent through eye gaze for shared autonomy. AAAI fall symposium series: shared autonomy in research and practice. AAAI Press, London, pp 298–303
- Narayanan VK, Spalanzani A, Babel M (2016) A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance. *Proc IEEE/RSJ Int Conf Intell Robot Syst.* <https://doi.org/10.1109/IROS.2016.7759691>
- Kris H (2013) Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Auton Robot* 35(4):241–254. <https://doi.org/10.1007/s10514-013-9350-3>
- Yu W, Alqasemi R, Dubey R, Pernalet N (2005) Telemanipulation assistance based on motion intention recognition. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ROBOT.2005.1570266>
- Aarno D, Kragic D (2008) Motion intention recognition in robot assisted applications. *Robot Auton Syst* 56(8):692–705. <https://doi.org/10.1016/j.robot.2007.11.005>
- Schultz C, Gaurav S, Monfort M, Zhang L, Ziebart BD (2017) Goal-predictive robotic teleoperation from noisy sensors. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ICRA.2017.7989633>
- Rakita D, Mutlu B, Gleicher M, Hiatt LM (2019) Shared control-based bimanual robot manipulation. *Sci Robot* 4(30):eaaw0955. <https://doi.org/10.1126/scirobotics.aaw0955>
- Li S, Bowman M, Nobarani H, Zhang X (2020) Inference of manipulation intent in teleoperation for robotic assistance. *J Intell Robot Syst* 99(1):29–43. <https://doi.org/10.1007/s10846-019-01125-8>
- Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. *Proc AAAI Conf Artif Intell.* <https://doi.org/10.5555/1620270.1620297>
- Siméon T, Laumond JP, Nissoux C (2000) Visibility-based probabilistic roadmaps for motion planning. *Adv Robot* 14(6):477–493. <https://doi.org/10.1163/156855300741960>
- Siméon T, Laumond JP, Cortés J, Sahnani A (2004) Manipulation planning with probabilistic roadmaps. *Int J Robot Res* 23(7–8):729–746. <https://doi.org/10.1177/0278364904045471>
- LaValle SM (1998) Rapidly-exploring random trees: a new tool for path planning. TR 98–11, Computer Science Department, Iowa State University. <http://msl.cs.illinois.edu/~lavalle/papers/Lav98c.pdf>. Accessed 22 Sept 2021
- Kuffner JJ, LaValle SM (2000) RRT-connect: an efficient approach to single-query path planning. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ROBOT.2000.844730>
- Kalakrishnan M, Chitta S, Theodorou E, Pastor P, Schaal S (2011) STOMP: stochastic trajectory optimization for motion planning.



- Proc IEEE Int Conf Robot Autom. <https://doi.org/10.1109/ICRA.2011.5980280>
23. Zucker M, Ratliff N, Dragan AD, Pivtoraiko M, Klingensmith M, Dellin CM, Bagnell JA, Srinivasa SS (2013) CHOMP: covariant hamiltonian optimization for motion planning. *Int J Robot Res* 32(9–10):1164–1193. <https://doi.org/10.1177/0278364913488805>
 24. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S (2013) Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput* 25(2):328–373. https://doi.org/10.1162/NECO_a_00393
 25. Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and generalization of motor skills by learning from demonstration. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ROBOT.2009.5152385>
 26. Calinon S (2016) A tutorial on task-parameterized movement learning and retrieval. *Intel Serv Robot* 9(1):1–29. <https://doi.org/10.1007/s11370-015-0187-9>
 27. Havoutis I, Calinon S (2019) Learning from demonstration for semi-autonomous teleoperation. *Auton Robot* 43:713–726. <https://doi.org/10.1007/s10514-018-9745-2>
 28. Zhang T, Mccarthy Z, Jow O, Lee D, Chen X, Goldberg K, Abbeel P (2017) Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *Proc IEEE Int Conf Robot Autom.* <https://doi.org/10.1109/ICRA.2018.8461249>
 29. Hersch M, Guenter F, Calinon S, Billard A (2008) Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Trans Robot* 24(6):1463–1467. <https://doi.org/10.1109/TRO.2008.2006703>
 30. Khansari-Zadeh SM, Billard A (2012) A dynamical system approach to realtime obstacle avoidance. *Auton Robot* 32(4):433–454. <https://doi.org/10.1007/s10514-012-9287-y>
 31. Khansari-Zadeh SM, Billard A (2011) Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Trans Robot* 27(5):943–957. <https://doi.org/10.1109/TRO.2011.2159412>
 32. Li L, Ji H, Gao X (2006) Maximum entropy fuzzy clustering with application to real-time target tracking. *Signal Process* 86(11):3432–3447. <https://doi.org/10.1016/j.sigpro.2006.03.007>

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