Human-in-the-loop Robot Control for Human-Robot Collaboration

Human intention estimation and safe trajectory tracking control for collaborative tasks

Ashwin P. Dani, Iman Salehi, Ghananeel Rotithor, Daniel Trombetta,
and Harish Ravichandar
POC: A. P. Dani (ashwin.dani@uconn.edu)
January 24, 2024

The prospect of a collaborative work environment for humans and robotic automation in a manufacturing setting [1] provides the motivation for finding innovative solutions to human-in-the-loop control for safe, efficient, and trustworthy human-robot collaboration (HRC) or human-robot interaction (HRI) in cyber-physical-human systems (CPHS) [2], [3]. Studies in [4] show that collaborative automation can be beneficial to 90% of ˜300,000 small-to-medium scale enterprises (SMEs) in the United States (US). In the paradigm of human-centered automation (see [5]), human safety, ergonomics, and collaborative efficiency of the work are given an utmost importance. Traditional methods to ensure the safety of humans around factory robots involve the use of cages. Some recent works have looked beyond cage-based safety to provide robot control and sensing-driven solutions for human safety around robots [6]–[8]. The purpose of this article is to provide a tutorial on human-in-the-loop estimation and control methods to achieve human safety in the context of close-proximity HRC. Examples of current close-proximity interaction include collaborative assembly, collaborative carrying of loads, and robot assistants working in close proximity to human workers in manufacturing plants such as automotive, aerospace, and electronics industry. The article is broadly divided into two main sections: the first section describes methods for human action intention inference based on sensor data, and the second section describes safe robot control policy generation based on the inferred human's action intention. State-of-the-art methods in both human action intention and safe robot control are first discussed and the mathematics behind them are provided in adequate detail, supported by simulation and experimental results in certain cases. For the intention estimation section, an overview of the topic is first provided by presenting existing literature in a tutorial manner, followed by the approaches that focus and expand on the prior work on intention estimation. The modifications to the algorithms are mentioned at the appropriate locations. New experiments

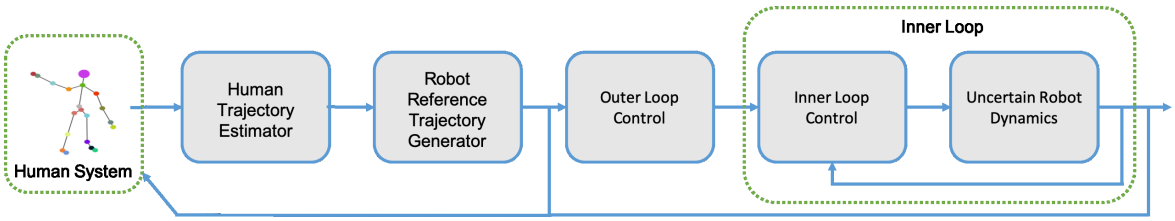

Figure 1: Block diagram of the human-in-the-loop control with human intention estimation and safe trajectory tracking of robot for collaborative tasks. The human dynamics block uses skeletal tracking using RGB-D sensors. Based on the skeletal tracking of one of the arms, human trajectory estimator generates an estimate of human intention of reaching actions. The robot reference trajectory is generated using the estimated human trajectory such that the robot reference trajectory is safe.

are conducted and the results are added to the manuscript. Compared to the prior work in [9], a robot control design method for generating safe robot reference trajectory is discussed that uses control barrier functions (CBF) in the HRC context. Experiments are conducted using a Kinect sensor and the Baxter research robot platform where the human's trajectory forecast is used to generate safe trajectories for robots to follow. A block diagram of the human-in-the-loop control with human intention estimation and safe trajectory tracking of a robot for collaborative tasks is shown in Fig. 1. Use cases of such collaborative tasks include – (1) robot assisted assembly where different components require different installation methods (different models). The robot observes the human(s) to determine the current model and hence the robot's own desired actions; (2) wire harness assembly in which a human and a robot both carry out certain steps in the assembly process. For instance, for a wire stripping task, the human may reach for a wire stripping tool, indicating to the robot that it must grasp wire and bring it to the human; (3) assisted construction/medical surgery wherein the robot is responsible for acquiring the correct tool for the task the human is currently trying to perform; (4) collaboratively moving of loads, an experiment is performed using this use case; (5) repair of vehicles in uninhabitable environments (underwater, space) wherein a robot may be assisting a human who has limited ability given the environment and has to interpret the human's intention from their motion profiles in order to assist them in making repairs in undesirable situations.

Intention Estimation: Providing sensing-based human action intention inference solutions is one of the important steps toward achieving human safety during HRC in manufacturing automation, automotive (self-driving cars) [10], space robotics [11], and assistive robotics [12]–[16]. Studies in psychology suggest that when two humans interact, they infer each other's intended actions for safe interaction and collaboration [17], [18]. An optimal control model

of human response and its applications are studied in [19], [20]. Deriving inspiration from these studies about the principles of human interactions, the safety, operational efficiency, and task reliability in HRC could be greatly improved if robots are provided with the capability to infer human action intentions. For instance, in [12], [21]–[23], inference of the human partner's intention is shown to improve the overall performance of tasks requiring HRC.

The human intention is inferred via various sensing modalities by measuring different human cues using sensors that are on the human body, that is, wearable sensors or sensors that are not on the human body, that is, non-wearable sensors. Using wearable sensors, physiological information measured using heart rate and skin response sensors [24], [25], and electromyography (EMG) sensors [26] is typically used for measuring human motion intention. In [24], human intention is represented using valence/arousal characteristics which are measured by using physiological signals, such as heart rate and skin response. The valence/arousal representation of human intention only indicates the degree of approval to a given stimulus.

Using non-wearable sensors, cues such as human emotion [27], a human's approval response [24], body posture [28], gestures [29], eye gaze [30], [31], facial expressions [32], human skeletal movement [33], [34] are measured. Imaging sensors such as Red-Green-Blue (RGB) cameras are also commonly used. In [35], the human's intention to handover an object is predicted by using key features extracted from a RGB camera sensor.

For estimating human intention, the dynamics of the human motion is modeled using probabilistic models where hidden Markov models (HMMs) [24], [36], [37], dynamic Bayesian networks (DBNs) [38], [39], growing HMMs (GHMMs) [40], conditional random fields (CRFs) [41]–[43] are employed to estimate the human intention. In [44], the activities of the humans are inferred by using a HMM of the human actions and its interaction with the autonomous mobile robots. In [45], a human intent estimation algorithm based on the fuzzy inference engine is presented. The intention estimation problem is formulated as estimating relationship between attention and physiological measurements only and online inference of human intention is not performed. In [24], an online algorithm to estimate the affective state of the person based on valence/arousal representation using HMM is developed. The intention in these methods is formulated as a classification problem. Since the future state prediction in methods based on HMMs or its variants are dependent only on the current state, these approaches quickly react to changes in intent. However, if the change in intent is not frequent, the Markov assumption can be overly restrictive, as it prevents these algorithms from becoming more certain of human intent with additional observations [46].

In another popular paradigm, the human motion is represented using continuous/discrete

dynamics that are parameterized using dynamic neural networks in case of deterministic modeling with noise, or Gaussian processes (GPs), Gaussian mixture models (GMMs) in case of probabilistic modeling. In [23], a neural network (NN) is used to approximate the endpoint of the human hand for representing human motion and intent in physical HRC applications. In [33], a latent variable model called intention-driven dynamic model (IDDM) is proposed to infer intentions from the observed human movements. Robot table tennis and human activity classification are demonstrated using a belief propagation algorithm coupled with the IDDM. In [47], the human intention inference algorithm is developed using unsupervised GMMs where the parameters of GMMs are learned using expectation-maximization (E-M) algorithm. The framework presented in [47] provides an unsupervised online learning algorithm while the algorithms presented in [33] do not involve online learning. The aforementioned approaches use the entire observed trajectory in the prediction of future state. Thus, the certainty in estimated intent tends to converge over time. These models are typically not good to detect sudden changes in intention. Additionally, the models that are learned by all the above mentioned methods are not learned considering the stability and convergence to the reaching point of the human actions/motion.

In yet another paradigm, the action plans are represented as policies in terms of state-action pairs. Inverse optimal control (IOC) or inverse reinforcement learning (IRL) algorithms are used to model the intention-driven behavior where the intended motion maximizes an unknown objective or reward function. An inverse linear quadratic regulator (iLQR) approach is developed in [48] to predict the intent and trajectory forecast of the human motion. The applicability of human intention estimation using inverse approaches can be found in various applications. For example, human's motor control intent estimation in rehabilitation applications is presented in [49] using inverse Model Predictive Control (iMPC) strategy. Predicting pilot behavior by modeling their goals using reinforcement learning (RL) and game theory approach is presented in [50]. In [51], human motion during collaborative manipulation is predicted by using an IOC approach. The IOC-based approaches typically require exploration of the state space, thus, they require large amount of data for converging to the correct solutions. A table depicting categorization of the intention estimation algorithm is shown in Table 1.

The *intention* is defined as a goal location of reaching motions and associated trajectory forecast. Knowing an estimate of the goal location is sufficient in many applications. In such cases, a maximum likelihood (ML) estimator can be used. When a distribution of the intention is required, a maximum-a-posteriori (MAP) estimator can be used. Thus, two approaches to human intention estimation are discussed that are based on ML and MAP estimation techniques. The ML estimator requires more data compared to the MAP estimator which uses a prior. When

TABLE 1: Categorization of the intention estimation algorithms based on intention estimation problem formulation. Modeling methods and sensor type used are presented along with the references.

	Method	Sensor Type	Reference
Intention Parameter Estimation	NN	RGB-D	Ravichandar et al. [34]
	GP	RGB	Wang et al. [33]
	GMM	RGB-D	Mainprice and Berenson [52]
	NN	Force/Torque	Li et al. [23]
Action Recognition and Intention Classification	IMM	RGB-D	Ravichandar et al. [53]
	HMM	Physiological	Kulic et al. [24]
	Anticipatory Temporal CRF	RGB-D	Koppula et al. [54]
	HMM	RGB+Laser	Kelley et al. [44]
	HMM	Motion Capture	Ding et al. [37]
	Hybrid DBN	RGB	Gehrig et al. [38]
	Hybrid DBN	Agnostic	Schrempf et al. [39]
	GHMM	RGB	Elfring et al. [40]
	Linear-Chain CRF	RGB-D	Hu et al. [43]
	GP-Latent CRF	RGB-D	Jiang et al. [42]
	GMM	Motion Capture	Luo et al. [47]
IOC/IRL-based Intention Estimation	Path Integral IRL	Motion Capture	Mainprice et al. [51]
	iLQR	RGB-D	Monfort et al. [48]
	iMPC	Encoder	Ramadan et al. [49]
Misc	Fuzzy Logic	Physiological	Kulic et al. [45]
	Fuzzy Logic	Force/Torque	Carli et al. [36]

the priors are uniform MAP estimator gives a ML estimate. Hence, ML estimation is a special case of the MAP estimation. In the *first technique*, the goal-reaching intention is modeled as a parameter of the continuous dynamics. A ML estimation technique called approximate E-M algorithm is applied to estimate the goal-reaching intention. To accommodate changes in motion, an online learning method for NN weight learning is also developed. NN modeling of human motion can also accommodate personalized factors such as age and other physical characteristics of the person. The algorithm is called as adaptive neural intention estimator

(ANIE) (see, [34]). In addition to the ANIE algorithm presented in [34], we have added gaze information to provide more accurate initialization to the E-M algorithm. In the *second technique,* individual goal-reaching motions are modeled as multiple dynamical system (DS) models. A MAP estimation of intention is obtained using the interacting multiple model (IMM) framework that computes probabilities of each goal reaching motion model based on the motion observations. The changes/switches in intention can also be detected using the IMM framework. A good prior to the IMM algorithms can improve its performance. Hence, additional cues such as the person's gaze direction computed by estimating the head orientation from RGB images are used. The algorithm is called as Gaze-based Multiple Model Intention Estimator (GMMIE) (see, [9]).

In the first method based on ML estimation, the complex dynamic motion of the human arm is represented by using a nonlinear system model. The positions and velocities of the joints of human arm are used as the states. The uncertain system dynamics is modeled using a dynamic neural network (DNN) [55] to represent the state propagation. Intentions are modeled as the goal locations of human arm reaching motions, which are represented by the unknown parameters of the state space model. The problem of intention inference is solved as a parameter inference problem given noisy motion data using an approximate E-M algorithm [56]. The NN approximation can potentially allow the consideration of user-specific or object-specific characteristics, such as the size and the shape of the object to be included as a part of the dynamics.

Different humans may reach the same point in 3-dimensions (3D) space in different ways based on their physical characteristics. This brings a challenge in using the model learned from the demonstration data to represent joint position trajectories of the other subjects. One way of updating the model in real-time is to use the E-M algorithm by optimizing the $\mathbf{Q}$ function over the model parameters along with the intention. A closed-form expression for the model update using E-M exists, if the model is linear or represented using a radial basis function neural network (RBF-NN) [57]. However, the human arm motion dynamics are highly nonlinear and RBF may not always be the best choice for the basis functions of the NN to represent human motion. To overcome this challenge, an identifier-system-based algorithm presented in [58] is used for online the model update. The identifier-system is designed using a robust feedback term, called Robust Integral of the Signum of the Error (RISE) [59]. Based on the Lyapunov analysis, the parameter update laws for model update are derived using the error between the state estimate generated by the identifier system and the state estimate from the original system model. The analysis ensures the asymptotic convergence of the state identification errors and their derivatives between the learned model and the true model. The guarantees on online learning can be very useful in tasks where the training data are limited and predictions have to be made about new

users with varying motion dynamics in new environments. For instance, consider a wire harness assembly task in a manufacturing environment, where the NN is trained using data obtained from a user assembling parts to build an object. If the trained NN is used for a new user assembling parts of a similar object, the NN approximation error is likely to be high as the motion profiles will be different for different users assembling different parts. However, as new data become available, the presence of the feedback term allows the identifier-system to implicitly learn the network weights and minimize the effects of NN approximation errors in real-time [58]. The inference algorithm is then used with the updated model for early prediction of the intentions.

In the second method based on MAP estimation, the organizing principles of motion generated in humans are used which states that the generated motion is inherently closed-loop stable and smooth while performing various tasks [60]. The problem of learning the human arm's motion dynamics is considered. It is formulated as a parameter learning problem under goal convergence constraints, derived using contraction analysis of nonlinear systems [61], that aid in learning stable nonlinear dynamics with respect to the reaching goal location. Details of the learning algorithm can be found in [62]. Such modeling of human motion is also useful when the person is represented as a point and its motion in 2D/3D space is observed as a point motion reaching to different locations. For instance, factory workers moving to different working stations. Using the learned model, intention inference is carried out by using the multiple-model-intention estimator (MMIE) presented in [63]. The MMIE algorithm uses an interacting multiple model (IMM) filtering approach in which the posterior probabilities of candidate goal locations are computed through model-matched filtering (see [64]). When the number of models are very high, it is well known that the performance of IMM filter degrades. A variable structure IMM (VSIMM) filter can be used in such cases [65]. A limiting case of VSIMM when the mode space is continuous is presented in [66]. In this article, probability priors of the finite number of available models are computed using gaze-based prior computation that helps in reducing the number of possible candidates.

A set of demonstrations capturing human arm joint position trajectories for reaching motions is collected by using a RGB-depth (RGB-D) camera (Microsoft Kinect). Each recorded joint position trajectory is labeled according to the corresponding true intention, that is, the 3D goal location of the reaching motion. NN models are learned by using the labeled demonstrations of the joint position trajectories.

Human-in-the-loop Safety Control of Robot: When the robot is collaborating with a human in close proximity, one of the important problems is to make the robot aware of human's motion intentions so that the robot can adapt its motion controller to move safely around human and perform a collaborative task. In [21]–[23], inference of the human's intention is shown to

improve the overall performance of collaborative tasks. In [67], human goal intention is used in conjunction with admittance controller to achieve HRC. Many contributions in HRI has targeted motion intention estimation for tasks where direct physical interaction between human and robot is involved. For control strategies, the literature focuses on designing impedance control laws [12], [23] or admittance control laws [67], [68] for adapting the interaction forces exerted by the human on the robot, when the robot is physically interacting with the human. In [69], a controller is developed for human robot handover interaction based on dynamic movement primitives. In HRC, where there may not be a direct contact with the human, the human motion intention estimation and robot path planning, and control design becomes a more important technical challenge.

There are studies in the literature that address the problem of robot motion control when humans are present in the vicinity of robots or autonomous systems [46], [70]. Most of these studies view the problem as a collision avoidance problem and solve the collision avoidance using potential field approach [71]. These control actions are purely reactionary in nature. Anticipatory skills are required when humans work in a close-proximity of the robots to improve efficiency and safety in performing close-proximity collaborative tasks [46], [70]. To achieve pro-activeness, the controller and motion planner must incorporate the probabilistic information about the possible intentions of human actions. In [46], predictive modeling of pedestrian motions with changing intentions is proposed to plan safe robot trajectories. Integrated estimation and control for HRI involving an industrial robot has been proposed in [71]. In [23], a feedback controller for HRI is developed that is based on the NN model of the human intention. In [52], a stochastic trajectory optimizer for motion planning is used for planning robot arm motion based on human intentions. In [72], scheduling, planning and control algorithms are presented that adapt to the changing preferences of a human co-worker, while providing strong guarantees for synchronization and timing of activities. In [73], new hierarchical planners based on Hierarchical Goal Networks are developed for assembly planning in human-robot team. In [74], an empirical study of human-human interaction is conducted to investigate the ways human teammates perform in a coordinated behavior. This study is targeted towards scheduling and planning tasks rather than studying human behavior in the context of robot motion planning for safety and proactive control for close proximity operations.

In controls literature, stability studies of human-in-the-loop adaptive controllers are presented using the inner-outer loop control structure in [75]. Stability studies of human-in-the-loop telerobotics with time-delay is presented in [76]. These studies do not explicitly consider safety aspects of the human-in-the-loop systems. Providing safety guarantees on the learned controller of machine/robot is typically achieved by adjusting the reference command using a

pre-filter called a reference governor [77]–[79] or by using optimal control under uncertainty in a differential game setting.

In [80], a reinforcement learning (RL) method that guarantees stability and safety by exploring the state space to collect new data for learning is presented. In [81], a safe, online, model-free approach to path planning with Q-learning is discussed. A general safety framework for learning-based control using reachability analysis is presented in [82]. In [83], a receding horizon safe path planning approach using mixed integer linear programming (MILP) is presented. Safe trajectory generation for autonomous operation of spacecraft using convex optimization formulation is proposed in [84]. When the region is non-convex, successive convexification can be performed [85]. An detailed survey and tutorial of $\mathcal{L}_1$ adaptive control architecture for safety critical systems is presented in [86].

Other methods of achieving the safety property of controller synthesis are to employ a barrier function/certificate (BF) or CBF, which ensures the closed-loop system's trajectories remain inside a prescribed safe set [87]. There are two candidates to construct BFs, namely, Reciprocal BFs and Zeroing BFs. The Reciprocal BFs can be of inverse-type and logarithmic-type. Similar extensions to CBFs have also been developed in the literature. Applications of BFs or CBFs in many autonomous robotic systems, such as robot manipulators, autonomous vehicles, and walking robots, are shown in [88]–[91]. In [88], [90], [92], BFs were successfully applied to dynamical systems where ensuring safety conditions are critical. In [92], time-varying BFs and CBFs for avoiding moving and static obstacles are derived, and their application to flying quadcopter that avoid unsafe obstacle regions is shown. Robustness properties of the CBFs are studied in [93], which shows that if a perturbation (or model error) makes it impossible to satisfy the invariance condition for a reciprocal barrier function, then the solution of the model must cease to exist because the control input becomes unbounded. For the Zeroing CBFs, Input-to-State stability (ISS) result holds in the presence of model uncertainties. A concept of exponential BFs and CBFs is introduced in [94]. The method of CBFs is extended to position-based constraints with relative degree 2 in [95] to address the safety constraints for systems with a higher relative degree. Furthermore, a backstepping based design method to design CBFs with a higher relative degree is also introduced. However, achieving a backstepping-based CBF design for systems with a higher relative degree is challenging. In [94], a concept of exponential CBFs is introduced that can handle state-dependent constraints for systems with a higher relative degree. In [96], a safety aware RL framework using BFs is proposed. However, the application of BFs or CBFs to HRC is new, and there are still many unaddressed technical challenges. In the recent work presented in [97], a methodology is proposed to learn system dynamics that can be used to generate the robot's desired trajectories, which are strictly bounded within a prescribed safety

set. An adaptive controller that takes the human intention estimation in the loop for trajectory synchronization is presented in the recent work in [9].

In this article, a safe trajectory generation algorithm that uses the inferred human trajectory from the intention estimator is developed. The safe trajectory is then provided for tracking to the robot using a feedback adaptive controller for nonlinear Euler-Lagrange (EL) system that adapts to the uncertainties in the model. EL dynamics are widely used to represent the dynamics of robot manipulators. In the controller stability analysis errors due to the inferred reference trajectory of the robot, which are generated using human trajectory forecast computed using the intention estimation algorithm, are considered. The controller is shown to be asymptotically stable under disturbances and uncertainties. A robot with an understanding of human intent is more equipped to act to improve safety of the joint tasks. An experiment is carried out that computes intention of the person in a collaborative moving task of an object along with the robot. To this end, a safe reference trajectory is generated for the robot and an adaptive controller is designed to track the estimated safe reference trajectory.

There are many broader challenges in human-in-the-loop robot control from the perspectives of both estimation and control. The major challenges include the development of fundamental theoretical guarantees and limits on long-term intention estimation, using predictive forecasting methodologies for learning models of humans and/or robots, the fusion of multiple sensing modalities, such as vision, ultrasound, LiDAR in the context of autonomous vehicles and/or physiological signals in the context of manufacturing applications. Incorporating human trust, workload [98]–[102], attention allocation [103], cognitive factors [104] in the human-robot collaboration brings many potential avenues for research. Including human ergonomic preferences in CPHS also has a great potential to make the HRI more comfortable for humans (see [105], [106]). Some recent trends in control and decision-making in human-robot collaborative systems are found in [107]. For the ML-based intention estimation technique of approximate E-M algorithm, various approximations for the E-step can be considered to improve the performance of the estimator when the process and sensor noise characteristics are non-Gaussian. Integration of information provided by the intention inference algorithms into safe control methodologies using CBF formulation that deal with the uncertainty in estimation is another important area that requires further investigation. In addition to safety verification there are temporal properties that need verification in the HRC setting, e.g., eventuality, avoidance, reachability (a dual problem to safety property) and composition of these properties. Development of methods for verification of these properties in HRC context is also important. Compensating time-delays in the communication of information between the team of humans and robots brings in many challenges, specifically in the context of network of human and robot agents [108]. HRC in

a virtual reality setting also brings many challenges from estimation and controls perspective. For example, if a rehabilitation application, a human who is undergoing a walking or biking rehabilitation can get benefited from interactions with a virtual environment. The interaction with the virtual environment can provide an important feedback to the human user via the robot it is interacting with.

The remainder of the article is organized as follows. Firstly, a human intent estimation method based on ML estimation using an approximate E-M algorithm is presented. Then, another intention estimation method based on MAP estimator that employs IMM filter with priors computed using the human gaze map is described. A human-in-the-loop control strategy that uses CBF to compute safe reference trajectory for the robot to follow is designed and an adaptive robot control is developed for the robot manipulator to track the safe desired trajectory while the collaborative task is in progress. Simulations and experiments are conducted to validate the performance of the proposed intention estimation and control results. Note that in the subsequent development, the dependency of variables on time is dropped for the compactness of notation, unless it is necessary for clarity.

Human Action Intention Estimation Scheme

Intention Estimation as a ML Estimation Problem

In this section, the human goal-reaching intention is modeled as a parameter of the nonlinear dynamics. The motion of the human is modeled as a nonlinear differential equation approximated as a NN. An approximate E-M algorithm is designed to estimate the reaching intention of the human's action (see, Fig. 2). To facilitate the discussion, a problem scenario is first described. **Problem Scenario:** Consider a 3D workspace with a human performing tasks, such as picking up objects placed on a table, shelf or walking towards certain locations in the workspace. The human subject reaches out to different objects placed on a table and a robot watches the human through a 3D camera sensor mounted on it. The problem of inferring the human hand's reaching goal location is addressed. Since the human motion is highly nonlinear and uncertain, a NN approximation of the nonlinear function of the dynamic system is used to model the human arm motion. The NN is trained by using a dataset containing skeletal tracking of human reaching for predefined target locations in a given workspace observed using a RGB-D camera. When a set of new measurements become available, the trained NN is used to estimate the reaching goal intention (goal location in 3D) using an approximate E-M algorithm. An online NN model weight learning method is also developed using an identifier-based algorithm to adapt to the variations of motions in different human subjects.

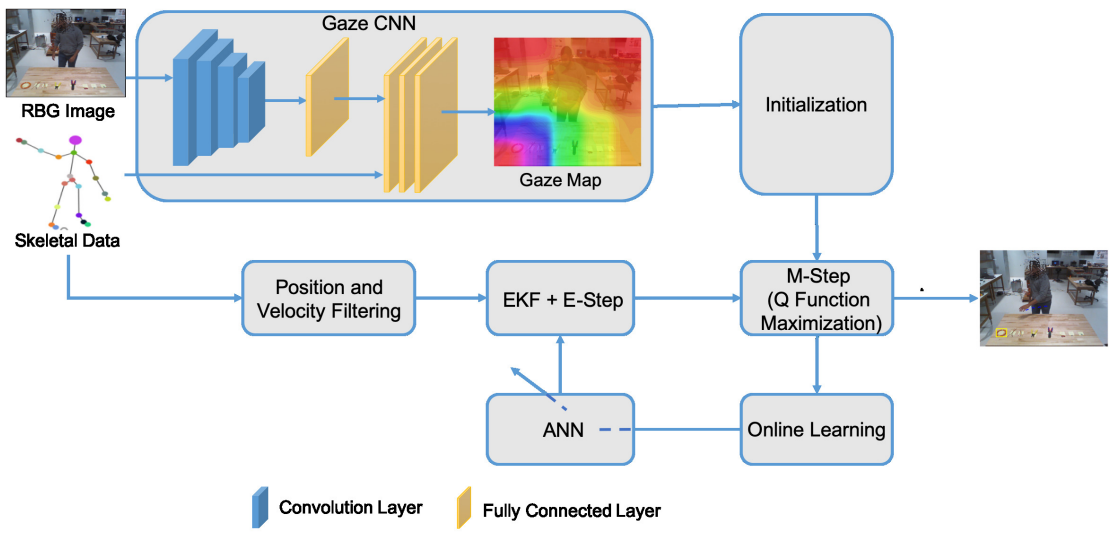

Figure 2: Block diagram of the Approximate E-M based intention estimation algorithm with gaze-based estimates used for initialization of the M-step of the E-M. The gaze estimator block uses RGB images along with skeletal data to estimate the gaze map of the person which determines the most probable objects that the person is looking at. The gaze map is used to provide an initial estimate of the parameter for the E-M algorithm.

Human Motion Dynamic Model and Measurement Model

The dynamics of human arm motion are modeled using a continuous nonlinear dynamic model of joint positions, velocities as states and intention parameters represented as the reaching goal location of the motion. The human intention is denoted by $g \in \mathcal{G}$, where $\mathcal{G} = \{g_1, g_2, ..., g_n\}$ and $g_i \in \mathbb{R}^3$ represents a 3D location of an object on a table. The true intention g is one of the goal locations $g_i's$ which can be finite (represented by a discrete variable) or very large (represented by a continuous variable). The state $x(t) \in \mathbb{R}^{24}$ represents the positions and velocities of four points on the arm (shoulder, elbow, wrist, and palm) that describe the motion of the arm in the robot reference frame, and $z(t) \in \mathbb{R}^{12}$ represents the measurement obtained from the RGB-D camera sensor data. The modeling of g as a continuous variable would be suitable in scenarios where it is not possible to obtain all possible object/goal locations.

State Transition Model: The state transition model is described by the following equation

$$\dot{x}(t) \quad = \quad f_c^*(x(t), g) + \omega(t), \tag{1}$$

where $\omega(t) \sim \mathcal{N}(0, Q_c) \in \mathbb{R}^{24}$ is a zero-mean Gaussian random process with a covariance matrix $Q_c \in \mathbb{R}^{24\times24}$. Let $\mathbb{S}$ be a compact simply connected set of $\mathbb{R}^{24} \times \mathbb{R}^3$. $f_c^*(x(t), g) : \mathbb{S} \to \mathbb{R}^{24}$ is a

Lipschitz continuous function. Since an explicit form of a nonlinear function $f_c^*(x(t), g)$ for the human arm motion is not known, $f_c^*(x(t), g)$ is approximated using a feedforward NN. There exist weights and biases such that the function $f_c^*(x(t), g)$ can be represented by a three-layered NN as

$$f_c^*(x(t), g) = W^T \sigma(U^T s(t)) + \epsilon(s(t)), \tag{2}$$

where $s(t) = [[x^T(t), g^T], 1]^T \in \mathbb{R}^{28}$ is the input vector to the NN, $U \in \mathbb{R}^{28 \times n_h}$ and $W \in \mathbb{R}^{(n_h+1)\times 24}$ are the bounded constant ideal weight matrices, that is, $\|W\|_F \leq \bar{\sigma}_w$, $\|U\|_F \leq \bar{\sigma}_u$, $\sigma(U^T s(t)) \in \mathbb{R}^{n_h+1}$ is an activation function that can be represented by using a vector-sigmoid function, a radial basis function (RBF), or a rectilinear unit, $\left(U^T s(t)\right)_i$ is the i^{th} element of the vector $\left(U^T s(t)\right)$, $\epsilon(s(t)) \in \mathbb{R}^{24}$ is the function reconstruction error, and $n_h \in \mathbb{Z}^+$ is the number of neurons in the hidden layer of the NN.

Measurement Model: The measurements of human arm's joint positions are obtained using a RGB-D camera sensor. The measurements are obtained in the camera's reference frame. Let $p^c(t) = (x^c(t), y^c(t), z^c(t))^T$ be a point in the camera reference frame and $p^r(t) = (x^r(t), y^r(t), z^r(t))^T$ be a point in the robot reference frame. The points $p^c(t)$ and $p^r(t)$ are related by

$$p^c = R_r^c p^r + T_r^c, \tag{3}$$

where $R_r^c \in \mathrm{SO}(3)$ and $T_r^c \in \mathbb{R}^3$ are the rotation matrix and the translation vector, respectively, between the robot reference frame and the camera's reference frame. The camera sensor measures the 3D locations of the skeleton's joints.

The measurement model is given by

$$y(t) = h(x(t)) + \upsilon(t), \tag{4}$$

where $y(t) \in \mathbb{R}^{12}$ is the position of the skeletal joints of the arm in the camera reference frame, $h(x(t)) = Hx(t) + b$, $b = \left[[T_r^c]^T, [T_r^c]^T, [T_r^c]^T, [T_r^c]^T\right]^T$, $H = [\mathrm{blkdiag}\{R_r^c, R_r^c, R_r^c, R_r^c\}, \mathbf{0}_{12\times 12}] \in \mathbb{R}^{12\times 24}$, and $\upsilon(t) \sim \mathcal{N}(0, \Sigma_z) \in \mathbb{R}^{12}$ is a zero-mean Gaussian noise with a covariance matrix $\Sigma_z \in \mathbb{R}^{12\times 12}$. The measurement noise $\upsilon(t)$ is assumed to be independent of the process noise $\omega(t)$ defined in (1). The measurement model of the shifted measurement vector $z(t) = y(t) - b$ is given by

$$z(t) = Hx(t) + \upsilon(t). \tag{5}$$

For RGB-D camera the Gaussian assumption of the measurement noise $\upsilon(t)$ is a standard assumption in the literature [109], [110]. The measurement noise of $z^c(t)$ components of $y(t)$ is assumed to be normally distributed in [109] and the variance is modeled as a quadratic function $\sigma_{z^c}^2(t) = c_1(z^c(t))^2 + c_2 z^c(t) + c_3$, where $c_1 \in \mathbb{R}, c_2 \in \mathbb{R}, c_3 \in \mathbb{R}$. The other components of Σ_z

can be calculated from the calibration parameters, individual pixel variances $\sigma^2_{u_c}(t), \sigma^2_{v_c}(t)$ and depth variance $\sigma^2_{z_c}(t)$.

NN Model Training

The training of the NN is carried out by using the data consisting of the human arm's joint locations, velocities, and accelerations along with the reaching goal locations. The baseline NN is trained using Bayesian regularization. The objective function used to train the NN using Bayesian regularization is given by $J(\hat{U}, \hat{W}) = K_\alpha E_D + K_\beta E_W$, where $\hat{U} \in \mathbb{R}^{28\times n_h}$, $\hat{W} \in \mathbb{R}^{(n_h+1)\times 24}$ are the estimated NN weight matrices, $E_D = \sum_i \|y_i(t) - a_i(t)\|_2^2$ is the sum of squared errors, $y_i(t)$ is the target output, $a_i(t)$ is the network's output, $E_W = \|\hat{W}\|_2^2 + \|\hat{U}\|_2^2$ is the sum of the squares of the NN weights, and K_α and K_β are the parameters of regularization that can be used to change the emphasis between reducing the reconstruction errors and model complexity, respectively.

Approximate E-M Algorithm for Estimating the Intention

An approximate E-M algorithm is presented to estimate the intention parameter (see [56]) using the state transition model learned using a NN. The intention inference algorithm on an offline trained model is presented first. The extension of the intention estimation algorithm with the online model learning is discussed subsequently. We start by discretizing the continuous model using a first order Euler approximation. It is assumed that the states are sampled at a high rate so that the first order Euler approximation is valid. The discretization of the state transition model defined in (1) yields

$$x(t) = f(x(t-1), g) + \omega(t)\delta t, \tag{6}$$

where $f(x(t-1), g) = x(t-1) + W^T\sigma(U^T s(t-1))\delta t$ and δt is the sampling period. Let $Z_T = \{z(t=0), ..., z(t=T)\}$ be a collective set of observations and $X_T = \{x(t=0), ..., x(t=T)\}$ be a collective representation of states from time $t = 0$ to $t = T$. In order to infer the intention, the posterior probability of Z_T given the intention g is maximized using a ML criterion. The process noise of the discretized system in (6) is given by $Q = \delta t^2 Q_c$. The log-likelihood function of the intention g is given by

$$l(g) = \log\ p(Z_T|g), \tag{7}$$

which can be obtained after marginalizing the joint distribution $p(X_T, Z_T|g)$ over X_T. In general, analytically evaluating this integral is very difficult. In this article, an approximate E-M algorithm is presented that uses state transition models trained using the NN. Using the fact that $\mathbb{E}_{X_T}\{\log\,[p(Z_T|g)]|Z_T, \hat{g}(t)\} = \log\ p(Z_T|g)$, the log-likelihood defined in (7) is decomposed in

the following way

$$\log\ p(Z_T|g) = \mathbf{Q}(g, \hat{g}(t)) - \mathbf{H}(g, \hat{g}(t)), \tag{8}$$

where $\mathbf{Q}(g, \hat{g}(t)) = \mathbb{E}_{X_T}\{\log\ [p(Z_T, X_T|g)]|Z_T, \hat{g}(t)\}$ is the expected value of the complete data log-likelihood given all the measurements and intentions, $\mathbf{H}(g, \hat{g}(t)) = \mathbb{E}_{X_T}\{\log[p(X_T|Z_T, g)]|Z_T, \hat{g}(t)\}$, $\mathbb{E}_{X_T}(\cdot)$ is the expectation operator, and $\hat{g}(t)$ is the estimate of g at time t. It can be shown using Jensen's inequality that,$\mathbf{H}(g, \hat{g}(t)) \leq \mathbf{H}(\hat{g}(t), \hat{g}(t))$. Thus, in order to iteratively increase the log-likelihood, g has to be chosen such that $\mathbf{Q}(g, \hat{g}(t)) \geq \mathbf{Q}(\hat{g}(t), \hat{g}(t))$. The **E**-Step involves the computation of the auxiliary function $\mathbf{Q}(g, \hat{g}(t))$ given the observations Z_T and the current estimate of the intention $\hat{g}(t)$. The **M**-Step involves the computation of the next intention estimate $\hat{g}(t+1)$ by finding the value of g that maximizes $\mathbf{Q}(g, \hat{g}(t))$.

The **E-Step** involves the evaluation of the expectation of the complete data log-likelihood which can be rewritten as

$$\mathbf{Q}(g, \hat{g}(t)) = \mathbb{E}_{X_T}\{V_0 + \sum_{t=1}^{T} V(x(t), x(t-1), g)|Z_T, \hat{g}(t)\}. \tag{9}$$

In the case of $v(t)$ and $w(t)$ are Gaussian, the computation of $\mathbf{Q}(g, \hat{g}(t))$ can be simplified.

The **M-step** involves the optimization of $\mathbf{Q}(g, \hat{g}(t))$ over g as described by the following expression

$$\hat{g}(t+1) = \arg\ \max_{g} \mathbf{Q}(g, \hat{g}(t)). \tag{10}$$

This step can be carried out in two different ways, namely numerical optimization or direct evaluation as described below.

One way to maximize the $\mathbf{Q}$ function is to use the GradEM algorithm for M-step which in turn uses first iteration of Newton's algorithm [56]. Since the Newton's method often converges quickly, the local convergence properties of GradEM algorithm are identical to the E-M algorithm. More details of the convergence properties of GradEM algorithm can be found in [111]. This method involves optimizing the $\mathbf{Q}$ function over $\mathbb{R}^3$. The update equation for $\hat{g}$, through GradEM algorithm, is given by

$$\hat{g}(t+1) = \hat{g}(t) - \mathcal{H}(\mathbf{Q})^{-1}\Delta(\mathbf{Q}), \tag{11}$$

where $\hat{g}(t)$ is the estimate of g at the current time t of the optimization algorithm, $\mathcal{H}(\mathbf{Q})$ and $\Delta(\mathbf{Q})$ are the Hessian and Gradient of the $\mathbf{Q}$ function respectively. Note that numerical optimization methods need to run at every time step of the E-M algorithm. For the real time implementations, the number of iterations for the optimization in (11) could be chosen based on computational capabilities. More details of the computation of the Hessian of the $\mathbf{Q}$ function

and expression for the gradient of the $\mathbf{Q}$ function can be found in [34]. General details of E-M algorithm are described in "Expectation-Maximization Algorithm".

Another way to infer g is to evaluate the $\mathbf{Q}$ function for all possible g_i's (the goal locations) in $\mathcal{G}$ and obtain $\hat{g}(t+1)$ as described by the following expression

$$\hat{g}(t+1) = \arg \max_{g \in \mathcal{G}} \mathbf{Q}(g, \hat{g}(t)). \tag{12}$$

This method involving direct evaluation of the $\mathbf{Q}$ function is possible if all possible goal locations are known *a priori* and are finite. This is not an unusual case in the context of human intention estimation in practical applications such as manufacturing assembly, space robotics or assisted construction. Image processing algorithms, such as Region convolutional neural network (R-CNN), Faster R-CNN, MaskRCNN, YoLo can be used to detect the objects on the workspace and extract the 3D locations using the camera data.

Online Model Weight update

In this section, the online learning algorithm is described that is used to update the weights of the NN model. The online learning of the NN weights is important to make the inference framework robust to variations in starting arm positions and various motion trajectories taken by different people. The NN weights are updated iteratively as new data become available. A state identifier is developed that computes an estimate of the state derivative based on the current state estimates obtained from the EKF and the current NN weights. The identifier state error is computed based on the state estimate and the measurement. The error in the state identifier is used to update the NN weights for the next time instance. Note that the state identifier can run at a higher sampling rate compared to the E-M algorithm, which involves optimizing the $\mathbf{Q}$ function that may have a slower convergence rate. Hence, the state identifier is presented in the continuous form. The identifier uses a RISE feedback [59] to ensure asymptotic convergence of the state estimates and their derivatives to the true values. The weight update equations are designed using Lyapunov-based stability analysis.

The state identifier is given by

$$\dot{\hat{x}}_{id} = \hat{W}^T(t)\sigma(\hat{U}^T(t)\hat{s}(t)) + \mu(t), \tag{13}$$

where $\hat{U}(t) \in \mathbb{R}^{28 \times n_h}$, $\hat{W}(t) \in \mathbb{R}^{n_h \times 24}$, $\hat{s}(t) = [[\hat{x}_{id}^T(t), \hat{g}(t)], 1]^T \in \mathbb{R}^{28}$, $\hat{g}(t) \in \mathbb{R}^3$ is the current estimate of g from the E-M algorithm, $\hat{x}_{id}(t) \in \mathbb{R}^{24}$ is the current identifier state, and $\mu(t) \in \mathbb{R}^{24}$ is the RISE feedback term defined as $\mu(t) = k\tilde{x}(t) - k\tilde{x}(0) + \nu(t)$, where $\tilde{x}(t) = x(t) - \hat{x}_{id}(t)$ is the state identification error and $\nu(t) \in \mathbb{R}^{24}$ is the Filippov generalized solution [58] to the following differential equation

$$\dot{\nu}(t) = (k\alpha + \gamma)\tilde{x}(t) + \beta_1 \text{sgn}(\tilde{x}(t)), \quad \nu(0) = 0, \tag{14}$$

where k, α, γ, $\beta_1 \in \mathbb{R}^+$ are positive constant control gains and $\mathrm{sgn}(\cdot)$ denotes a vector signum function. The weight update equations are given by

$$\begin{aligned} \dot{\hat{W}}(t) &= \mathrm{proj}(\Gamma_w \hat{\sigma}'(t) \hat{U}_{x(t)}^T \dot{\hat{x}}_{id}(t), \tilde{x}^T(t)), \\ \dot{\hat{U}}_{x(t)} &= \mathrm{proj}(\Gamma_{u_x} \dot{\hat{x}}_{id}(t) \tilde{x}^T(t) \hat{W}^T(t) \hat{\sigma}'(t)), \\ \dot{\hat{U}}_{g(t)} &= \mathrm{proj}(\Gamma_{u_g} \dot{\hat{g}}(t) \tilde{x}^T(t) \hat{W}^T(t) \hat{\sigma}'(t)), \end{aligned} \tag{15}$$

where $\mathrm{proj}(\cdot)$ is a projection operator defined in [112], $\hat{U}_{x(t)}$ and $\hat{U}_{g(t)}$ are the sub-matrices of $\hat{U}(t)$ formed by taking the rows corresponding to $\hat{x}_{id}(t)$ and $\hat{g}(t)$ respectively, $\hat{\sigma}'(t)$ is the first order derivative of the sigmoid function with respect to its input $\hat{U}^T \hat{s}(t)$, and Γ_w, Γ_{u_x}, and Γ_{u_g} are constant weighting matrices of appropriate dimensions. In the online learning algorithm $\hat{g}(t)$ from the E-M algorithm is used. Hence, for the online learning step, $\hat{g}(t)$ is assumed to be a known signal. The derivative of the intention estimate $\dot{\hat{g}}(t)$ is computed using the finite difference method. It can be shown using Lyapunov analysis that the identifier defined in (13) along with the update equations defined in (15) is asymptotically stable and the state identification error converges to zero.

Gaze Map Computation: This section briefly describes the convolutional neural network (CNN), introduced in [113], used to extract gaze information from an RGB image. To this end, a deep architecture of CNN is employed. The input (features) to the CNN is a $D_w \times D_h$ RGB image of the subject looking at an object and the relative position of the subject's head in that image. The output is the gaze map $\mathcal{G}$ of size $D_w \times D_h$ containing the probabilities of each pixel being the gaze point.

Data: The dataset used for training the CNN model, as described in [113], is created by concatenating images from six different sources: 1548 images from SUN; 33790 images from MS coco; 9135 images from Actions40; 7791 images from PASCAL; 508 images from the ImageNet detection challenge; and 198097 images from the Places dataset.

Implementation of CNN: The five-layered CNN shown in Fig. 2 is implemented using Caffe library. Images of size $224 \times 224 \times 3$ are used for training the CNN. These input images are filtered by 96 convolution kernels of size $11 \times 11 \times 3$ and fed into the first convolution layer of size $55 \times 55 \times 96$. The output of the first layer is filtered with 256 convolution kernels of size $5 \times 5 \times 48$ and fed to the second convolution layer. The subsequent three layers are connected to one another without any pooling layers between them. The third convolution layer has 384 convolution kernels of size $3 \times 3 \times 256$ connected to the normalized and pooled outputs of the second convolution layer. The fourth convolution layer has 384 convolution kernels of size $3 \times 3 \times 192$ and the fifth convolution layer has 256 convolution kernels of size $3 \times 3 \times 192$. The

remaining four layers used in the network are fully connected (FC) and are of sizes 100, 400, 200, and 169. See [113] for a more in-depth description of the CNN framework.

The CNN is used to compute a gaze map, which is an image that assigns a probability value to each pixel to represent the likelihood of the pixel being looked at in the image based on the head orientation data of the person. To compute the probability of g_j being the initial goal location for the optimization problem of E-M based approach, the average probability $\bar{p}_j(0)$ of the j^{th} object in the scene is calculated as follows

$$\bar{p}_j(0) = \sum_{i \in \mathcal{G}P_j} \left(\mathcal{G}(i)/NP_j \right), \tag{16}$$

where $\mathcal{G}(i)$ is the probability of the i^{th} pixel being the gaze point, NP_j is the number of pixels associated with the j^{th} object, and $\mathcal{G}P_j$ is the set of all pixel locations associated with the j^{th} object. Of all the objects in the scene, the object that corresponds to the highest average probability is chosen as the initial goal location for the optimization of the E-M algorithm of the ANIE method.

Experimental Results for E-M based Intention Estimation

In this section, two experiments are presented on real data obtained from Kinect sensor tracking a human's movements. In both experiments, the reaching motion data used for training and testing is collected from different human subjects.

NN Training: The starting positions of the human arm and the possible goal locations of the test trajectories are different from each other. In the training phase, some of the trajectories involved reaching objects that are randomly placed close to each other in a cluttered manner and some of the recorded arm motions consisted of the subject initially moving the hand close to an object, but finally reaching another object. Each trajectory contained roughly 40 to 60 frames of skeletal data. A set of 8 trajectories is used for training a NN. The raw position measurements obtained from the RGB-D camera sensor are processed using a Kalman filter, such as the one in [114], to obtain the position and velocity estimates. The number of neurons in the hidden layer is empirically chosen to be 50.

NN Testing: Once the NN is trained, the test data from a different subject is used as measurements to infer the underlying reaching intentions of the reaching motion. During the inference, the NN weights are learned online to adapt to the motion performed by the test subject. It should be noted that the total number of frames for each reaching motion is not fixed and the intended object is reached at varying frame numbers. The $\mathbf{Q}$ function is evaluated for all the possible intentions to find the intention that led to the maximum $\mathbf{Q}$ value (direct evaluation method). The initial mean of the state $\mu(0)$ is assumed to be a zero vector. The initial state covariance $P_{EM}(0)$, the process noise covariance Q, and the measurement noise covariance Σ_z are selected

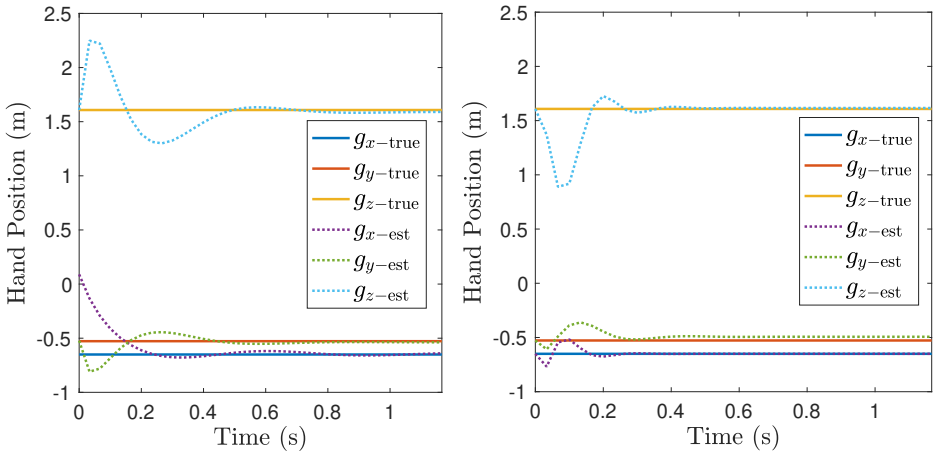

Figure 3: Estimated goal location using ANIE algorithm by numerically optimizing the **Q** function. a) Convergence plot of the estimated intention (X, Y, Z locations) when the initial estimate is chosen randomly. b) Convergence plot of the estimated intention (X, Y, Z locations) when the initial estimate for E-M optimization is chosen using gaze cues.

to be $0.2I_{24\times24}$, $0.1I_{24\times24}$, and $0.2I_{24\times24}$, respectively, where I denotes the identity matrix. The gains for the online learning algorithm defined in (13) and (15) are selected to be $k = 25$, $\alpha = 5$, $\gamma = 25$, $\beta_1 = 4$, and the adaptation gains are chosen to be $\Gamma_W = 0.75I_{50\times50}$, $\Gamma_{U_x} = 0.75I_{24\times24}$, $\Gamma_{U_g} = 0.75I_{3\times3}$. The state estimates are initialized to the same value as the first measurement z_1. The sampling time for discretization is $\frac{1}{30}$ sec.

When the intention $\hat{g}$ is modeled as a continuous variable, GradEM algorithm is used for evaluating the intention estimate. In Fig. 3, the convergence of the estimated goal location to the true goal location is shown. The numerical optimization of the **Q** function is done for five iterations at every time step. The state estimates are initialized to the initial measurements and the intention estimates are initialized using two methods, namely, random selection and gaze-based selection. In the first set of experiments the goal location is randomly chosen from the possible eight goal locations and the convergence of the estimated intention to the true intention is shown in Fig. 3a. In the second set of experiments the estimated intention is initialized using gaze cues and the convergence of the estimated intention to the true intention is shown in Fig. 3b. From the following plots it can be observed that the intention estimates $\hat{g}_x, \hat{g}_y$ and $\hat{g}_z$ for the gaze-based initialization converge faster and show less transients compared to the random initialization.

When the intention $\hat{g}$ is modeled as a discrete variable, a direct evaluation of **Q** function is performed to estimate the intention. In the first method, one of the possible n goal locations is selected randomly as the initial estimate. In Fig. 4, the intention estimate progression and

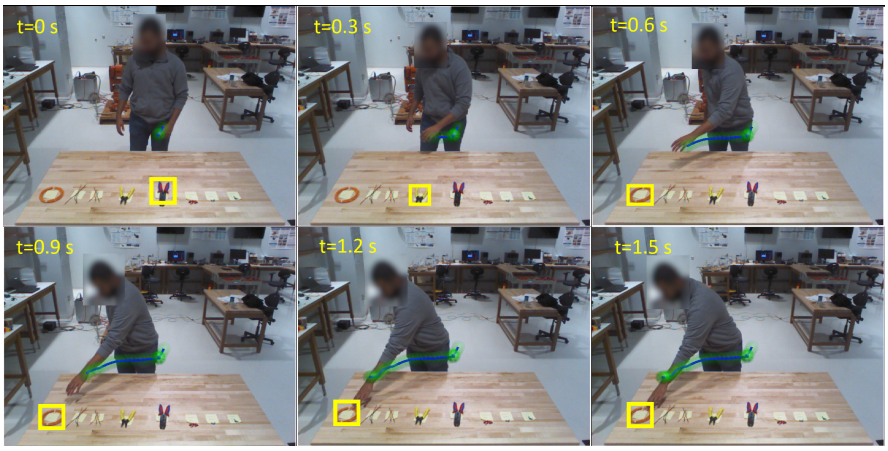

Figure 4: Image sequence showing the online inference of intention and the evolution of the human hand (wrist) trajectory (shown in green) using an approximate E-M algorithm with online model learning. The initial estimate of the reaching goal intention is chosen randomly among the finite number of objects available on the table.

trajectory evolution for such a randomly chosen estimate is shown. Error statistics with a large number of experiments are reported in the prior work in [34]. Due to the random assignment, the initial estimate can be any of the possible goal locations including the ones far away from the true intention. It is observed that in some cases the E-M algorithm requires more observations to converge to the true intention when the intention estimate is randomly initialized. To overcome this issue, a gaze-based selection of the goal location for optimization of E-M is tested in the second experiment which provides better cues of reaching-goal location of the person. In the second method, a dense gaze probability map is obtained using the method described in [53], [113] and a probability is assigned to each of the possible goal locations using the formula in (16). The goal location with the highest probability is chosen as the initial estimate $\hat{g}(0)$. In Fig. 5, the dense gaze probability map is shown along with the probabilities computed for each of the possible goal locations. The object with the maximum probability value, 0.2 in this case, is chosen as the initial intention estimate. In Fig. 6, the intention estimate progression is shown along with the trajectory evolution for the initial intention estimate computed from the gaze map. For the case of the sequence shown in Fig. 6, the goal location with the maximum gaze probability happens to be the true goal location and as a result the algorithm predicts the intention correctly throughout the sequence. In the case of random initialization of intention, the ML estimation is equivalent to the MAP estimation with uniform prior. The observations in previous experiments that the ML method gives faster convergence when initialized with gaze

cues compared to the random initialization could be a coincidence. If the human is looking toward the opposite side of the direction of the arm motion, the gaze-based goal initialization will not be very useful. This may lead to a bad initialization for the goal location of the E-M algorithm. If the initialization of the goal location is bad, the E-M algorithm may require more measurements before it correctly converges to the true goal location because E-M is a ML estimation method. As shown in Fig. 3, the E-M algorithm requires observations up to 0.6s to correctly predict the goal location for a 1.5 sec sequence.

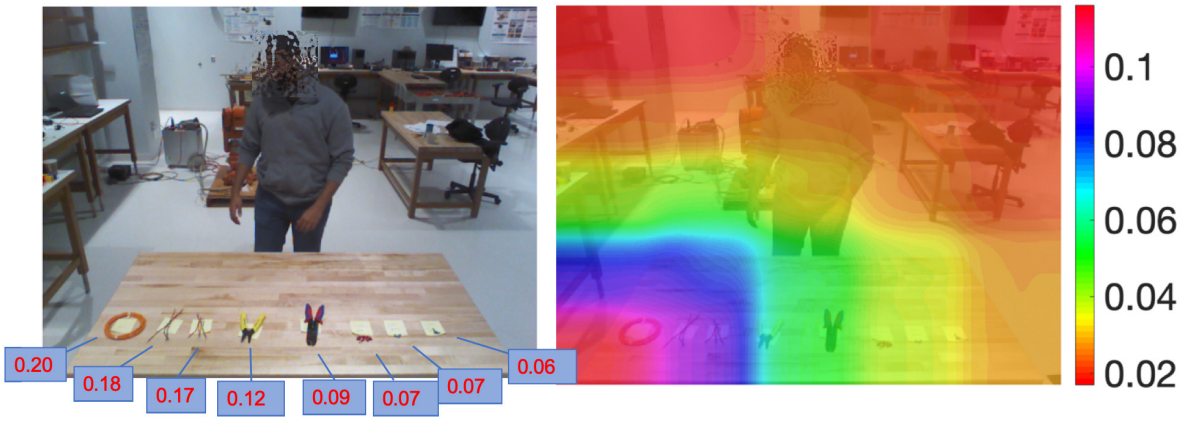

Figure 5: (a) Goal location probabilities computed from the gaze map. The numbers shown inside the labels represent the probability of that object being reached by the person based on the initial head orientation. (b) Dense gaze map showing gaze probability associated with each pixel.

Intention Estimation as a Multiple Model Estimation Problem

In this section, a MAP estimation of the human goal-reaching intention is computed using a multiple model estimation problem. Since the human reaching motion is generated by an inherent stable motion strategy, each reaching motion is modeled as a continuous DS whose solutions converge to the goal location. To ensure that the models reach the goal location, the dynamics that are approximated using dynamic NN are trained subject to convergence to the goal location constraints. The intention estimation strategy is then to select the correct model based on the currently observed motion trajectory. A block diagram of the multiple model-based intention estimation methodology is shown in Fig. 7. More details of the algorithm can also be found in [53].

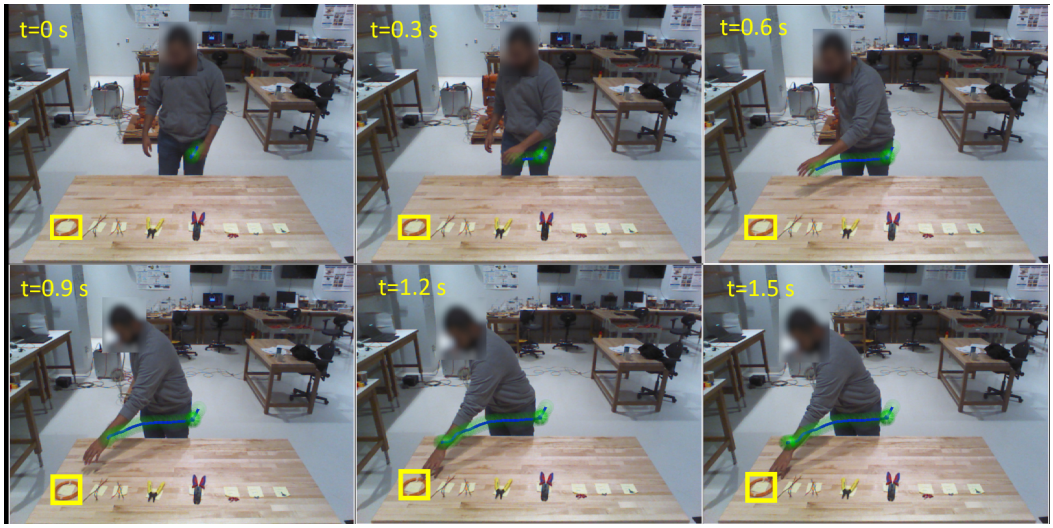

Figure 6: Image sequence showing the online inference of reaching intention and the evolution of the human hand (wrist) trajectory for an initial intention chosen using the gaze map. The gaze cue aids in narrowing down the human reaching motion to the object being reaching for this test case.

Human Motion Dynamic Model

In this section, a method for learning the nonlinear dynamics of human arm reaching motion is presented. Consider a state variable $x\left(t\right) \in \mathbb{R}^n$ and a set of $N_{\mathcal{D}}$ demonstrations $\{\mathcal{D}_i\}_{i=1}^{N_{\mathcal{D}}}$ representing reaching motions to various goal locations. Each demonstration would consist of the trajectories of the state $\{x\left(t\right)\}_{t=0}^{t=T}$ and the trajectories of the state derivative $\{\dot{x}\left(t\right)\}_{t=0}^{t=T}$ from time $t = 0$ to $t = T$. All state trajectories of the demonstrations are translated such that they converge to the origin. Let the translated demonstrations be solutions to the underlying DS governed by the first-order differential equation

$$\dot{x} = f_e(x(t)), \tag{17}$$

where $f_e(x(t))$ is a Lipschitz continuous function. Since all the trajectories of the translated demonstrations converge to the origin, the system defined in (17) could be seen as a globally contracting system. The nonlinear function $f_e\left(x(t)\right)$ is approximated by using a NN similar to the one in (2) without the intention variable g. Note that only one NN is used to represent the dynamics of reaching motion trajectories that converge to the origin. Arm motion trajectories pertaining to different goal locations can be obtained by corresponding linear translations of the solutions to the DS in (1).

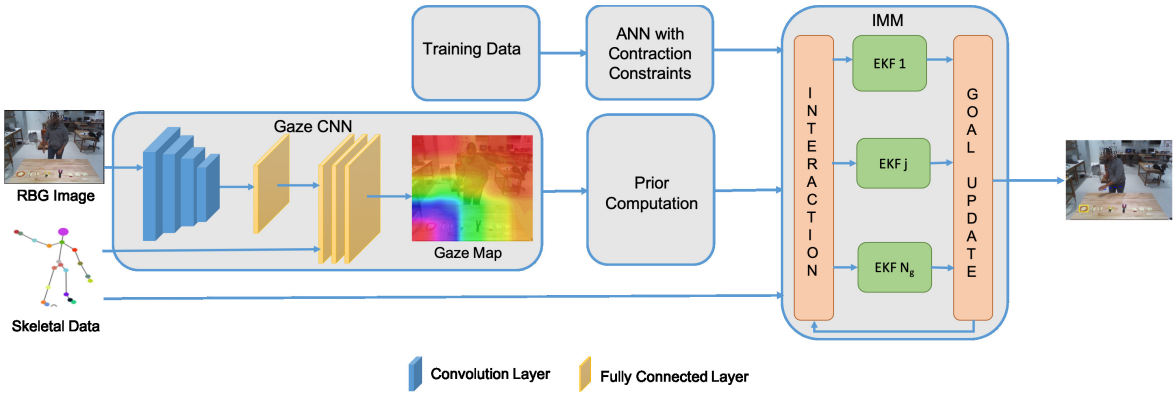

Figure 7: Block diagram of the multiple model based intention estimation algorithm with gaze priors. The gaze estimator block uses RGB images along with head position to estimate the gaze map of the person which determines the most probable objects that the person is looking at. The information from gaze map is used to compute prior model probability for the IMM algorithm.

Learning Contracting Nonlinear Dynamics of Human Reaching Motion

To learn the NN weights from the sample reaching motion data, the following NN weight training algorithm is used. The weights are trained such that the states of the dynamics converge to a given reaching goal location. To achieve this, a constrained optimization problem is solved subject to goal-reaching terminal constraints enforced using contraction analysis of nonlinear dynamics. The following optimization problem is set up in order to learn the weights of the NN

$$\{\hat{W}, \hat{U}\} = \arg\min_{W,U} \{\alpha E_D + \beta E_W\} \tag{18}$$

$$\text{s.t.} \quad \frac{\partial f_e}{\partial x}^T M + M \frac{\partial f_e}{\partial x} \prec -\gamma M, \; M \succ 0 \tag{19}$$

where $E_D = \sum_{i=1}^{D} [y_i(t) - a_i(t)]^T [y_i(t) - a_i(t)]$, $y_i(t) \in \mathbb{R}^n$ and $a_i(t) \in \mathbb{R}^n$ represent the target and the network's output of the i^{th} demonstration, E_W is the sum of the squares of the NN weights, $\alpha, \beta \in \mathbb{R}$ are scalar parameters of regularization, $\gamma \in \mathbb{R}$ is a strictly positive constant, and $M \in \mathbb{R}^{n \times n}$ represents a constant positive symmetric matrix. The details of the computation of Jacobian, $\frac{\partial f_e(x(t))}{\partial x}$ for the NN approximation of $f_e(x(t))$ can be found in [62].

Multiple Model Estimation Algorithm for Intention Estimation

Based on multiple dynamic NN models that reach different goal locations, an IMM algorithm is developed that selects the most probable model, which is a representation of the current trajectory of human's reaching motion observed through sensor data. The IMM estimator is first described, and the computation of priors using gaze-based cues is subsequently explained.

Interacting Multiple Model Estimator

Given the trained network and a trajectory of the reaching hand, the problem involves inferring the goal location ahead in time. Let a fixed set of candidate goal locations that the human can reach be $G = \{g_1, g_2, \cdots, g_{N_g}\}$. The NN weights learned from human demonstrations is used to represent human motion. For each goal location g_j, the state vector and the corresponding dynamics are defined as $x^j(t) = [[x_{pos}(t) - g_j]^T, x_{vel}^T(t)]^T$ and $\dot{x}^j(t) = f_e(x^j(t))$. Similarly, for a fixed set of N_g goal locations, a set of N_g dynamic systems is formed. The discretized versions of these systems are given by

$$x^j(t+1) = x^j(t) + f_e\left(x^j(t)\right)\delta t + \omega_e^j(t)\delta t, \tag{20}$$

where $j = 1, \cdots, N_g$, δt is the sampling period, and $\omega_e^j(t) \sim \mathcal{N}(0, Q_e^j) \in \mathbb{R}^{2n}$ is a zero-mean Gaussian random process with a covariance matrix $Q_e^j \in \mathbb{R}^{2n\times 2n}$. For this section, consider $n = 3$, that is, only the last joint of the arm's skeleton is tracked with X_e, Y_e, and Z_e positions. The measurement model is given by

$$z_e(t) = h_e(x^j(t)) + v_e^j(t), \qquad j = 1, 2, ..., N_g, \tag{21}$$

where $z_e(t)$ is the measurement vector, $v_e(t) \sim \mathcal{N}(0, R_e)$ is a zero mean Gaussian process with covariance R_e and $h_e(x^j(t)) = \begin{bmatrix} 1&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&0&1&0 \end{bmatrix} x^j(t)$ is the measurement function.

Let $M_1, M_2, ..., M_{N_g}$ represent the N_g models defined in (20) and (21) for the set of candidate goal locations G. The posterior probability of model j being correct is denoted by $P(g_j|Z_{1:t})$. The expression $P(g_j|Z_{1:t})$ denotes the posterior probability of each g_j being the correct goal location given a set of measurements $Z_{1:t} = [z_e(1), z_e(2), \cdots, z_e(t)]$. Note that $P(g_j|Z_{1:t}) = P(M_j|Z_{1:t})$ since the models and goal locations have a one-to-one correspondence. Hence, in order to obtain the posterior probabilities $P(g_j|Z_{1:t})$, $j = 1, .., N_g$, the posterior probabilities of the models $P(M_j|Z_{1:t})$, $j = 1, .., N_g$ are computed. The posterior probability $P(M_j|Z_{1:t})$ is calculated using the Bayes' formula as follows $P(M_j|Z_{1:t}) = P(M_j|z(t), Z_{1:t-1}) = \frac{p(z(t)|Z_{1:t-1}, M_j)P(M_j|Z_(1:t-1))}{\sum_{i=1}^{N} p(z(t)|Z_{1:t-1}, M_i)P(M_i|Z+(1:t-1)}$, where $p(z(t)|Z_{1:t-1}, M_j)$ is the likelihood function of mode j at time t, and $P(M_j)|Z_{1:t-1}$ is the prior probability of M_j being correct. In the IMM framework with N_g models, the likelihood function $p(z(t)|Z_{1:t-1}, M_j)$ of mode j at time t which can be computed using $\Lambda_j(t) = p(z(t)|Z_{1:t-1}, M_j) = p(v_e^j(t)) = \mathcal{N}(z_e(t) - \hat{z}_e^j(t|t-1); 0; S_e^j)$. The innovation $\nu_j = z_e(t) - \hat{z}_e^j(t|t-1)$ and its covariance S_j are computed from the mode-matched filter corresponding to mode j. The G-MMIE algorithm uses EKFs matched to each mode. Other filters based on state-dependent coefficient form parametrization of nonlinear systems, e.g., see [115], can also be used for each individual filter. Each iteration of the IMM filter for intention inference is divided into four main steps, namely, interaction/mixing stage, model matched filtering, model probability update and model switch detection. More details about the

static IMM filter can be found in [64]. Details specific to multiple model filter in the context of gaze-based intention estimation can be found in [53].

In many real world applications small number of models may not be sufficient to describe all the modes. When the number of models is large the performance of IMM filter can degrade and the computational burden increases. A variable structure IMM (VSIMM) filter can be used in such cases [65]. A limiting case of VSIMM when the mode space is continuous is presented in [66]. In certain HRC applications large number of models are required to represent the application context. Next, computation details of prior distribution of models for IMM using gaze map are shown.

Computation of Prior Distribution using Gaze Map for IMM filter: Using the gaze based prior computation procedure, and the average prior probability $\bar{p}_j(0)$ computed using (16), the probability of each of the N_g candidate locations being the goal location is calculated as follows $\mu_j(0) = \frac{\bar{p}_j(0)}{\sum_{j=1}^{N_g} \bar{p}_j(0)}$, where $\mu_j(0)$ is the prior probability of g_j being the goal location for the IMM filter and g_j refers to the location of the j^{th} object.

Experimental Results for Multiple Model-based Intention Estimation

In order to validate the multiple model-based intention estimation algorithm a set of 10 demonstrations collected from a subject are used for training the NN under contraction analysis constraints. For training the NN each demonstration is labeled based on the ground truth goal location. Note that the ground truth labeling is done only for the training data. All data are collected by using a Microsoft Kinect for Windows. The joint position data obtained from the subjects are preprocessed to obtain the velocity and acceleration estimates using a Kalman filter (see [114] for details). In all the experiments, the position and velocity of the hand in 3D Cartesian space are considered to be the elements of the state vector $x(t) \in \mathbb{R}^6$, and the number of possible goal locations $N_g = 8$. An IMM filter for computing intention estimate is implemented using the following parameters. The initial state estimate covariance $\hat{P}_e^j(0), j = 1, 2, .., N_g$, the process noise covariance $Q_e^j, j = 1, 2, .., N_g$, and the measurement noise covariance R_e for the EKF of the IMM filter are selected to be $0.2I_{6\times6}, 0.1I_{6\times6}$, and $0.2I_{6\times6}$, respectively. The state estimates $\hat{x}^j, j = 1, 2, \cdot\cdot, N_g$, are initialized using the first two measurement $z_e(1)$ and $z_e(2)$ (a finite difference method is used for the velocity initialization). The model transition matrix for IMM is chosen to be a 8×8 matrix $\prod_{ij}$ with diagonal elements to be 0.79 and off-diagonal elements to be 0.03. For computing the IMM priors, uniform distribution and gaze-based cues are used.

In the first experiment, a uniform distribution is used as a prior which corresponds to

all possible goal locations having an equal probability of being the true intention. In Fig. 8, the intention estimate is shown along with the trajectory evolution for the uniform prior. The initial goal location is set to object 5 and after 0.3s the predicted goal location matches the true intention. In the second experiment, the prior distribution is computed from the dense gaze

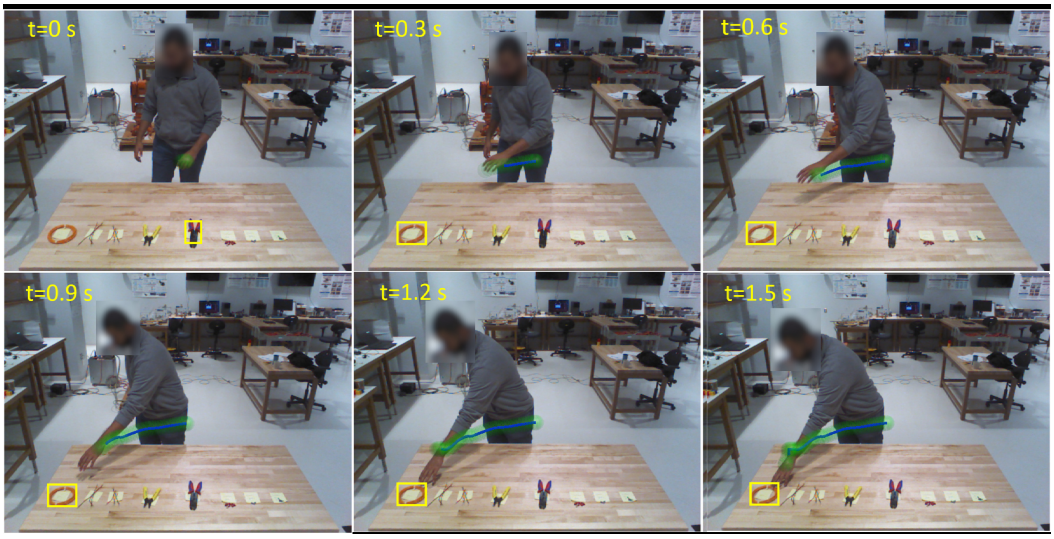

Figure 8: Image sequence showing the online inference of intention and the evolution of the human hand trajectory for a uniform prior distribution.

map as shown in Fig. 5 by using the formula in (16). The gaze-based prior assigns a higher prior probability to the goal locations where the human subject is likely to be looking. In Fig. 9, the intention estimate and trajectory evolution for prior distribution computed using a dense gaze map is shown. The goal location with the highest prior probability happens to be the true intention. Gaze-based prior computation assigns a higher probability value to the true intention which results in a better prediction with less number of observations of the human hand trajectory. An interested reader is referred to the prior work in [53], where the error statistics of a large number of experiments are reported.

Safe Robot Controller based on Human Intention Inference

In this section, we will describe the robot control design that takes into consideration the human trajectory generated by the intention estimators to generate its own motion that is safe around the human. First, a robot desired trajectory generation algorithm is presented that determines the robot desired trajectory based on human's predicted trajectory from the intention

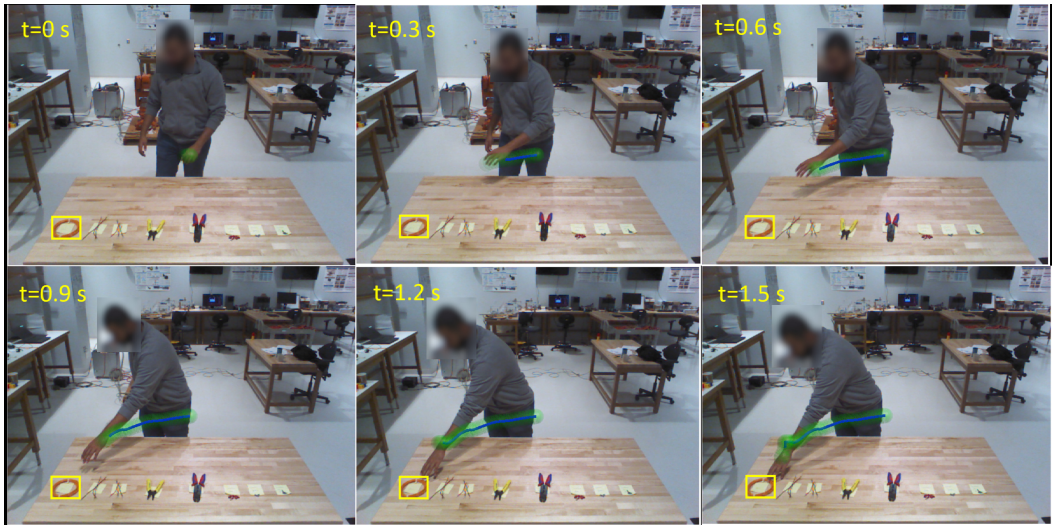

Figure 9: Image sequence showing the online inference of intention and the evolution of human hand trajectory for prior distribution computed using gaze map.

estimator. A barrier function formulation is then used to modify the robot's desired trajectories, which cross the boundary of the safety ellipsoid drawn around the human. In a subsequent section, a torque controller that follows the robot's desired trajectory is discussed. Note that the dynamics in continuous-time is used for the math development in this section because the controller sampling rate is typically much higher compared to the sampling rate of the data obtained from the sensors used for the intention estimation.

Problem Scenario: An object carrying task is used as a test case where the person is holding one side of the object, and the robot is holding the other side. While carrying the object, the robot end-effector trajectories may cross the safety ellipsoid around the human, and that is when the CBF is used to modify the robot's end-effector desired trajectories. In Fig. 10, an example scenario of the task is shown.

Generation of Robot Desired Trajectories for joint Human-Robot Task

A method is presented that generates the robot desired trajectories as a function of the human's inferred trajectory. If the robot's desired trajectory enters the prescribed safety ellipsoids around the human, then a barrier function based formulation is used to modify the robot's desired trajectory such that it always stays outside the safety ellipsoid. In Fig. 11, the details of the reference trajectory generation algorithm are shown in the form of a block diagram. Let

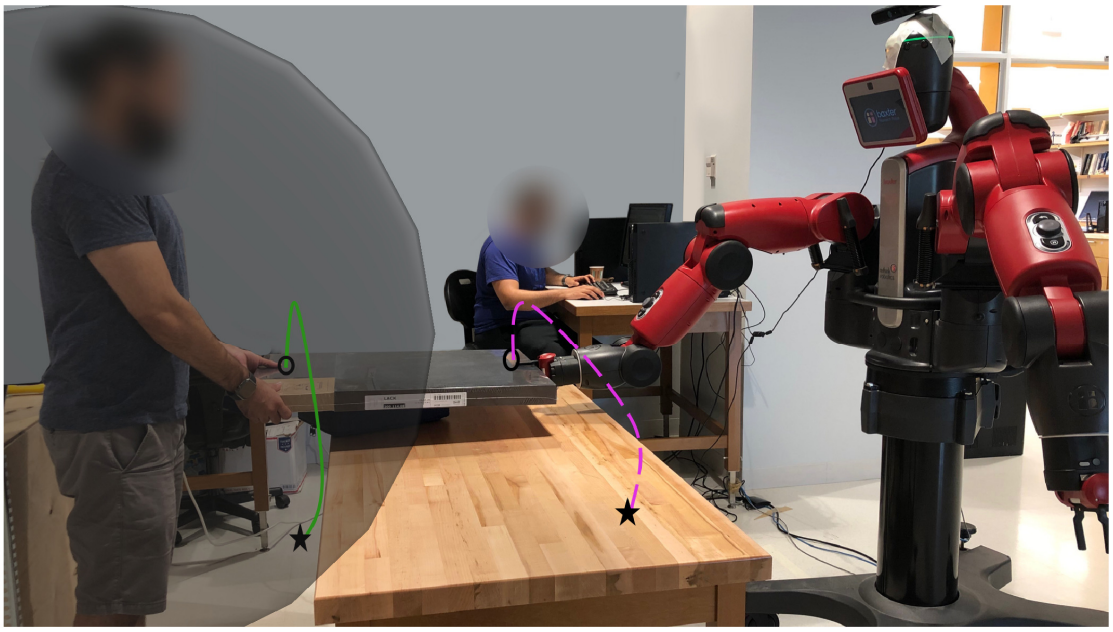

Figure 10: An example scenario showing human and robot working closely together to carry an object. The gray sphere around the human operator corresponds to the region where the robot should not enter during the task execution. The magenta (dashed) and green (solid) lines show the desired motion trajectory of the object-carrying task. The stars and circles represent the initial and target locations of the motion, respectively.

$x_{Rd}(t) \in \mathbb{R}^3$ be the robot's end-effector desired position state and $x_H(t) \in \mathbb{R}^3$ be the human's position trajectory generated by the intention estimation algorithm. The robot trajectory $x_{Rd}(t)$ is generated by

$$x_{Rd}(t) \triangleq T\left(x_H(t)\right) = A_H x_H(t) + b_H, \tag{22}$$

where $T(\cdot)$ is a generic transformation between $x_H(t)$ and $x_{Rd}(t)$. In particular, an affine transformation with known task-specific parameters $A_H : \mathbb{R}^{3\times3}$ and $b_H \in \mathbb{R}^3$ are used in this article. If the generated robot desired trajectory $x_{Rd}(t)$ enters an unsafe zone represented by an ellipsoid drawn around the human, then the robot's desired trajectory is required to be modified such that the robot does not collide with the human. To modify the robot's desired trajectory, a dynamic model of $x_{Rd}(t)$ is formulated with a control input such that two main objectives are satisfied; (1) the controller is able to modify $x_{Rd}(t)$ such that $x_{Rd}(t)$ does not enter the safety ellipsoid around the human, and (2) the modified $x_{Rd}(t)$ still keeps track of the original desired trajectory $x_{Rd}(t)$ closely. In order to achieve this, consider the following robot's desired end-effector motion dynamics $\dot{x}_{Rd}(t) = f_{Rn}(x_{Rd}(t))$, where $f_{Rn} : \mathbb{R}^3 \to \mathbb{R}^3$ is a

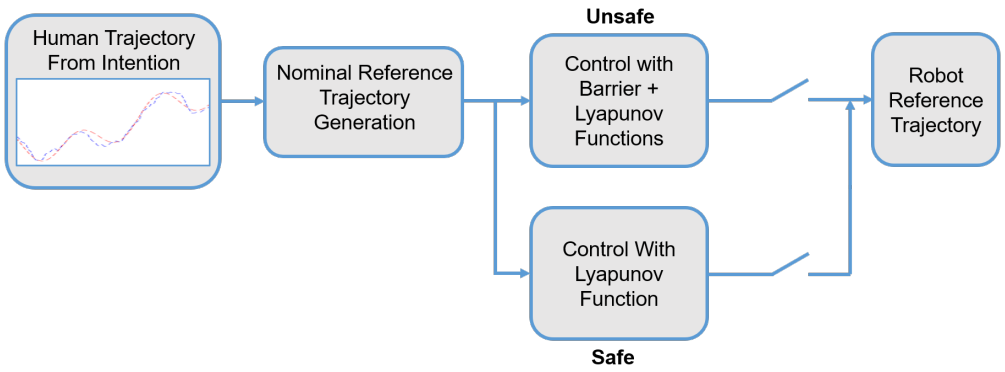

Figure 11: Block diagram of the reference trajectory generator for the robot based on observed and inferred human action. The human trajectory generated from human intention estimator is fed into a reference trajectory generator block for the robot. If the robot reference trajectory enters the unsafe zone around the human then a CBF approach is used to modify the robot's reference trajectory. Otherwise, a CLF approach is used to generate a bounded and convergent reference trajectory for the robot.

nonlinear continuous function that governs robot's end-effector motion dynamics. Based on the estimates of the human trajectory intention $\hat{x}_H(t)$ and its time derivative $\dot{\hat{x}}_H(t)$, and using function approximation methods such as extreme learning machine (ELM), a multi-layered neural network (NN), the function $\dot{\hat{x}}_{Rd}(t) = \hat{f}_{Rn}\left(\hat{x}_{Rd}(t) = T\left(\hat{x}_H(t)\right)\right)$ can be learned, where $\hat{f}_{Rn} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is an approximation of $f_{Rn}(\cdot)$. See "Extreme Learning Machines" for more details of ELM. Gaussian Mixture Models (GMMs) can also be used to approximate the nonlinear function $\hat{f}_{Rn}(\cdot)$ for capturing the uncertainty in the data. To ensure that the trajectories $\hat{x}_{Rd}(t)$ generated by the nominal model $\hat{f}_{Rn}(\hat{x}_{Rd}(t))$ avoids the working region of the human operator, represented by a 3D ellipsoid, a controller $\bar{u}(t)$ is designed to provide modifications to $\hat{x}_{Rd}(t)$ using CBF and CLF theory. In order the achieve this, a modified robot's desired end-effector dynamics can be written as

$$\dot{\hat{x}}_{Rd}(t) = \hat{f}_{Rn}(\hat{x}_{Rd}(t)) + \bar{u}(t), \tag{23}$$

where $\bar{u}(t) \in \mathbb{R}^3$ is a control input that is designed to avoid the unsafe zones. In next section, the design of $\bar{u}(t)$ term using CBF and CLF theory is discussed.

Control Barrier Function Formulation

A CBF defines a forward invariant region such that solutions of the DS which start in that region remain in that region for all time. The choice of an invariant region is application specific as long as the CBF conditions defined in "CBF-CLF" is a valid choice. In this article, we use

the CBF to avoid the obstacle region, that is, the robot trajectories that start outside the unsafe region stay outside the unsafe region for all the time. Ellipse, circle, and ellipsoid are commonly used shapes to represent obstacles or a region of operation where it is unsafe for the robot's end-effector to enter. For the application, we use a 3D ellipsoid shape around a human as an unsafe zone for the robot to enter. The ellipsoidal barrier function is defined as a continuously differentiable function $b : \mathbb{R}^3 \to \mathbb{R}$, given by

$$b(x(t)) = \frac{(x_1(t) - x_{1g})^2}{a_1^2} + \frac{(x_2(t) - x_{2g})^2}{a_2^2} + \frac{(x_3(t) - x_{3g})^2}{a_3^2} - 1, \tag{24}$$

where $x(t) = [x_1(t), x_2(t), x_3(t)]^T$, x_{1g}, x_{2g}, and x_{3g} are the coordinates of the center of gravity of the ellipse, a_1, a_2, and a_3 are the semi-axes of an ellipsoid. Using the Barrier function in (24), the CBF candidate is defined as $B(x(t)) = \frac{1}{b(x(t))}$ that satisfies the properties given in "CBF-CLF".

Control Lyapunov Function Formulation

The CLF is used in order to make sure that the desired trajectories of the robot remain stable and converge to a point in the 3D space, where the object is being placed. Consider a quadratic Lyapunov function candidate of the form $V(\hat{x}_{Rd}(t)) = \frac{1}{2}(\hat{x}_{Rd}(t) - x^*_{Rd})^T(\hat{x}_{Rd}(t) - x^*_{Rd})$, $\forall \hat{x}_{Rd}(t) \in \mathcal{X}_u$. On taking the time derivative of the CLF and using (23), we have

$$\dot{V}(\hat{x}_{Rd}) = (\hat{x}_{Rd} - x^*_{Rd})^T \left(\hat{f}_{Rn}(\hat{x}_{Rd}) + \bar{u} \right) \leq -\gamma_l V(\hat{x}_{Rd}), \tag{25}$$

where $\gamma_l \in \mathbb{R}$ is a positive constant.

Control Design using CBF and CLF Constraints

Given the approximated nonlinear function $\hat{f}_{Rn}(\hat{x}_{Rd}(t))$, an online controller learning problem with safety and stability constraints is now discussed. Two cases are considered: (1) when the robot's desired trajectories are not crossing the safety ellipsoid, and (2) when the robot's desired trajectories are crossing the safety ellipsoid.

Case I: When the robot's desired trajectories are not crossing the safety ellipsoid, then a controller that uses CLF constraints is synthesized. The controller with CLF constraints will ensure that the trajectories $\hat{x}_{Rd}(t)$ generated by (23) remain stable with respect to the equilibrium point x^*_{Rd}. To this end, the following quadratic program (QP) is solved to synthesize the controller

$$\begin{aligned} \bar{u}^*(x) = \arg \min_{\bar{u} \in \mathbb{R}^n} & \frac{1}{2} \bar{u}^T P \bar{u} + p\delta^2 \\ \text{s.t.} \quad & (\hat{x}_{Rd} - x^*_{Rd})^T \bar{u} \leq -(\hat{x}_{Rd} - x^*_{Rd})^T \hat{f}_{Rn}(\hat{x}_{Rd}) - \gamma_l V(\hat{x}_{Rd}) + \delta, \quad \forall \hat{x}_{Rd} \in \mathcal{X}_0, \end{aligned} \tag{26}$$

where $\delta \in \mathbb{R}$ is a positive constant added as a relaxation variable that ensures solvability of the QP as penalized by $p > 0$, and $P \in \mathbb{R}^{3\times 3}$ is a positive definite weight matrix.

Case II: When the robot's desired trajectories are crossing the safety ellipsoid, then the optimal controller is synthesized using both CLF and CBF constraints. The CBF constraint will ensure that the trajectories $\hat{x}_{Rd}(t)$ generated by (23) will not enter the unsafe ellipsoidal zone and the CLF constraint will ensure that the trajectories remain stable with respect to x^*_{Rd}. The optimization problem subject to the CBF and CLF is given by

$$\bar{u}^*(x) = \arg\min_{\bar{u}\in\mathbb{R}^n} \frac{1}{2}\bar{u}^T P\bar{u} + p\delta^2$$

$$\text{s.t.} \quad \frac{\partial B(\hat{x}_{Rd})}{\partial \hat{x}_{Rd}}\bar{u} \leq \frac{\gamma_B}{B(\hat{x}_{Rd})} - \frac{\partial B(\hat{x}_{Rd})}{\partial \hat{x}_{Rd}}\hat{f}_{Rn}(\hat{x}_{Rd}), \quad \forall \hat{x}_{Rd} \in \mathcal{X}_0, \tag{27}$$

$$(\hat{x}_{Rd} - x^*_{Rd})^T\bar{u} \leq -(\hat{x}_{Rd} - x^*_{Rd})^T\hat{f}_{Rn}(\hat{x}_{Rd}) - \gamma_l V(\hat{x}_{Rd}) + \delta, \quad \forall \hat{x}_{Rd} \in \mathcal{X}_0, \tag{28}$$

where $\delta \in \mathbb{R}$ is a positive constant, $\gamma_l > 0$ and $\gamma_B > 0$ are positive constants.

Numerical Simulation Results

Given the position estimates of the human demonstrated motion trajectory, the robot desired trajectory can be obtained from equation (23). However, the position trajectory generated from the affine function does not guarantee the human workspace will be avoided. As can be seen in Fig. 12, the robot desired trajectory (in solid green lines) intersects with the human workspace. An ellipsoid is used to represent the unsafe region for the robot's end-effector. In that region the human carries out their own tasks. Based on the robot's desired end-effector states (position and velocity), an ELM parametrization is used to approximate the nonlinear function that governs the underlying dynamics. A controller is synthesized by optimizing the objective function subject to the constraints derived using the CBF-CLF. To ensure the position trajectory generated from the learned model avoids the human workspace. In Fig. 12, the simulated result are shown in which the controller applies corrections to the trajectory generated by the model as to avoid the human workspace while still reaching the target location. In Fig. 13, an example is illustrated in which the robot's desired end-effector trajectory is far away from the human workspace, thus, the robot poses no harm to the human operator. Therefore, a CLF is used to learn a controller, which ensures that the robot's end-effector reaches to the target location without interfering with the human workspace.

Adaptive Robot Controller

In this section, an adaptive robot controller for the robot arm is designed that follows the desired trajectory generated by the robot desired trajectory generator block. First a robot

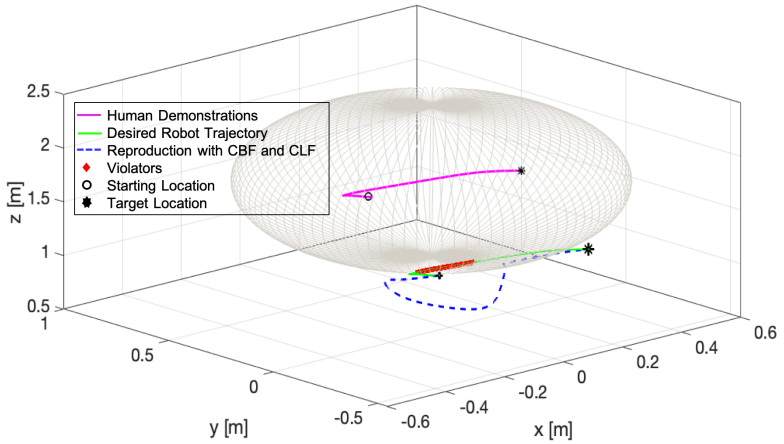

Figure 12: Human demonstrated motion trajectory is enclosed by an ellipsoid to represent the unsafe region for a robot's end-effector to enter. The desired robot's end-effector position trajectory is modified by a controller that avoids the ellipsoid region while reaching to the target location.

dynamic model represented by EL dynamics is discussed. Then an adaptive controller for the robot manipulator that tracks the desired trajectory generated for the robot is presented with a Lyapunov stability analysis.

Robot Dynamic Model

Consider the following robot manipulator dynamics, represented by an EL dynamics, given by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau. \tag{29}$$

In (29), $q(t),\ \dot{q}(t),\ \ddot{q}(t) \in \mathbb{R}^d$ denote the generalized states, $M(q) \in \mathbb{R}^{d\times d}$ denotes a generalized inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{d\times d}$ denotes a generalized centripetal-Coriolis matrix, $G(q) \in \mathbb{R}^d$ denotes a generalized gravity vector, $\tau(t) \in \mathbb{R}^d$ represents the generalized input control vector. It is well known that, for the robot manipulator dynamics in (29), the following properties hold.

Property 1: The matrix $M\left(q\right)$ is positive definite and $\lambda_{min}I \leq M\left(q\right) \leq \lambda_{max}I$ where λ_{min} and λ_{max} are positive constants.

Property 2: For any differentiable vector $\xi(t) \in \mathbb{R}^n$, the dynamics in (29) are linearly parameterizable as follows $M\left(q\right)\dot{\xi} + C\left(q,\dot{q}\right)\xi + g\left(q\right) = Y\left(q,\dot{q},\xi,\dot{\xi}\right)\Theta$, where $\Theta \in \mathbb{R}^k$ is a set of robot-specific parameters and $Y : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n\times k}$ is a matrix of known functions of the generalized coordinates and their higher directives.

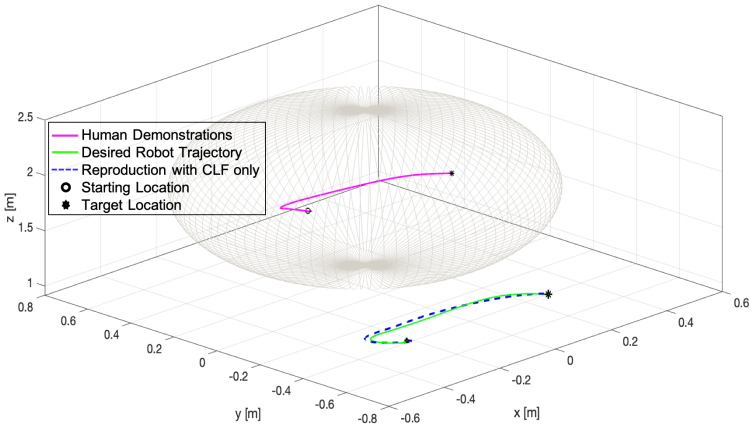

Figure 13: A scenario is shown where the robot's desired end-effector position trajectory is far from the unsafe region dictated by the human operator. A controller is learned using CLF that avoids the region where human is present (represented by an ellipsoid) while reaching to the target location.

Property 3: The inertia and centripetal-Coriolis matrices satisfied the property $\xi^T\left(\dot{M}-2C\right)\xi = 0,\ \forall\xi(t)\in\mathbb{R}^n$, that is, $\left(\dot{M}-2C\right)$ is skew-symmetric matrix. See [112], [116] for more details.

Let $x_R(t)\in\mathcal{W}\subset\mathbb{R}^m$ be the the robot's end-effector state, and $q(t)\in Q_J\subset\mathbb{R}^{n_R}$ be the joint angles of the robot. The forward and velocity kinematic relationships are given by $x_R(t)=h_{\mathrm{FR}}(q(t))$, $\dot{x}_R(t)=J(q(t))\dot{q}(t)$, where $h_{\mathrm{FR}}:\mathbb{R}^{n_R}\to\mathbb{R}^m$ is the mapping between the joint space Q_J and the task space $\mathcal{W}$, and $J(q(t)):\mathbb{R}^n\to\mathbb{R}^{m\times n}$ is the known Jacobian.

Controller Design

The desired trajectory $x_{Rd}(t)\in\mathbb{R}^m$ of the robot end-effector state is generated based on the robot desired trajectory generator block. In order to design the robot controller, consider the signals $a(t)$, $v(t)$, and $s(t)$ for non-redundant robots given by $v = J^{-1}(q)\left(\dot{x}_{Rd}-\Lambda e\right)$, $a=\dot{J}^{-1}(q)\left(\dot{x}_{Rd}-\Lambda e\right)+J^{-1}(q)\left(\ddot{x}_{Rd}-\Lambda\dot{e}\right)$, and $s=J^{-1}(q)\left(-\dot{x}_{Rd}+\Lambda e\right)+\dot{q}$, where $e(t)\triangleq\left(x_R(t)-x_{Rd}(t)\right)\in\mathbb{R}^m$ is the tracking error, $v=\dot{q}-s$ and $a=\dot{v}$, $\Lambda\in\mathbb{R}^m\times\mathbb{R}^m$ is a positive definite diagonal matrix. Also, consider the filtered error in the task space, $r(t)\triangleq J(q)s(t)$. Using (29), $r(t)$ can be expressed as

$$r=-\dot{x}_{Rd}+\Lambda\left(x_R-x_{Rd}\right)+\dot{x}_R=\dot{e}+\Lambda e. \tag{30}$$

The control input in (29) is designed to be

$$\tau=\hat{Y}\hat{\Theta}-K_t\hat{s}-J^TK_J^T\hat{e}, \tag{31}$$

where $\hat{Y} = Y\left(q, \dot{q}, \hat{v}, \hat{a}\right)$ is a regressor matrix, and $\hat{\Theta}$ consists of estimates of the parameters M, C, and g, respectively, $\hat{e} = x_R - \hat{x}_{Rd}$ is the estimated tracking error, $K_t \in \mathbb{R}^{n_R} \times \mathbb{R}^{n_R}$ and $K_J \in \mathbb{R}^m \times \mathbb{R}^m$ are diagonal matrices and $\hat{v} = J^{-1}(q)\left(\dot{\hat{x}}_{Rd} - \Lambda\hat{e}\right)$, $\hat{a} = \dot{J}^{-1}(q)\left(\dot{\hat{x}}_{Rd} - \Lambda\hat{e}\right) + J^{-1}(q)\left(\ddot{\hat{x}}_{Rd} - \Lambda\dot{\hat{e}}\right)$, and $\hat{s} = J^{-1}(q)\left(-\dot{\hat{x}}_{Rd} + \Lambda\hat{e}\right) + \dot{q}$. The parameter update rule is given by

$$\dot{\hat{\Theta}} = \mathrm{proj}\left(-\Gamma^{-1}\hat{Y}^T\hat{s}\right), \tag{32}$$

where $\Gamma \in \mathbb{R}^{k\times k}$ is a positive definite matrix and $\mathrm{proj}\left(\cdot\right)$ is a standard projection operator, which ensures that the parameter estimates are bounded (see [117] for details).

Remark 1: The parameter estimation error $\tilde{\Theta}(t) \triangleq \Theta(t) - \hat{\Theta}(t)$ is uniformly continuous since $\hat{\Theta}(t)$ evolves according to (32).

Substituting (31) in (29) yields the closed-loop system as given by

$$M\dot{s} + Cs + K_t s = -Y\tilde{\Theta} - J^T K_J^T e + \tilde{Y}\tilde{\Theta} - J^T K_J^T \tilde{x}_{Rd} - M\tilde{a} - C\tilde{v} + K_t\tilde{s}. \tag{33}$$

More details of the signals in (33) can be found in [9].

Assumption 1: The signals $\tilde{x}_{Rd}(t)$, $\dot{\tilde{x}}_{Rd}(t)$, and $\ddot{\tilde{x}}_{Rd}(t)$ are uniformly continuous, and $\|\tilde{x}_{Rd}(t)\|$, $\|\dot{\tilde{x}}_{Rd}(t)\|$, $\|\ddot{\tilde{x}}_{Rd}(t)\| \to 0$ as $t \to \infty$.

Remark 2: Based on Assumption 1 and formulas for $\tilde{v}(t)$, $\tilde{a}(t)$, $\tilde{s}(t)$, $\tilde{Y}(t)$, it can be seen that $\|\tilde{v}(t)\|$, $\|\tilde{a}(t)\|$, $\|\tilde{s}(t)\|$, $\|\tilde{Y}(t)\| \to 0$ as $t \to \infty$.

Stability Analysis

A Lyapunov stability theorem is provided for the closed-loop system defined in (33).

Theorem 1: The closed-loop system in (33) is stable and the tracking error is globally uniformly ultimately bounded in the sense that $\|e\left(t\right)\| \to \epsilon_0 \exp\left(-\epsilon_1 t\right) + \epsilon_2$, where $\epsilon_0,\ \epsilon_1,\ \epsilon_2 \in \mathbb{R}$ denote positive bounding constants, if the following conditions about the gains of the controller defined in (31) are satisfied: $K_t > 0$, $K_J > 0$.

Proof: See [9] for more details. ■

Extensions of the adaptive controller design for redundant manipulators can be found in [118]. The stability analysis can be extended to the modified controller and redundant manipulator can be shown to asymptotically track the desired trajectory.

Numerical Simulation Results

The results of the tracking controller, shown in Figs. 14, 15 were generated using a simulation of Rethink Robotics' 7-DoF Baxter Robot in the Gazebo simulation platform. The reference trajectories and their derivatives, which are obtained by observing the motion of the human hand and then modified according to the CBF and CLF constraints, are used as $\hat{x}_{Rd}(t)$ and $\dot{\hat{x}}_{Rd}(t)$ for the controller defined in (31). During execution, the true robot task space end-effector positions $x_R(t)$ and velocities $\dot{x}_R(t)$ are read from virtual sensors provided in Gazebo.

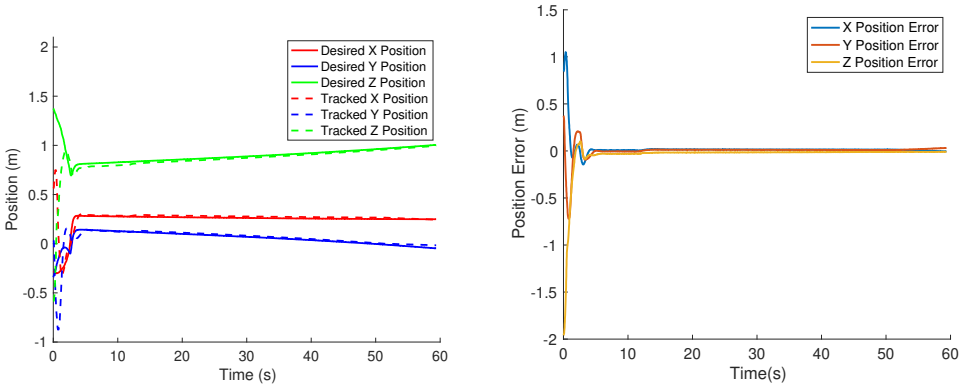

Figure 14: (a) The desired trajectories of the robot end-effector being tracked by the robot controller. (b) The position tracking errors.

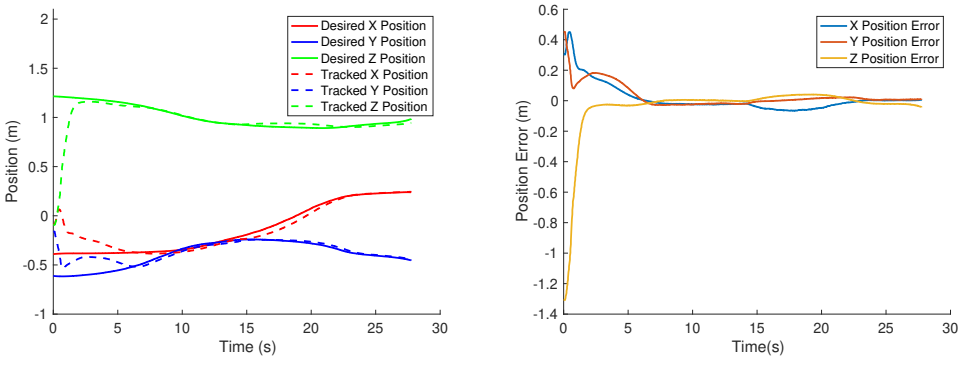

Figure 15: (a) The desired trajectories of the robot end-effector being tracked by the robot controller. (b) The position tracking errors.

Experimental Results

A collaborative task involving a human and a robot jointly moving a heavy object is designed for experimental evaluation. In the designed task the robot must infer the motion intention of the human online and adjust its trajectory to coordinate with the human in order to move a rectangular box from a workbench to an elevated surface. While doing so, the robot must follow a trajectory which does not enter the human workspace to ensure the safety of the human. The human is assumed to move between two different dynamical models (one for moving the box parallel to the surface of the workbench and the other for lifting the object and placing it on the elevated surface). The experiment is demonstrated through a sequence of images as shown in Fig. 16.

Four demonstrations of the human hand motion training data for each motion are collected using the Microsoft Kinect skeletal tracking to train two NN models (corresponding to the two dynamical models) subject to contraction constraints. The single layer NN model for the translational motion consists of 12 neurons and the model for the vertical motion consists of 15 neurons. The trained NNs are then used as motion models for EKFs which are used in the IMM framework. A Microsoft Kinect sensor placed on top of Baxter's head is used for acquiring the human hand position data online during the experiment. The IMM estimator is used to estimate the 3D position of the human hand and likelihood of each model being the true model at each time instance. A controller $\bar{u}(t)$ is designed by the solving the optimization problem in (26)-(28) which generates the robot desired trajectory $\hat{x}_{Rd}(t)$. The controller $\bar{u}(t)$ ensures that the robot desired trajectory stays outside the human workspace ensuring the safety of the human. The matrix H is chosen to be identity of for the optimization. The adaptive tracking controller in (31) is used to track the safe trajectory generated by solving the optimization problem. The experiments are carried out using the Baxter research robot and Baxter API along with ROS. The gains chosen for the adaptive controller are as $K_J = \mathrm{diag}\{27, 30, 37.5, 0, 0, 0\}$, $K_t = \mathrm{diag}\{3, 6, 3, 3.25, 3, 3, 1.5\}$ and $\Lambda = \mathrm{diag}\{1, 1, 1, 0, 0, 0\}$. The orientation of the robot end-effector was constant throughout the experiment. The results of the experiment are summarized in Figs. 17(a) and 17(b). The robot end-effector trajectories converge and track the desired trajectories as shown in Fig. 17(a). The error between the actual and desired trajectory is shown in Fig. 17(b).

Conclusion

For assistive robots to integrate seamlessly into human environments, they should be able to understand the intentions of the human agents and adapt to their motion. Many HRC tasks in advanced manufacturing operations require humans and robots to perform joint tasks, such as carry a heavy load or a large flexible material together or perform a joint welding operation,

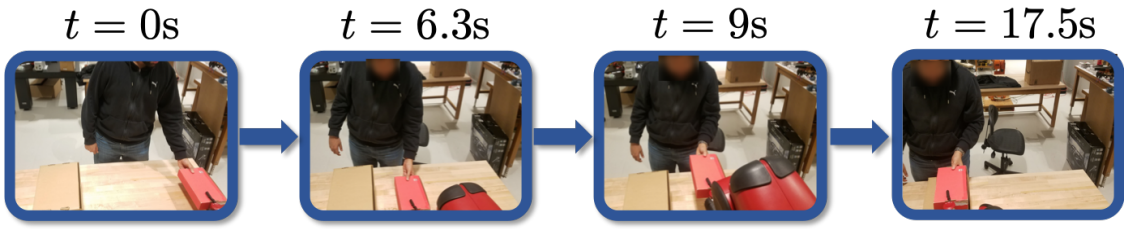

Figure 16: Experimental results for moving an object collaboratively using a Baxter research robot and a person.

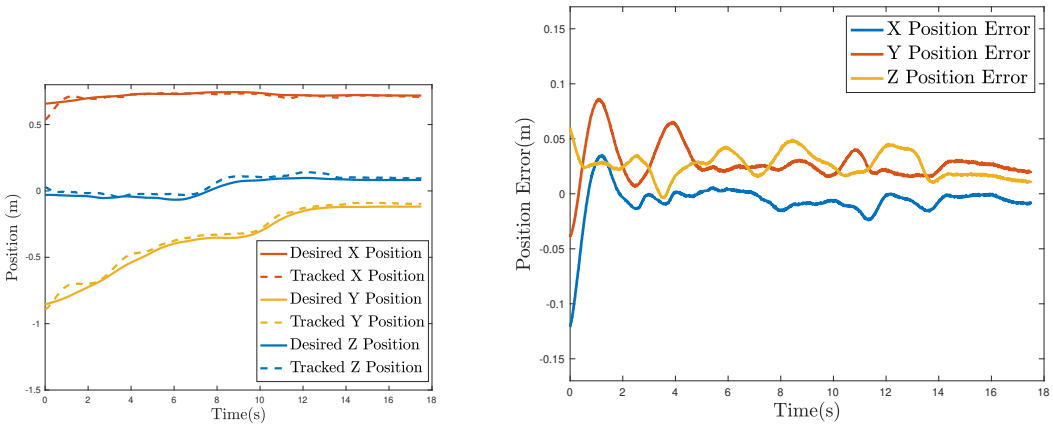

Figure 17: Experimental results from a Baxter research robot (a) the desired trajectories of the Baxter robot end-effector being tracked by the robot controller; (b) the position tracking errors.

for which the robot can act as an assistant to learn the motion behavior of the humans based on the RGB-D sensor data. Then the robot can adapt its motions to carry out a joint task. In this article, a survey of the topic is first provided by presenting existing literature in a tutorial manner, and then approaches that focus and expand on the work are discussed that infer the human motion trajectory and estimate its reaching goal intention. The first method is based on a ML estimation technique, called approximate E-M that uses online model learning to accommodate uncertainties in the human's motion. The second method is based on a multiple model estimator that switches between multiple nonlinear human motion models. Short tutorials on approximate E-M and multiple model estimation methods are presented that uses nonlinear models of the human motion learned using NN function approximation from the labeled sensor data. For ML-based estimator, the optimizer is initialized using human gaze cues as well as using random initialization. For MAP-based estimator, a gaze-based prior is used which is computed by analyzing RGB image data using a deep NN and the results are compared with the uniform prior. A human-in-the-loop control strategy is discussed that uses the estimated human motion

trajectory and intended reaching goal location to determine a safe robot reference trajectory which is tracked using an adaptive controller for the robot. For the desired trajectory generation, a CBF formulation is used to generate trajectories that are safe around humans. The desired trajectories are generated with two objectives - (1) the robot end-effector trajectories do not enter a safety region around the human, (2) the robot end-effector trajectories are able to synchronize with the human motion as closely as possible. The adaptive controller converges to the desired trajectories generated by the trajectory generation block. A case study of human and robot carrying an object together is discussed. In future, challenges such as ensuring safety of humans in the presence of actuator or sensor failures can be a potential topic to explore, for example, via human supervision of automation to achieve resilient control [119]. Estimating intention using sensors such as ultrasound imaging can be another avenue for potential research.

Acknowledgment

This work was in part supported by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program, and in part by Office of Naval Research Award No. N00014-20-1-2040. The authors would like to thank Tanisha Mitra for her help with figure formatting.

References

[1] S. Berger, *Making in America: From Innovation to Market*. MIT Press, 2013.

[2] H. Modares, I. Ranatunga, B. AlQaudi, F. L. Lewis, and D. O. Popa, "Intelligent human–robot interaction systems using reinforcement learning and neural networks," in *Trends in control and decision-making for human–robot collaboration systems*, 2017, pp. 153–176.

[3] J. Mumm and B. Mutlu, "Human-robot proxemics: physical and psychological distancing in human-robot interaction," in *Proceedings of the 6th international conference on Human-robot interaction*, 2011, pp. 331–338.

[4] C. Nobile, "Robotics business review perspectives 2013: Outlook for next-gen, new-gen industrial co-worker robotics," 2013.

[5] M. Sampath and P. P. Khargonekar, "Socially responsible automation: A framework for shaping the future," *National Academy of Engineering Bridge*, vol. 48, no. 4, pp. 45–52, 2018.

[6] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.

[7] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and*

trends in human-computer interaction, vol. 1, no. 3, pp. 203–275, 2007.

[8] E. Martinez and A. P. del Pobil, "Safety for human-robot interaction in dynamic environments," in *IEEE International Symposium on Assembly and Manufacturing*, 2009, pp. 327–332.

[9] H. C. Ravichandar, D. Trombetta, and A. P. Dani, "Human intention-driven learning control for trajectory synchronization in human-robot collaborative tasks," in *IFAC Cyber-Physical Human Systems*, vol. 51, no. 34, 2019, pp. 1–7.

[10] M. Sodhi, B. Reimer, and I. Llamazares, "Glance analysis of driver eye movements to evaluate distraction," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 529–538, 2002.

[11] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 370–382, 2007.

[12] B. Alqaudi, H. Modares, I. Ranatunga, S. M. Tousif, F. L. Lewis, and D. O. Popa, "Model reference adaptive impedance control for physical human-robot interaction," *Control Theory and Technology*, vol. 14, no. 1, pp. 68–82, 2016.

[13] C.-S. Tsai, J.-S. Hu, and M. Tomizuka, "Ensuring safety in human-robot coexistence environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 4191–4196.

[14] D.-J. Kim, Z. Wang, N. Paperno, and A. Behal, "System design and implementation of UCF-MANUS – an intelligent assistive robotic manipulator," *IEEE/ASME transactions on mechatronics*, vol. 19, no. 1, pp. 225–237, 2012.

[15] F. Zhang and H. Huang, "Source selection for real-time user intent recognition toward volitional control of artificial legs," *IEEE journal of biomedical and health informatics*, vol. 17, no. 5, pp. 907–914, 2012.

[16] T. G. Sugar, J. He, E. J. Koeneman, J. B. Koeneman, R. Herman, H. Huang, R. S. Schultz, D. Herring, J. Wanberg, S. Balasubramanian *et al.*, "Design and control of rupert: a device for robotic upper extremity repetitive therapy," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 15, no. 3, pp. 336–346, 2007.

[17] D. A. Baldwin and J. A. Baird, "Discerning intentions in dynamic human action," *Trends in cognitive sciences*, vol. 5, no. 4, pp. 171–178, 2001.

[18] M. A. Simon, *Understanding human action: Social explanation and the vision of social science*. SUNY Press, 1982.

[19] D. L. Kleinman, S. Baron, and W. Levison, "An optimal control model of human response part i: Theory and validation," *Automatica*, vol. 6, no. 3, pp. 357–369, 1970.

[20] S. Baron, D. Kleinman, and W. Levison, "An optimal control model of human response

part ii: prediction of human performance in a complex task," *Automatica*, vol. 6, no. 3, pp. 371–383, 1970.

[21] R. B. Warrier and S. Devasia, "Inferring intent for novice human-in-the-loop iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1698–1710, 2017.

[22] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, "Goal inference improves objective and perceived performance in human-robot collaboration," in *International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 940–948.

[23] Y. Li and S. Ge, "Human-robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.

[24] D. Kulic and E. A. Croft, "Affective state estimation for human–robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 991–1000, 2007.

[25] E. Meisner, V. Isler, and J. Trinkle, "Controller design for human-robot interaction," *Autonomous Robots*, vol. 24, no. 2, pp. 123–134, 2008.

[26] Y. S. Razin, K. Pluckter, J. Ueda, and K. Feigh, "Predicting task intent from surface electromyography using layered hidden markov models," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1180–1185, 2017.

[27] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction." in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW).*, vol. 5, 2003, pp. 53–53.

[28] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, and S. Srinivasa, "Learning the communication of intent prior to physical collaboration," in *IEEE International Symposium on Robot and Human Interactive Communication*, September 2012.

[29] Y. Matsumoto, J. Heinzmann, and A. Zelinsky, "The essential components of human-friendly robot systems," in *International Conference on Field and Service Robotics*, 1999, pp. 43–51.

[30] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making service robots human-safe," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 696–701.

[31] B. Mutlu, J. Forlizzi, and J. Hodgins, "A storytelling robot: Modeling and evaluation of human-like gaze behavior," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 518–523.

[32] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 143–166, 2003.

[33] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[34] H. Ravichandar and A. P. Dani, "Human intention inference using expectation-maximization algorithm with online model learning," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 855–868, 2017.

[35] K. W. Strabala, M. K. Lee, A. D. Dragan, J. L. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli, "Towards seamless human-robot handovers," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 112–132, 2013.

[36] D. De Carli, E. Hohert, C. A. Parker, S. Zoghbi, S. Leonard, E. Croft, and A. Bicchi, "Measuring intent in human-robot cooperative manipulation," in *IEEE International Workshop on Haptic Audio visual Environments and Games.* IEEE, 2009, pp. 159–163.

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

[38] D. Gehrig, P. Krauthausen, L. Rybok, H. Kuehne, U. D. Hanebeck, T. Schultz, and R. Stiefelhagen, "Combined intention, activity, and motion recognition for a humanoid household robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4819–4825.

[39] O. C. Schrempf and U. D. Hanebeck, "A generic model for estimating user intentions in human-robot cooperation." in *International Conference on Informatics in Control, Automation and Robotics*, 2005, pp. 251–256.

[40] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, 2014.

[41] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[42] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs," in *Robotics: Science and Systems (RSS)*, 2014.

[43] N. Hu, Z. Lou, G. Englebienne, and B. Kröse, "Learning to recognize human activities from soft labeled data," in *Robotics: Science and Systems, Berkeley, USA*, 2014.

[44] R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis, "Understanding human intentions via hidden markov models in autonomous mobile robots," in *ACM/IEEE international conference on Human robot interaction*, 2008, pp. 367–374.

[45] D. K. E. Croft, "Estimating intent for human-robot interaction," in *IEEE International Conference on Advanced Robotics*, 2003, pp. 810–815.

[46] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 161–177.

[47] R. Luo and D. Berenson, "A framework for unsupervised online human reaching motion recognition and early prediction," in *International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2426–2433.

[48] M. Monfort, A. Liu, and B. D. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation." in *AAAI Conference on Artificial Intelligence*, 2015, pp. 3672–3678.

[49] A. Ramadan, J. Choi, C. J. Radcliffe, J. M. Popovich, and N. P. Reeves, "Inferring control intent during seated balance using inverse model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 224–230, 2018.

[50] Y. Yildiz, A. Agogino, and G. Brat, "Predicting pilot behavior in medium scale scenarios using game theory and reinforcement learning," in *AIAA Modeling and Simulation Technologies (MST) Conference*, 2013, p. 4908.

[51] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 885–892.

[52] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Intelligent Robots and Systems, 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 299–306.

[53] H. C. Ravichandar, A. Kumar, and A. P. Dani, "Gaze and motion information fusion for human intention inference," *International Journal of Intelligent Robotics and Applications*, vol. 2, no. 2, pp. 136–148, 2018.

[54] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response." in *Robotics: Science and Systems*, 2013.

[55] H. T. Dinh, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust observers for uncertain nonlinear systems," *Neural Networks*, vol. 60, pp. 44–52, 2014.

[56] G. C. Goodwin and J. Aguero, "Approximate E-M algorithms for parameter and state estimation in nonlinear stochastic models," in *IEEE Conference on Decision and Control, and European Control Conference*. IEEE, 2005, pp. 368–373.

[57] G. Goodwin and A. Feuer, "Estimation with missing data," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 5, no. 3, pp. 220–244, 1999.

[58] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. E. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 187–192, 2013.

[59] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and rise feedback control structure," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

[60] N. Hogan, "An organizing principle for a class of voluntary movements," *Journal of Neuroscience*, vol. 4, no. 11, pp. 2745–2754, 1984.

[61] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[62] H. Ravichandar, I. Salehi, and A. Dani, "Learning partially contracting dynamical systems from demonstrations," in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78. PMLR, 2017, pp. 369–378.

[63] H. Ravichandar and A. P. Dani, "Human intention inference through interacting multiple model filtering," in *IEEE Conference on Multisensor Fusion and Integration (MFI)*, 2015, pp. 220–225.

[64] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[65] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic control*, vol. 41, no. 4, pp. 478–493, 1996.

[66] X. R. Li, V. P. Jilkov, J. Ru, and A. Bashi, "Expected-mode augmentation algorithms for variable-structure multiple-model estimation," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 175–180, 2002.

[67] I. Ranatunga, F. L. Lewis, D. O. Popa, and S. M. Tousif, "Adaptive admittance control for human–robot interaction using model reference design and adaptive inverse filtering," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 278–285, 2016.

[68] N. Hogan, "Impedance control: An approach to manipulation: Part II-Implementation," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 8–16, 1985.

[69] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 249–254.

[70] G. Hoffman and C. Breazeal, "Effects of anticipatory perceptual simulation on practiced human-robot tasks," *Autonomous Robots*, vol. 28, no. 4, pp. 403–423, 2010.

[71] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 4000–4007.

[72] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing," *Robotics*, pp. 441–449, 2013.

[73] V. Shivashankar, K. N. Kaipa, D. S. Nau, and S. K. Gupta, "Towards integrating hierarchical goal networks and motion planners to support planning for human-robot

teams," in *AAAI Fall Symposium: Artificial Intelligence for Human-Robot Interaction*, 2014, pp. 13–15.

[74] J. Shah and C. Breazeal, "An empirical analysis of team coordination behaviors and action planning with application to human–robot teaming," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 52, no. 2, pp. 234–245, 2010.

[75] T. Yucelen, Y. Yildiz, R. Sipahi, E. Yousefi, and N. Nguyen, "Stability limit of human-in-the-loop model reference adaptive control architectures," *International Journal of Control*, vol. 91, no. 10, pp. 2314–2331, 2018.

[76] E. Yousefi, Y. Yildiz, R. Sipahi, and T. Yucelen, "Stability analysis of a human-in-the-loop telerobotics system with two independent time-delays," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6519–6524, 2017.

[77] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.

[78] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conference*. IEEE, 2014, pp. 226–241.

[79] U. V. Kalabić, I. V. Kolmanovsky, and E. G. Gilbert, "Reduced order extended command governor," *Automatica*, vol. 50, no. 5, pp. 1466–1472, 2014.

[80] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Advances in Neural Information Processing Systems*, 2017, pp. 908–919.

[81] G. P. Kontoudis and K. G. Vamvoudakis, "Kinodynamic motion planning with continuous-time q-learning: An online, model-free, and safe navigation framework," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3803–3817, 2019.

[82] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[83] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the American Control Conference*, 2004, pp. 5576–5581.

[84] L. S. Breger and J. P. How, "Safe trajectories for autonomous rendezvous of spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1478–1489, 2008.

[85] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems with state constraints," in *IFAC-PapersOnLine*, 2017, pp. 4063–4069.

[86] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. M. Gregory, "l_1 adaptive control

for safety-critical systems," *IEEE Control Systems Magazine*, vol. 31, no. 5, pp. 54–104, 2011.

[87] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2004, pp. 477–492.

[88] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[89] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes," in *IEEE 55th Conference on Decision and Control*. IEEE, 2016, pp. 4661–4666.

[90] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 2460–2465.

[91] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 827–834.

[92] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *American Control Conference*. IEEE, 2016, pp. 2252–2258.

[93] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[94] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328.

[95] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.

[96] Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-aware reinforcement learning framework with an actor-critic-barrier structure," in *American Control Conference*. IEEE, 2019, pp. 2352–2358.

[97] I. Salehi, G. Yao, and A. P. Dani, "Active sampling based safe identification of dynamical systems using extreme learning machines and barrier certificates," in *IEEE International Conference on Robotics and Automation*, 2019.

[98] K. Akash, K. Polson, T. Reid, and N. Jain, "Improving human-machine collaboration through transparency-based feedback–part i: Human trust and workload model," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 315–321, 2019.

[99] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from

demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2, pp. 79–91, 2004.

[100] S. Rahman, B. Sadrfaridpour, and Y. Wang, "Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing," in *ASME 2015 Dynamic Systems and Control Conference*, 2015.

[101] S. M. Rahman and Y. Wang, "Mutual trust-based subtask allocation for human–robot collaboration in flexible lightweight assembly in manufacturing," *Mechatronics*, vol. 54, pp. 94–109, 2018.

[102] S. K. Jayaraman, C. Creech, L. P. Robert Jr, D. M. Tilbury, X. J. Yang, A. K. Pradhan, and K. M. Tsui, "Trust in av: An uncertainty reduction model of av-pedestrian interactions," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 133–134.

[103] V. Srivastava, R. Carli, C. Langbort, and F. Bullo, "Attention allocation for decision making queues," *Automatica*, vol. 50, no. 2, pp. 378–388, 2014.

[104] J. R. Peters, V. Srivastava, G. S. Taylor, A. Surana, M. P. Eckstein, and F. Bullo, "Human supervisory control of robotic teams: integrating cognitive modeling with engineering design," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 57–80, 2015.

[105] A. Bestick, R. Pandya, R. Bajcsy, and A. D. Dragan, "Learning human ergonomic preferences for handovers," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–9.

[106] A. M. Bestick, S. A. Burden, G. Willits, N. Naikal, S. S. Sastry, and R. Bajcsy, "Personalized kinematics for human-robot collaborative manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1037–1044.

[107] Y. Wang and F. Zhang, *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*. Springer, 2017.

[108] J. Yamauchi, M. W. S. Atman, T. Hatanaka, N. Chopra, and M. Fujita, "Passivity-based control of human-robotic networks with inter-robot communication delays and experimental verification," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2017, pp. 628–633.

[109] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 688–694.

[110] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[111] K. Lange, "A gradient algorithm locally equivalent to the E-M algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 425–437, 1995.

[112] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhäuser: Boston, 2003.

[113] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba, "Where are they looking?" in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[114] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward safe human robot collaboration by using multiple kinects based real-time human tracking," *ASME Journal of Computing and Information Science in Engineering*, vol. 14, no. 1, pp. 011 006–9, 2014.

[115] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer design for stochastic nonlinear systems via contraction-based incremental stability," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 700–714, 2015.

[116] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[117] W. E. Dixon, "Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics," *IEEE Trans. Automat. Control*, vol. 52, pp. 488–493, 2007.

[118] Y.-C. Liu and N. Chopra, "Controlled synchronization of heterogeneous robotic manipulators in the task space," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 268–275, 2012.

[119] A. B. Farjadian, B. Thomsen, A. M. Annaswamy, and D. D. Woods, "Resilient flight control: An architecture for human supervision of automation," *IEEE Transactions on Control Systems Technology*, 2020.

Sidebar: Summary

This article provides a perspective on problems involved in cyber-physical human systems (CPHS) that work at the intersection of cyber-physical systems (CPS), e.g., an autonomous robot and human systems, and discusses solutions to some of the problems involved in CPHS. One example of a CPHS is a close-proximity Human-Robot Collaboration (HRC) in a manufacturing setting. The issues of efficiency of the joint operation, and human factors such as safety of the human workers, attention, mental state and comfort, naturally arise in the HRC context. By considering human factors, robots' actions can be controlled to achieve the objectives of safe operations, human comfort, etc. On the other hand, some questions arise considering robot factors (RFs), e.g., can we provide direct inputs to humans or provide information to the humans about the environment or other robots in the environment such that the objectives of safety, efficiency, comfort can be satisfied by taking into consideration the current capabilities of the robots. The article discusses some specific problems involved in HRC, which controls the robot's motion, a CPS, by taking the current actions of the human worker in the loop with the robot system. To this end, two main challenges are discussed, (1) intention inference of the human actions by analyzing the motion of the human observed through skeletal tracking and human gaze data, (2) based on the inferred human intention, a robot controller is developed that keeps the robot motion constrained to a boundary in 3-dimensional (3D) space using control barrier functions. The utility of the presented methods is demonstrated through a manufacturing task, where a human and a robot work collaboratively to move a pallet load. The estimation method fuses skeleton joint tracking data obtained using Microsoft Kinect sensor and human gaze data obtained using RGB images to estimate the direction of human's hand reaching motion while performing a joint pick and place task. The hand motion dynamics are learned using skeletal data. The trajectory of the human hand estimated forward in time based on the gaze and hand motion data at the current time instance. The robot controller is then used to control the two-arm Baxter robot such that the robot does not enter a 3D region where the human is likely to be present, making the robot operation safer for the humans. In addition, the trajectory forecast data is also used to generate robot trajectories to carry out a collaborative moving of an object using the human and the robot. An adaptive controller is then used to track the reference trajectories using the Baxter robot in Gazebo simulation environment.

Sidebar: Expectation-Maximization Algorithm

The E-M algorithm is summarized in this sidebar [S1]. Consider a log-likelihood function

$$l(\theta) = \log p(\, Z_T |\, \theta), \tag{S1}$$

where $Z_T = \{z_1, z_2, ..., z_T\}$ denotes the given measurements. Let $X_T = \{x_1, x_2, ..., x_T\}$ be a collection of states. The likelihood function in (S1) can be written as the following integral

$$l(\theta) = \log \int p(\, X_T, Z_T |\, \theta) dX_T, \tag{S2}$$

based on the joint probability density function $p(\, X_T, Z_T |\, \theta)$. The evaluation of the integral in (S2) is difficult and hence the approximation are required. To avoid integration of (S2), an E-M algorithm can be utilized. In the E-M algorithm two main steps are involved, E-step and M-step. The algorithm is summarized below.

1) Choose an initial estimation $\hat{\theta}_0 \in \mathcal{S}$, where $\mathcal{S}$ is a constrained parameter set.
2) **E-step:** Compute an auxiliary function $Q(\theta, \hat{\theta}_i)$, which is the expected value of the complete data log-likelihood with respect to the random variable X_N given the observed data Y_N and the estimate $\hat{\theta}_i$. The $Q(\theta, \hat{\theta}_i)$ is given by

$$Q(\theta, \hat{\theta}_i) = \mathrm{E}_{X_N}\{\log[p(\, X_T, Z_T |\, \theta)|\, Y_T, \hat{\theta}_i]\} \tag{S3}$$

3) **M-step:** Set $\hat{\theta}_{i+1} = \underset{\theta \in \mathcal{S}}{\mathrm{argmax}} \mathrm{Q}(\theta, \hat{\theta}_\mathrm{i})$.
4) Repeat steps 2 and 3 until convergence.

The basic idea of EM is to decompose the log-likelihood function

$$\begin{aligned} \log[p(\, Z_T |\, \theta)] &= \mathrm{E}_{X_N}\{\log[p(\, X_T, Z_T |\, \theta)|\, Y_T, \theta_i]\} - \mathrm{E}_{X_N}\{\log[p(\, X_T |\, Z_T, \theta)|\, Y_T, \theta_i]\} \\ &= Q(\theta, \theta_i) - H(\theta, \theta_i). \end{aligned} \tag{S4}$$

By using Jensen inequality, $H(\theta, \theta_i) \leq H(\theta_i, \theta_i)$. Hence, if we choose θ such that $Q(\theta, \theta_i) \geq Q(\theta_i, \theta_i)$ then from (S4), the log-likelihood function $l(\theta) = \log[p(\, Z_T |\, \theta)]$ can be maximized.

References

[S1] G. C. Goodwin and J. Aguero, "Approximate E-M algorithms for parameter and state estimation in nonlinear stochastic models," in *IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 368–373.

Sidebar: Constructing Control Barrier and Control Lyapunov Functions

Barrier Function

Consider a continuous nonlinear DS of the form

$$\dot{x}(t) = f(x(t)), \tag{S5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz continuous nonlinear function and $x(t) \in \mathcal{S} \subseteq \mathbb{R}^n$ is the state of the system. A set $\mathcal{S} \in \mathbb{R}^n$ is called *(forward) invariant* with respect to (S5) if for any initial condition $x(0) := x(t_0) \in \mathcal{S}$ implies that $x(t) \in \mathcal{S}$, $\forall t \in \mathbb{R}^+$ [S1]. BFs define a forward invariant safe region, where the solutions of a DS in this region remain in the region for all time [S2] and [S3].

Constructing the Barrier Function

Given a closed set $\mathcal{S} \subset \mathbb{R}^n$, its interior and its boundary are defined as follows

$$\mathcal{S} = \{x \in \mathbb{R}^n : \ b(x) \geq 0\}, \tag{S6}$$

$$\partial\mathcal{S} = \{x \in \mathbb{R}^n : \ b(x) = 0\}, \tag{S7}$$

$$Int(\mathcal{S}) = \{x \in \mathbb{R}^n : \ b(x) > 0\}, \tag{S8}$$

where $b(x(t)) : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. An example of $b(x(t))$ is given in (24).

Definition 1: ([S3, Definition 1]) Given the continuous system (S5), the closed set $\mathcal{S}$ defined by (S6)-(S8), and continuously differentiable function $b : \mathbb{R}^n \to \mathbb{R}$, a real-valued function $B : Int(\mathcal{S}) \to \mathbb{R}$ that is differentiable with respect to its argument is said to be a reciprocal BF, if there exist class $\mathcal{K}$ functions η_1, η_2, η_3 such that for all $x \in Int(\mathcal{S})$

$$\frac{1}{\eta_1(b(x))} \leq B(x) \leq \frac{1}{\eta_2(b(x))} \tag{S9}$$

$$\frac{\partial B(x)}{\partial x} f(x) \leq \eta_3(b(x)) \tag{S10}$$

There are two popular candidates to construct the BFs $B(x(t))$ from an invariant set, namely, Reciprocal BFs and Zeroing BFs. Some candidate Reciprocal BFs are *inverse-type* and *logarithmic-type* BFs given by $B(x(t)) = \frac{1}{b(x(t))}$ and $B(x(t)) = -\log \frac{b(x(t))}{1+b(x(t)}$, respectively [S3]. Note that both candidates are unbounded on the set boundary, $B(x(t)) \to \infty$ as $x(t) \to \partial\mathcal{S}$.

Conventionally safety verification in the form of trajectory invariance with respect to a given closed set relies on the existence of a BF satisfying conditions on both the function itself

and its time derivative along solution trajectories of the DS, namely $\dot{B}(x) \leq 0$ [S2]. However, as it is mentioned in [S4], the existence of a BF is just a sufficient condition to guarantee the safety property to be verified. In other words, the invariance verification of all sublevels of the closed set $\mathcal{S}$ may not be required. Therefore, authors in [S3] relaxed the condition to $\dot{B} \leq \frac{\gamma}{B}$ for Reciprocal BFs, where γ is a positive constant. The modification enables only a single sublevel set to be invariant.

Control Barrier Functions

BFs are essential means to verify invariance of a set but they cannot be used in its direct form to design a controller [S3]. In other words, to make sure that the set $Int(\mathcal{S})$ is forward invariant under the dynamics of the system (S5), a controller that guarantees the invariance of the set is required. Similar on how Lyapunov functions are extended to Control Lyapunov functions [S5], the concept of BFs can be extended to the case of control systems through the use of CBFs. Given the DS in (S5), in cases where the solutions of a DS do not stay in an invariant set $\mathcal{S}$, a CBF can be specified that will assure the solutions to remain inside the invariant set.

Constructing the Control Barrier Function

Suppose that we have the following DS with an external control input

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{S11}$$

where f and g are locally Lipschitz, $x(t) \in \mathbb{R}^n$, and $u(t) \in \mathbb{R}^m$. To find a suitable CBF such that the solution trajectories of the DS in (S11) avoid possible obstacles within its state space, the constraint on the system state $x(t)$ is encoded in a smooth constraint function $b(x(t))$. A value $b(x(t)) \geq 0$ indicates adherence, whereas $b(x(t)) < 0$ indicates a violation. This means that the constraint function has to be designed depending on the shape of the obstacle such that $b(x(t)) = 0$ defines its boundary. The set of admissible state $\mathcal{X}_0$ is defined by

$$\begin{aligned} \mathcal{X}_0 &= \{x \in \mathbb{R}^n : \ b(x) \geq 0\}, \\ \partial\mathcal{X}_0 &= \{x \in \mathbb{R}^n : \ b(x) = 0\}, \\ Int(\mathcal{X}_0) &= \{x \in \mathbb{R}^n : \ b(x) > 0\}. \end{aligned} \tag{S12}$$

A Reciprocal CBF $B : \ Int(\mathcal{X}_0) \to \mathbb{R}$ is a non-negative function, if there exist class $\mathcal{K}$ functions α_1, α_2, and α_3 such that for all $x(t) \in Int(\mathcal{X}_0)$,

$$\frac{1}{\alpha_1(b(x))} \leq B(x) \leq \frac{1}{\alpha_2(b(x))}, \tag{S13}$$

$$\inf_{u \in \mathbb{R}^m} \left\{ \mathcal{L}_f B(x) + \mathcal{L}_g B(x) u - \alpha_3 \left(\frac{1}{B(x)} \right) \right\} \leq 0, \tag{S14}$$

where $\mathcal{L}_f B(x(t))$ is the Lie-Derivative $\frac{\partial B(x(t))}{\partial x} f(x(t))$ along the vector field $f(x(t))$ and $\mathcal{L}_g B(x(t))$ is the Lie-Derivative $\frac{\partial B(x(t))}{\partial x} g(x(t))$ along the vector field $g(x(t))$. Hence for the system in (S11), any locally Lipschitz controller $u : \mathcal{X}_0 \to \mathbb{R}^m$ that is selected form (S14) assures the closed-set $\mathcal{X}_0 \subset \mathbb{R}^n$ is forward invariant.

Control Lyapunov Function

A continuous nonlinear DS of the form (S11) is said the be globally asymptotically stable at equilibrium x^* if there exist a real-valued continuously differentiable function $V(x(t)) : \mathbb{R}^n \to \mathbb{R}$ also known as control Lyapunov function (CLF) that is differentiable with respect to its argument and there exists a set of controls $\bar{U} \subset \mathbb{R}^m$ and class $\mathcal{K}$ functions α_{1v},α_{2v}, and α_{3v} such that for all $x(t) \in Int(\mathcal{X}_0)$ the following conditions are satisfied

$$\alpha_{1v}(|x|) \leq V(x) \leq \alpha_{2v}(|x|), \tag{S15}$$

$$\inf_{u \in \bar{U}} \{\mathcal{L}_f V(x) + \mathcal{L}_g V(x)\bar{u} + \alpha_{3v}(|x|)\} \leq 0, \tag{S16}$$

where $\mathcal{L}_f V(x)$ and $\mathcal{L}_g V(x)$ are the Lie derivatives.

References

[S1] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[S2] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*. Heidelberg: Springer-Verlag, 2004.

[S3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[S4] L. Dai, T. Gan, B. Xia, and N. Zhan, "Barrier certificates revisited," *Journal of Symbolic Computation*, vol. 80, pp. 62–86, 2017.

[S5] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.

Sidebar: Extreme Learning Machine

ELM is a learning algorithm that uses least squares approach to estimate the parameters of the single layer neural network (SLNN). According to the ELM theory [S1], from the function approximation point of view, input weights and all the hidden node parameters are randomly assigned. ELM theory claims, unlike the conventional learning methods, that parameter tuning of input weights, slopes, and biases is not required in learning. Therefore, the choice of ELM over traditional gradient-based algorithm for SLNN is because of the learning speed of ELM is faster and the gradient-based methods may have issues such as local minima and improper learning rate etc. [S1]. Consider the following multilayer perceptron with n_h hidden nodes

$$y(t) = \sum_{i}^{n_h} G_i(x(t), U_i, a_i, b_i) \cdot W_i^T, \tag{S17}$$

where $y(t) \in \mathbb{R}^n$, $G_i(\cdot)$ is a scalar, which denotes the i^{th} hidden node activation function whose range depends on the choice of the nonlinear function, $U_i \in \mathbb{R}^n$ is the input weight vector connecting the input layer to the i^{th} hidden neuron, $a_i, b_i \in \mathbb{R}$ are slope and bias corresponding to the i^{th} hidden neuron, and $W_i^T \in \mathbb{R}^n$ is the output weight vector connecting the output layer to the i^{th} hidden neuron. The ELM is created by randomly initializing the input weight matrix $U \in \mathbb{R}^{n \times n_h}$, the slopes (usually are set to one) $a \in \mathbb{R}^{n_h}$, and the biases $b \in \mathbb{R}^{n_h}$. The sigmoid function is typically chosen for the activation function. Although the ELM algorithm is fast, random initialization may lead to having saturated or constant neurons, which are not desired while learning the model [S2]. To circumvent this problem, an intrinsic plasticity (IP) learning rule can be used. IP is an online learning method that optimizes the information transmission of a single neuron by adapting the slopes and biases such that the output of the activation function becomes exponentially distributed. Inspired by IP method, a computationally efficient batch version of the method, called BIP, was introduced in [S2].

References

[S1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[S2] K. Neumann and J. J. Steil, "Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity," *Neurocomputing*, vol. 102, pp. 23–30, 2013.

Author Biography

Ashwin Dani (SM' 18, M'11) received the M.S. and Ph.D. degrees from the University of Florida, Gainesville, FL. He was a Post-Doctoral Research Associate at the University of Illinois, Urbana-Champaign, IL. He is currently an Associate Professor at the University of Connecticut, Storrs, CT. He has authored over 50 technical papers and 4 book chapters. His current research interests include nonlinear estimation and control, machine learning for control, human–robot collaboration, and autonomous navigation.

Iman Salehi received his Master's degree from the Department of Electrical and Computer Engineering at the University of Hartford in 2015. He is currently working toward his Ph.D. in the Electrical and Computer Engineering department at the University of Connecticut, Storrs, CT. His research interests include learning for control, human-robot interaction, and system identification.

Ghananeel Rotithor received the M.S. degree in Biomedical engineering from the University of Florida, in 2017. He is currently pursuing the Ph.D. degree from the Electrical and Computer Engineering department, University of Connecticut, CT. His research interests include deep learning for control, computer vision, robotics, vision based estimation and control.

Daniel Trombetta received his B.S. in Electrical and Computer Engineering from the University of Connecticut in 2018. He is currently pursuing a Ph.D. from the Electrical and Computer Engineering department at the University of Connecticut. His research interests include machine learning, human-robot collaboration, and estimator design.

Harish Ravichandar (M' 18) is a Research Scientist in the School of Interactive Computing at Georgia Institute of Technology. He received his M.S. degree in Electrical and Computer Engineering from the University of Florida, Gainesville, FL, in 2014 and his Ph.D. in Electrical and Computer Engineering from the University of Connecticut, Storrs, CT, in 2018. His current research interests span the areas of robot learning, human-robot interaction, and multi-agent systems.