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

he prospect of a collaborative work environment between humans and robotic automation in a manufacturing setting [1] provides the motivation for finding innovative solutions to human-inthe-loop control for safe, efficient, and trustworthy human–robot collaboration (HRC), or HR interaction

Digital Object Identifier 10.1109/MCS.2020.3019725 Date of current version: 13 November 2020 (HRI), in cyberphysical human systems (CPHSs) [2], [3]. Studies in [4] show that collaborative automation can be beneficial to 90% of approximately 300,000 small-to-medium-scale enterprises in the United States. In the paradigm of human-centered automation [5], human safety, ergonomics, and the collaborative efficiency of the work are given the utmost importance. Traditional methods to ensure the safety of humans around factory robots involve the use of cages. Recent work looked beyond cage-based

1066-033X/20©2020IEEE

DECEMBER 2020 « IEEE CONTROL SYSTEMS 29

safety to provide robot control and sensing-driven solutions for human safety around robots [6]-[8].

The purpose of this article (see "Summary") is to provide a tutorial on human-in-the-loop estimation and control methods to achieve worker safety in the context of close-proximity HRC. Examples of current close-proximity interaction include collaborative assembly, the cooperative carrying of loads, and robot assistants working near humans in manufacturing plants in industries such as automotive, aerospace, and electronics.

The article is broadly divided into two main sections. The first describes methods for human-action intention inference based on sensor data, and the second describes safe robot-control policy generation based on an inferred human-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). An overview of intention estimation is first provided by presenting existing literature in a tutorial manner, followed by the approaches that focus and expand on prior work. Modifications to the algorithms are mentioned at the appropriate locations. New experiments 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 trajectories is discussed that uses control barrier functions (CBFs) in the HRC context. Experiments are conducted using a Kinect sensor and the Baxter research robot platform, where a human's trajectory forecast is employed to generate safe courses for robots to follow.

A block diagram of human-in-the-loop control with human-intention estimation and safe trajectory tracking of a

Summary

his article provides a perspective on estimation and con-trol problems in cuborphysical in trol problems in cyberphysical human systems (CPHSs) that work at the intersection of cyberphysical systems and human systems. The article also discusses solutions to some of the problems in CPHSs. One example of a CPHS is a closeproximity human-robot collaboration (HRC) in a manufacturing setting. The issue of the joint operation's efficiency and human factors, such as safety, attention, mental states, and comfort, naturally arise in the HRC context. By considering human factors, robots' actions can be controlled to achieve objectives, including safe operations and human comfort. Alternately, questions arise when robot factors are considered. For example, can we provide direct inputs and information to humans about an environment and the robots in the area such that the objectives of safety, efficiency, and comfort can be satisfied by considering the robots' current capabilities?

The article discusses specific problems involved in HRC related to controlling a robot's motion by taking the current actions

robot for collaborative tasks is shown in Figure 1. Use cases of such collaborative tasks include the following:

- 1) robot-assisted assembly, where different components require various installation methods (distinct models); a robot observes a human to determine the current model and hence the robot's own desired actions
- 2) wire harness assembly, in which a human and a robot complete certain steps in the fabrication process; for instance, for a wire stripping task, a human may reach for a tool, indicating to a robot that it must grasp wire and take it to the human
- 3) assisted construction/surgery, where a robot is responsible for acquiring the correct tool for the task the human is currently trying to perform
- 4) collaboratively moving loads (an experiment is performed for this use case)
- 5) the repair of vehicles in uninhabitable environments (such as underwater and in space), where a robot may assist a human who has limited ability given the conditions, and the robot has to interpret the human's intention from motion profiles.

INTENTION ESTIMATION

Providing sensing-based human-action intention inference solutions is one of the important steps toward achieving safety during HRC in manufacturing automation, automotive applications (self-driving cars) [10], space robotics [11], and assistive robotics [12]-[16]. Studies in psychology suggest that when two humans interact, those people infer one another's intended actions for safe interaction and collaboration [17], [18]. An optimal control model of human response and its applications is examined in [19] and [20]. Based on these

of the human in the loop with the robot's control system. To this end, two main challenges are discussed: 1) inferring the intention behind human actions by analyzing a person's motion as observed through skeletal tracking and gaze data and 2) a controller design that keeps robot motion constrained to a boundary in a 3D space by using control barrier functions. The intention inference method fuses skeleton-joint tracking data obtained using the Microsoft Kinect sensor and human gaze data gathered from red-green-blue Kinect images. The direction of a human's handreaching motion and a goal-reaching point is estimated while performing a joint pick-and-place task. The trajectory of the hand is estimated forward in time based on the gaze and hand motion data at the current time instance. A barrier function method is applied to generate safe robot trajectories along with forecast hand movements to complete the collaborative displacement of an object by a person and a robot. An adaptive controller is then used to track the reference trajectories using the Baxter robot, which is tested in a Gazebo simulation environment.

studies about the principles of human interactions, the safety, operational efficiency, and task reliability in HRC could be greatly improved if robots were provided the capability to infer human-action intentions. For instance, in [12] and [21]–[23], inferring a human partner's intention is shown to improve the overall performance of tasks requiring HRC.

Human intention is inferred via various sensing modalities by measuring different cues captured by sensors that are on a person's body (wearable sensors) or mounted elsewhere (nonwearable sensors). Using wearable devices, physiological information measured through heart rate, skin response [24], [25], and electromyography sensors [26] is typically used for gauging human motion intention. In [24], human intention is represented using valence/arousal characteristics that are measured by physiological signals, such as heart rate and skin response. The valence/arousal representation of human intention indicates only the degree of approval to a given stimulus. Using nonwearable sensors, cues such as human emotion [27], approval responses [24], body posture [28], gestures [29], eye gaze [30], [31], facial expressions [32], and skeletal movement [33], [34] are measured. Imaging sensors, such as red-green-blue (RGB) cameras, are also commonly used. In [35], a human's intention to hand over an object is predicted by key features extracted from an RGB camera sensor.

For estimating human intention, the dynamics of human motion are modeled using probabilistic representations, where hidden Markov models (HMMs) [24], [36], [37], dynamic Bayesian networks [38], [39], growing HMMs [40], and conditional random fields (CRFs) [41]–[43] are employed. In [44], human activities are inferred by an HMM of people's actions and interactions with autonomous mobile robots. In [45], a human intent estimation algorithm based on a fuzzy inference engine is presented. The intentionestimation problem is formulated only as a relationship between attention and physiological measurements, and online inference of human intention is not performed.

In [24], an online algorithm to estimate the affective state of a person is developed based on valence/arousal representation using an HMM. The intention in these methods is formulated as a classification problem. Since the future state prediction in methods based on HMMs and their variants is 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 through additional observations [46].

In another popular paradigm, human motion is represented using continuous/discrete dynamics that are parameterized using dynamic neural networks (NNs) in the case of deterministic modeling with noise and through Gaussian processes (GPs) and Gaussian mixture models (GMMs) in the case of probabilistic modeling. In [23], an NN is used to approximate the endpoint of a human hand for representing human motion and intent in physical HRC applications. In [33], a latent variable representation called the intention-driven dynamic model (IDDM) is proposed to infer intentions from observed human movements. Robot table tennis and human activity classification are demonstrated using a belief propagation algorithm coupled with the IDDM. In [47], a human-intention inference algorithm is developed using unsupervised GMMs, where the parameters of the GMMs are learned using the expectation-maximization (EM) 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 states. Thus, the certainty in the estimated intent tends to converge through time. These models are typically not good for detecting sudden changes in intention. Additionally, the models developed through the preceding methods are not all learned, considering the stability and convergence to the reaching point of human actions/motion.

In yet another paradigm, action plans are represented as policies in terms of state-action pairs. Inverse optimal control (IOC) and inverse reinforcement learning (RL) algorithms are used to model the intention-driven behavior, where the intended motion maximizes an unknown



FIGURE 1 A block diagram of human-in-the-loop control, with human-intention estimation and safe trajectory tracking for robots participating in collaborative tasks. The human dynamics block uses skeletal tracking via red-green-blue-depth sensors. Based on the skeletal tracking of one arm, the human trajectory estimator generates an approximation of a person's reaching action intentions. The robot-reference trajectory is produced using the estimated human trajectory so that the robot's movement is safe.

objective or reward function. An inverse linear quadratic regulator approach is developed in [48] to predict the intent and trajectory forecast of human motion. The applicability of human-intention estimation using inverse approaches can be found in various applications. For example, human motor-control intent estimation in rehabilitation applications is presented in [49] using an inverse model predictive control strategy. Predicting pilots' behavior by modeling their goals through RL and game theory is presented in [50]. In [51], human motion during collaborative manipulation is predicted via an IOC approach. The IOC-based techniques typically require exploration of the state space; thus, they require large amounts of data for converging to the correct solutions. A categorization of intention estimation algorithms is given in Table 1.

The *intention* is defined as a goal location of reaching motions and an 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 priori information). When the priors are uniform, the MAP estimator gives an ML approximation. Hence, ML estimation is a special case of MAP estimation.

In the *first technique*, the goal-reaching intention is modeled as a parameter of the continuous dynamics. An ML estimation technique called the approximate EM algo*rithm* is applied to estimate the goal-reaching intention. To accommodate changes in motion, an online 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. The algorithm is called the adaptive neural intention estimator (ANIE) [34]. In addition to the ANIE algorithm presented in [34], gaze information is included to provide a more accurate initialization to the EM algorithm. In the second technique, individual goal-reaching motions are represented as multiple dynamical system (DS) models. A MAP estimation of intention is obtained through an interacting multiple-model (IMM) framework that computes probabilities for each goal-reaching motion model based on observations. Changes/switches in intention can also be detected using the IMM framework. A good prior to the IMM algorithms can improve their performance. Hence, additional cues, such as the

TABLE 1 A categorization of the intention estimation algorithms based on intention estimation problem formulation.			
	Method	Sensor Type	Reference
Intention parameter estimation	Neural network (NN)	Red-green-blue depth (RGB-D)	Ravichandar et al. [34]
	Gaussian process (GP)	RGB	Wang et al. [33]
	Gaussian mixture model (GMM)	RGB-D	Mainprice and Berenson [52]
	NN	Force/torque	Li et al. [23]
Action recognition and intention classification	Interacting multiple model	RGB-D	Ravichandar et al. [53]
	Hidden Markov model (HMM)	Physiological	Kulic et al. [24]
	Anticipatory temporal conditional random field (CRF)	RGB-D	Koppula et al. [54]
	НММ	RGB + laser	Kelley et al. [44]
	НММ	Motion capture	Ding et al. [37]
	Hybrid dynamic Bayesian network (DBN)	RGB	Gehrig et al. [38]
	Hybrid DBN	Agnostic	Schrempf et al. [39]
	Growing HMM	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]
Inverse optimal control/inverse reinforcement learning (IRL)-based intention estimation	Path integral IRL	Motion capture	Mainprice et al. [51]
	Inverse linear quadratic regulator	RGB-D	Monfort et al. [48]
	Inverse model predictive control	Encoder	Ramadan et al. [49]
Miscellaneous	Fuzzy logic	Physiological	Kulic et al. [45]
	Fuzzy logic	Force/torque	Carli et al. [36]

32 IEEE CONTROL SYSTEMS >> DECEMBER 2020

direction of a person's gaze, computed by estimating the head orientation from RGB images are used. The algorithm is called the gaze-based multiple-model intention estimator (G-MMIE) [9].

In the first method, based on ML estimation, the complex dynamic motion of the human arm is represented by a nonlinear system model. The positions and velocities of the joints are used as the states. The uncertain system dynamics are modeled using a dynamic NN [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 issue of intention inference is solved as a parameter inference problem, given noisy motion data through an approximate EM algorithm [56]. The NN estimation can potentially enable the consideration of user- and 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 3D space in various ways based on their physical characteristics. This brings a challenge to using the model learned from the demonstration data to represent joint position trajectories of other subjects. One way of updating the representation in real time is to use the EM algorithm by optimizing the ${\bf Q}$ function across the model parameters, along with the intention. A closed-form expression for the model update using EM exists if the model is linear or represented using a radial basis function (RBF) NN [57]. However, arm-motion dynamics are highly nonlinear, and the RBF may not always be the best choice for the basis functions of the NN to represent human movement. To overcome this challenge, an identifier system-based algorithm presented in [58] is used for online model updates. The identifier system is designed using a robust feedback term: the robust integral of the signum of the error (RISE) [59]. Based on the Lyapunov analysis, the parameter update laws for the 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 of 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 novel environments. For instance, consider a wire harness fabrication 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 employed for a new user handling parts of a similar object, the NN approximation error is likely to be high, as the motion profiles will vary for different users assembling dissimilar parts. However, as new data become available, the presence of the feedback term enables the identifier system to implicitly learn the network weights and minimize the effects of NN approximation errors in real time [58]. An 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. The principles state that the generated motion is inherently closed-loop stable and smooth during 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 a 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 a person is represented as a point and his or her motion in the 2D/3D space is observed as a point motion reaching to varying locations (for instance, factory workers moving to different work stations).

Using the learned model, the intention inference is completed by the multiple-model intention estimator (MMIE) presented in [63]. The MMIE algorithm employs an 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 is very high, it is well known that the performance of the IMM filter degrades. A variable-structure IMM (VS-IMM) filter can be applied in such cases [65]. A limiting case of the VS-IMM 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 a 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 an 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 the labeled demonstrations of the joint position trajectories.

HUMAN-IN-THE-LOOP SAFETY CONTROL OF ROBOTS

When a robot is collaborating with a human in close proximity, one of the important problems is to make the robot aware of the human's movement intentions so that the robot can adapt its motion controller to operate safely and perform a collaborative task. In [21]–[23], inferring a human's intention is shown to improve the overall performance of collaborative tasks. In [67], human goal intention is applied in conjunction with an admittance controller to achieve HRC. Many contributions to HRI have targeted motion-intention estimation for tasks where direct physical interaction between humans and robots is involved. For control strategies, the literature focuses on designing impedance control laws [12], [23] and admittance control laws [67], [68] for adapting the interaction forces exerted by a human on a robot when the robot is physically interacting with the person. In [69], a controller is developed for HR handover interaction based on dynamic movement primitives. In HRC (where there may not be direct contact with a human), the human motion intention estimation, robot path planning, and control design become 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 and autonomous systems [46], [70]. Most of these studies view the problem as a collision avoidance issue and solve it using the potential field approach [71]. These control actions are purely reactionary in nature. Anticipatory skills are required to improve efficiency and safety during collaborative tasks when humans work in close proximity to robots [46], [70]. To achieve proactiveness, the controller and motion planner must incorporate probabilistic information about the possible 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 was proposed in [71]. In [23], a feedback controller for HRI is developed based on an NN model of human intention. In [52], a stochastic trajectory optimizer for motion planning is employed 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 coworker while providing strong guarantees for the synchronization and timing of activities. In [73], new hierarchical planners based on hierarchical goal networks are developed for assembly planning in the HR team. In [74], an empirical study of human–human interaction is conducted to investigate the ways in which human teammates perform in a coordinated behavior. This study is targeted toward scheduling and planning tasks rather than studying human behavior in the context of robot motion planning for safety and the proactive control of close proximity operations.

In control literature, stability studies of human-in-theloop adaptive controllers are presented using the innerouter loop control structure in [75]. Stability examinations of human-in-the-loop telerobotics with a time delay are presented in [76]. These works do not explicitly consider safety aspects of the human-in-the-loop systems. Providing safety guarantees in the learned controller of the machine/robot is typically achieved by adjusting the reference command via a prefilter called a reference governor [77]–[79] and by using optimal control under uncertainty in a differential game setting.

In [80], an 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 introduced in [82]. In [83], a receding horizon safe-path planning approach using mixed integer linear programming is presented. Safe trajectory generation for autonomous operation of spacecraft using convex optimization formulation is proposed in [84]. When the region is nonconvex, successive convexification can be performed [85]. A detailed survey of, and tutorial on, an \mathcal{L}_1 -adaptive control architecture for safety critical systems appears in [86].

Other methods of achieving the safety property of controller synthesis are to employ a BF/certificate or a CBF, which ensures that 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. Reciprocal BFs can be of the inverse type and the logarithmic type. Similar extensions to CBFs have also been developed in the literature. Applications of BFs and CBFs in many autonomous robotic systems (such as robot manipulators, autonomous vehicles, and walking robots) are shown in [88]-[91]. In [88], [90], and [92], BFs were successfully applied to DSs, where ensuring safety conditions is critical. In [92], time-varying BFs and CBFs for avoiding moving and static obstacles are derived, and their application to quadcopters 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 BF, then the solution of the model must cease to exist because the control input becomes unbounded.

For zeroing CBFs, input-to-state stability results hold 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 two in [95] to address the safety constraints for systems with a higher relative degree. Furthermore, a backstepping-based 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 presented 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 and 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 harnessed to generate a robot's desired trajectories, which are strictly bounded within a prescribed safety set. An adaptive controller that accepts the humanintention estimation in the loop for trajectory synchronization appears 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 to the robot for tracking via a feedback adaptive controller for the 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 is generated using human trajectory forecasts computed through 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 the safety of joint tasks. An experiment is completed that computes the intention of a person performing a collaborative task of moving an object with a 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-theloop robot control from the perspectives of both estimation and control. The major challenges include the development of fundamental theoretical guarantees and limits on longterm intention estimation, using predictive forecasting methodologies for learning models of humans and/or robots, and the fusion of multiple sensing modalities (such as vision, ultrasound, and lidar in the context of autonomous vehicles and/or physiological signals in the context of manufacturing applications). Incorporating human trust, workloads [98]–[102], attention allocation [103], and cognitive factors [104] in the HR collaboration brings many potential avenues for research. Including human ergonomic preferences in CPHSs also has great potential to make the HRI more comfortable for humans [105], [106].

Some recent trends in control and decision making in HR collaborative systems are found in [107]. For the ML-based intention estimation technique of applying the approximate EM 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. Integrating information provided by intention inference algorithms into safe control methodologies that use CBF formulation and address the uncertainty in estimation is another important area that requires further investigation. In addition to safety confirmation, there are temporal properties that need verification in the HRC setting, such as eventuality, avoidance, reachability (a problem related to the safety property), and the composition of these properties. Developing methods for corroborating these properties in the HRC context is also important.

Compensating for time delays during the communication of information between humans and robots brings many challenges, specifically in the context of networks of human and robot agents [108]. HRC in a virtual reality setting also presents many obstacles from an estimation and control perspective. For example, in a rehabilitation application, a human who is undergoing walking or biking rehabilitation can benefit from interactions with a virtual environment. The interaction with the virtual setting can provide important feedback to the human user via the robot he or she is interacting with.

The remainder of the article is organized as follows. First, a human intent estimation method based on ML estimation using an approximate EM algorithm is presented. Then, another intention approximation method based on a MAP estimator that employs an IMM filter (with priors computed using the human gaze map) is described. A human-in-the-loop control strategy that uses CBFs to compute a safe reference trajectory for a robot to follow is designed, and an adaptive robot control is developed for a robot manipulator to track a safe desired trajectory during a collaborative task. 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 Machine Learning Estimation Problem

In this section, the human goal-reaching intention is modeled as a parameter of the nonlinear dynamics. The motion of a human is modeled as a nonlinear differential equation approximated as an NN. An approximate EM algorithm is designed to estimate the reaching intention of the human's action (see Figure 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 or a shelf or walking toward certain locations in the area. The human subject reaches out to different objects placed on a table, and a robot watches the person through a 3D camera sensor. The problem of inferring the human hand's reaching goal location is addressed. Since the human motion is highly nonlinear and uncertain, an NN approximation of the nonlinear function of the dynamic system is used to model the movement. The NN is trained using a data set containing skeletal tracking of a human reaching for predefined target locations in a given workspace, observed using an RGB-D camera. When a set of new measurements becomes available, the trained NN is employed to estimate the reaching goal intention (the goal location in 3D) via an approximate EM algorithm. An online NN model weight learning method is also developed through an identifier-based algorithm to adapt to the variations of motions in different human subjects.

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 $z(t) \in \mathbb{R}^{12}$ denotes 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:

$$\dot{x}(t) = f_c^*(x(t), g) + \omega(t),$$
 (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 $Q_c \in \mathbb{R}^{24 \times 24}$. Let \mathbb{S} be a compact, simply connected set of $\mathbb{R}^{24} \times \mathbb{R}^3$. Here, $f_c^*(x(t), g)$: $\mathbb{S} \to \mathbb{R}^{24}$ 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

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

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, $||W||_F \leq \bar{\sigma}_w$ and $||U||_F \leq \bar{\sigma}_u$;

 $\sigma(U^T s(t)) \in \mathbb{R}^{n_h+1}$ is an activation function that can be represented by a vector sigmoid function, an RBF, or a rectilinear unit; $(U^T s(t))_i$ is the *i*th element of the vector $(U^T s(t))$; $\epsilon(s(t)) \in \mathbb{R}^{24}$ 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 joint positions are gathered by using an 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

$$p^c = R^c_r p^r + T^c_r, \tag{3}$$

where $R_r^c \in SO(3)$ 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 reference frame. The camera sensor measures the 3D locations of the skeleton's joints.

The measurement model is given by

$$y(t) = h(x(t)) + v(t),$$
 (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, $b = [[T_r^c]^T, [T_r^c]^T, [T_r^c]^T, [T_r^c]^T]^T$, $H = [blkdiag\{R_r^c, R_r^c, R_r^c, R_r^c\}, 0_{12 \times 12}] \in \mathbb{R}^{12 \times 24}$, and $v(t) \sim \mathcal{N}(0, \Sigma_z) \in \mathbb{R}^{12}$ is zero-mean Gaussian noise with a covariance matrix $\Sigma_z \in \mathbb{R}^{12 \times 12}$. The measurement noise v(t) is assumed to be independent of



FIGURE 2 A block diagram of the approximate expectation-maximization (EM)-based intention estimation algorithm with gaze-based approximates that are used for initializing the M step of the EM. The gaze estimator block uses red-green-blue (RGB) images along with skeletal data to estimate the gaze map of a person, which determines the most probable objects that the person is looking at. The gaze map is applied to provide an initial estimate of the parameter for the EM algorithm. CNN: convolutional neural network; EKF: extended Kalman filter; ANN: artificial neural network.

36 IEEE CONTROL SYSTEMS >> DECEMBER 2020

the process noise $\omega(t)$ defined in (1). The measurement model of the shifted measurement vector z(t) = y(t) - b is given by

$$z(t) = Hx(t) + v(t).$$
 (5)

For the RGB-D camera, the Gaussian assumption of the measurement noise v(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 $\sigma_{u_{c}}^{2}(t)$, $\sigma_{v_{c}}^{2}(t)$; and depth variance $\sigma_{z_{c}}^{2}(t)$.

Neural Network Model Training

The training of the NN is completed 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 through Bayesian regularization. The objective function applied 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 the model complexity, respectively.

Approximate Expectation-Maximization Algorithm for Estimating the Intention

An approximate EM algorithm is presented to estimate the intention parameter [56] using the state transition model learned through the 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. To infer the intention, the posterior probability of Z_T given the intention g is maximized using an 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 $p(X_T, Z_T|g)$ across X_T . In general, analytically evaluating this integral is very difficult. In this article, an approximate EM algorithm is presented that uses state transition models trained using the NN. Given 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, to iteratively increase the log likelihood, g must 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

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

If 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 $\mathbf{Q}(g, \hat{g}(t))$ across *g*, as described by

$$\hat{g}(t+1) = \operatorname{argmax} \mathbf{Q}(g, \hat{g}(t)).$$
(10)

This step can be completed in two different ways, namely, numerical optimization and direct evaluation. One way to maximize the **Q** function is to use the gradient EM (GradEM) algorithm for the M step, which, in turn, uses the first iteration of Newton's method [56]. Since Newton's method often converges quickly, the local convergence properties of the GradEM algorithm are identical to the EM algorithm. More details of the convergence properties of the GradEM algorithm can be found in [111]. This method involves optimizing the **Q** function across \mathbb{R}^3 . The update equation for \hat{g} , through the GradEM algorithm, is given by

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

where $\hat{g}(t)$ is the estimate of g at the current time t of the optimization algorithm and $\mathcal{H}(\mathbf{Q})$ and $\Delta(\mathbf{Q})$ are the Hessian and the gradient of the \mathbf{Q} function, respectively. Note

that numerical optimization methods need to run at every time step of the EM algorithm. For 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 the expression for the gradient of the \mathbf{Q} function can be found in [34]. General details of the EM algorithm are described in "Expectation-Maximization Algorithm."

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

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

This method involving direct evaluation of the **Q** function is feasible if all possible goal locations are known a priori and finite. This is not an unusual case in the context of humanintention estimation in practical applications, such as manufacturing assembly, space robotics, and assisted construction. Image processing algorithms, such as the region convolutional NN (R-CNN), faster R-CNN, mask R-CNN, and You Only Look Once, can be used to detect objects in the workspace and extract the 3D locations using camera data.

Online Model Weight Update

In this section, an online learning algorithm is described and 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

Expectation-Maximization Algorithm

he expectation-maximization (EM) algorithm is summarized here [S1]. Consider a log-likelihood function

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

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

$$I(\theta) = \log \int p(X_T, Z_T | \theta) dX_T, \qquad (S2)$$

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

- Choose an initial estimation θ̂₀ ∈ S, where S is a constrained parameter set.
- For the E step, compute an auxiliary function Q(θ, θ̂_i), which is the expected value of the complete data log like-lihood with respect to the random variable X_N, given the

people. The NN weights are iteratively updated 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 extended Kalman filter (EKF) and the current NN weights. The identifier state error is computed from 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 EM algorithm, which involves optimizing the Q function that may have a slower convergence rate. Hence, the state identifier is presented in the continuous form. The identifier uses RISE feedback [59] to ensure the 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

$$\dot{\hat{x}}_{id} = \hat{W}^{T}(t) \,\sigma(\hat{U}^{T}(t)\hat{s}(t)) + \mu(t),$$
(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 EM 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]. Here, $\nu(t) \in \mathbb{R}^{24}$ is the Filippov generalized solution [58] to the differential equation

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

observed data Y_N and the estimate $\hat{\theta}_i$. The $Q(\theta, \hat{\theta}_i)$ is given by

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

- 3) For the M step, set $\hat{\theta}_{i+1} = \operatorname{argmax} Q(\theta, \hat{\theta}_i)$.
- 4) Repeat steps 2 and 3 until convergence.

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

$$\log[p(Z_{\tau}|\theta)] = \mathsf{E}_{X_N} \{ \log[p(X_{\tau}, Z_{\tau}|\theta)|Y_{\tau}, \theta_i] \} - \mathsf{E}_{X_N} \{ \log[p(X_{\tau}|Z_{\tau}, \theta)|Y_{\tau}, \theta_i] \} = Q(\theta, \theta_i) - H(\theta, \theta_i).$$
(S4)

Using the Jensen inequality, $H(\theta, \theta_i) \leq H(\theta_i, \theta_i)$. Hence, θ is chosen such that $Q(\theta, \theta_i) \geq Q(\theta_i, \theta_i)$. Then, from (S4), the log-likelihood function $I(\theta) = \log[p(Z_T | \theta)]$ can be maximized.

REFERENCE

[S1] G. C. Goodwin and J. Aguero, "Approximate EM algorithms for parameter and state estimation in nonlinear stochastic models," in *Proc. 44th IEEE Conf. Decision Control*, 2005, pp. 368–373. doi: 10.1109/CDC.2005.1582183.

where *k*, α , γ , and $\beta_1 \in \mathbb{R}^+$ are positive constant control gains and sgn(·) denotes a vector signum function.

The weight update equations are given by

$$\begin{split} \hat{W}(t) &= \operatorname{proj}\left(\Gamma_{w} \hat{\sigma}'(t) \, \hat{U}_{x(t)}^{T} \dot{\hat{x}}_{id}(t), \, \tilde{x}^{T}(t)\right), \\ \dot{\hat{U}}_{x(t)} &= \operatorname{proj}\left(\Gamma_{u_{x}} \dot{\hat{x}}_{id}(t) \, \tilde{x}^{T}(t) \, \hat{W}^{T}(t) \, \hat{\sigma}'(t)\right), \\ \dot{\hat{U}}_{g(t)} &= \operatorname{proj}\left(\Gamma_{u_{g}} \dot{\hat{g}}(t) \, \tilde{x}^{T}(t) \, \hat{W}^{T}(t) \, \hat{\sigma}'(t)\right), \end{split}$$
(15)

where proj(·) is a projection operator defined in [112]; $\hat{U}_{x(t)}$ and $\hat{U}_{g(t)}$ are the submatrices of $\hat{U}(t)$ formed by taking the rows corresponding to $\hat{x}_{id}(t)$ $\hat{g}(t)$, 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 EM 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 $\dot{g}(t)$ is computed through the finite difference method. It can be shown using Lyapunov analysis that the identifier defined in (13) and the update equations defined in (15) are asymptotically stable and that the state identification error converges to zero.

Gaze Map Computation

This section briefly describes the CNN introduced in [113], which is applied to extract gaze information from an RGB image. To this end, a deep CNN architecture 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 is the gaze point.

Data

The data set used for training the CNN model, as described in [113], is created by concatenating images from six different sources: 1548 images from Scene Understanding; 33,790 images from Microsoft Common Objects in Context; 9135 images from Actions40; 7791 images from Pattern Analysis, Statistical Modeling, and Computational Learning; 508 images from the ImageNet detection challenge; and 198,097 images from the Places data set.

Implementation of Convolutional Neural Network

The five-layered CNN shown in Figure 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 and 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 a person. To compute the probability of g_j being the initial goal location for the optimization problem of the EM-based approach, the average probability $\tilde{p}_j(0)$ of the *j*th object in the scene is calculated as

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

where 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 GP_j is the set of all pixel locations associated with the *j*th object. Of all the objects in the scene, the one that corresponds to the highest average probability is chosen as the initial goal location for the optimization of the EM algorithm of the ANIE method.

Experimental Results for Expectation-Maximization-Based Intention Estimation

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

Neural Network Training

The starting positions of the human arm and the possible goal locations of the test trajectories are different. In the training phase, some of the trajectories involved reaching for objects that were randomly placed close to each other in a cluttered manner, and some of the recorded arm motions consisted of the subject initially moving his or her hand close to an object and finally reaching another item. Each trajectory contained roughly 40–60 frames of skeletal data. A set of eight trajectories is used for training an NN. The raw position measurements obtained from the RGB-D camera sensor are processed using a KF, 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.

Neural Network Testing

Once the NN is trained, the test data from a different subject are used as measurements to infer the underlying 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 **Q** function is evaluated for all the possible intentions to find the one that leads to the maximum *Q* value (the direct evaluation method). The initial mean of the state $\mu(0)$ is assumed to be a zero vector. The initial state covariance $P_{\text{EM}}(0)$, the process noise covariance *Q*, and the measurement noise covariance Σ_z are selected 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, $\alpha = 5$, $\gamma = 25$, and $\beta_1 = 4$, and the adaptation gains are chosen to be $\Gamma_W = 0.75I_{50\times50}$, $\Gamma_{U_x} = 0.75I_{24\times24}$, and $\Gamma_{U_g} = 0.75I_{3\times3}$. The state estimates are initialized to the same value as the first measurement z_1 . The sampling time for discretization is 1/30 s.

When the intention \hat{g} is modeled as a continuous variable, the GradEM algorithm is used for evaluating the intention estimate. In Figure 3, the convergence of the estimated goal location to the true goal location is shown. The numerical optimization of the **Q** function is completed for five iterations at every time step. The state and intention estimates are initialized using two methods, namely, random selection and gaze-based selection. In the first set of experiments, the goal location is arbitrarily chosen from the eight possible ones, and the convergence of the estimated intention to the true intention is shown in Figure 3(a). In the second set, the approximated intention is initialized using gaze cues, and the convergence of the forecast intention to the true one is shown in Figure 3(b). 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 fewer transients, compared to the random initialization. When the intention \hat{g} is modeled as a discrete variable, a direct evaluation of the **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 Figure 4, the intention estimate progression and 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 arbitrary assignment, the initial approximation 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 EM 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 optimizing the EM is tested in the second experiment, which provides better cues of the reaching goal location of the person. A dense gaze probability map is obtained using the method described in [53] and [113], and a probability is assigned to each of the possible goal locations by using the formula in (16). The goal location with the highest probability is chosen as the initial estimate $\hat{g}(0)$.

In Figure 5, the dense gaze probability map is presented 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 Figure 6, the intention estimate progression is shown, along with the trajectory evolution for the initial



FIGURE 3 The estimated goal location using the adaptive neural intention estimator algorithm by numerically optimizing the \mathbf{Q} function. (a) The convergence plot of the estimated intention (X, Y, and Z locations) when the initial estimate is randomly chosen. (b) The convergence plot of the estimated intention (X, Y, and Z locations) when the initial estimate for expectation-maximization optimization is chosen using gaze cues.

40 IEEE CONTROL SYSTEMS >> DECEMBER 2020

intention estimate computed from the gaze map. For the case of the sequence in Figure 6, the goal location with the maximum gaze probability is the true goal location. As a result, the algorithm predicts the intention correctly throughout the sequence. In the case of the random initialization of the intention, the ML estimation is equivalent to the MAP approximation with a uniform prior.

The observations in previous experiments that the ML method gives a faster convergence when initialized with gaze cues compared to the random initialization could be a

coincidence. If a 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 EM algorithm. If the initialization of the goal location is bad, the EM algorithm may require more measurements before it correctly converges to the true goal location because EM is a ML estimation method. As demonstrated in Figure 3, the EM algorithm requires observations of up to 0.6 s to correctly predict the goal location for a 1.5-s sequence.



FIGURE 4 An image sequence showing the online inference of intention and the evolution of the human hand (wrist) trajectory (shown in green) through an approximate expectation-maximization algorithm with online model learning. The initial estimate of the reaching goal intention is randomly chosen from the finite number of objects available on the table.



FIGURE 5 (a) Goal location probabilities computed from the gaze map. The numbers shown inside the labels represent the probability of an object being reached by the person, based on the person's initial head orientation. (b) A dense gaze map showing the 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 movement 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 through NNs are trained subject to convergence to the goal location constraints. The intention estimation strategy is to select the correct model based on the currently observed motion trajectory. A block diagram of the multiple-model-based intention estimation methodology is given in Figure 7. More details of the algorithm can also be found in [53].

Human Motion Dynamic Model

In this section, a method for learning the nonlinear dynamics of the human arm-reaching motion is presented. Consider a state variable $x(t) \in \mathbb{R}^n$ and a set of N_D demonstrations $\{\mathcal{D}_i\}_{i=1}^{N_D}$ representing reaching motions to various goal locations. Each demonstration would consist of the trajectories of the state $\{x(t)\}_{t=0}^{t=T}$ and the trajectories of the state derivative $\{\dot{x}(t)\}_{t=0}^{t=T}$ from time t = 0 to t = T. All state trajectories of 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 one. The nonlinear function $f_e(x(t))$ is approximated by an 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).

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 to learn the weights of the NN:

$$\{\hat{W}, \hat{U}\} = \underset{W, U}{\operatorname{argmin}} \{\alpha E_D + \beta E_W\}$$
(18)

subject to
$$\frac{\partial f_e}{\partial x}^T M + M \frac{\partial f_e}{\partial x} \prec -\gamma M, \quad M \succ 0$$
 (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



FIGURE 6 An image sequence showing the online inference of a person's 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 the human's reaching motion to the object being selected.

42 IEEE CONTROL SYSTEMS >> DECEMBER 2020

NN weights, α , $\beta \in \mathbb{R}$ 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 the Jacobian $\partial f_e(x(t)) / \partial x$ 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 a 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 in advance. Let a fixed set of candidate goal locations that the human can reach be $G = \{g_1, g_2, ..., g_{N_g}\}$. The NN weights learned from human demonstrations are 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

$$x^{j}(t+1) = x^{j}(t) + f_{e}(x^{j}(t))\delta t + \omega_{e}^{j}(t)\delta t,$$
(20)

where $j = 1, ..., N_g$, δt is the sampling period, and $\omega_e^j(t) \sim N(0, Q_e^j) \in \mathbb{R}^{2n}$ 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 the X_e , Y_e , and Z_e positions. The measurement model is

$$z_e(t) = h_e(x^j(t)) + v_e^j(t), \quad j = 1, 2, \dots, N_g,$$
(21)

where $z_e(t)$ is the measurement vector, $v_e(t) \sim \mathcal{N}(0, R_e)$ is a zero-mean GP with covariance R_e , and

$$h_e(x^j(t)) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x^j(t)$$

is the measurement function.

Let $M_1, M_2, \ldots, M_{N_g}$ 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 $P(g_i | Z_{1:t})$. The expression $P(g_i | Z_{1:t})$ indicates the posterior probability of each g_i being the correct goal location, given a set of measurements $Z_{1:t} = [z_e(1), z_e(2), \dots, z_e(t)]$. Note that $P(g_i | Z_{1:t}) = P(M_i | Z_{1:t})$ since the models and goal locations have a one-to-one correspondence. Hence, 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 $P(M_i | Z_{1:t}) = P(M_i | z(t), Z_{1:t-1}) =$ $p(z(t) | Z_{1:t-1}, M_i) P(M_i | Z(1:t-1)) / \sum_{i=1}^{N} p(z(t) | Z_{1:t-1}, M_i) \times$ $P(M_i | Z + (1:t-1))$, where $p(z(t) | Z_{1:t-1}, M_j)$ is the likelihood function of mode *j* at time *t* and $P(M_i) | Z_{1:t-1}$ is the prior probability of M_i 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* is $\Lambda_i(t) = p(z(t) | Z_{1:t-1}, M_i) = p(v_e^j(t)) =$ $\mathcal{N}(z_e(t) - \hat{z}_e^j(t|t-1); 0; S_e^j)$. The innovation $\nu_j = z_e(t) - \hat{z}_e^j(t|t-1)$ and its covariance S_i 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 the state-dependent coefficient form parameterization of nonlinear systems [115] can also



FIGURE 7 A block diagram of the multiple-model-based intention estimation algorithm with gaze priors. The gaze estimator block uses red-green-blue (RGB) images along with the head position to estimate the gaze map of a person, which determines the most probable objects that the person is looking at. The information from the gaze map is used to compute the prior model probability for the interacting multiple-model (IMM) algorithm. CNN: convolutional neural network; ANN: artificial neural network; EKF: extended Kalman filter.

be used for each individual filter. Each iteration of the IMM filter for intention inference is divided into four main steps: the interaction/mixing stage, model-matched filtering, the model probability update, and model switch detection. More details about the static IMM filter can be found in [64]. Details specific to the multiple-model filter in the context of gaze-based intention estimation can be reviewed in [53].

In many real-world applications, a small number of models may not be sufficient to describe all the modes. When the number of models is large, the performance of the IMM filter can degrade, and the computational burden increases. A VS-IMM filter can be used in such cases [65]. A limiting case of the VS-IMM when the mode space is continuous is presented in [66]. In certain HRC applications, a significant number of models is required to represent the application context.

Computation of Prior Distribution the Using Gaze Map for the Interacting Multiple-Model Filter

Using the gaze-based prior computation procedure and the average prior probability $\bar{p}_j(0)$ computed through (16), the probability of each of the N_g candidate locations being the goal site is $\mu_j(0) = \bar{p}_j(0) / \sum_{j=1}^{N_g} \bar{p}_j(0)$, where $\mu_j(0)$ is the prior probability of g_j being the goal area for the IMM filter and g_j refers to the positioning of the *j*th object.

Experimental Results for

Multiple-Model-Based Intention Estimation

To validate the multiple-model-based intention estimation algorithm, a set of 10 demonstrations collected from a

subject is 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 a Microsoft Kinect for Windows. The joint position data obtained from the subjects are preprocessed to obtain the velocity and acceleration estimates using a KF (see [114] for details). In all the experiments, the position and velocity of the hand in the 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 the 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 $0.2I_{6\times 6}$, $0.1I_{6\times 6}$, and $0.2I_{6\times 6}$, respectively. The state estimates \hat{x}^j , $j = 1, 2, ..., N_g$ are initialized using the first two measurements $z_e(1)$ and $z_e(2)$ (a finite difference method is used for the velocity initialization). The model transition matrix for the IMM is chosen to be an 8×8 matrix Π_{ij} with diagonal elements of 0.79 and off-diagonal elements of 0.03. For computing the IMM priors, a uniform distribution and gaze-based cues is applied.

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 Figure 8, the intention estimate is displayed along with the trajectory evolution for the uniform prior. The initial goal location is set



FIGURE 8 An image sequence showing the online inference of intention and the evolution of the human hand trajectory for a uniform prior distribution.

44 IEEE CONTROL SYSTEMS >> DECEMBER 2020

```
Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on October 29,2022 at 00:12:54 UTC from IEEE Xplore. Restrictions apply.
```



FIGURE 9 An image sequence showing the online inference of intention and the evolution of the human hand trajectory for a prior distribution computed using the gaze map.

to object 5, and, after 0.3 s, the predicted goal location matches the true intention. In the second experiment, the prior distribution is computed from the dense gaze map, as shown in Figure 5, using (16). The gaze-based prior assigns a higher prior probability to the goal locations where the human subject is likely to be looking. In Figure 9, the intention estimate and trajectory evolution for a prior distribution computed by using a dense gaze map are presented. The goal location with the highest prior probability is the true intention. Gaze-based prior computation assigns a higher probability value to the true intention, which results in a better prediction with fewer observations of the human hand trajectory. The 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

This section describes robot control design that takes into consideration the human trajectory generated by the intention estimators to produce motion that is safe around the person. First, a robot's desired trajectory generation algorithm is presented to determine the desired trajectory based on the human's predicted course from the intention estimator. A BF 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 are 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 a person is holding one side of an object and the robot is holding the other side. While carrying the object, the robot's end effector trajectories may cross the safety ellipsoid around the human, which is when the CBF is used to modify the robot's desired movements. An example scenario of the task is in Figure 10.



FIGURE 10 An example scenario showing a human and a robot working closely together to carry an object. The gray sphere around the human operator corresponds to the region 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.

Generation of the Robot's Desired Trajectories for the Human–Robot Task

A method is presented that generates the desired robot trajectories as a function of the human's inferred movement. If the robot's desired trajectory enters the prescribed safety ellipsoid around the human, then a BF-based formulation is used to modify the robot's movement such that it always stays outside the danger area. In Figure 11, the details of the reference trajectory generation algorithm are illustrated in the form of a block diagram. Let $x_{\text{Rd}}(t) \in \mathbb{R}^3$ be the desired position state of the robot's end effector and $x_H(t) \in \mathbb{R}^3$ be the human's position trajectory generated by the intention estimation algorithm. The robot trajectory $x_{\text{Rd}}(t)$ is

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

where $T(\cdot)$ is a generic transformation between $x_H(t)$ and $x_{\text{Rd}}(t)$. Specifically, an affine transformation with known task-specific parameters $A_H : \mathbb{R}^{3\times3}$ and $b_H \in \mathbb{R}^3$ is used in this article. If the robot's generated desired trajectory $x_{\text{Rd}}(t)$ enters an unsafe zone represented by an ellipsoid drawn around the human, then the robot's preferred course is required to be modified such that the robot does not collide with the person.

To modify the robot's desired trajectory, a dynamic model of $x_{\text{Rd}}(t)$ is formulated with a control input so that two main objectives are satisfied: 1) the controller is able to modify $x_{\text{Rd}}(t)$ such that $x_{\text{Rd}}(t)$ does not enter the safety ellipsoid around the human, and 2) the modified $x_{\text{Rd}}(t)$ still closely tracks the original desired trajectory $x_{\text{Rd}}(t)$. To achieve this, consider the robot's desired end effector motion dynamics $\dot{x}_{\text{Rd}}(t) = f_{\text{Rn}}(x_{\text{Rd}}(t))$, where $f_{\text{Rn}}: \mathbb{R}^3 \to \mathbb{R}^3$ is a nonlinear continuous function that governs the robot's end effector motion dynamics. Based on the estimates of the human trajectory intention $\hat{x}_H(t)$ and its

time derivative $\hat{x}_H(t)$ [and using function approximation methods, such as the extreme learning machine (ELM)], the function $\hat{x}_{Rd}(t) = \hat{f}_{Rn}(\hat{x}_{Rd}(t) = T(\hat{x}_H(t)))$ can be learned, where $\hat{f}_{Rn} : \mathbb{R}^3 \to \mathbb{R}^3$ is an approximation of $f_{Rn}(\cdot)$. See "Extreme Learning Machine" for more details of the ELM.

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 $\hat{f}_{Rn}(\hat{x}_{Rd}(t))$ avoid 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 CBFs and control Lyapunov function (CLF) theory. To execute this, a robot's modified desired end effector dynamics can be written as

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

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

Control Barrier Function Formulation

A CBF defines a forward invariant region such that solutions of the DS that start in that area remain there permanently. The choice of an invariant region is application specific as long as the CBF conditions specified in "Constructing Control Barrier and Control Lyapunov Functions" are valid choices. In this article, the CBF is used to avoid the obstacle region; that is, the robot trajectories that start outside the unsafe region permanently remain outside that zone. Ellipses, circles, and ellipsoids are commonly used shapes to represent obstacles and regions of operation where it is unsafe for a robot's end effector to enter. The application uses a 3D ellipsoid shape around a human as an unsafe zone for a robot to enter. The ellipsoidal BF is defined as a continuously differentiable function $b: \mathbb{R}^3 \to \mathbb{R}$ given by



FIGURE 11 A block diagram of the reference trajectory generator for the robot, based on observed and inferred human action. The human movement approximation produced by 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 control barrier function approach is used to modify the robot's movement. Otherwise, a control Lyapunov function approach is used to generate a bounded and convergent reference trajectory for the robot.

46 IEEE CONTROL SYSTEMS >> DECEMBER 2020

$$b(x(t)) = \frac{(x_1(t) - x_{1g})^2}{a_1^2} + \frac{(x_2(t) - x_{2g})^2}{a_2^2} + \frac{(x_3(t) - x_{3g})^2}{a_3^2} - 1,$$
(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; and a_1 , a_2 , and a_3 are the semiaxes of an ellipsoid. Using the BF in (24), the CBF candidate is defined as B(x(t)) = 1/b(x(t)), which satisfies the properties given in "Constructing Control Barrier and Control Lyapunov Functions."

Control Lyapunov Function Formulation

The CLF is used to ensure 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 $V(\hat{x}_{\text{Rd}}(t)) = 1/2(\hat{x}_{\text{Rd}}(t) - x_{\text{Rd}}^*)^T(\hat{x}_{\text{Rd}}(t) - x_{\text{Rd}}^*), \forall \hat{x}_{\text{Rd}}(t) \in \mathcal{X}_u$. Using the time derivative of the CLF and (23),

$$\dot{V}(\hat{x}_{\rm Rd}) = (\hat{x}_{\rm Rd} - x_{\rm Rd}^*)^T (\hat{f}_{\rm Rn}(\hat{x}_{\rm Rd}) + \bar{u}) \le -\gamma_l V(\hat{x}_{\rm Rd}), \qquad (25)$$

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

Control Design Using Control Barrier Function and Control Lyapunov Function Constraints

Given the approximated nonlinear function $f_{Rn}(\hat{x}_{Rd}(t))$, 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 1

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:

$$\begin{split} \bar{u}^*, \delta^* &= \operatorname*{argmin}_{\bar{u} \in \mathbb{R}^n} \frac{1}{2} \bar{u}^T P \bar{u} + p \delta^2 \\ \text{subject to} \quad (\hat{x}_{\text{Rd}} - x_{\text{Rd}}^*)^T \bar{u} \leq -(\hat{x}_{\text{Rd}} - x_{\text{Rd}}^*)^T \hat{f}_{\text{Rn}}(\hat{x}_{\text{Rd}}) \\ \quad -\gamma_l V(\hat{x}_{\text{Rd}}) + \delta, \quad \forall \hat{x}_{\text{Rd}} \in \mathcal{X}_0, \end{split}$$
(26)

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

Case 2

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 guarantee that the trajectories remain stable with respect to x_{Rd}^* . The optimization problem subject to the CBF and CLF is

Extreme Learning Machine

The extreme learning machine (ELM) is a learning algorithm that uses a least-squares approach to estimate the parameters of a single-layer neural network (SLNN). According to ELM theory [S2], from the function approximation perspective, the input weights and all the hidden node parameters are randomly assigned. ELM theory claims that, unlike conventional learning methods, parameter tuning the input weights, slopes, and biases is not required in learning. Therefore, the choice of the ELM over traditional gradient-based algorithms for SLNN is because the learning speed of the ELM is faster and because the gradient-based methods may have issues, such as local minima and improper learning rates [S2].

Consider the following multilayer perceptron with n_h hidden nodes:

$$y(t) = \sum_{i}^{n_h} G_i(x(t), U_i, a_i, b_i) \cdot W_i^{\mathsf{T}},$$
(S5)

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 the 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 $U \in \mathbb{R}^{n \times n_b}$, the slopes (which are usually set to one) $a \in \mathbb{R}^{n_b}$, and the biases $b \in \mathbb{R}^{n_b}$. The sigmoid function is typically chosen for the activation function.

Although the ELM algorithm is fast, random initialization may lead to saturated or constant neurons, which are not desired while learning the model [S3]. 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 the IP method, a computationally efficient batch version of the method, called BIP, was introduced in [S3].

REFERENCES

[S2] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006. doi: 10.1016/j.neucom.2005.12.126.
[S3] K. Neumann and J. J. Steil, "Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity," *Neurocomputing*, vol. 102, pp. 23–30, Feb. 2013. doi: 10.1016/j.neucom.2012.01.041.

DECEMBER 2020 « IEEE CONTROL SYSTEMS 47

$$\bar{u}^{*}, \delta^{*} = \operatorname*{argmin}_{\bar{u} \in \mathbb{R}^{n}} \frac{1}{2} \bar{u}^{T} P \bar{u} + p \delta^{2}$$

subject to
$$\frac{\partial B(\hat{x}_{\text{Rd}})}{\partial \hat{x}_{\text{Rd}}} \bar{u} \leq \frac{\gamma_{B}}{B(\hat{x}_{\text{Rd}})} - \frac{\partial B(\hat{x}_{\text{Rd}})}{\partial \hat{x}_{\text{Rd}}} \hat{f}_{\text{Rn}}(\hat{x}_{\text{Rd}}),$$
$$\forall \hat{x}_{\text{Rd}} \in \mathcal{X}_{0}, \qquad (27)$$

$$(\hat{x}_{\rm Rd} - x^*_{\rm Rd})^T \bar{u} \leq -(\hat{x}_{\rm Rd} - x^*_{\rm Rd})^T f_{\rm Rn}(\hat{x}_{\rm Rd}) - \gamma_1 V(\hat{x}_{\rm Rd}) + \delta, \forall \hat{x}_{\rm Rd} \in \mathcal{X}_0,$$
(28)

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

Numerical Simulation Results

Given the position estimates of the human-demonstrated motion trajectory, the robot's desired movement can be obtained from (23). However, the position trajectory generated from the affine function does not guarantee that the human workspace will be avoided. As shown in Figure 12, the robot's desired trajectory (in solid green lines) intersects with the human's workspace. An ellipsoid is used to represent the unsafe region for the robot's end effector. In that area, the human completes his or her own tasks. Based on the robot's desired end effector states (position and velocity), an ELM parameterization is employed 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 and the CLF, to ensure the position trajectory generated from the learned model avoids the human workspace.

In Figure 12, the simulated results are shown in which the controller applies corrections to the trajectory generated by the model to avoid the human workspace while still reaching the target location. In Figure 13, an example is illustrated in which the robot's desired end effector trajectory is far 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 the target location without interfering with the person's workspace.

Constructing Control Barrier and Control Lyapunov Functions

BARRIER FUNCTION

C onsider a continuous nonlinear dynamical system (DS) of the form

$$\dot{x}(t) = f(x(t)), \tag{S6}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz-continuous nonlinear function and $x(t) \in S \subseteq \mathbb{R}^n$ is the state of the system. A set $S \in \mathbb{R}^n$ is called *(forward) invariant* with respect to (S6) if, for any initial condition, $x(0) := x(t_0) \in S$ implies that $x(t) \in S, \forall t \in \mathbb{R}^+$ [S4]. Barrier functions (BFs) define a forwardinvariant safe region where the solutions of a DS remain permanently [S5], [S6].

CONSTRUCTING THE BARRIER FUNCTION

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

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

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

$$Int(\mathcal{S}) = \{ x \in \mathbb{R}^n : b(x) > 0 \},$$
(S9)

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, Def. 1]

Given the continuous system (S6), the closed set S defined by (S7)–(S9), and the continuously differentiable function $b : \mathbb{R}^n \to \mathbb{R}$, a real-valued function $B : Int(S) \to \mathbb{R}$ 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 , and η_3 such that, for all $x \in Int(S)$,

$$\frac{1}{\eta_1(b(x))} \le B(x) \le \frac{1}{\eta_2(b(x))},$$
 (S10)

$$\frac{\partial B(x)}{\partial x}f(x) \le \eta_3(b(x)). \tag{S11}$$

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 the *inverse type* and the *logarithmic type*, given by B(x(t)) = 1/(b(x(t))) and $B(x(t)) = -\log(b(x(t))/(1 + b(x(t)))$, respectively [S6]. Note that both candidates are unbounded on the set boundary $B(x(t)) \to \infty$ as $x(t) \to \partial S$.

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 $\dot{B}(x) \leq 0$ [S5]. However, as noted in [S7], the existence of a BF is a sufficient condition to guarantee the safety property to be verified. That is, the invariance verification of all sublevels of the closed set S may not be required. Therefore, the authors of [S6] relaxed the condition to $\dot{B} \leq \gamma/B$ 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 an essential means to verify the invariance of a set. However, they cannot be used in a direct form to design a controller [S6]. That is, to ensure that the set Int(S) is forward invariant under the dynamics of the system (S6), a controller that guarantees the invariance of the set is required. Similar to how Lyapunov functions are extended to control Lyapunov functions [S8], the concept of BFs can be extended to the case of

48 IEEE CONTROL SYSTEMS >> DECEMBER 2020

ADAPTIVE ROBOT CONTROLLER

In this section, an adaptive controller for the robot arm is designed that follows the preferred movement created by the desired trajectory generator block. First, a robot dynamic model represented by EL dynamics is discussed. An adaptive controller for the robot manipulator that tracks the desired trajectory is presented with a Lyapunov stability analysis.

Robot Dynamic Model

Consider the following robot manipulator dynamics (represented by EL dynamics) given by

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

In (29), q(t), $\dot{q}(t)$, and $\ddot{q}(t) \in \mathbb{R}^d$ denote the generalized states; $M(q) \in \mathbb{R}^{d \times d}$ represents a generalized inertia matrix; $C(q, \dot{q}) \in \mathbb{R}^{d \times d}$ designates a generalized centripetal Coriolis matrix; $G(q) \in \mathbb{R}^d$ indicates a generalized gravity vector;

control systems through the use of control BFs (CBFs). Given the DS in (S6), in cases where the solutions of a DS do not stay in an invariant set S, a CBF can be specified that will that assure the solutions remain inside the invariant set.

CONSTRUCTING THE CONTROL BARRIER FUNCTION

Assume the following DS with an external control input:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t),$$
 (S12)

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 (S12) avoid possible obstacles within the DS's state space, the constraint on the system state x(t) is encoded in a smooth constraint function b(x(t)). A value $b(x(t)) \ge 0$ indicates adherence, whereas b(x(t)) < 0 represents a violation. This means that the constraint function must be designed depending on the shape of the obstacle such that b(x(t)) = 0 describes the obstacle's boundary. The set of admissible states X_0 is defined by

$$\begin{aligned} & \mathcal{X}_{0} = \{ x \in \mathbb{R}^{n} : b(x) \geq 0 \}, \\ & \partial \mathcal{X}_{0} = \{ x \in \mathbb{R}^{n} : b(x) = 0 \}, \\ & \text{Int}(\mathcal{X}_{0}) = \{ x \in \mathbb{R}^{n} : b(x) > 0 \}. \end{aligned}$$
(S13)

A reciprocal CBF B : Int(X_0) $\rightarrow \mathbb{R}$ is a nonnegative function if there exist class \mathcal{K} functions α_1, α_2 , and α_3 such that, for all $x(t) \in Int(X_0)$,

$$\frac{1}{\alpha_1(b(x))} \le B(x) \le \frac{1}{\alpha_2(b(x))}, \qquad (S14)$$

$$\inf_{u\in\mathbb{R}^m}\left\{\mathcal{L}_t B(x) + \mathcal{L}_g B(x) u - \alpha_3\left(\frac{1}{B(x)}\right)\right\} \le 0,$$
(S15)

where $\mathcal{L}_t B(x(t))$ is the Lie derivative $(\partial B(x(t))/\partial x)f(x(t))$ along the vector field f(x(t)) and $\mathcal{L}_g B(x(t))$ is the Lie derivative

and $\tau(t) \in \mathbb{R}^d$ signifies 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*(*q*) is positive definite, and $\lambda_{\min}I \leq M(q) \leq \lambda_{\max}I$, 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 $M(q)\dot{\xi} + C(q,\dot{q})\xi + g(q) = Y(q,\dot{q},\xi,\dot{\xi})\Theta$, where $\Theta \in \mathbb{R}^k$ is a set of robot-specific parameters and $Y : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times k}$ is a matrix of known functions of the generalized coordinates and their higher directives.
- » *Property 3*: The inertia and centripetal Coriolis matrices satisfied the property $\xi^T (\dot{M} 2C) \xi = 0$, $\forall \xi(t) \in \mathbb{R}^n$; that is, $(\dot{M} 2C)$ is a skew-symmetric matrix. See [112] and [116] for more details.

Let $x_R(t) \in W \subset \mathbb{R}^m$ be 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

 $(\partial B(x(t))/\partial x)g(x(t))$ along the vector field g(x(t)). Hence, for the system in (S12), any locally Lipschitz controller $u : X_0 \to \mathbb{R}^m$ that is selected from (S15) assures the closed set $X_0 \subset \mathbb{R}^n$ is forward invariant.

CONTROL LYAPUNOV FUNCTION

A continuous nonlinear DS of the form (S12) is said to be globally asymptotically stable at equilibrium x^* if there exists a real-valued, continuously differentiable function $V(x(t)) : \mathbb{R}^n \to \mathbb{R}$ (which is also known as the control Lyapunov function) that is differentiable with respect to its argument, and there exists a set of controls $\tilde{U} \subset \mathbb{R}^m$ and class \mathcal{K} functions α_{1v}, α_{2v} , and α_{3v} such that, for all $x(t) \in Int(X_0)$, the following conditions are satisfied:

$$\alpha_{1\nu}(|x|) \le V(x) \le \alpha_{2\nu}(|x|),$$
 (S16)

$$\inf_{u\in U} \{\mathcal{L}_f V(x) + \mathcal{L}_g V(x) \bar{u} + \alpha_{3\nu}(|x|)\} \le 0,$$
(S17)

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

REFERENCES

[S4] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[S5] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Proc. Int. Workshop Hybrid Systems, Computation and Control*, 2004, pp. 477–492. doi: 10.1007/978-3-540-24743-2_32.

[S6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.

[S7] L. Dai, T. Gan, B. Xia, and N. Zhan, "Barrier certificates revisited," *J. Symbolic Comput.*, vol. 80, pp. 62–86, 2017. doi: 10.1016/j. jsc.2016.07.010.

[S8] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, 1983. doi: 10.1137/0321028. forward and velocity kinematic relationships are given by $x_R(t) = h_{FR}(q(t)), \ \dot{x}_R(t) = J(q(t))\dot{q}(t)$, where $h_{FR} : \mathbb{R}^{n_R} \to \mathbb{R}^m$ 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 $x_{\text{Rd}}(t) \in \mathbb{R}^m$ of the robot end effector state is produced based on the robot's desired trajectory generator block. To design the robot controller, consider the signals a(t), v(t), and s(t) for nonredundant robots given by $v = J^{-1}(q)(\dot{x}_{\text{Rd}} - \Lambda e)$, $a = \dot{J}^{-1}(q)(\dot{x}_{\text{Rd}} - \Lambda e) + J^{-1}(q)(\ddot{x}_{\text{Rd}} - \Lambda \dot{e})$, and $s = J^{-1}(q)(-\dot{x}_{\text{Rd}} + \Lambda e) + \dot{q}$, where $e(t) \triangleq (x_R(t) - x_{\text{Rd}}(t)) \in$ \mathbb{R}^m is the tracking error, $v = \dot{q} - s$, $a = \dot{v}$, and $\Lambda \in \mathbb{R}^m \times \mathbb{R}^m$ 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

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

The control input in (29) is designed to be

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

where $\hat{Y} = Y(q, \dot{q}, \dot{v}, \hat{a})$ is a regressor matrix; $\hat{\Theta}$ 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; $\hat{v} = J^{-1}(q)(\hat{x}_{Rd} - \Lambda \hat{e})$, $\hat{a} = \dot{J}^{-1}(q)(\hat{x}_{Rd} - \Lambda \hat{e}) + J^{-1}(q)(\hat{x}_{Rd} - \Lambda \hat{e})$; and $\hat{s} = J^{-1}(q)(-\hat{x}_{Rd} + \Lambda \hat{e}) + \dot{q}$. The parameter update rule is

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

where $\Gamma \in \mathbb{R}^{k \times k}$ is a positive definite matrix and proj(·), 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

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

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

Assumption 1

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

Remark 2

Based on Assumption 1 and formulas for $\tilde{v}(t)$, $\tilde{a}(t)$, $\tilde{s}(t)$ and $\tilde{Y}(t)$, 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 and uniformly bounded in the sense that



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 robot's desired end effector position trajectory is modified by a controller that avoids the ellipsoid zone while reaching the target location. CBF: control barrier function; CLF: control Lyapunov function.

50 IEEE CONTROL SYSTEMS >> DECEMBER 2020

 $||e(t)|| \rightarrow \epsilon_0 \exp(-\epsilon_1 t) + \epsilon_2$, where ϵ_0 , ϵ_1 , $\epsilon_2 \in \mathbb{R}$ denote positive bounding constants if the following conditions for the gains of the controller defined in (31) are satisfied: $K_t > 0$ and $K_J > 0$. See [9] for the proof and 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 the redundant manipulator can be shown to asymptotically track the desired trajectory.

Numerical Simulation Results

The results of the tracking controller (shown in Figures 14 and 15) were generated using a simulation of Rethink

Robotics' seven-degrees-of-freedom Baxter Robot in the Gazebo simulation platform. The reference trajectories and their derivatives (which are obtained by observing the motion of a human hand and then modified according to the CBF and CLF constraints) are used as $\hat{x}_{Rd}(t)$ and $\hat{x}_{Rd}(t)$ 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.

Experimental Results

A collaborative task involving a human and a robot moving a heavy object is designed for experimental evaluation. In



FIGURE 13 A scenario 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 the control Lyapunov function (CLF) that avoids the region where a human is present (represented by an ellipsoid) while reaching the target location.



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

DECEMBER 2020 « IEEE CONTROL SYSTEMS 51

it, the robot must infer the motion intention of the human online and adjust its trajectory to coordinate with the human to move a rectangular box from a workbench to an elevated surface. While doing so, the robot must follow a trajectory that does not enter the human workspace to ensure the safety of the person. 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 Figure 16.

Four demonstrations of the human hand motion training data for each movement are collected using 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 incorporated into the IMM framework. A Microsoft Kinect sensor placed on top of Baxter's head is employed for acquiring the human hand position data online during the experiment. The IMM estimator is used to approximate the 3D position of the human hand and the likelihood of each model being the true representation at each time instance.

A controller $\bar{u}(t)$ is designed by the solving the optimization problem in (26)-(28), which generates the robot's desired trajectory $\hat{x}_{Rd}(t)$. The controller $\bar{u}(t)$ ensures that the robot's desired trajectory stays outside the human workspace, ensuring the safety of the person. The matrix His chosen to be identity for the optimization. The adaptive tracking controller in (31) is used to monitor the safe trajectory generated by solving the optimization problem. The experiments are completed with the Baxter research robot and the Baxter application programming interface along with the Robot Operating System. The gains chosen for the adaptive controller are $K_I = \text{diag} \{27, 30, 37.5, 0, 0, 0\},\$ $K_t = \text{diag} \{3, 6, 3, 3.25, 3, 3, 1.5\}, \text{ and } \Lambda = \text{diag} \{1, 1, 1, 0, 0, 0\}.$ The orientation of the robot end effector was constant throughout the experiment. The results of the experiment are summarized in Figure 17(a) and (b). The robot end effector trajectories converge and track the desired trajectories, as shown in Figure 17(a). The error between the actual and desired trajectories appears in Figure 17(b).



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



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

52 IEEE CONTROL SYSTEMS >> DECEMBER 2020



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.

CONCLUSION

For assistive robots to integrate seamlessly into human environments, they should be able to understand the intentions of human agents and adapt to those people's motions. Many HRC tasks in advanced manufacturing operations require humans and robots to perform joint tasks, such as carrying heavy loads and large, flexible materials and completing welding operations, for which the robot can act as an assistant to learn people's motion behaviors, based on RGB-D sensor data. A robot can then adapt its motions to complete a cooperative task. In this article, a survey of the topic was first provided by presenting existing literature in a tutorial manner, and approaches that focus and expand this article were discussed for inferring human motion trajectories and estimating reaching goal intentions.

The first method is based on an ML estimation technique called *approximate EM* that uses online model learning to accommodate uncertainties in human motions. The second is based on a multiple-model estimator that switches between multiple nonlinear human motion models. Short tutorials on approximate EM and multiple-model estimation methods were presented that use nonlinear models of human motion learned through NN function estimation from labeled sensor data. For the ML-based estimator, the optimizer is initialized using human gaze cues and random initialization. For the MAP-based estimator, a gaze-based prior is employed that is computed by analyzing RGB image data using a deep NN. The results were compared with the uniform prior.

A human-in-the-loop control strategy was discussed that uses the estimated human motion trajectory and intended reaching goal location to determine a safe robot-reference trajectory, which is tracked via an adaptive controller for the robot. For the desired trajectory generation, a CBF formulation is used to produce movements that are safe around humans. The desired motions are created with two objectives: 1) the robot end effector trajectories do not enter a safety region around a human, and 2) the robot end effector trajectories are able to synchronize with human motion as closely as possible. The adaptive controller converges to the desired trajectories produced by the generation block.

A case study of humans and robots carrying an object together was discussed. Future challenges, such as ensuring the safety of humans in the presence of actuator and sensor failures, can be a potential topic to explore, for example, via human supervision of automation to achieve resilient control [119]. Estimating intention by using sensors, such as ultrasound imaging, can be another avenue for potential research [120].

ACKNOWLEDGMENTS

This work was supported, in part, by a Space Technology Research Institutes grant (80NSSC19K1076) from the NASA Space Technology Research Grants Program, by Office of Naval Research Award N00014-20-1-2040, and by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award DE-EE0007613. The authors would like to thank Tanisha Mitra for her help with figure formatting.

AUTHOR INFORMATION

Ashwin P. Dani (ashwin.dani@uconn.edu) received the M.S. and Ph.D. degrees from the University of Florida, Gainesville. He was a postdoctoral research associate at the University of Illinois at Urbana–Champaign. He is currently an associate professor at the University of Connecticut, Storrs. He has authored more than 50 technical articles and four book chapters. His current research interests include nonlinear estimation and control, machine learning for control, human-robot collaboration, and autonomous navigation. He is a Senior Member of IEEE.

Iman Salehi received the M.S. degree from the Department of Electrical and Computer Engineering, University of Hartford, Connecticut, in 2015. He is currently working toward the Ph.D. degree in electrical and computer engineering at the University of Connecticut, Storrs. 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 in electrical and computer engineering at the University of Connecticut, Storrs. His research interests include deep learning for control, computer vision, robotics, and vision-based estimation and control.

Daniel Trombetta received the B.S. and M.S. degrees in electrical and computer engineering from the University of Connecticut, Storrs, in 2018. His research interests include machine learning, human–robot collaboration, and estimator design.

Harish Ravichandar received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2014 and the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, in 2018. He is a research scientist in the School of Interactive Computing, Georgia Institute of Technology. His current research interests include robot learning, human–robot interaction, and multiagent systems. He is a Member of IEEE.

REFERENCES

[1] S. Berger, Making in America: From Innovation to Market. Cambridge, MA: 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*, Y. Wang and F. Zhang, Eds. Switzerland: Springer, 2017, pp. 153–176.

[3] J. Mumm and B. Mutlu, "Human-robot proxemics: Physical and psychological distancing in human-robot interaction," in *Proc. 6th Int. Conf. Human-Robot Interaction*, 2011, pp. 331–338. doi: 10.1145/1957656.1957786.
[4] C. Nobile, "Robotics business review perspectives 2013: Outlook for nextgen, new-gen industrial co-worker robotics," *Robot. Business Rev.* 2013. [Online]. Available: https://www.roboticsbusinessreview.com/manufacturing/

outlook_for_next_gen_new_gen_industrial_co_worker_robotics/ [5] M. Sampath and P. P. Khargonekar, "Socially responsible automation: A framework for shaping the future," *Nat. Acad. Eng. 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, "Robot architecture," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 377–390, 2001. doi: 10.1109/70.954751.

[7] M. A. Goodrich and A. C. Schultz, "Human–robot interaction: A survey," *Found. Trends Hum.-Comput. Interaction*, vol. 1, no. 3, pp. 203–275, 2007. doi: 10.1561/110000005.

[8] E. Martinez and A. P. del Pobil, "Safety for human-robot interaction in dynamic environments," in *Proc. IEEE Int. Symp. Assembly and Manufacturing*, 2009, pp. 327–332. doi: 10.1109/ISAM.2009.5376949.

[9] H. C. Ravichandar, D. Trombetta, and A. P. Dani, "Human intentiondriven learning control for trajectory synchronization in human-robot collaborative tasks," *IFAC Cyber-Phys. Hum. Syst.*, vol. 51, no. 34, pp. 1–7, 2019. doi: 10.1016/j.ifacol.2019.01.001.

[10] M. Sodhi, B. Reimer, and I. Llamazares, "Glance analysis of driver eye movements to evaluate distraction," *Behav. Res. Methods, Instrum., Comput.*, vol. 34, no. 4, pp. 529–538, 2002. doi: 10.3758/BF03195482. [11] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *J. Guid., Control, Dyn.*, vol. 30, no. 2, pp. 370–382, 2007. doi: 10.2514/1.26750.

[12] B. Alqaudi, H. Modares, I. Ranatunga, S. M. Tousif, F. L. Lewis, and D. O. Popa, "Model reference adaptive impedance control for physical humanrobot interaction," *Control Theory Technol.*, vol. 14, no. 1, pp. 68–82, 2016. doi: 10.1007/s11768-016-5138-2.

[13] C.-S. Tsai, J.-S. Hu, and M. Tomizuka, "Ensuring safety in human-robot coexistence environment," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems (IROS)*, 2014, pp. 4191–4196. doi: 10.1109/IROS.2014.6943153.

[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 Trans. Mechatron.*, vol. 19, no. 1, pp. 225–237, 2012. doi: 10.1109/TMECH.2012.2226597.

[15] F. Zhang and H. Huang, "Source selection for real-time user intent recognition toward volitional control of artificial legs," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 5, pp. 907–914, 2012. doi: 10.1109/JBHI.2012. 2236563.

[16] T. G. Sugar et al., "Design and control of RUPERT: A device for robotic upper extremity repetitive therapy," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 3, pp. 336–346, 2007. doi: 10.1109/TNSRE.2007.903903.

[17] D. A. Baldwin and J. A. Baird, "Discerning intentions in dynamic human action," *Trends Cogn. Sci.*, vol. 5, no. 4, pp. 171–178, 2001. doi: 10.1016/ S1364-6613(00)01615-6.

[18] M. A. Simon, Understanding Human Action: Social Explanation and the Vision of Social Science. New York: 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. doi: 10.1016/0005-1098(70)90051-8.

[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. doi: 10.1016/0005-1098(70)90052-X.

[21] R. B. Warrier and S. Devasia, "Inferring intent for novice human-in-theloop iterative learning control," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1698–1710, 2017. doi: 10.1109/TCST.2016.2628769.

[22] C. Liu et al., "Goal inference improves objective and perceived performance in human–robot collaboration," in *Proc. Int. Conf. Autonomous Agents Multiagent Systems*, 2016, pp. 940–948.

[23] Y. Li and S. Ge, "Human-robot collaboration based on motion intention estimation," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 3, pp. 1007–1014, 2014. doi: 10.1109/TMECH.2013.2264533.

[24] D. Kulic and E. A. Croft, "Affective state estimation for human–robot interaction," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 991–1000, 2007. doi: 10.1109/TRO.2007.904899.

[25] E. Meisner, V. Isler, and J. Trinkle, "Controller design for human–robot interaction," *Autonom. Robot.*, vol. 24, no. 2, pp. 123–134, 2008. doi: 10.1007/s10514-007-9054-7.

[26] Y. S. Razin, K. Pluckter, J. Ueda, and K. Feigh, "Predicting task intent from surface electromyography using layered hidden Markov models," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1180–1185, 2017. doi: 10.1109/ LRA.2017.2662741.

[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 *Proc. IEEE Conf. Computer Vision Pattern Recognition Workshop (CVPRW)*, 2003, vol. 5, p. 53. doi: 10.1109/CVPRW.2003.10057.

[28] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, and S. Srinivasa, "Learning the communication of intent prior to physical collaboration," in *Proc. IEEE Int. Symp. Robot Human Interactive Communication*, Sept. 2012, pp. 968– 973. doi: 10.1109/ROMAN.2012.6343875.

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

[30] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making service robots human-safe," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2000, pp. 696–701. doi: 10.1109/IROS.2000.894685.

[31] B. Mutlu, J. Forlizzi, and J. Hodgins, "A storytelling robot: Modeling and evaluation of human-like gaze behavior," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 518–523. doi: 10.1109/ICHR.2006.321322.

[32] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robot. Autonom. Syst.*, vol. 42, nos. 3–4, pp. 143–166, 2003. doi: 10.1016/S0921-8890(02)00372-X.

54 IEEE CONTROL SYSTEMS >> DECEMBER 2020

[33] Z. Wang et al., "Probabilistic movement modeling for intention inference in human-robot interaction," *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 841– 858, 2013. doi: 10.1177/0278364913478447.

[34] H. Ravichandar and A. P. Dani, "Human intention inference using expectation-maximization algorithm with online model learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 855–868, 2017. doi: 10.1109/TASE.2016.2624279.
[35] K. W. Strabala et al., "Towards seamless human–robot handovers," *J. Hum.–Robot Interact.*, vol. 2, no. 1, pp. 112–132, 2013. doi: 10.5898/JHRI.2.1.Strabala.

[36] D. De Carli et al., "Measuring intent in human-robot cooperative manipulation," in *Proc. IEEE Int. Workshop Haptic Audio Visual Environments Games*, 2009, pp. 159–163. doi: 10.1109/HAVE.2009.5356124.

[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 *Proc. IEEE Int. Conf. Robotics Automation*, 2011, pp. 5875–5880. doi: 10.1109/ICRA.2011.5980248.

[38] D. Gehrig et al., "Combined intention, activity, and motion recognition for a humanoid household robot," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems (IROS)*, 2011, pp. 4819–4825. doi: 10.1109/IROS.2011.6095118.

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

[40] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robot. Autonom. Syst.*, vol. 62, no. 4, pp. 591–602, 2014. doi: 10.1016/j.robot.2014.01.003.

[41] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 951–970, 2013. doi: 10.1177/0278364913478446.

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

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

[44] R. Kelley, M. Nicolescu, A. Tavakkoli, M. Nicolescu, C. King, and G. Bebis, "Understanding human intentions via Hidden Markov Models in autonomous mobile robots," in *Proc. ACM/IEEE Int. Conf. Human Robot Interaction*, 2008, pp. 367–374. doi: 10.1145/1349822.1349870.

[45] D. K. E. Croft, "Estimating intent for human-robot interaction," in *Proc. IEEE Int. Conf. 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*, H. Akin, N. Amato, V. Isler, and A. van der Stappen, Eds. Berlin: Springer-Verlag, 2015, pp. 161–177.
[47] R. Luo and D. Berenson, "A framework for unsupervised online human reaching motion recognition and early prediction," in *Proc. Int. Conf. Intelligent Robots Systems (IROS)*, 2015, pp. 2426–2433. doi: 10.1109/IROS.2015.7353706.

[48] M. Monfort, A. Liu, and B. D. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation," in *Proc. AAAI Conf. 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 Robot. Automat. Lett.*, vol. 4, no. 2, pp. 224–230, 2018. doi: 10.1109/LRA.2018.2886407.

[50] Y. Yildiz, A. Agogino, and G. Brat, "Predicting pilot behavior in medium scale scenarios using game theory and reinforcement learning," in *Proc. AIAA Modeling Simulation Technologies (MST) Conf.*, 2013, p. 4908. doi: 10.2514/6.2013-4908.

[51] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *Proc. IEEE Int. Conf. Robotics Automation (ICRA)*, 2015, pp. 885–892. doi: 10.1109/ICRA.2015.7139282.

[52] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. IEEE/ RSJ Int. Conf. Intelligent Robots Systems*, 2013, pp. 299–306. doi: 10.1109/ IROS.2013.6696368.

[53] H. C. Ravichandar, A. Kumar, and A. P. Dani, "Gaze and motion information fusion for human intention inference," *Int. J. Intell. Robot. Appl.*, vol. 2, no. 2, pp. 136–148, 2018. doi: 10.1007/s41315-018-0051-0.

[54] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," in *Proc. Robotics: Science 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 Netw.*, vol. 60, pp. 44–52, Dec. 2014. doi: 10.1016/j.neunet.2014.07.009.

[56] G. C. Goodwin and J. Aguero, "Approximate EM algorithms for parameter and state estimation in nonlinear stochastic models," in *Proc. IEEE Conf. Decision Control, European Control Conf.*, 2005, pp. 368–373. doi: 10.1109/CDC.2005.1582183.

[57] G. Goodwin and A. Feuer, "Estimation with missing data," *Math. Comput. Model. Dyn. Syst.*, vol. 5, no. 3, pp. 220–244, 1999. doi: 10.1076/mcmd.5.3.220.3681.
[58] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. E. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 187–192, 2013. doi: 10.1109/TAC.2012.2203452.

[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 Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, 2008. doi: 10.1109/TAC.2008.930200.

[60] N. Hogan, "An organizing principle for a class of voluntary movements," J. Neurosci., vol. 4, no. 11, pp. 2745–2754, 1984. doi: 10.1523/JNEU-ROSCI.04-11-02745.1984.

[61] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998. doi: 10.1016/S0005-1098(98)00019-3.

[62] H. Ravichandar, I. Salehi, and A. Dani, "Learning partially contracting dynamical systems from demonstrations," in *Proc. 1st Annu. Conf. Robot Learning*, 2017, vol. 78, pp. 369–378.

[63] H. Ravichandar and A. P. Dani, "Human intention inference through interacting multiple model filtering," in *Proc. IEEE Conf. Multisensor Fusion Integration (MFI)*, 2015, pp. 220–225. doi: 10.1109/MFI.2015.7295812.

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

[65] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, 1996. doi: 10.1109/9.489270.

[66] X. R. Li, V. P. Jilkov, J. Ru, and A. Bashi, "Expected-mode augmentation algorithms for variable-structure multiple-model estimation," *IFAC Proc. Vol.*, vol. 35, no. 1, pp. 175–180, 2002. doi: 10.3182/20020721-6-ES-1901.00440.

[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 Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 278–285, 2016. doi: 10.1109/TCST.2016.2523901.

[68] N. Hogan, "Impedance control: An approach to manipulation: Part II-Implementation," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 107, no. 1, pp. 8–16, 1985. doi: 10.1115/1.3140713.

[69] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2010, pp. 249–254. doi: 10.1109/IROS.2010.5648931.

[70] G. Hoffman and C. Breazeal, "Effects of anticipatory perceptual simulation on practiced human-robot tasks," *Autonom. Robot.*, vol. 28, no. 4, pp. 403–423, 2010. doi: 10.1007/s10514-009-9166-3.

[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 *Proc. IEEE Int. Conf. Robotics Automation*, 2013, pp. 4000–4007. doi: 10.1109/ICRA.2013.6631141.

[72] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing," in *Proc. Robot. Sci. Syst.*, 2012.

[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 *Proc. AAAI Fall Symp., Artificial Intelligence 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," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 52, no. 2, pp. 234–245, 2010. doi: 10.1177/0018720809350882.

[75] T. Yucelen, Y. Yildiz, R. Sipahi, E. Yousefi, and N. Nguyen, "Stability limit of human-in-the-loop model reference adaptive control architectures," *Int. J. Control*, vol. 91, no. 10, pp. 2314–2331, 2018. doi: 10.1080/00207179.2017.1342274.
[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. doi: 10.1016/j.ifacol.2017.08.596.
[77] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 3, pp. 415–419, 1998. doi: 10.1109/9.661611.
[78] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *Proc. American Control Conf.*, 2014, pp. 226–241. doi: 10.1109/ACC.2014.6859176.

[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. doi: 10.1016/j.automatica.2014.03.012.

[80] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe modelbased reinforcement learning with stability guarantees," in *Proc. Advances 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 Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3803–3817, 2019. doi: 10.1109/TNNLS.2019.2899311.

[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 Trans. Autom. Control*, vol. 64, no. 7, pp. 2737–2752, 2018. doi: 10.1109/TAC.2018.2876389.

[83] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proc. American Control Conf.*, 2004, pp. 5576–5581. doi: 10.23919/ACC.2004.1384742.

[84] L. S. Breger and J. P. How, "Safe trajectories for autonomous rendezvous of spacecraft," *J. Guid., Control, Dyn.,* vol. 31, no. 5, pp. 1478–1489, 2008. doi: 10.2514/1.29590.

[85] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4063–4069, 2017. doi: 10.1016/j.ifacol.2017.08.789.

[86] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. M. Gregory, "l₁ adaptive control for safety-critical systems," *IEEE Control Syst. Mag.*, vol. 31, no. 5, pp. 54–104, 2011. doi: 10.1109/MCS.2011.941961.

[87] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Proc. Int. Workshop Hybrid Systems, Computation Control*, 2004, pp. 477–492. doi: 10.1007/978-3-540-24743-2_32.

[88] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017. doi: 10.1109/TAC .2016.2638961.

[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 *Proc. IEEE 55th Conf. Decision Control*, 2016, pp. 4661–4666. doi: 10.1109/CDC.2016.7798979.

[90] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *Proc. IEEE Int. Conf. Robotics Automation*, 2018, pp. 2460–2465. doi: 10.1109/ICRA.2018.8460471.

[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 *Proc. IEEE 55th Conf. Decision Control (CDC)*, 2016, pp. 827–834. doi: 10.1109/CDC.2016.7798370.

[92] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *Proc. American Control Conf.*, 2016, pp. 2252–2258. doi: 10.1109/ ACC.2016.7525253.

[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. doi: 10.1016/j.ifacol.2015.11.152.

[94] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc.* 2016 *American Control Conf. (ACC)*, pp. 322–328. doi: 10.1109/ACC.2016.7524935.

[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. doi: 10.1016/j.ifacol.2015.11.167.

[96] Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-aware reinforcement learning framework with an actor-criticbarrier structure," in *Proc. American Control Conf.*, 2019, pp. 2352–2358. doi: 10.23919/ACC.2019.8815335.

[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 *Proc. IEEE Int. Conf. Robotics Automation*, 2019, pp. 22–28. doi: 10.1109/ICRA.2019.8793891.

[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. doi: 10.1016/j.ifacol.2019.01.028.

[99] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robot. Autonom. Syst.*, vol. 47, nos. 2–3, pp. 79–91, 2004. doi: 10.1016/j.robot.2004.03.003. [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 *Proc. ASME 2015 Dynamic Systems Control Conf.*, pp. 1–10. doi: 10.1115/DSCC2015-9850.

[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, Oct. 2018. doi: 10.1016/j.mechatronics.2018.07.007.

[102] S. K. Jayaraman et al., "Trust in AV: An uncertainty reduction model of AV-pedestrian interactions," in *Proc. Companion 2018 ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 133–134. doi: 10.1145/3173386.3177073.

[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. doi: 10.1016/j.automatica.2013.11.028.

[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 Syst. Mag.*, vol. 35, no. 6, pp. 57–80, 2015. doi: 10.1109/MCS.2015.2471056.

[105] A. Bestick, R. Pandya, R. Bajcsy, and A. D. Dragan, "Learning human ergonomic preferences for handovers," in *Proc. 2018 IEEE Int. Conf. Robotics Automation (ICRA)*, pp. 3257–3264. doi: 10.1109/ICRA.2018.8461216.

[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 *Proc. 2015 IEEE/RSJ Int. Conf. Intelligent Robots Systems (IROS)*, 2015, pp. 1037–1044. doi: 10.1109/IROS.2015.7353498.

[107] Y. Wang and F. Zhang, *Trends in Control and Decision-Making for Hu*man-Robot Collaboration Systems. New York: Springer-Verlag, 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 *Proc. 2017 IEEE Int. Conf. Advanced Intelligent Mechatronics (AIM)*, pp. 628–633. doi: 10.1109/ AIM.2017.8014087.

[109] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2015, pp. 688–694. doi: 10.1109/IROS.2015.7353447.
[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. doi: 10.3390/s120201437.

[111] K. Lange, "A gradient algorithm locally equivalent to the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 57, no. 2, pp. 425–437, 1995. doi: 10.1111/j.2517-6161.1995.tb02037.x.

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

[113] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba, "Where are they looking?" in *Proc. Advances Neural Information Processing Systems (NIPS)*, 2015, pp. 199–207.

[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 J. Comput. Inform. Sci. Eng.*, vol. 14, no. 1, pp. 011006–011009, 2014. doi: 10.1115/1.4025810.

[115] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer design for stochastic nonlinear systems via contraction-based incremental stability," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, 2015. doi: 10.1109/ TAC.2014.2357671.

[116] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3. Hoboken, NJ: Wiley, 2006.

[117] W. E. Dixon, "Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics," *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 488–493, 2007. doi: 10.1109/TAC.2006.890321.

[118] Y.-C. Liu and N. Chopra, "Controlled synchronization of heterogeneous robotic manipulators in the task space," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 268–275, 2012. doi: 10.1109/TRO.2011.2168690.

[119] A. B. Farjadian, B. Thomsen, A. M. Annaswamy, and D. D. Woods, "Resilient flight control: An architecture for human supervision of automation," *IEEE Trans. Control Syst. Technol.*, early access, 2020.

[120] Q. Zhang, K. Kim, and N. Sharma. "Prediction of ankle dorsiflexion moment by combined ultrasound sonography and electromyography," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 318–327, 2019.



56 IEEE CONTROL SYSTEMS >> DECEMBER 2020