Human Intention Inference Using Expectation-Maximization Algorithm With Online Model Learning

Harish Chaandar Ravichandar and Ashwin P. Dani

***Abstract*— An algorithm called adaptive-neural-intention estimator (ANIE) is presented to infer the intent of a human operator's arm movements based on the observations from a 3-D camera sensor (Microsoft Kinect). Intentions are modeled as the goal locations of reaching motions in 3-D space. Human arm's nonlinear motion dynamics are modeled using an unknown nonlinear function with intentions represented as parameters. The unknown model is learned by using a neural network. Based on the learned model, an approximate expectation-maximization algorithm is developed to infer human intentions. Furthermore, an identifier-based online model learning algorithm is developed to adapt to the variations in the arm motion dynamics, the motion trajectory, the goal locations, and the initial conditions of different human subjects. The results of experiments conducted on data obtained from different users performing a variety of reaching motions are presented. The ANIE algorithm is compared with an unsupervised Gaussian mixture model algorithm and an Euclidean distance-based approach by using Cornell's CAD-120 data set and data collected in the Robotics and Controls Laboratoy at UConn. The ANIE algorithm is compared with the inverse LQR and ATCRF algorithms using a labeling task carried out on the CAD-120 data set.**

***Note to Practitioners*—This paper addresses the problem of inferring the goal location of a human hand motion observed using an RGB-D sensor while performing reaching tasks, such as picking up objects from a table. A Microsoft Kinect (3-D camera) sensor is used to track the joints of the human skeleton. The dynamics of the human arm motion are learned from the demonstrations of a human reaching for different objects on a workbench. An algorithm is presented that uses the learned dynamic model to infer the goal location of the reaching hand ahead of time. The goal location inference can be useful for path planning and collision avoidance in applications involving human–robot collaboration. The inference algorithm does not depend on the human subject or the number of objects and their placement.**

***Index Terms*— Human-robot interaction, intention inference, expectation-maximization, neural network modeling.**

Manuscript received August 18, 2015; revised March 22, 2016 and September 15, 2016; accepted October 25, 2016. Date of publication December 27, 2016; date of current version April 5, 2017. This paper was recommended for publication by Associate Editor M. Oishi and Editor D. Tilbury upon evaluation of the reviewers' comments.

H. C. Ravichandar is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: harish.ravichandar@uconn.edu).

A. P. Dani is with the Department of Electrical and Computer Engineering and the Management and Engineering for Manufacturing Program, University of Connecticut, Storrs, CT 06269 USA (e-mail: ashwin.dani@uconn.edu).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The Supplementary Material is a video and contains segments of demonstrations and experiments conducted using Cornell's CAD120 dataset and skeletal data collected from five subjects in order to validate the proposed algorithm. This material is 14.7 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2624279

I. Introduction

HUMAN intention inference is the first natural step for achieving safety in human–robot collaboration (HRC), e.g., manufacturing assembly operations [1]–[4]. Studies in psychology show that when two humans interact, they infer the intended actions of the other person and decide which proactive actions to take based on this inference for safe interaction and collaboration [5], [6]. In this paper, an inference algorithm called adaptive-neural-intention estimator (ANIE) is presented to estimate the intentions of human actions.

The complex dynamic motion of the human arm is represented by using a state space model, where a neural network (NN) model is used to represent the state propagation [7], [8]. The positions and velocities of the joints of human arm are used as the states. Intentions are modeled as the goal locations of human arm reaching motions, which are represented by the parameters of the state space model. The problem of intention inference is solved as a parameter inference problem using an approximate expectation-maximization (E-M) algorithm [9]. There are three sources of uncertainty in the human arm motion model: the uncertain system dynamics, the sensor measurement noise, and the unknown human intent. The NN approximation can potentially allow considering user-specific or object-specific characteristics, such as the size and the shape of the object to be included as a part of the dynamics. No specific results including object size and shape are presented in this paper. This paper will be pursued in the future.

A set of demonstrations capturing human arm joint position trajectories for reaching motions is collected by using a 3-D camera (Microsoft Kinect). Each recorded joint position trajectory is labeled according to the corresponding true intention, i.e., the 3-D goal location of the reaching motion. An NN model is learned by using the labeled demonstrations of the joint position trajectories. The learned NN model is then used to infer the intention parameter using the ANIE algorithm. The ANIE algorithm is an approximate E-M algorithm based on the parameter estimation algorithm in [9] for the transition models learned using NNs.

Different humans may reach the same point in 3-D space in different ways based on their physical characteristics. This

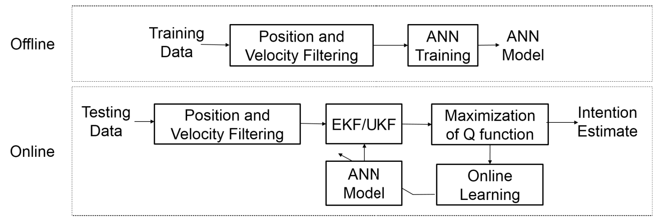

Fig. 1. Block diagram representation of the ANIE algorithm.

brings a challenge in using the model learned from the demonstration data to represent joint position trajectories of the other subjects. One way of updating the model in real time is to use the E-M algorithm by optimizing the **Q** function over the model parameters along with the intention. A closed-form expression for model update using E-M exists, if the model is linear or represented using a radial basis function NN (RBF-NN) [10], [11]. However, the human arm motion dynamics are highly nonlinear and the basis functions of the NN are not restricted to RBFs in this paper. The ANIE algorithm uses NNs with nonlinear sigmoid basis functions. To overcome this challenge, an identifier-system-based algorithm [12] is used for online model update. The identifier system is designed using a robust feedback term, called robust integral of the signum of the error (RISE). Based on the Lyapunov analysis, the parameter update laws for model update are derived using the error between the state estimate generated by the identifier system and the state estimate from the original system model. The inference algorithm is then used with the updated model for early prediction of the intentions. In Fig. 1, a block diagram of the ANIE algorithm is shown.

Experiments are conducted using data collected from a 3-D camera as well as the publicly available CAD-120 data set. The ANIE algorithm is compared with the unsupervised Gaussian mixture model (GMM) algorithm presented in [13] and the Euclidean distance-based approach. For the data sets used in testing, it is observed that the ANIE algorithm outperforms both the unsupervised GMM algorithm and the Euclidean distance-based approach in terms of the intention inference accuracy, time of inference, and trajectory prediction accuracy. Furthermore, experiments conducted with and without the identifier-based model update show that the identifier-based model update significantly improves the intention inference accuracy. The ANIE algorithm shows comparable performance in terms of accuracy and precision, and recall with ATCRF [14] and inverse LQR (I-LQR) [15] algorithms. I-LQR algorithm is better in prediction with 20% and 40% of trajectory observed, whereas the ANIE algorithm shows better performance when 60% and 80% trajectories are observed.

A. Background

Algorithms for human intention estimation are studied in human–computer interaction [16] and human–robot interaction [17]. The human intention is represented via modalities, such as natural language instructions [18], human emotion [19], human's approval response [20], and human's activity [21]–[24]. The intentions are inferred by estimating/measuring information about body posture [25], [26], gestures [27], voice commands [18], eye gaze [28], facial expressions [19], [29], object affordances [14], human skeletal movement [23], [24], and physiological parameters (heart rate and skin response) [20], [30].

Human intention inference has been studied by using hidden Markov models (HMMs) [31]–[33], dynamic Bayesian networks [34], [35], growing HMMs [36], conditional random fields [14], [21], [37], [38], Gaussian processes (GPs) [24], [39], GMMs [13], [40], and I-LQR [15]. The ANIE algorithm models the human arm motion as a continuous nonlinear dynamical system (DS) of human skeletal joints approximated using an NN, where reaching intentions are modeled as the parameters.

In [41], human intention, represented by grasping configuration, is predicted by visually observing the hand–object interaction in grasping tasks. In [26], the human's intention to handover an object is predicted by using key features extracted from a vision sensor. In [14] and [21], the intended activities of human subjects are inferred by modeling spatial-temporal relations through object affordances. In [22], the activities of human agents are inferred by using the HMM of the robot's experience and its interaction with the environment. In [31], a human intent estimation algorithm based on the fuzzy inference logic is presented. In [20], the affective state estimation algorithm based on the HMM is developed. Both in [31] and [20], human intention is represented using valence/arousal characteristics, which are measured by using physiological signals, such as heart rate and skin response. The valence/arousal representation of human intention only indicates the degree of approval to a given stimulus. However, the human motion is not modeled using a continuous DS of human skeletal joints.

In [24], a latent variable model called intention-driven dynamic model is proposed to infer intentions from the observed human movements. Robot table tennis and human activity classification are demonstrated using a belief propagation algorithm coupled with the intention-driven dynamical model (IDDM). In [23], human motion during collaborative manipulation is predicted by using an inverse optimal control approach. In [13], human intention inference algorithm is developed using unsupervised GMMs, where the parameters of GMMs are learned using E-M algorithm. The framework presented in [13] provides an unsupervised online learning algorithm, while the algorithms presented in [23] and [24] do not involve online learning. However, the methods presented in [23], [24], and [13] do not provide any theoretical guarantees on the model learning. In this paper, Lyapunov-based stability analysis is developed to derive the parameter update laws for identifier-based online model update. The analysis ensures the asymptotic convergence of the state identification errors and their derivatives between the learned model and the true model. The guarantees on online learning could be very useful in tasks, where the training data are limited and predictions have to be made about new users with varying motion dynamics in new environments. For instance, consider an assembly task in a manufacturing environment, where the NN is trained using data obtained from a user assembling parts to build an object. If the trained

NN is used for a new user assembling parts of a similar object, the NN approximation error is likely to be high as the motion profiles will be different for different users assembling different parts. However, as new data become available, the presence of the feedback term (RISE) allows the identifier system to implicitly learn the network weights and minimize the effects of NN approximation errors [12].

II. Problem Description and Solution Approach

Consider a 3-D workspace with human performing tasks, such as picking up objects placed on a table. The human operator reaches out to different objects placed on a table and a robot watches the human through a 3-D camera sensor mounted on its head. This paper addresses the problem of inferring the goal location, where the human hand is intended to reach. For human motion is highly nonlinear and uncertain, an NN is used to model the human arm motion. The NN is trained by using a data set containing RGB-D demonstrations of a human reaching for predefined target locations in a given workspace. When a set of new measurements become available, the trained NN is used to estimate the intention (goal location) using an approximate E-M algorithm that is adapted for dynamic models learned using the NNs. Furthermore, the weights of the NN model are updated iteratively using an identifier-based algorithm to adapt to variations in the start locations and trajectories of the human arm.

III. System Modeling

The dynamics of human arm motion are modeled using a nonlinear transition function of joint positions, velocities, and intentions, which are represented by the goal location of the human hand. For the above-mentioned problem scenario, the human intention is denoted by $g \in \mathcal{G}$, where $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$, and $g_i \in \mathbb{R}^3$ represents a 3-D location of an object on a table. The true intention g is one of the finite number of goal locations (target objects) $g_i's$. 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 position of the arm at a given time t, and $z_t \in \mathbb{R}^{24}$ represents the measurement obtained after filtering the camera sensor data (See Section VI for details) at a given time t. All locations are specified in the 3-D Cartesian space. It should be noted that the ANIE algorithm can also support g defined as a continuous variable. 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.

A. State Transition Model

The state transition model is described by the following equation:

$$\dot{x}_t = f_c^*(x_t, g) + \omega_t \quad (1)$$

where $\{\omega_t\} \sim \mathcal{N}(0, Q_c) \in \mathbb{R}^{24}$ is a zero-mean Gaussian random process with a covariance matrix $Q_c \in \mathbb{R}^{24\times 24}$, $f_c^*(x_t, g) : \mathbb{R}^{24} \times \mathbb{R}^3 \to \mathbb{R}^{24}$ is assumed to be a analytical function. The nonlinear function $f_c^*(x_t, g)$, defined in (1), is modeled using an NN given by

$$f_c^*(x_t, g) = W^T \sigma(U^T s_t) + \epsilon(s_t) \quad (2)$$

where $s_t = [[x_t^T, g^T], 1]^T \in \mathbb{R}^{28}$ is the input vector to the NN, $\sigma(U^T s_t) = [(1/(1 + \exp((-U^T s_t)_1))), (1/(1 + \exp((-U^T s_t)_2))), \ldots \ (1/(1 + \exp((-U^T s_t)_i))), \ldots (1/(1 + \exp((-U^T s_t)_{n_h})))]^T$ is the vector-sigmoid activation function, and $(U^T s_t)_i$ is the ith element of the vector $(U^T s_t)$; $U \in \mathbb{R}^{28\times n_h}$ and $W \in \mathbb{R}^{n_h\times 24}$ are the bounded constant weight matrices, $\epsilon(s_t) \in \mathbb{R}^{24}$ is the function reconstruction error, and $n_h \in \mathbb{Z}^+$ is the number of neurons in the hidden layer of the NN.

B. Brief Review of Offline Model Training

The training of the NN is done using the data consisting of the human arm's joint locations, joint velocities, and joint accelerations along with the intended target locations. The NN is trained using Bayesian regularization [42]. The objective function used to train an NN using Bayesian regularization is given by $J(U, W) = K_\alpha E_D + K_\beta E_W$, where $E_D = \sum_i \|y_i - a_i\|_2^2$ is the sum of squared errors, y_i is the target output, a_i is the network's output, E_W 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 reducing the weight sizes, respectively. Details pertaining to gathering training data from human subjects are described in Section VI.

C. Measurement Model

The measurements of human skeleton's joint positions are obtained using a camera sensor. The human skeleton is defined using 20 joints. The measurements are obtained in the camera's reference frame. Let $p^c = (x^c, y^c, z^c)^T$ be a point in the camera reference frame and $p^r = (x^r, y^r, z^r)^T$ be a point in the robot reference frame. The points p^c and p^r are related by

$$p^c = R_r^c p^r + T_r^c \quad (3)$$

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's reference frame. The camera sensor measures the 3-D locations of the skeleton's joints. The raw position measurements obtained from the camera sensor are fed to a Kalman filter, such as the one in [43], to obtain the position and velocity estimates, which are used as measurements in the intention inference algorithm. Design and implementation details of the Kalman filter can be found in Appendix A.

The measurement model is given by

$$y_t = h(x_t) + \upsilon_t \quad (4)$$

where $h(x_t) = Hx_t + b$, $b = [[T_r^c]^T, [T_r^c]^T, [T_r^c]^T, [T_r^c]^T, 0_{1\times 12}]^T$, and $H = \text{diag}\{R_r^c, R_r^c, \ldots, R_r^c\} \in \mathbb{R}^{24\times 24}$ is a block diagonal matrix, and $\{\upsilon_t\} \sim \mathcal{N}(0, \Sigma_z) \in \mathbb{R}^{24}$ is a zero-mean Gaussian noise with a covariance matrix $\Sigma_z \in \mathbb{R}^{24\times 24}$. The measurement noise $\{\upsilon_t\}$ is assumed to be independent of the process noise $\{\omega_t\}$ defined in (1). The measurement model of the shifted measurement vector[1] $z_t = y_t - b$ at time t is given by

$$z_t = Hx_t + \upsilon_t. \quad (5)$$

[1]Note that b is a known constant.

IV. Intention Inference

Our approximate E-M algorithm extends the work in [9] to the state transition models learned using NNs. Once the NN model is trained, the intention g can be inferred iteratively as new measurements become available. The E-M algorithm requires the state transition model to be in the discrete form. The state transition model defined in (1) is discretized using first-order Euler approximation yielding

$$x_t = f(x_{t-1}, g) + \omega_t T_s \tag{6}$$

where $f(x_{t-1}, g) = x_{t-1} + W^T \sigma(U^T s_{t-1}) T_s$, and T_s is the sampling period. In order to infer intention, the posterior probability of Z_T given the intention g is maximized using a maximum-likelihood criterion, where $Z_T = z_{1:T}$[2] is a set of observations from time $t = 1$ to $t = T$. The process noise of the discretized system in (6) is given by $Q = T_s^2 Q_c$. The log-likelihood function of the intention g is given by

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

which can be obtained after marginalizing the joint distribution $p(X_T, Z_T|g)$ over X_T, where $X_T = x_{1:T}$ is a collective representation of states from time $t = 1$ to $t = T$. In general, analytically evaluating this integral is extremely difficult. The E-M algorithm and other approximation techniques based on particle filtering are used to circumvent this problem. In this paper, an approximate E-M algorithm is used with modifications for handling state transition models trained using the NN. Using the fact that $\mathbb{E}_{X_T}\{\log[p(Z_T|g)]|Z_T, \hat{g}_t\} = \log p(Z_T|g)$, the log-likelihood defined in (7) is decomposed in the following way:

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

where $\mathbf{Q}(g, \hat{g}_t) = \mathbb{E}_{X_T}\{\log[p(Z_T, X_T|g)]|Z_T, \hat{g}_t\}$ is the expected value of the complete data log-likelihood, given all the measurements and intentions, $\mathbf{H}(g, \hat{g}_t) = \mathbb{E}_{X_T}\{\log[p(X_T|Z_T, g)]|Z_T, \hat{g}_t\}$, $\mathbb{E}_{X_T}(\cdot)$ is the expectation operator, and $\hat{g}_t$ is the estimate of g at time t. It can be shown using Jensen's inequality that $\mathbf{H}(g, \hat{g}_t) \le \mathbf{H}(\hat{g}_t, \hat{g}_t)$ [44]. Thus, in order to iteratively increase the log-likelihood, g has to be chosen such that $\mathbf{Q}(g, \hat{g}_t) \ge \mathbf{Q}(\hat{g}_t, \hat{g}_t)$. The **E**-Step involves the computation of the auxiliary function $\mathbf{Q}(g, \hat{g}_t)$ given the observations Z_T and the current estimate of the intention $\hat{g}_t$. The **M**-Step involves the computation of the next intention estimate $\hat{g}_{t+1}$ by finding the value of g that maximizes $\mathbf{Q}(g, \hat{g}_t)$.

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

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

In the case of $\{v_t\}$ and $\{\omega_t\}$ being Gaussian, V_0 and $V_t(x_t, x_{t-1}, g)$ are given by

$$\begin{aligned} V_0 &= \log[p(x_0|g)] = \log[p(x_0)] \\ &= \text{const} - \frac{1}{2}\log[|P_0|] - \frac{1}{2}(x_0 - \mu_0)^T P_o^{-1}(x_0 - \mu_0) \end{aligned}$$

$$\begin{aligned} &V_t(x_t, x_{t-1}, g) \\ &\quad = \log[p(z_t|x_t)] + \log[p(x_t|x_{t-1}, g)] \end{aligned} \tag{10}$$

where μ_0 and P_0 are the initial state mean and covariance, and $|\cdot|$ is the determinant operator

$$\begin{aligned} \log[p(z_t|x_t)] = &-\frac{1}{2}\log[|\Sigma_z|] \\ &- \frac{1}{2}\left\{(z_t - h(x_t))^T \Sigma_z^{-1}(z_t - h(x_t))\right\} \end{aligned} \tag{11}$$

$$\begin{aligned} \log[p(x_t|x_{t-1}, g)] = &-\frac{1}{2}\log[|Q|] \\ &- \frac{1}{2}\{(x_t - f(x_{t-1}, g))^T \\ &\qquad \times Q^{-1}(x_t - f(x_{t-1}, g))\}. \end{aligned} \tag{12}$$

Note that in (10), $\log[p(z_t|x_t, g)]$ is replaced by $\log[p(z_t|x_t)]$. This is because, in (4), the measurement z_t does not depend on the intention g. When attempting to optimize (10), the main difficulty arises because of the nonlinearity of the state transition model. The nonlinear state transition model is represented by an NN in our case. In order to compute the expectation of the log-likelihood in (12), the expression given in the second line of (12) (terms inside the curly brackets) is linearized about $\bar{x}_t$ and $\bar{x}_{t-1}$ using the Taylor series expansion. In practice, the points of linearization $\{\bar{x}_t\}$ are obtained from the measurements $\{z_t\}$ by ignoring the measurement noise and inverting the measurement function given in (5). Let $\tilde{V}_t = (x_t - f(x_{t-1}, g))^T Q^{-1}(x_t - f(x_{t-1}, g))$, and the Taylor series expansion of $\tilde{V}_t$ is given by

$$\begin{aligned} \tilde{V}_t \approx\ & \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}) + \left[\frac{\partial \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t}\right]^T [x_t - \bar{x}_t] \\ &+ \left[\frac{\partial \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_{t-1}}\right]^T [x_{t-1} - \bar{x}_{t-1}] \\ &+ \frac{1}{2}[x_t - \bar{x}_t]^T \frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t \partial x_t}[x_t - \bar{x}_t] \\ &+ \frac{1}{2}[x_{t-1} - \bar{x}_{t-1}]^T \frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_{t-1} \partial x_{t-1}}[x_{t-1} - \bar{x}_{t-1}] \\ &+ \frac{1}{2}[x_t - \bar{x}_t]^T \frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t \partial x_{t-1}}[x_{t-1} - \bar{x}_{t-1}] + \cdots. \end{aligned} \tag{13}$$

The derivatives of $\tilde{V}_t$ are given by the following equations:

$$\frac{\partial \tilde{V}_t}{\partial x_t} = (Q^{-1} + Q^{-T})(x_t - f(x_{t-1}, g)) \tag{14}$$

$$\frac{\partial \tilde{V}_t}{\partial (x_{t-1})_i} = \left[\frac{\partial \tilde{V}}{\partial f}\right]^T \frac{\partial f}{\partial (x_{t-1})_i} \tag{15}$$

$$\frac{\partial^2 \tilde{V}_t}{\partial x_t \partial x_t} = Q^{-1} + Q^{-T} \tag{16}$$

$$\frac{\partial^2 \tilde{V}_t}{\partial x_t \partial x_{t-1}} = -(Q^{-1} + Q^{-T})\left[\frac{\partial f}{\partial x_{t-1}}\right] \tag{17}$$

[2] T is not fixed and could be different for training and testing data

$$\frac{\partial^2 \tilde{V}_t}{\partial (x_{t-1})_i \partial (x_{t-1})_j} = \left[\frac{\partial^2 \tilde{V}}{\partial f (\partial x_{t-1})_i}\right]^T \frac{\partial f}{\partial (x_{t-1})_j} + \frac{\partial^2 f}{\partial (x_{t-1})_j \partial (x_{t-1})_i}\left[\frac{\partial \tilde{V}}{\partial f}\right] \tag{18}$$

where $[\partial \tilde{V}/\partial f] = -[Q^{-1} + Q^{-T}][x_t - f(x_{t-1}, g)]$ and $[(\partial^2 \tilde{V})/(\partial f(\partial x_{t-1})_i)] = [Q^{-1}+Q^{-T}]^T(\partial f/(\partial (x_{t-1})_i))$. Note that $(\partial f/(\partial x_{t-1}))$ is the submatrix of the Jacobian of the NN that can be obtained by ignoring the rows pertaining to $(\partial f/\partial g)$. Thus, the Jacobian $(\partial f/(\partial x_{t-1}))$ can be derived by taking the first n columns of $(\partial f/\partial s_{t-1})$, where n is the number of states. The Hessian $((\partial^2 f)/(\partial (x_t)\partial (x_t)))$ can be derived in a similar fashion. The analytical expressions for the Jacobian and Hessian are provided in Appendix B. Using (13)–(18), the expectation in (9) can be written as

$$\begin{aligned}
&\mathbf{Q}(g, \hat{g}_t) \\
&= -\frac{1}{2}\log[|P_0|] - \frac{T}{2}\log[|\Sigma_z|] - \frac{T}{2}\log[|Q|] \\
&\quad - \frac{1}{2}\mathrm{tr}\{P_0(\hat{P}_0 + (\hat{x}_0 - \mu_0)(\hat{x}_0 - \mu_0)^T)\} \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\big\{\Sigma_z^{-1}([z_t - H\hat{x}_t][z_t - H\hat{x}_t]^T + H\hat{P}_t H^T)\big\} \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}\tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g) \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}\left[\left[\frac{\partial \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t}\right]^T [\hat{x}_t - \bar{x}_t]\right] \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}\left[\left[\frac{\partial \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_{t-1}}\right]^T [\hat{x}_{t-1} - \bar{x}_{t-1}]\right] \\
&\quad - \frac{1}{4}\sum_{t=1}^{T}\mathrm{tr}\left\{\frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t \partial x_t}\right. \\
&\qquad\qquad \left. \times (\hat{P}_t + [\hat{x}_t - \bar{x}_t][\hat{x}_t - \bar{x}_t]^T)\right\} \\
&\quad - \frac{1}{4}\sum_{t=1}^{T}\mathrm{tr}\left\{\frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_{t-1} \partial x_{t-1}}\right. \\
&\qquad\qquad \left. \times (\hat{P}_{t-1} + [\hat{x}_{t-1} - \bar{x}_{t-1}][\hat{x}_{t-1} - \bar{x}_{t-1}]^T)\right\} \\
&\quad - \frac{1}{4}\sum_{t=1}^{T}\mathrm{tr}\left\{\frac{\partial^2 \tilde{V}_t(\bar{x}_t, \bar{x}_{t-1}, g)}{\partial x_t \partial x_{t-1}}\right. \\
&\qquad\qquad \left. \times (\hat{P}_{t,t-1} + [\hat{x}_t - \bar{x}_t][\hat{x}_{t-1} - \bar{x}_{t-1}]^T)\right\} - \cdots
\end{aligned} \tag{19}$$

where $\hat{x}_t$ and $\hat{P}_t$ are the state estimate and its covariance, respectively, $\hat{x}_0$ and $\hat{P}_0$ are their initial values, and $\hat{P}_{t,t-1}$ is the cross covariance of the state estimates at times t and $t-1$. The state estimate $\hat{x}_t$ and the covariances $\hat{P}_t$ and $\hat{P}_{t,t-1}$ are obtained by using an extended Kalman filter (EKF). In order to linearize the transition model for the EKF at the current time step t, the state estimate $\hat{x}_{t-1}$ values from the previous time step are used as the point of linearization. The equation in (19) can be written in an iterative form to calculate the value of the $\mathbf{Q}$ function at every iteration.

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

$$\hat{g}_{t+1} = \arg\max_{g} \mathbf{Q}(g, \hat{g}_t). \tag{20}$$

This step can be carried out in two different ways, viz., numerical optimization or direct evaluation as described in the following.

A. Numerical Optimization of the Q Function

One way to maximize the $\mathbf{Q}$ function is to use the GradEM algorithm [45], where the first few iterations of Newton's algorithm are used for the M-step. This method involves optimizing the $\mathbf{Q}$ function over $\mathbb{R}^3$. The update equation for $\hat{g}$, through GradEM algorithm, is given by

$$\hat{g}_{k+1} = \hat{g}_k - \mathcal{H}(\mathbf{Q})^{-1}\Delta(\mathbf{Q}) \tag{21}$$

where $\hat{g}_k$ is the estimate of g at the kth iteration of the optimization algorithm, and $\mathcal{H}(\mathbf{Q})$ and $\Delta(\mathbf{Q})$ are the Hessian and Gradient of the $\mathbf{Q}$ function, respectively. Note that numerical optimization methods need to run at every time step of the E-M algorithm. In real-time implementations, the number of iterations for the optimization in (21) could be chosen based on computational capabilities. This is similar to using the first iteration of Newton's method. The Hessian of the $\mathbf{Q}$ function can be numerically approximated and the analytical expression for the gradient of the $\mathbf{Q}$ function is provided in Appendix C.

B. Direct Evaluation of the Q Function

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

$$\hat{g}_{t+1} = \arg\max_{g\in\mathcal{G}} \mathbf{Q}(g, \hat{g}_t). \tag{22}$$

This method involving direct evaluation of the $\mathbf{Q}$ function is possible if all possible goal locations are known *a priori* and are finite. This is not an unusual case in the context of the problem scenario described in Section II. Image processing algorithms can be used to detect the objects on the workbench and extract the 3-D locations using the camera data.

V. Online Model Learning

This section describes the online learning algorithm used to update the weights of the NN model. The online learning of the NN weights is important to make the inference framework robust to variations in starting arm positions and various motion trajectories taken by different people. The NN weights are updated iteratively as new data become available. To this end, a state identifier is developed that computes an estimate of the state derivative based on the current state estimates obtained from the EKF and the current NN weights. The identifier state error is computed based on its estimate and measurement. The error in the state is used to update the NN weights for the next time instance. The identifier uses a robust integral of the signum of the error (RISE) feedback [46] to ensure asymptotic convergence of the state estimates and their derivatives to the true values. The weight update equations are computed using Lyapunov-based stability analysis.

The state identifier is given by

$$\dot{\hat{x}}_{id_t} = \hat{W}_t^T \sigma\left(\hat{U}_t^T \hat{s}_t\right) + \mu_t \tag{23}$$

where $\hat{U}_t \in \mathbb{R}^{28\times n_h}$, $\hat{W}_t \in \mathbb{R}^{n_h\times 24}$, $\hat{s}_t = [[\hat{x}_{id_t}^T, \hat{g}_t^T], 1]^T \in \mathbb{R}^{28}$, $\hat{g}_t \in \mathbb{R}^3$ is the current estimate of g from the E-M algorithm, $\hat{x}_{id_t} \in \mathbb{R}^{24}$ is the current identifier state, and $\mu_t \in \mathbb{R}^{24}$ is the RISE feedback term defined as $\mu_t = k\tilde{x}_t - k\tilde{x}_0 + \nu_t$, where $\tilde{x}_t = x_t - \hat{x}_{id_t}$ is the state identification error at time t, and $\nu_t \in \mathbb{R}^{24}$ is the Filippov generalized solution [12] to the following differential equation:

$$\dot{\nu}_t = (k\alpha + \gamma)\tilde{x}_t + \beta_1 \mathrm{sgn}(\tilde{x}_t); \quad \nu_0 = 0 \tag{24}$$

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

$$\begin{aligned} \dot{\hat{W}}_t &= \mathrm{proj}\left(\Gamma_w \hat{\sigma}_t' \hat{U}_{x_t}^T \dot{\hat{x}}_{id_t} \tilde{x}_t^T\right) \\ \dot{\hat{U}}_{x_t} &= \mathrm{proj}\left(\Gamma_{u_x} \dot{\hat{x}}_{id_t} \tilde{x}_t^T \hat{W}_t^T \hat{\sigma}_t'\right) \\ \dot{\hat{U}}_{g_t} &= \mathrm{proj}\left(\Gamma_{u_g} \dot{\hat{g}}_t \tilde{x}_t^T \hat{W}_t^T \hat{\sigma}_t'\right) \end{aligned} \tag{25}$$

where $\mathrm{proj}(\cdot)$ is a projection operator defined in [47], $\hat{U}_{x_t}$ and $\hat{U}_{g_t}$ are the submatrices of $\hat{U}_t$ formed by taking the rows corresponding to $\hat{x}_{id_t}$ and $\hat{g}_t$, respectively, $\hat{\sigma}_t'$ is the first-order derivative of the sigmoid function with respect to its input $\hat{U}^T \hat{s}_t$, and Γ_w, Γ_{u_x}, and Γ_{u_g} are constant weighting matrices of appropriate dimensions. In the online learning algorithm, $\hat{g}_t$ from the E-M algorithm is used. Hence, for the online learning step, $\hat{g}_t$ is assumed to be a known parameter. The derivative of the intention estimate $\dot{\hat{g}}_t$ is computed using the finite difference method. It is shown in Appendix D that the identifier defined in (23) along with the update equations defined in (25) is asymptotically stable, and the state identification error converges to zero. The learning algorithm is described in Algorithm 1.

Algorithm 1: Intention Inference With Online Model Update Algorithm

Obtain demonstrations;
Using a Kalman filter, obtain position, velocity, and acceleration estimates for the demonstrations;
Label the training data based on the correspoding goal locations;
Learn the NN model defined in (2) using the demonstrations;
Obtain test data from a new subject using camera;
Filter the position measurements of the test data using a Kalman filter to obtain position and velocity estimates and use them as measurentments Z_T;
Initialize $\hat{x}_0$, $\hat{P}_0$, $\hat{x}_{id_0}$, and $\hat{g}_0$;
Define the parameters of the system: μ_0, P_0, Q, and Σ_z;
Define the gains for the online update algorithm: $k, \alpha, \gamma, \beta_1, \Gamma_W, \Gamma_{U_x}$, and Γ_{U_g};
while *data for the current time step is present* **do**
    Read the current measurement z_t;
    E-step:
    Using the current NN model and the previous intention estimate $\hat{g}_{t-1}$, compute $\hat{x}_t$, $\hat{P}_t$, $\hat{P}_{t,t-1}$ using the EKF;
    M-step:
    if *Numerical optimization* **then**
        Using the estimates obtained from the E-step, compute $\hat{g}_t$ by iteratively maximizing the **Q** function defined in (19) over $\mathbb{R}^3$ using (21);
    end
    if *Direct evaluation* **then**
        Using the estimates obtained from the E-step, compute $\hat{g}_t$ by maximizing the **Q** function defined in (19) over $\mathcal{G}$ using (22);
    end
    Online model update:
    Using the intention estimate $\hat{g}_t$ from the M-step, compute the identifier output $\dot{\hat{x}}_{id_t}$ using (23);
    Update the current NN model by changing the weights according to the adaptation laws given in (25);
end

VI. Experimental Results

In order to validate the ANIE algorithm, four different experiments are conducted using data obtained from a Microsoft Kinect (3-D camera) for Windows VI in the Robotics and Controls laboratory at UConn and using the publicly available Cornell's CAD-120 data set [14]. The joint position data obtained from the subjects are preprocessed to obtain the velocity and acceleration estimates using a Kalman filter (See Appendix A). Each demonstration is labeled based on the ground truth goal location in the filtered data. Note that the goal location labeling is required and done only for demonstrations that are a part of the training data. The measurements are processed on a standard desktop computer running Intel i3 processor and 8 GB of memory. The algorithm is coded in MATLAB 2014a. The average computation time for processing each frame and giving out an estimate is 0.05 sec. The average computation time is computed over a sample trajectory consisting of 78 frames. In the first three experiments, the success of a test is determined based on two criteria: 1) a test is considered successful, if the algorithm converges to the true intention in half the time it took the subject to reach the goal location (SC1) and 2) a test is considered successful, if the algorithm converges to the true intention before the subject's hand reaches a sphere around the goal location with radius equal to the half of the straight line distance between the start and the goal locations (SC2). The performance of the algorithm is also evaluated based on the percentage of tests with correctly inferred intentions with respect to the percentage of trajectory observed. The aim of each experimental study is described in the following.

1) The first experiment is conducted to show that the learned NN model can be used to infer the intention from new data with different characteristics, such as the

starting positions of the arm, motion profiles, clutter, and number of target locations.

2) The second experiment is conducted to test the ANIE algorithm's ability to adapt to the motions of new subjects. The test trajectories are collected from four different subjects, whose data are not used to train the NN offline.
3) The third experiment is conducted to validate the ANIE algorithm on an independent data set. The Cornell's CAD-120 data set is used for this purpose.
4) The fourth experiment is conducted to evaluate the ability of the ANIE algorithm to predict subtask labels in the Cornell's CAD-120 data set.

The ANIE algorithm is compared with the two-layer unsupervised GMM algorithm [13] and the Euclidean distance-based algorithm in the first three experiments, and I-LQR and ATCRF algorithms in the fourth experiment. At every iteration, for the Euclidean distance-based algorithm, the goal location that has the least Euclidean distance to the reaching hand of the tracked human skeleton is chosen as an intention estimate.[3] The comparisons are made based on the intention inference accuracy, the average time of inference, and the trajectory prediction accuracy. The trajectory prediction accuracy is evaluated using the dynamic time warping (DTW) distance [48] between the hand trajectories and the trajectories predicted by the algorithms after observing different percentages of the trajectories. In Experiment 4, performances are evaluated using accuracy, precision, and recall. A video containing some of the results presented in this section can be found at https://goo.gl/wgMhqN.

A. Experiment 1

In this experiment, the training and testing data are collected from the same person (Subject 1). A set of 130 arm motion trajectories reaching for different objects are recorded. The starting positions of the human arm and the possible goal locations of the test trajectories are different from each other. Some of the trajectories involved reaching objects that are randomly placed close to each other in a cluttered manner. Some of the recorded arm motions consisted of the subject initially moving the hand close to an object, but finally reaching another object. Each trajectory contained roughly 100–125 frames of skeletal data. A set of ten of these trajectories are used for training an NN. The number of neurons in the hidden layer is empirically chosen to be 50. Once the NN is trained, the test data (the remaining 120 trajectories) are used as measurements to infer the underlying intentions. 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 intention that led to the maximum **Q** value (direct evaluation method). The initial mean of the state μ_0 is assumed to be a zero vector. The initial state covariance P_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 (23) and (25) are selected to be $k = 20$, $\alpha = 5$, $\gamma = 50$, and $\beta_1 = 1.25$, and the adaptation gains are chosen to be $\Gamma_W = 0.1I_{50\times50}$, $\Gamma_{U_x} = 0.2I_{24\times24}$, and $\Gamma_{U_g} = 0.2I_{3\times3}$. The state estimates are initialized to the same value as the first measurement z_1.

[3]The Euclidean distance-based method does not consider momentum information, which can potentially improve its prediction results.

TABLE I

TEST STATISTICS OF EXPERIMENT 1

	Euclidean distance-based	Unsupervised GMM [13]	ANIE
No. of training sets	0	0	10
No. of test sets	120	120	120
No. of successful tests (SC1)	71	96	108
No. of successful tests (SC2)	65	91	106
Average time of inference (sec.)	1.12	0.87	0.71

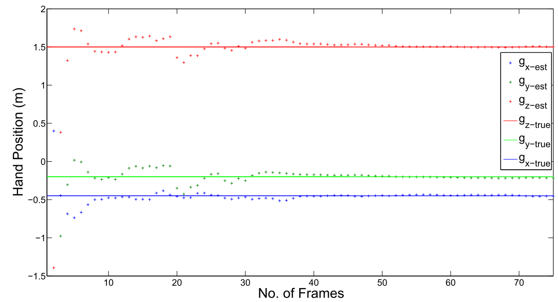

Fig. 2. Intention inference by numerical optimization of **Q** function over $\mathbb{R}^3$ for Subject 1.

The sampling time for discretization is $\frac{1}{30}$ s. For a set of 20 trajectories, intentions are also inferred by numerical optimization of the **Q** function, where g is considered to be a continuous variable. The intention estimate $\hat{g}$ is randomly initialized to one of the four possible intentions. A sample result of intention estimation using numerical optimization is shown in Fig. 2. The numerical optimization algorithm at each time step is run for five iterations. For different sample tests, Figs. 3–5 show the sequences of images at various time instances and indicate the corresponding intention estimates for various scenarios considered for Experiment 1. At every frame, the trajectory of the subject's hand is predicted by integrating the model forward in time and is overlaid in Fig. 4. The test statistics of Experiment 1 are given in Table I. The quadratic deviation of each of the trajectories from corresponding straight lines between the start and the goal locations is computed. The quadratic deviation between any two sequences $l = [l_1, l_2, \ldots, l_N]$ and $m = [m_1, m_2, \ldots, m_N]$, where $l_i, m_i \in \mathbb{R}^n$, is given by $D_q = \sum_i ||l_i - m_i||_2^2$. It is found that the average quadratic deviation of the trajectories in the success cases is 18.51 m^2, while the failure cases have an average quadratic deviation of 31.67 m^2.

B. Experiment 2

The second experiment uses data collected from Subjects 2–5, while only the data collected from Subject 1 are used for training the NN offline. Intention is inferred using the online learning algorithm to show that the online learning algorithm can adapt to novel scenarios. A set of 134 arm trajectories reaching for different objects are recorded for this experiment. The set of trajectories consists of 24

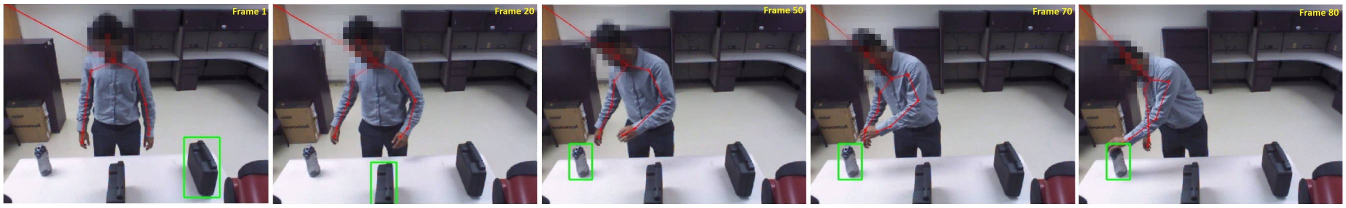

Fig. 3. Image sequence showing skeletal tracking (red line) and online inference of the goal location (green box). The training and testing data, collected from the same person, are mutually exclusive and have different initial conditions.

Fig. 4. Image sequence showing the comparison between the ANIE algorithm and the Euclidean distance-based algorithm. The inferred intention of the ANIE algorithm in each frame is marked by red solid box, while that of the Euclidean distance-based algorithm is marked by a yellow dashed box. The trajectories predicted by the NN at the specified frames are also overlaid (red dashed line).

Fig. 5. Image sequence showing the intention inferred by the ANIE algorithm (red solid box) and the Euclidean distance-based technique (yellow dashed box). Four objects are placed in new locations close to each other in a cluttered manner and no new training data are used.

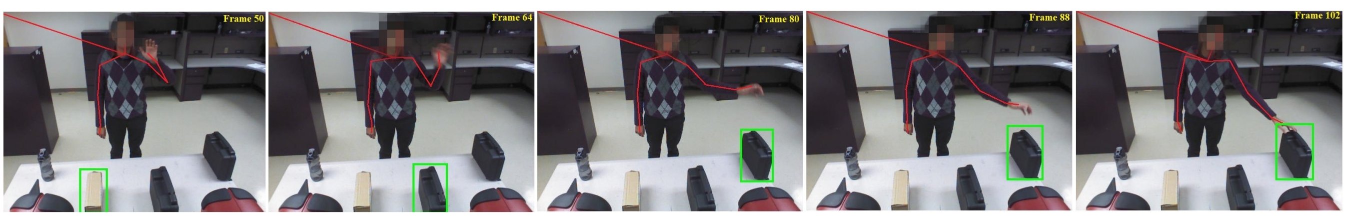

Fig. 6. Image sequence showing skeletal tracking (red line) and online inference of intention, i.e., the goal location (green box) with online model update. Data from Subject 2, with different initial conditions and motion profiles, are used to test the ANIE algorithm.

reaching trajectories from Subject 2, 18 reaching trajectories from Subject 3, 12 reaching trajectories from Subject 4, and 80 trajectories from a collaborative desk drawer assembly task performed by Subjects 4 and 5. The gains for the online learning algorithm defined in (23) and (25) are selected to be $k = 20$, $\alpha = 5$, $\gamma = 50$, and $\beta_1 = 1.25$, and the adaptation gains are chosen to be $\Gamma_W = 0.1 I_{50\times 50}$, $\Gamma_{U_x} = 0.2 I_{24\times 24}$, and $\Gamma_{U_g} = 0.2 I_{3\times 3}$. The initial state covariance P_0, the process noise covariance Q, and the measurement noise covariance Σ_z are selected to be $0.2 I_{24\times 24}$, $0.1 I_{24\times 24}$, and $0.2 I_{24\times 24}$, respectively. The $\mathbf{Q}$ function is optimized using the direct evaluation method. The test statistics of Experiment 2 are given in Table II. In Figs. 6–8, sequences of images showing the inferred intentions are given. The quadratic deviation of each of the trajectories from corresponding straight lines between the start and the goal locations are computed. It is found that the average quadratic deviation of the trajectories in the failure cases is 35.4 m^2, while that in the successful cases is 18.67 m^2.

TABLE II
TEST STATISTICS OF EXPERIMENT 2

	Euclidean distance-based	Unsupervised GMM [13]	ANIE
No. of test sets	134	134	134
No. of successful tests (SC1)	91	115	121
No. of successful tests (SC2)	88	113	120
Average time of inference (sec.)	1.23	0.81	0.75

C. Experiment 3

A set of 20 sequences (five sequences from each of the four subjects) with reaching motions are randomly chosen from Cornell's CAD-120 data set. Due to the fact that the CAD-120 data set has only three joints (shoulder, elbow, and hand) for each arm, the state vector is redefined as $x_t \in \mathbb{R}^{18}$ by removing the wrist joint position and velocity from the

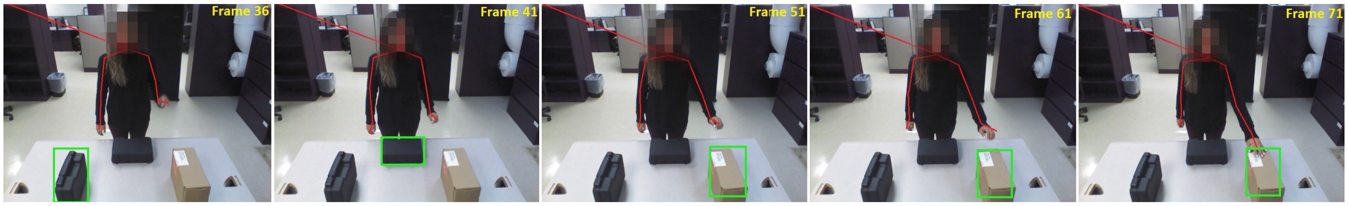

Fig. 7. Image sequence showing skeletal tracking (red line) and online inference of intention (green box) with online model update (the motion starts from frame 36). Data from Subject 3, with new and different goal locations, are used to test the ANIE algorithm.

Fig. 8. Image sequence showing the inferred intentions of Subject 4 (yellow dashed box) and Subject 5 (red dashed box) as they are performing a collaborative desk draw assembly task.

Fig. 9. Image sequence showing the intention inferred by the ANIE algorithm (red solid box) using Cornell's CAD-120 data set. No part of the CAD-120 data set is used to train the NN offline.

TABLE III
TEST STATISTICS OF EXPERIMENT 3

	Euclidean distance-based	Unsupervised GMM [13]	ANIE
No. of test sets	20	20	20
No. of successful tests (SC1)	12	17	18
No. of successful tests (SC2)	11	17	18
Average time of inference (sec.)	0.92	0.69	0.65

original state definition. An NN is trained using the same set of ten trajectories collected from Subject 1 for Experiment 1 with the measurements of the wrist joint removed. However, no part of Cornell's CAD-120 data set is used to train the NN offline. The gains for the online learning algorithm defined in (23) and (25) are selected to be $k = 20$, $\alpha = 5$, $\gamma = 50$, and $\beta_1 = 1.25$, and the adaptation gains are chosen to be $\Gamma_W = 0.1I_{35\times35}$, $\Gamma_{U_x} = 0.2I_{18\times18}$, and $\Gamma_{U_g} = 0.2I_{3\times3}$. The initial state covariance P_0, the process noise covariance Q, and the measurement noise covariance Σ_z are selected to be $0.2I_{18\times18}$, $0.1I_{18\times18}$, and $0.2I_{18\times18}$, respectively. The **Q** function is optimized using the direct evaluation method. The possible goal locations are chosen to be the objects on the table for each sequence. In Fig. 9, a sequence of images overlaid with the inferred intentions is shown. The ANIE algorithm is able to infer the correct intention in 18 tests according to both SC1 and SC2. The test statistics of Experiment 3 are given in Table III.

In Fig. 10, the percentage of tests, where the intention is correctly inferred, is shown as a function of time for the first three experiments. In order to evaluate the importance of online learning, the intention inference accuracies of the ANIE algorithm are compared with and without the online learning component. The results are shown in Fig. 11. In addition to the intention inference accuracy, trajectory prediction accuracy of the ANIE algorithm is evaluated in the first three experiments. This evaluation is realized by computing the DTW distance [48] between the hand trajectories predicted by the NN and the corresponding true trajectories for all the test trajectories. The DTW distance is computed at various instances in time based on the percentage of trajectory that is observed (20%, 40%, 60%, and 80%). The same metric is also computed for the unsupervised GMM algorithm. The predicted trajectories are computed by forward propagating the models in time until the corresponding goal locations are reached. In Fig. 12, the relationship between the average DTW distance and the percentage of trajectory observed is shown for the first three experiments.

D. Experiment 4

The fourth experiment is conducted on the CAD-120 data set. In this experiment, the ANIE algorithm is used for labeling subactivities and compared with the I-LQR [15] and the ATCRF [14] algorithms. For the purpose of comparison, modifications to the CAD-120 data set are made following the steps described in [15]. The original CAD-120 data set has ten subactivities: *reaching, moving, pouring, eating, drinking, opening, placing, closing, cleaning*, and *null*. Each subactivity is considered to be associated with a goal location in order to map a goal location predicted by the ANIE algorithm to a subactivity label. The *moving* subactivity is considered to be

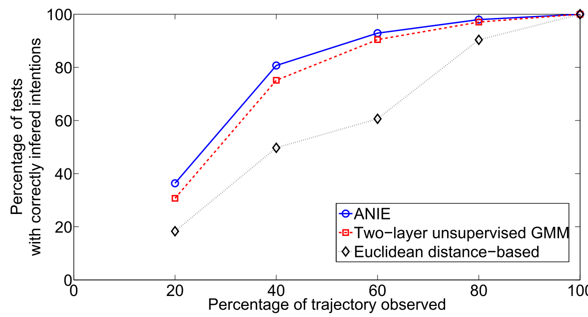

Fig. 10. Percentage of tests with correctly inferred intention as a function of the percentage of trajectory observed over 274 trajectories from Experiments 1–3.

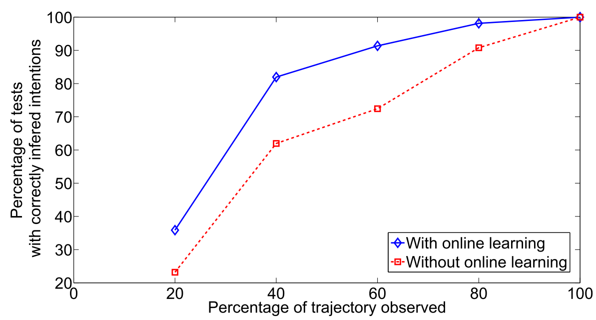

Fig. 11. Percentage of tests with correctly inferred intention as a function of the percentage of trajectory observed for the ANIE algorithm with online learning, and without online learning over 274 trajectories from Experiments 1–3.

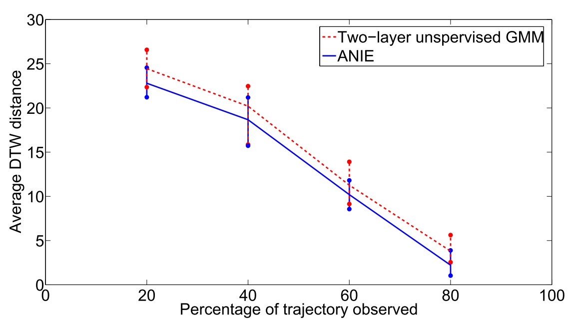

Fig. 12. Average DTW distance as a function of percentage of trajectories observed over 274 trajectories from Experiments 1–3.

a part of the succeeding subactivity. For instance, if *moving* proceeds *pouring*, the goal of the moving subactivity will be the location above the container that is being poured into. Hence the *moving* subactivity is merged with the subactivity it proceeds. The *null* subactivity is ignored, since it is not driven by a goal location. The *opening* subactivity is divided into two subactivities, namely, *opening the microwave* and *opening a jar*, since they have different goal locations. These modifications result in a total of nine subactivities. The goal locations of *eating* and *drinking* subactivities are chosen to be the head joint of the tracked human skeleton. The goal locations of the other subactivities are computed by averaging over the observed goal locations of the respective subactivity in the training set. More details about the modifications and setup can be found in [15]. The data set is randomly divided into testing and training set with 10% of trajectories being

TABLE IV
COMPARISON RESULTS FROM EXPERIMENT 4

	Accuracy	Macro Precision	Macro Recall
ANIE 20% sequence	58.21	40.34 ± 21.95	37.04 ± 27.67
ANIE 40% sequence	76.12	70.02 ± 26.28	63.22 ± 28.89
ANIE 60% sequence	92.54	97.57 ± 4.81	86.3 ± 13.56
ANIE 80% sequence	95.52	98.59 ± 0.28	92.22 ± 11.76
ANIE 100% sequence	100	100 ± 0.0	100 ± 0.0
I-LQR 20% sequence [15]	80.9	65.0 ± 3.1	77.3 ± 2.4
I-LQR 40% sequence [15]	82.5	73.4 ± 2.2	91.4 ± 0.6
I-LQR 60% sequence [15]	84.1	79.1 ± 2.5	94.2 ± 0.6
I-LQR 80% sequence [15]	90.4	87.5 ± 1.8	96.2 ± 0.3
I-LQR 100% sequence [15]	100	100 ± 0.0	100 ± 0.0
ATCRF 100% sequence [14]	86.0	84.2 ± 1.3	76.9 ± 2.6

in the test set. The gains of the online learning component are selected to be the same as Experiment 3. The ANIE algorithm's ability to classify the subactivities is evaluated at different percentages of trajectory that is observed (20%, 40%, 60%, 80%, and 100%). The comparison results are summarized in Table IV. The performance statistics of the I-LQR and the ATCRF algorithms reported in [15] are used for comparison.

VII. DISCUSSION

Four sets of experiments are conducted to evaluate the performance of the ANIE algorithm with real data collected using a camera sensor and the CAD-120 data set. In the first three experiments, the ANIE algorithm outperforms both the unsupervised GMM algorithm and the Euclidean distance-based approach. In the first experiment, the ANIE algorithm resulted in 12 (according to SC1) and 14 (according to SC2) unsuccessful tests out of 120 tests as opposed to the unsupervised GMM algorithm that resulted in 24 (according to SC1) and 29 (according to SC2) unsuccessful tests. The test data for Experiment 1 involve objects randomly placed close to each other in a cluttered manner and confusing trajectories that approach a certain location initially and then change course to ultimately reach a different location. The ANIE algorithm is still able to infer the correct intention ahead of time in most cases. The second experiment involved data collected from four new subjects. The ANIE algorithm resulted in only 13 (according to SC1) and 14 (according to SC2) unsuccessful tests out of 134 tests as opposed to the unsupervised GMM algorithm's 19 (according to SC1) and 21 (according to SC2) unsuccessful tests. In Experiment 3, conducted on the CAD-120 data set, the ANIE algorithm resulted in 2 (according to SC1 and SC2) unsuccessful tests out of 20 tests as opposed to 3 (according to SC1 and SC2) unsuccessful tests of the unsupervised GMM algorithm. In addition to intention inference accuracy, the ANIE algorithm also performs better in terms of trajectory prediction accuracy. The ANIE algorithm's average DTW distances, computed over the first three experiments, between the predicted and the true trajectories at the given time instances are 22.79, 18.66, 10.19, and 2.21, while those of the unsupervised GMM algorithm are 24.44, 20.20, 11.26, and 3.80.

For the first three experiments, the NN is trained using the data collected from Subject 1 of Experiment 1. Hence, it is challenging to learn mappings that are generalizable to new instances. The online learning component improves the performance of the ANIE algorithm for the novel cases. The initial hand locations of the testing data are considerably different from that of the training data. The average Euclidean distance between the initial hand locations of all 254 test trajectories and the average of the initial hand locations of the 10 trajectories of the training data is found to be 0.46 m. The experimental results show that the ANIE algorithm can be used in generic scenarios, where the subject and other characteristics are different than what the NN is trained for. In the first three experiments, the ANIE algorithm and the unsupervised GMM algorithm outperform the Euclidean distance-based method. This comparison points out the need for learning the dynamics of reaching motion. The Euclidean distance-based approach failed in many cases, where the objects are placed close to each other and where objects are placed on the way to reach the target object. In contrast, the ANIE algorithm and the unsupervised GMM algorithm are able to infer the true goal locations ahead of time in such instances by learning the new trajectories online. The failure cases (according to SC1 and SC2) of the ANIE algorithm in our experiments can be related to complicated motions that are not typical arm motions represented by the large average quadratic deviation of the arm trajectories from a straight line connecting the starting and end locations. The average quadratic deviations are found to be 18.51 m^2 (Experiment 1) and 18.51 m^2 (Experiment 2) for the successful tests compared with 31.67 m^2 (Experiment 1) and 35.4 m^2 (Experiment 2) for the unsuccessful tests. The nontypical arm motions are the outliers in the data representation to the NN.

In Experiment 4, the ANIE algorithm is shown to be capable of labeling the subtasks on the CAD-120 data set. The comparison results indicate that the ANIE and I-LQR algorithms perform better than the ATCRF algorithm. Furthermore, the I-LQR algorithm performs better than the ANIE algorithm when 20% and 40% of the trajectory is observed, while the ANIE algorithm does better after observing 60% and 80% of the trajectory. It is believed that since I-LQR algorithm is a maximum *a posteriori* estimator with heuristically designed priors, it works better for 20% and 40% observed trajectory cases. Whereas the online learning component of the ANIE algorithm is believed to be a reason for improved performance of ANIE algorithm for 60% and 80% of trajectory observed.

VIII. Conclusion

A new methodology called the ANIE is presented to infer human intentions denoted by the goal locations of reaching motions using an NN-based approximate E-M algorithm with online model learning. NNs are used to model the nonlinear human arm motion dynamics. An identifier-based online learning algorithm is developed to iteratively learn new motion dynamics as new measurements become available. The experimental results show that online learning can improve the intention inference results for new human subjects with different initial conditions, motion profiles, and goal locations. Comparison of the ANIE algorithm on Cornell's CAD-120 data set with unsupervised GMM and Euclidean distance-based approach shows better performance of ANIE algorithm. It is observed that the ANIE algorithm is capable of predicting subtask labels in the Cornell's CAD-120 data set. The labeling results are compared with the I-LQR and the ATCRF algorithms. A real-time implementation of the ANIE algorithm on the Baxter robot and sensor fusion strategies to take advantage of other cues, such as head pose and eye gaze, will be considered in future work.

Appendix A
Kalman Filter Implementation

The simple model of human motion from [43] is used to design and implement a standard Kalman filter. The state transition model is given by the Taylor series expansion for position, velocity, and acceleration along all the three axes: $X_{kf_{t+1}} = \mathbf{F}_{kf} X_{kf_t} + W_{kf_t}$, where $X_{kf_t} = [x_{kf_t}, \dot{x}_{kf_t}, \ddot{x}_{kf_t}, y_{kf_t}, \dot{y}_{kf_t}, \ddot{y}_{kf_t}, z_{kf_t}, \dot{z}_{kf_t}, \ddot{z}_{kf_t}]^T$ at time t, and

$$\mathbf{F} = \begin{bmatrix} B_1 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & B_1 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & B_1 \end{bmatrix}, \quad \mathbf{B_1} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

and W_{kf_t} is the GP noise with covariance Q_{kf_t} given by

$$Q_{kf_t} = q_{kf} \begin{bmatrix} B_2 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & B_2 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & B_2 \end{bmatrix}$$

$$\mathbf{B_2} = \begin{bmatrix} 1+T_s^2 & T_s & T_s^2 \\ T_s & 1+T_s^2 & T_s \\ T_s^2 & T_s & 1 \end{bmatrix}$$

where $q_{kf} = 0.02$ is the noise strength. The measurement model for the camera sensor is given by $Z_{kf_t} = H_{kf} X_{kf_t} + V_{kf_t}$, where

$$H_{kf} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

and V_{kf_t} is the zero-mean Gaussian measurement noise with covariance [43]

$$\Sigma_{kf} = \begin{bmatrix} 0.06 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.06 \end{bmatrix}$$

Given these models, a standard Kalman filter is used to obtain the state estimates $\hat{X}_{kf_t}$.

Appendix B
Analytical Jacobian and Hessian of the NN

The analytical Jacobian and Hessian of the trained NN can be derived to be $(\partial f/\partial s_{t-1}) = W^T((\partial\sigma(U^T s_{t-1}))/\partial s_{t-1}) = W^T\Sigma'(a)U^T$ and $((\partial^2 f)/(\partial((s_{t-1})_i)\partial((s_{t-1})_j))) = W^T[(\Sigma''(a)U_i^T)\cdot U_j^T]$, respectively, where $a = U^T s_{t-1}$; $\Sigma'(a)$ is a diagonal matrix with elements $((\partial\sigma(a_i))/(\partial a_i)) = \sigma(a_i)(1-\sigma(a_i))$; U_i and U_j are the ith and jth rows of the matrix U, respectively; $\Sigma''(a)$ is a diagonal matrix with elements $((\partial^2\sigma(a_i))/(\partial a_i^2)) = \sigma(a_i)(1-\sigma(a_i))(1-2\sigma(a_i))$; the product $(\Sigma''(a)U_i^T)\cdot U_j^T$ is a Hadamard product.

APPENDIX C
GRADIENT OF THE $\mathbf{Q}$ FUNCTION

The gradient of the $\mathbf{Q}$ function used in the optimization can be derived as shown in the following:

$$
\begin{aligned}
\nabla_{g_i}\mathbf{Q} &= -\frac{1}{2}\sum_{t=1}^{T}\left[\left[\frac{\partial\tilde{V}}{\partial f}\right]^T\frac{\partial f}{\partial g_i}\right] \\
&\quad+\frac{1}{2}\sum_{t=1}^{T}\left[\left[\frac{\partial f}{\partial g_i}\right]^T[Q^{-T}+Q^{-1}][\hat{x}_t-\bar{x}_t]\right] \\
&\quad-\frac{1}{2}\sum_{t=1}^{T}\left\{\left[\frac{\partial^2\tilde{V}}{\partial f\partial g_i}\right]^T\frac{\partial f}{\partial x_{t-1}}+\left[\frac{\partial\tilde{V}}{\partial f}\right]^T\right. \\
&\qquad\left.\times\frac{\partial^2 f}{\partial x_{t-1}\partial g_i}\right\}[\hat{x}_{t-1}-\bar{x}_{t-1}] \\
&\quad+\frac{1}{4}\sum_{t=1}^{T}\mathrm{tr}\left\{\left[[Q^{-1}+Q^{-T}]\frac{\partial^2 f}{\partial x_{t-1}\partial g}\right]\right. \\
&\qquad\left.\times\ [\hat{P}_{t,t-1}+[\hat{x}_t-\bar{x}_t][\hat{x}_{t-1}-\bar{x}_{t-1}]^T]\right\} \\
&\quad-\frac{1}{4}\sum_{t=1}^{T}\mathrm{tr}\left\{\left[\left[\frac{\partial f}{\partial x_{t-1}}\right]^T[Q^{-1}+Q^{-T}]\frac{\partial^2 f}{\partial x_{t-1}\partial g_i}\right.\right. \\
&\qquad+\left[\frac{\partial^2 f}{\partial x_{t-1}\partial g_i}\right]^T[Q^{-1}+Q^{-T}]\frac{\partial f}{\partial x_{t-1}} \\
&\qquad\left.+\frac{\partial^3 f}{\partial x_{t-1}\partial x_{t-1}^T\partial g_i}\frac{\partial\tilde{V}}{\partial f}+\frac{\partial^2 f}{\partial x_{t-1}\partial x_{t-1}^T}\frac{\partial^2\tilde{V}}{\partial f\partial g_i}\right] \\
&\qquad\left.\times\left[\hat{P}_{t-1}+[\hat{x}_{t-1}-\bar{x}_{t-1}][\hat{x}_{t-1}-\bar{x}_{t-1}]^T\right]\right\}.
\end{aligned}\tag{26}
$$

APPENDIX D
STABILITY ANALYSIS

Assumption 1: The ideal weights of the NN are bounded by known positive constants [49].

Assumption 2: The function reconstruction error $\epsilon(\cdot)$ and its derivatives with respect to its arguments are assumed to be bounded [49].

Remark 1: Assumption 2 can lead to conservative bounds on the approximation error and the respective partial derivatives. However, in practice, the bound could be decreased by increasing the number of neurons in the hidden layer and using the knowledge of how the reconstruction error changes with increasing the number of neurons in the hidden layer.

The identification error dynamics can be described by

$$\dot{\tilde{x}}_t = W^T\sigma(U^Ts_t)-\hat{W}_t^T\sigma(\hat{U}_t^T\hat{s}_t)+\epsilon(s_t)-\mu_t. \tag{27}$$

A filtered error is defined as

$$r_t \triangleq \dot{\tilde{x}}_t+\alpha\tilde{x}_t \tag{28}$$

and its derivative with respect to time is as follows:

$$
\begin{aligned}
\dot{r}_t &= W^T\sigma_t'U^T\dot{s}_t-\dot{\hat{W}}_t^T\sigma\left(\hat{U}_t^T\hat{s}_t\right)-\hat{W}_t^T\hat{\sigma}_t'\dot{\hat{U}}_t^T\hat{s}_t+\dot{\epsilon}(s_t) \\
&\quad-\hat{W}_t^T\hat{\sigma}_t'\hat{U}_t^T\dot{\hat{s}}_t-kr_t-\gamma\tilde{x}_t-\beta_1\mathrm{sgn}(\tilde{x}_t)-\alpha\dot{\tilde{x}}_t.
\end{aligned}\tag{29}
$$

Grouping similar terms in (29) yields

$$\dot{r}_t = \tilde{N}_t+N_{A_t}+\hat{N}_{B_t}-kr_t-\gamma\tilde{x}_t-\beta_1\mathrm{sgn}(\tilde{x}_t) \tag{30}$$

where $\tilde{N}_t \triangleq \alpha\dot{\tilde{x}}_t-\dot{\hat{W}}_t^T\sigma(\hat{U}_t^T\hat{s}_t)-\hat{W}_t^T\hat{\sigma}_t'\dot{\hat{U}}_t^T\hat{s}_t+\frac{1}{2}W^T\hat{\sigma}_t'\hat{U}_t^T\dot{\hat{s}}_t+\frac{1}{2}\hat{W}_t^T\hat{\sigma}_t'\hat{U}_t^T\dot{\tilde{s}}_t$, $N_{A_t} \triangleq W^T\sigma_t'U^T\dot{s}_t-\frac{1}{2}W^T\hat{\sigma}_t'\hat{U}_t^T\dot{s}_t-\frac{1}{2}\hat{W}_t^T\hat{\sigma}_t'U^T\dot{s}_t+\dot{\epsilon}(s_t)$, and $\hat{N}_{B_t} \triangleq \frac{1}{2}\tilde{W}_t^T\hat{\sigma}_t'\hat{U}_t^T\dot{\hat{s}}_t+\frac{1}{2}\hat{W}_t^T\hat{\sigma}_t'\tilde{U}_t^T\dot{\hat{s}}_t$. To facilitate stability analysis, an auxiliary term N_{B_t} is defined by replacing $\dot{\hat{s}}_t$ in $\hat{N}_{B_t}$ by $\dot{s}_t$, $\tilde{N}_{B_t} \triangleq \hat{N}_{B_t}-N_{B_t}$, and $N_{AB_t}=N_{A_t}+N_{B_t}$. Based on Assumptions 1 and 2, the following bounds can be obtained [12]:

$$
\begin{aligned}
&\|\tilde{N}_t\|\le\rho_1(\|z_t\|)\|z_t\|,\quad \|N_{A_t}\|\le\zeta_1,\ \ \|N_{B_t}\|\le\zeta_2 \\
&\|\dot{N}_t\|\le\zeta_3+\zeta_4\rho_2(\|z_t\|)\|z_t\|,\quad \left\|\dot{\tilde{x}}_t^T\tilde{N}_{B_t}\right\|\le\zeta_5\|\tilde{x}_t\|^2+\zeta_6\|r_t\|^2
\end{aligned}\tag{31}
$$

where $z_t=[\tilde{x}_t^T,r_t^T]^T$, $\rho_1(\cdot)$ and $\rho_2(\cdot)$ are positive, globally invertible, and nondecreasing functions, ζ_i, $i=1,2,\ldots,6$ are computable positive constants.

Theorem 1: If Assumptions 1 and 2 are satisfied, the identifier developed in (23) along with its weight update laws in (25) ensures asymptotic convergence,[4] in the sense that $\lim_{t\to\infty}\|\tilde{x}_t\|=0$ and $\lim_{t\to\infty}\|\dot{\tilde{x}}_t\|=0$, provided the gains k, γ, β_1, and β_2 satisfy the conditions $\gamma>(\zeta_5/\alpha)$, $k>\zeta_6$, $\beta_1>\max(\zeta_1+\zeta_2,\ \zeta_1+(\zeta_3/\alpha))$, and $\beta_2>\zeta_4$.

Proof: Let V_t be a locally Lipschitz function defined as

$$V_t=\frac{1}{2}r_t^Tr_t+\frac{1}{2}\gamma\tilde{x}_t^T\tilde{x}_t+P_{\upsilon_t}+Q_{\upsilon_t} \tag{32}$$

where $Q_{\upsilon_t} \triangleq (\alpha/4)(\mathrm{tr}(\tilde{W}_t^T\Gamma_W^{-1}\tilde{W}_t)+\mathrm{tr}(\tilde{U}_{x_t}^T\Gamma_{U_x}^{-1}\tilde{U}_{x_t})+\mathrm{tr}(\tilde{U}_{g_t}^T\Gamma_{U_g}^{-1}\tilde{U}_{g_t}))$

$$\dot{P}_{\upsilon_t}=-L_t,\quad P_{\upsilon_0}=\beta_1\sum_{i=1}^{n}|\tilde{x}_0(i)|-\tilde{x}_0^TN_{AB_0}. \tag{33}$$

P_{υ_0}, $\tilde{x}_0$, and N_{AB_0} denote the values of P_{υ_t}, $\tilde{x}_t$, and N_{AB_t}, respectively, at time $t=0$, $\tilde{x}_0(i)$ denotes ith component of $\tilde{x}_0$, and

$$
\begin{aligned}
L_t &\triangleq r_t^T(N_{A_t}-\beta_1\mathrm{sgn}(\tilde{x}_t))+\dot{\tilde{x}}_t^TN_{B_t} \\
&\quad-\beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\|
\end{aligned}\tag{34}
$$

where $\beta_2\in\mathbb{R}^+$. The function derivative $\dot{V}_t$ is given by

$$\dot{V}_t=r_t^T\dot{r}_t+\gamma\tilde{x}_t^T\dot{\tilde{x}}_t+\dot{P}_{\upsilon_t}+\dot{Q}_{\upsilon_t} \tag{35}$$

where $\dot{Q}_{\upsilon_t} = -\frac{\alpha}{2}(\mathrm{tr}(\tilde{W}_t^T\Gamma_W^{-1}\ \dot{\hat{W}}_t)+\mathrm{tr}(\tilde{U}_{x_t}^T\Gamma_{U_x}^{-1}\dot{\hat{U}}_{x_t})+\mathrm{tr}(\tilde{U}_{g_t}^T\Gamma_{U_g}^{-1}\dot{\hat{U}}_{g_t}))$. By substituting the expressions from (28), (29), (33), and (34), (35) can be rewritten as follows:

$$
\begin{aligned}
\dot{V}_t &= r_t^T(\tilde{N}_t+N_{A_t}+\hat{N}_{B_t}-kr_t-\gamma\tilde{x}_t-\beta_1\mathrm{sgn}(\tilde{x}_t)) \\
&\quad+\gamma\tilde{x}_t^T(r_t-\alpha\tilde{x}_t)+\dot{Q}_{\upsilon_t}-\dot{\tilde{x}}_t^TN_{B_t} \\
&\quad-r_t^T(N_{A_t}-\beta_1\mathrm{sgn}(\tilde{x}_t))+\beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\|.
\end{aligned}\tag{36}
$$

On cancelations and simplifications, (36) is given by

$$
\begin{aligned}
\dot{V}_t &= r_t^T\tilde{N}_t+\left(\dot{\tilde{x}}_t^T+\alpha\tilde{x}_t^T\right)\tilde{N}_{B_t}-k\|r_t\|^2-\alpha\gamma\|\tilde{x}_t\|^2 \\
&\quad-\dot{\tilde{x}}_t^TN_{B_t}+\beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\|+\dot{Q}_{\upsilon_t}.
\end{aligned}\tag{37}
$$

[4]For the stability analysis of the identifier, g is assumed to be known from the E-M algorithm and so it is treated as a known parameter.

By defining $\tilde{N}_{B_t} = \hat{N}_{B_t} - N_{B_t}$, (37) is rewritten as

$$\dot{V} = r^T \tilde{N}_t + \dot{\tilde{x}}_t^T \tilde{N}_{B_t} - k\|r_t\|^2 - \alpha\gamma \|\tilde{x}_t\|^2 + \beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\| + \alpha\tilde{x}_t^T \hat{N}_{B_t} + \dot{Q}_{v_t}. \quad (38)$$

On redefining $\hat{N}_{B_t} = \frac{1}{2}\tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{x_t}^T \dot{\hat{x}}_{id_t} + \frac{1}{2}\tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{g_t}^T \dot{\hat{g}}_t + \frac{1}{2}\hat{W}_t^T \hat{\sigma}_t' \tilde{U}_{x_t}^T \dot{\hat{x}}_{id_t} + \frac{1}{2}\hat{W}_t^T \hat{\sigma}_t' \tilde{U}_{g_t}^T \dot{\hat{g}}_t$ and substituting the update equations from (25), (38) is given by

$$\begin{aligned} \dot{V}_t &= r_t^T \tilde{N}_t + \dot{\tilde{x}}_t^T \tilde{N}_{B_t} - k\|r_t\|^2 - \alpha\gamma \|\tilde{x}_t\|^2 \\ &\quad + \beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\| \\ &\quad + \frac{\alpha}{2}\tilde{x}_t^T \big(\tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{x_t}^T \dot{\hat{x}}_{id_t} + \tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{g_t}^T \dot{\hat{g}}_t \\ &\qquad + \hat{W}_t^T \hat{\sigma}_t' \tilde{U}_{x_t}^T \dot{\hat{x}}_{id_t} + \hat{W}_t^T \hat{\sigma}_t' \tilde{U}_{g_t}^T \dot{\hat{g}}_t\big) \qquad (39) \\ &\quad - \frac{\alpha}{2}\big(\mathrm{tr}\big(\tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{x_t}^T \dot{\hat{x}}_{id_t} \tilde{x}_t^T\big) + \mathrm{tr}\big(\tilde{W}_t^T \hat{\sigma}_t' \hat{U}_{g_t}^T \dot{\hat{g}}_t \tilde{x}_t^T\big) \\ &\qquad + \mathrm{tr}\big(\tilde{U}_{x_t}^T \dot{\hat{x}}_{id_t} \tilde{x}_t^T \hat{W}_t^T \hat{\sigma}_t'\big) + \mathrm{tr}\big(\tilde{U}_{g_t}^T \dot{\hat{g}}_t \tilde{x}_t^T \hat{W}_t^T \hat{\sigma}_t'\big)\big). \end{aligned}$$

Using the cyclic property of the trace operator and the bounds defined in (31), (40) is rewritten as

$$\begin{aligned} \dot{V}_t \overset{a.e.}{\leq} &-\alpha\gamma \|\tilde{x}_t^2\| - k\|r_t^2\| + \rho_1(\|z_t\|)\|z_t\|\|r_t\| \\ &+ \zeta_5\|\tilde{x}_t\|^2 + \zeta_6\|r_t\|^2 + \beta_2\rho_2(\|z_t\|)\|z_t\|\|\tilde{x}_t\|. \quad (40) \end{aligned}$$

The right-hand side of (40) is continuous almost everywhere except the Lebesgue measure zero set of times when $\tilde{x}_t = 0$. Substituting for $k \triangleq k_1 + k_2$ and $\gamma \triangleq \gamma_1 + \gamma_2$ and completing the squares

$$\begin{aligned} \dot{V} \overset{a.e.}{\leq} &-(\alpha\gamma - \gamma_5)\|\tilde{x}_t^2\| - (k_1 - \gamma_6)\|r_t\|^2 \\ &+ \frac{\rho_1(\|z_t\|)^2}{4k_2}\|z_t\|^2 + \frac{\beta_2^2\rho_2(\|z_t\|)^2}{4\alpha\gamma_2}\|z_t\|^2. \quad (41) \end{aligned}$$

If the conditions $\gamma > (\zeta_5/\alpha)$, $k > \zeta_6$, $\beta_1 > \max(\zeta_1 + \zeta_2,\ \zeta_1 + (\zeta_3/\alpha))$, and $\beta_2 > \zeta_4$ are met, then $\dot{V}_t$ can be upper bounded as follows:

$$\dot{V}_t \overset{a.e.}{\leq} -\lambda\|z_t\|^2 + \frac{\rho(\|z_t\|)^2}{4\eta} \quad (42)$$

where $\lambda \triangleq \min\{\alpha\gamma_1 - \zeta_5, k_1 - \zeta_6\}$, $\eta \triangleq \min\{k_2, ((\alpha\gamma_2)/(\beta_2^2))\}$, and $\rho(\|z_t\|)^2 \triangleq \rho_1(\|z_t\|)^2 + \rho_2(\|z_t\|)^2$. A semiglobal asymptotic stability of the error dynamics in (27) can be shown using the inequalities in (41) and (42), which yields $\|\tilde{x}_t\| \to 0$, $\|\dot{\tilde{x}}_t\| \to 0$, and $\|r_t\| \to 0$ as $t \to \infty$ [12]. ∎

REFERENCES

[1] C.-S. Tsai, J.-S. Hu, and M. Tomizuka, "Ensuring safety in human-robot coexistence environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 4191–4196.

[2] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, Apr. 2016.

[3] H. C. Ravichandar and A. Dani, "Human intention inference through interacting multiple model filtering," in *Proc. IEEE Conf. Multisensor Fusion Integr. (MFI)*, Sep. 2015, pp. 220–225.

[4] S. Nikolaidis, K. Gu, R. Ramakrishnan, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, 2015, pp. 189–196.

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

[6] M. A. Simon, *Understanding Human Action: Social Explanation and the Vision of Social Science*. New York, NY, USA: SUNY Press, 1982.

[7] H. C. Ravichandar and A. Dani, "Human intention inference and motion modeling using approximate E-M with online learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1819–1824.

[8] H. C. Ravichandar and A. Dani, "Learning contracting nonlinear dynamics from human demonstration for robot motion planning," in *Proc. ASME Dyn. Syst. Control Conf. (DSCC)*, 2015, p. V002T27A008.

[9] G. C. Goodwin and J. C. Aguero, "Approximate EM algorithms for parameter and state estimation in nonlinear stochastic models," in *Proc. IEEE Conf. Decision Control, Eur. Control Conf.*, Dec. 2005, pp. 368–373.

[10] Z. Ghahramani and S. T. Roweis, "Learning nonlinear dynamical systems using an EM algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 431–437.

[11] G. C. Goodwin and A. Feuer, "Estimation with missing data," *Math. Comput. Model. Dyn. Syst.*, vol. 5, no. 3, pp. 220–244, 1999.

[12] 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, Jan. 2013.

[13] R. Luo and D. Berenson, "A framework for unsupervised online human reaching motion recognition and early prediction," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 2426–2433.

[14] 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, Jul. 2013.

[15] M. Monfort, A. Liu, and B. D. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 3672–3678.

[16] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*. Reading, MA, USA: Addison-Wesley, 1994.

[17] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends Human-Comput. Interact.*, vol. 1, no. 3, pp. 203–275, Feb. 2007.

[18] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*. Berlin, Germany: Springer-Verlag, 2013, pp. 403–415.

[19] 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. Comput. Vis. Pattern Recognit.*, vol. 5. Jun, 2003, p. 53.

[20] D. Kulic and E. A. Croft, "Affective state estimation for human–robot interaction," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 991–1000, Oct. 2007.

[21] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *Robot. Sci. Syst.*, vol. 38, no. 1, pp. 14–29, Jan. 2013.

[22] R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis, "Understanding human intentions via hidden Markov models in autonomous mobile robots," in *Proc. ACM/IEEE Int. Conf. Human Robot Interact.*, Mar. 2008, pp. 367–374.

[23] 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. Robot. Autom. (ICRA)*, May 2015, pp. 885–892.

[24] 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, Jun. 2013.

[25] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, and S. S. Srinivasa, "Learning the communication of intent prior to physical collaboration," in *Proc. IEEE Int. Symp. Robot Human Interact. Commun.*, Sep. 2012, pp. 968–973.

[26] K. W. Strabala *et al.*, "Towards seamless human-robot handovers," *J. Human-Robot Interact.*, vol. 2, no. 1, pp. 112–132, 2013.

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

[28] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making service robots human-safe," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Oct./Nov. 2000, pp. 696–701.

[29] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robot. Auto. Syst.*, vol. 42, nos. 3–4, pp. 143–166, Mar. 2003.

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

[31] D. Kulić and E. A. Croft, "Estimating intent for human-robot interaction," in *Proc. IEEE Int. Conf. Adv. Robot.*, 2003, pp. 810–815.

[32] D. De Carli *et al.*, "Measuring intent in human-robot cooperative manipulation," in *Proc. IEEE Int. Workshop Haptic Audio Vis. Environ. Games*, Nov. 2009, pp. 159–163.

[33] 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. Robot. Autom.*, May 2011, pp. 5875–5880.

[34] D. Gehrig *et al.*, "Combined intention, activity, and motion recognition for a humanoid household robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 4819–4825.

[35] O. C. Schrempf and U. D. Hanebeck, "A generic model for estimating user-intentions in human-robot cooperation," in *Proc. Int. Conf. Inform. Control, Autom. Robot.*, 2005, pp. 251–256.

[36] J. Elfring, R. van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robot. Auto. Syst.*, vol. 62, no. 4, pp. 591–602, Apr. 2014.

[37] N. Hu, Z. Lou, G. Englebienne, and B. Kröse, "Learning to recognize human activities from soft labeled data," in *Proc. Robot. Sci. Syst.*, Berkeley, CA, USA, 2014.

[38] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using Gaussian process latent CRFs," in *Proc. Robot. Sci. Syst. (RSS)*, 2014.

[39] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.

[40] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. Human-Robot Collaboration Ind. Manuf. Workshop Robot., Sci. Syst. Conf.*, Nov. 2014, pp. 299–306.

[41] D. Song *et al.*, "Predicting human intention in visual observations of hand/object interactions," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1608–1615.

[42] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[43] 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. Inf. Sci. Eng.*, vol. 14, no. 1, p. 011006, Jan. 2014.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[45] K. Lange, "A gradient algorithm locally equivalent to the EM algorithm," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 57, no. 2, pp. 425–437, 1995.

[46] 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, Oct. 2008.

[47] A. P. Dani, N. R. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 241–246, Jan. 2012.

[48] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Berlin, Germany: Springer-Verlag, 2007, pp. 69–84.

[49] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems With Actuator Nonlinearities*, vol. 24. Philadelphia, PA, USA: SIAM, 2002.

Harish Ravichandar received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.

His current research interests include machine learning, nonlinear dynamical systems and estimation, and human–robot interaction.

Ashwin P. Dani (M'11) received the M.S. and Ph.D. degrees from the University of Florida, Gainesville, FL, USA, in 2008 and 2011, respectively.

He was a Post-Doctoral Research Associate at the University of Illinois Urbana–Champaign, Champaign, IL, USA. In 2013, he joined the Faculty of Electrical and Computer Engineering (ECE), University of Connecticut, Storrs, CT, USA, as an Assistant Professor. He has co-authored over 30 refereed papers, three book chapters, and holds two patents in the area of robotics and vision-based estimation. His current research interests include nonlinear estimation and control, human–robot collaboration, and autonomous navigation.

Dr. Dani serves as a member of the Conference Editorial Board of the IEEE Control Systems Society (CSS). He was a recipient of the ASEM Dynamics Systems and Controls Conference Best Student Robotics Paper Award in 2015, the ISIF International Conference on Information Fusion Best Student Paper Award - 2nd runner up in 2016, the IEEE CSS Video Contest Award in 2015, and the UConn's Outstanding Teaching Award from ECE in 2015.