Proceedings of the ASME 2015 Dynamic Systems and Control Conference
DSCC2015
October 28-30, 2015, Columbus, OH, USA

DSCC2015-9870

LEARNING CONTRACTING NONLINEAR DYNAMICS FROM HUMAN DEMONSTRATION FOR ROBOT MOTION PLANNING

Harish Ravichandar
Department of Electrical and Computer Engineering
University of Connecticut
Storrs, CT, USA
Email: harish.ravichandar@uconn.edu

Ashwin Dani
Department of Electrical and Computer Engineering
Management and Engineering for Manufacturing Program
University of Connecticut
Storrs, CT, USA
Email: ashwin.dani@engr.uconn.edu

ABSTRACT

In this paper, we present an algorithm to learn the dynamics of human arm motion from the data collected from human actions. Learning the motion plans from human demonstrations is essential in making robot programming possible by non-expert programmers as well as realizing human-robot collaboration. The highly complex human reaching motion is generated by a stable closed-loop dynamical system. To capture the complexity a neural network (NN) is used to represent the dynamics of the human motion states. The trajectories of arm generated by humans for reaching to a place are contracting towards the goal location from various initial conditions with built-in obstacle avoidance. To take into consideration the contracting nature of the human motion dynamics the unknown motion model is learned using a NN subject to contraction analysis constraints. To learn the NN parameters an optimization problem is formulated by relaxing the non-convex contraction constraints to Linear matrix inequality (LMI) constraints. Sequential Quadratic Programming (SQP) is used to solve the optimization problem subject to the LMI constraints. For obstacle avoidance a negative gradient of the repulsive potential function is added to the learned contracting NN model. Experiments are conducted on Baxter robot platform to show that the robot can generate reaching paths from the contracting NN dynamics learned from human demonstrated data recorded using Microsoft Kinect sensor. The algorithm is able to adapt to situations for which the demonstrations are not available, e.g., an obstacle placed in the path.

1 Introduction

Learning the dynamics of the human motion and using the learned model to generate trajectories for robot that can adapt to new situations is an important problem in the context of training robots using non-expert operators [1–5] as well as for safe human-robot collaboration (HRC) [6–10]. For developing robot assistants, robots should be given an ability to learn from the user, e.g., in manufacturing context - a non-expert programmer operator should be able to program the robot by just demonstrating the task to the robot [5], or for elderly assistant robots - the user should be able to teach various tasks to robots. In this paper, we develop a method to learn the complex human motion dynamic motion model using a neural network (NN) subject to motion trajectory constraints for reaching tasks. Humans or animals generate inherently closed-loop stable limb motions to reach to different locations by incorporating sensory feedback [11, 12]. In this work, the human arm reaching motion is represented using a dynamic model $\dot{x} = f(x)$, where $f(x)$ is represented using a NN. The nonlinear function f is learned using a single or multiple demonstrations from a human. The problem of learning motion dynamics is formulated as a parameter learning problem of NNs under the stability constrains given by contraction analysis of nonlinear systems [13]. The contraction analysis yields a global exponential stability of nonlinear systems. The advantages of learning a globally contracting function are: (a) from any initial conditions the trajectories will converge to the goal location, (b) even the addition of obstacle avoidance feature will generate the trajectories that are converging to the goal location. To learn the

NN parameters, an optimization problem is formulated which computes the weights or parameters of the NN subject to the contraction condition of the underlying dynamics. The contraction condition yields a matrix inequality condition which is non-convex in the parameters of the NN. The contraction inequality constraint is reformulated as the bilinear inequality constraints by assuming the contraction metric to be a constant M[1]. The bilinear contraction conditions are then relaxed and converted to the linear matrix inequality (LMI) constraints by using Shor's relaxation [14]. A new back-propagation algorithm is developed which uses Sequential Quadratic Programming (SQP) algorithm subject to the relaxed convex contraction constraints. Good initial conditions for constrained SQP algorithm are selected based on the solutions obtained by solving an unconstrained optimization problem first.

The learning contracting dynamics method is enhanced with obstacle avoidance strategy by using repulsive potential function gradient [15,16]. We use the repulsive potential function gradient to modify the contracting attractor trajectory field learned from demonstrations. Since the contracting trajectories are globally converging, even if there is a change in trajectory due to repulsive field of obstacle the trajectories will always converge to the goal location. We also show that the proposed algorithm is robust to abrupt changes in trajectories caused due to sensor failures or malfunctioning on the robot. The learned motion model will always create paths that converge to the goal locations in spite of the abrupt changes. The other advantage of the approach is that the trajectories can be learned only based on a single demonstration although learning paths based on more demonstrations is beneficial if the single demonstration is a bad demonstration. The block diagram of the proposed algorithm is shown in Fig. 1. The contributions of this work are summarized below.

1. A new method to learn contracting dynamic motion model in state-space is presented. The learned model can be used to generate motion trajectories of robot based on human demonstrations. The proposed algorithm combines the advantages of global exponential stability property of the goal location due to contraction analysis (from dynamical systems theory) and the model represented using NN.
2. An obstacle avoidance feature can be naturally incorporated in the dynamic model, the global contracting (global exponential stability) nature of the dynamics makes the goal location globally attractive, which makes the dynamics robust to perturbations and sensor faults.

Related Work

The most related work to our work is learning from demonstration (LfD) using Gaussian Mixture Model (GMM) proposed in [3, 17]. The Lyapunov stability conditions are used to derive the parameters of GMMs with the global asymptotic stability conditions. The global exponential stability condition obtained using contraction provides robustness to perturbations, addition of obstacle avoidance features, as well as sensor faults. A comprehensive review of recent methods in LfD is presented in [18]. Our work falls in the category of learning the dynamic model from the data collected using human operators and use real-time sensor feedback to avoid the obstacles. In [19], nonlinear oscillators are used to learn the rhythmic movement by demonstrations. Another related research area is dynamic motion primitives (DMPs) [12, 20–22] which uses combination of simple linear dynamic models to represent complex motion to create goal-directed trajectories. The DMP method can also handle obstacle avoidance functions. In our approach, the attracting trajectories are created in state space, whereas in DMP the attracting trajectories are created in phase space.

An algorithm for physical human robot interaction (pHRI) is presented in [23] for learning model of human motion and bike bot interaction. The method uses axial linear embedding algorithm to learn the lower-dimensional latent dynamics for human limb motion. Compared to the method presented in our paper the learning dynamics goal is different. Safety in HRC is an important aspect. In [9, 24, 25], algorithms for Kinect array based safety in manufacturing assembly tasks are presented. Incorporating learning from failed demonstrations is studied in [26, 27]. An algorithm related to work space occupancy and human motion prediction is studied in [28]. In another line of research the objective functions for performing tasks are learned instead of the dynamic models which generate the trajectories. The methods are called as inverse reinforcement learning [29] or inverse optimal control [30, 31].

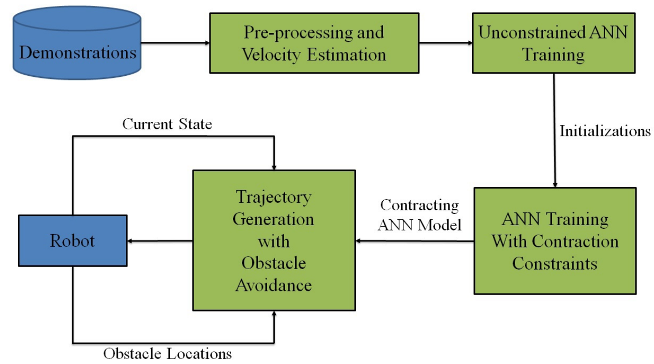

FIGURE 1. Block diagram representation of learning contracting dynamics using NNs for robot trajectory generation along with obstacle avoidance.

[1] Assumption of a constant contraction metric is not restrictive in the context of training NN with the contraction constraints

2 Preliminaries

A Brief Review of Contraction Analysis

In this section, contraction analysis [13] for analyzing exponential stability of nonlinear systems is briefly reviewed. Consider a nonlinear, non-autonomous system of the form

$$\dot{x} = f(x,t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is a state vector and $f: \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a continuously differentiable nonlinear function. With the assumed properties of (1), the exact relation $\delta\dot{x} = \frac{\partial f(x,t)}{\partial x}\delta x$ holds, where δx is an infinitesimal virtual displacement in fixed time. The squared virtual displacement between two trajectories of (1) in a symmetric, uniformly positive definite metric $M(x,t) \in \mathbb{R}^{n\times n}$ is given by $\delta x^T M(x,t)\,\delta x$ and its time derivative by

$$\frac{d}{dt}\left(\delta x^T M(x,t)\,\delta x\right) = \delta x^T \left(\frac{\partial f}{\partial x}^T M(x,t) + \dot{M}(x,t) + M(x,t)\frac{\partial f}{\partial x}\right)\delta x. \tag{2}$$

If the following inequality is satisfied

$$\frac{\partial f}{\partial x}^T M(x,t) + \dot{M}(x,t) + M(x,t)\frac{\partial f}{\partial x} \le -2\gamma M(x,t)\ \forall t,\ \forall x \tag{3}$$

for a strictly positive constant γ, then the system (1) is said to be contracting with the rate γ and all the system trajectories exponentially converge to a single trajectory irrespective of the initial conditions (hence, globally exponentially stable).

3 Problem Formulation and Solution Approach

3.1 Problem Formulation

Consider a state variable $x \in \mathbb{R}^n$ and let a set of N demonstrations $\{\mathscr{D}_i\}_{i=1}^N$ be solutions to the underlying dynamic model governed by the following first order differential equation

$$\dot{x}(t) = f(x(t)) \tag{4}$$

where $f: \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear continuous and continuously differentiable function. Each demonstration would consist of the trajectories of the state $\{x(t)\}_{t=1}^{t=T}$ and the trajectories of the state derivative $\{\dot{x}(t)\}_{t=1}^{t=T}$ from time $t=0$ to $t=T$. Since all the state trajectories of the demonstrations of a specific stable dynamic system would exponentially converge to a single trajectory or a single point, the system defined in (4) could be seen as a globally contracting system. The nonlinear function f is modeled using a NN given by

$$f(x(t)) = W^T\sigma\left(U^T s(t)\right) + \varepsilon(s(t)) \tag{5}$$

where $s(t) = \left[x(t)^T, 1\right]^T \in \mathbb{R}^{n+1}$ is the input vector to the NN, $\sigma(U^T s(t)) = [\frac{1}{1+\exp(-(U^T s(t))_1)}, \cdots \frac{1}{1+\exp(-(U^T s(t))_i)}, \cdots \frac{1}{1+\exp(-(U^T s(t))_{n_h})}]^T$ is the vector-sigmoid activation function and $\left(U^T s(t)\right)_i$ is the i^{th} element of the vector $\left(U^T s(t)\right)$, $U \in \mathbb{R}^{n+1\times n_h}$ and $W \in \mathbb{R}^{n_h\times n}$ are the bounded constant weight matrices, $\varepsilon(s(t)) \in \mathbb{R}^n$ is the function reconstruction error that goes to zero after the NN is fully trained, and n_h is the number of hidden layers of the NN. Given the demonstrations, this paper addresses the problem of learning the function f, which is modeled using a NN, under contraction conditions. This will help in generating exponentially converging trajectories, governed by a stable dynamical system, starting from a given arbitrary initial condition.

The constrained optimization problem to be solved in order to train a contracting NN can be written as

$$\{\hat{W}, \hat{U}\} = \arg\min_{W,U}\{\alpha E_D + \beta E_W\} \tag{6}$$

$$\text{such that } \frac{\partial f}{\partial x}^T M + M\frac{\partial f}{\partial x} \le -\gamma M,\ M > 0 \tag{7}$$

where $E_D = \sum_{i=1}^N [y_i - a_i]^T[y_i - a_i]$ is the sum of squared error, $y_i \in \mathbb{R}^n$ and $a_i \in \mathbb{R}^n$ represent the target and the network's output of the i-th demonstration, E_W is the sum of the squares of the NN weights, α and β are the parameters of regularization, γ is a strictly positive constant, and $M \in \mathbb{R}^{n\times n}$ represents a *constant* positive symmetric matrix. Thus, $\left(\frac{\partial f}{\partial x}\right)$ can be calculated as

$$\frac{\partial f}{\partial x} = W^T\frac{\partial\sigma\left(U^T s\right)}{\partial x} = W^T\left[\Sigma'\left(U^T s\right)\right]U_x^T \tag{8}$$

where for any $b \in \mathbb{R}^p$, $\Sigma'(b) \in \mathbb{R}^{n_h\times n_h}$ is a diagonal matrix given by

$$\Sigma'(b) = \operatorname{diag}\left(\sigma(b_1)(1-\sigma(b_1)),\ \sigma(b_2)(1-\sigma(b_2)), \cdots, \sigma(b_p)(1-\sigma(b_p))\right), \tag{9}$$

and $U_x \in \mathbb{R}^{n\times n_h}$ is a sub-matrix of U formed by taking the first n rows of U.

3.2 Learning Contracting NNs

The optimization problem defined in (6) and (7) can be written as

$$\{\hat{W}, \hat{U}\} = \arg\min_{W,U}\{\alpha\sum_{i=1}^N\left([y_i - a_i]^T[y_i - a_i]\right) + \beta\left(\operatorname{tr}\left(W^T W\right) + \operatorname{tr}\left(U^T U\right)\right)\} \tag{10}$$

such that

$$U_x\left[\Sigma'(U^T s)\right]^T WM + MW^T\left[\Sigma'(U^T s)\right]U_x^T \le -\gamma M,\ M > 0 \tag{11}$$

where $M \in \mathbb{R}^{n\times n}$ is the symmetric, positive definite contraction metric. We will show using Lemma 1 and Proposition 1 that the non-convex constraints (11) can be relaxed to LMI constraints.

Lemma 1. *The constraints defined in (11) is always satisfied if the following constraints are satisfied*

$$U_x W M + M W^T U_x^T + \gamma_1 M \leq 0,\ M > 0, \tag{12}$$

where $\gamma_1 = 4\gamma$.

Proof. The sigmoid function $\sigma(\cdot) \in [0,1]$ and hence its derivative $\sigma(\cdot)(1-\sigma(\cdot))$ has upper and lower bounds given by

$$0 \leq \sigma(\cdot)(1-\sigma(\cdot)) \leq 0.25. \tag{13}$$

Using (13) and the fact that $\Sigma^{'}(\cdot)$ is given by (9), each diagonal element of the matrix $\Sigma^{'}(U^T s)$ can be upper bounded by 0.25 and the upper bound of the whole matrix is given by

$$\Sigma^{'}(U^T s) \leq 0.25 I_{n_h \times n_h}. \tag{14}$$

On multiplying MW^T to the left and U_x^T to the right of both sides of (14), we have

$$MW^T \left[\Sigma^{'}(U^T s)\right] U_x^T \leq 0.25 \left[MW^T U_x^T\right] \tag{15}$$

and similarly we have

$$U_x \left[\Sigma^{'}(U^T s)\right]^T WM \leq 0.25 \left[U_x W M\right]. \tag{16}$$

Using (15) and (16), $U_x \left[\Sigma^{'}(U^T s)\right]^T WM + MW^T \left[\Sigma^{'}(U^T s)\right] U_x^T$ can be upper bounded as

$$\begin{aligned} & U_x \left[\Sigma^{'}(U^T s)\right]^T WM + MW^T \left[\Sigma^{'}(U^T s)\right] U_x^T \\ & \quad \leq 0.25 \left[U_x W M\right] + 0.25 \left[MW^T U_x^T\right]. \end{aligned} \tag{17}$$

Now, if the constraint defined in (12) holds, (12) and (17) together yield

$$\begin{aligned} & U_x \left[\Sigma^{'}(U^T s)\right]^T WM + MW^T \left[\Sigma^{'}(U^T s)\right] U_x^T \\ & \quad \leq 0.25 \left[U_x W M\right] + 0.25 \left[MW^T U_x^T\right] \leq -\gamma M, \end{aligned} \tag{18}$$

and the constraint in (11), a part of (18), is hence satisfied. □

Proposition 1. *The bilinear matrix inequality (BMI) defined in (12) can be converted to the following LMI equations by Shor's relaxation in terms of a new variable* $A = U_x W$

$$AM + MA^T + \gamma_1 M \leq 0, \tag{19}$$

and

$$\text{sym}\begin{bmatrix} I_{n_h \times n_h} & W \\ U_x & A \end{bmatrix} \leq 0,\ M > 0 \tag{20}$$

where $\text{sym}(\cdot)$ *is the symmetric part of a matrix.*

Proof. The first inequality in (12) can be written as $AM + MA^T + \gamma_1 M \leq 0$. To handle the equality constraint $A = U_x W$, new constraints on A, U_x, and W are given in (20) according to the Shor's relaxation [14]. □

Thus, the solution to the modified optimization problem is given by

$$\begin{aligned} \{\hat{W}, \hat{U}\} = \arg\min_{W,U} \{ & \alpha \sum_{i=1}^{N} \left([y_i - a_i]^T [y_i - a_i] \right) \\ & + \beta \left(\text{tr}\left(W^T W\right) + \text{tr}\left(U^T U\right) \right) \} \end{aligned} \tag{21}$$

$$\text{such that} \quad AM + MA^T + \gamma_1 M \leq 0,$$

$$\text{sym}\begin{bmatrix} I_{n_h \times n_h} & W \\ U_x & A \end{bmatrix} \leq 0,\ M > 0,$$

would also be a solution to the original optimization problem defined in (10) and (11). Note that the cost function to be minimized is a non-convex function with convex inequality constraints.

3.3 Obstacle Avoidance

The trajectory generated by the contracting NN does not take any obstacles into consideration. The only feedback the system considers is the current state of the system. In order to effectively generate trajectories to perform various reaching tasks, we also explore the capability of obstacle avoidance. To this end, we introduce an artificial repulsive potential field [15, 16] in the workspace to the contracting dynamics learned using NN. The repulsive potential V_r for the i^{th} obstacle and the control point (end effector) is given by

$$V_{ri}(x) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{d_i(x)} - \frac{1}{D_i^*} \right)^2, & d_i(x) \leq D_i^* \\ 0, & d_i(x) > D_i^* \end{cases} \tag{22}$$

The gradient of (22) w.r.t. the state x is given by

$$\nabla_x V_{ri}(x) = \begin{cases} \eta \left(\frac{1}{D_i^*} - \frac{1}{d_i(x)} \right) \frac{1}{d_i^2(x)} \nabla_x d_i(x), & d_i(x) \leq D_i^* \\ 0, & d_i(x) > D_i^* \end{cases} \tag{23}$$

where $d_i(x) = ||x - o_i||_2$ is the Euclidean distance from x to the location of the i^{th} obstacle o_i, D_i^* is the size of the domain of influence of the i^{th} obstacle, $\eta \in \mathbb{R} > 0$ is a positive constant, and $\nabla_x d_i(x)$ denotes the derivative of $d_i(x)$ with respect to x. The negative gradient of (22), given by the negative of (23), gives a repulsive force acting on the robot. The repulsive force that drives the robot away from the obstacles can be viewed as a force that acts along with an attractive force to drive the robot to the goal location. In our case, the attractive force is provided by the contracting NN. Hence, the combined dynamics is described by

$$\dot{x} = f(x(t)) - \sum_i \nabla_x V_{ri}(x),\ \text{for } i = \{1, .., n_o\} \tag{24}$$

where n_o is the number of obstacles.

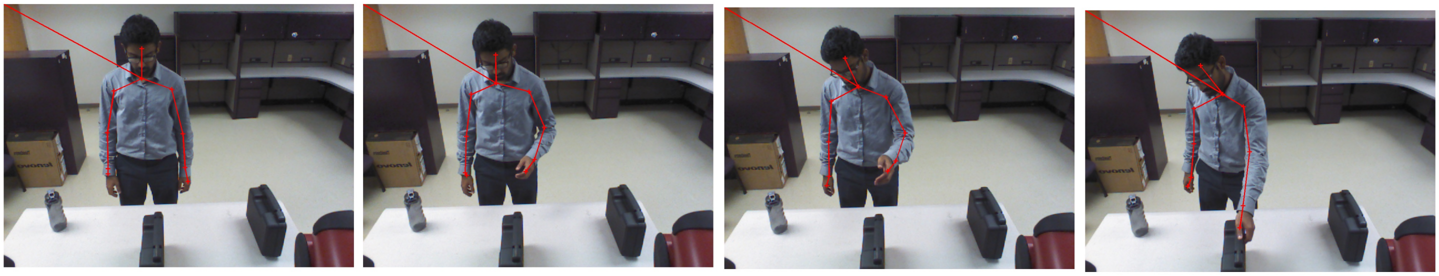

FIGURE 2. Demonstration of a human reaching for the target location.

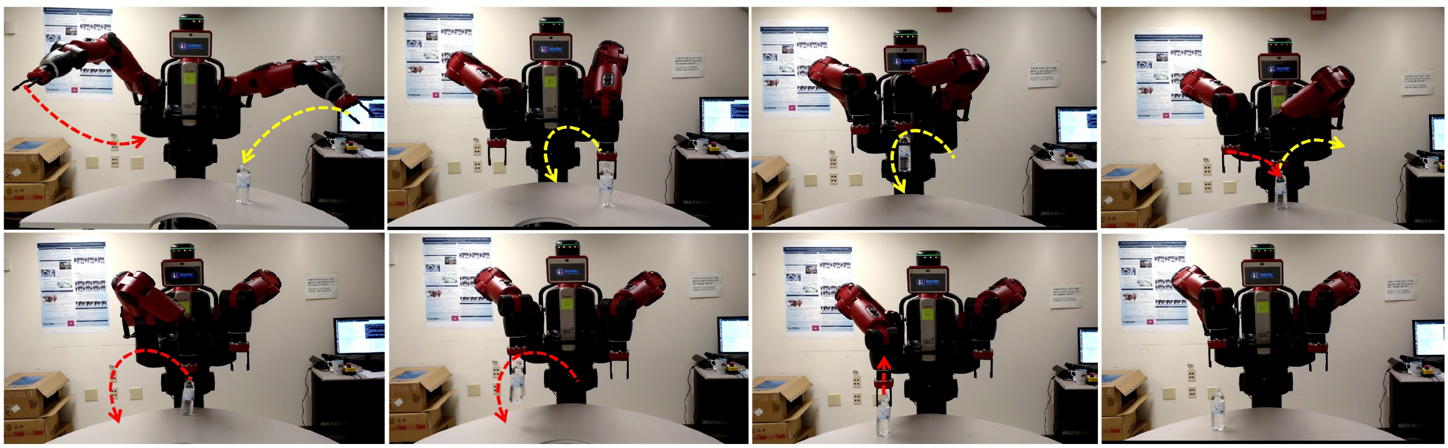

FIGURE 3. Robot motion planning using a contracting neural network for a pick and place task executed as a series of reaching tasks. The trajectories of the left arm (dotted yellow lines) and the trajectories of the right arm (dotted red lines) were generated independently using the same learned contracting NN.

4 Experimental Validation

A set of five experiments were conducted to learn the human arm's motion dynamics using a NN and demonstrate the performance of the proposed algorithm. All the experiments were conducted using a desktop computer running Intel i3 processor and 8 Gigabytes of memory. The algorithm was coded using Matlab 2014a. In all the experiments, the position of the hand in 3-dimensional (3D) Cartesian space was considered to be the state. The velocity estimates of the hand were computed from the position measurements using a finite difference method. The number of hidden layers of the NN was chosen to be six. The network weights of the constrained optimization algorithm were initialized to the weights obtained by learning the NN without the constraints. The metric M was chosen to be the identity matrix. Matlab's *fmincon* function was used to solve the optimization problem. The demonstrations were collected using the Microsoft Kinect for Windows and involved a human reaching for a target location to pick an object (see Fig. 2). The videos of the experiments conducted on the Baxter robot platform can be viewed at $https : //www.youtube.com/watch?v = z0WC_W c5AMk$. The objective of each experiment can be summarized as follows.

1. Experiment 1 was conducted to show that the learned model generates contracting trajectories from arbitrary initial conditions (See Fig. 4).
2. Experiment 2 was conducted to study the benefits of imposing the contraction conditions in the learning of the dynamic system as opposed to unconstrained learning (See Fig. 5).
3. Experiment 3 was conducted to study the behavior of the learned model under perturbations such as a sudden unintended change in the states (See Fig. 6).
4. Experiment 4 was conducted on the Baxter robot platform. The learned model was used to generate trajectories for both the arms of the robot in order to perform a pick and place task (See Fig. 3).
5. Experiment 5 was conducted to study the obstacle avoidance properties of the proposed algorithm (See Figs. 7 and 8).

The first set of experiment involved learning the parameters of the NN with contraction constraints. The set of *four* demonstrations were used to learn the dynamics. In Fig. 4, it can be seen that the learned NN indeed generates contracting trajectories that start from arbitrary initial conditions and end at the target.

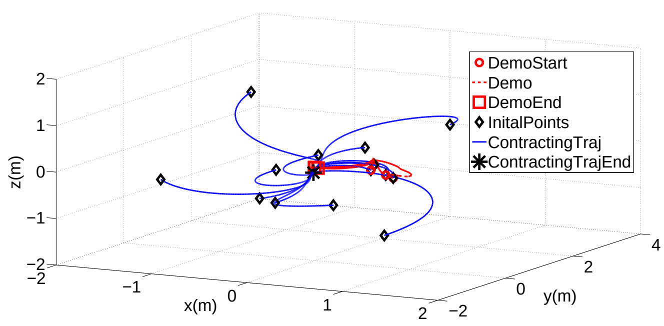

FIGURE 4. The learned NN generating contracting trajectories to the target starting from arbitrary initial points.

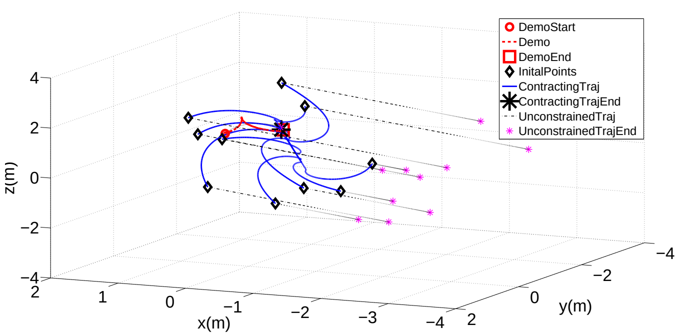

FIGURE 5. Comparison of models learned with (solid blue lines) and without (dashed black lines) contraction constraints using a single demonstration.

To demonstrate the effectiveness of the proposed algorithm, the second set of experiments involved the learning of the dynamics both with and without the enforcement of the constraints using a single demonstration (see Fig.5). The model learned without constraints generated trajectories that do not end at the target unless the initial point of the model was chosen to be the initial point of the demonstration. On the other hand, we can see that the model learned with the contraction constraints generated contracting trajectories to the target from arbitrary initial conditions. The initial conditions were chosen using Matlab's random number generator. It should be noted that the model learned with *four* demonstrations generated trajectories that were more "human-like" compared to the model learned with a single demonstration.

The third set of experiments were conducted to study the behavior of the learned model under perturbations. It is important to study the robustness of the learned model under possible perturbations such as a sudden spike in the sensor measurements. To simulate this issue, we introduced perturbations (translations) in the trajectories that were being generated by the model at the $50th$ iteration (see Fig. 6). The magnitude of the perturbations

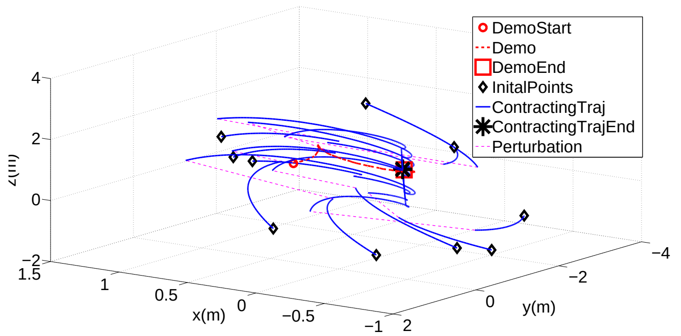

FIGURE 6. Robustness analysis of the proposed learning algorithm. Perturbations (translations) were introduced while the trajectories were being generated by the model.

were randomly sampled from a uniform distribution between 0 meters and 1 meters. It can be seen that even after the trajectories are perturbed, the model drives the trajectories to the target.

The fourth experiment involved motion planning for the Baxter robot in order to perform a pick and place task. The task was to pick an object from the table using one of the arms, place it in a location, have the other arm pick it up, and place it in the target location. The learned model was used to generate the trajectories that the arms were made to follow. The standard low level controller of Baxter was used to move the arms through the generated way points (see Fig. 3). The same learned model from experiment 1 was used and no separate demonstrations were required for generating the trajectories of both the arms of Baxter. The task was executed as a sequence of sub-tasks and the learned NN was used to generate the trajectories for each sub-task. This experiment was repeated five times with the arms starting from different initial conditions. The contracting NN was able to successfully generate the required trajectories on all five occasions.

The fifth experiment involved testing the obstacle avoidance ability of the proposed algorithm. The same learned NN from earlier experiments were used and (24) was used to generate the trajectories. The region of influence D_i^* of each obstacle was chosen to be 0.1 meters and the constant η was chosen to be 0.01. The trajectories generated from arbitrary initial points are shown in Fig. 8. The obstacle avoidance algorithm was also implemented on the Baxter platform for a pick and place task. The task involved picking up an object from a location and placing it in a given location while avoiding a box shaped obstacle. The results are shown as a sequence of images in Fig. 7.

5 Conclusion

The paper illustrates a method to learn contracting nonlinear dynamic systems represented using a NN. Arm motion trajectories generated by humans are contracting and reach a tar-

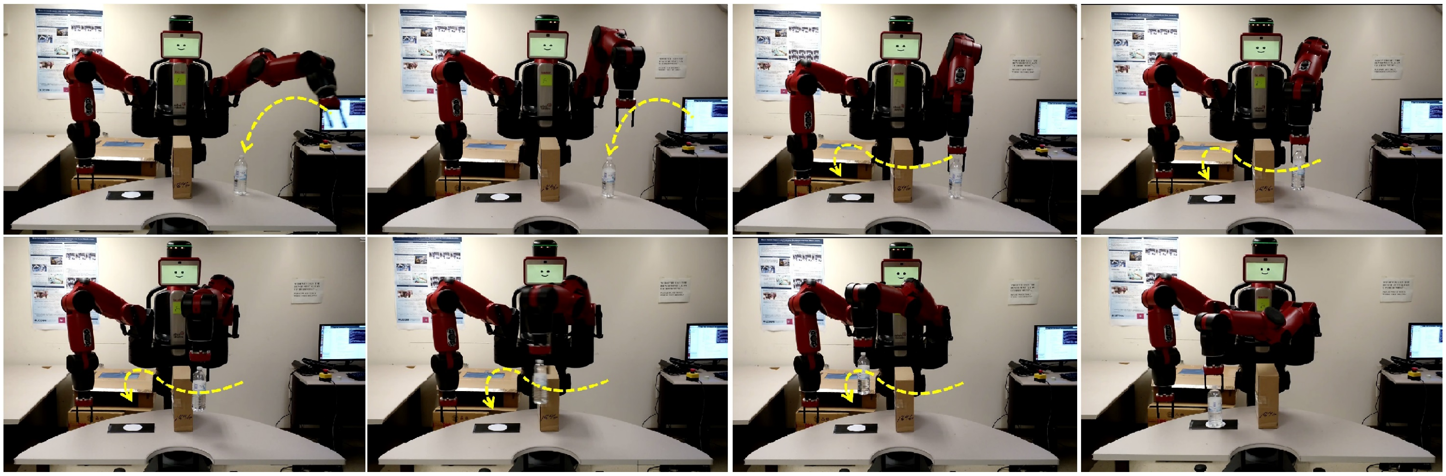

FIGURE 7. A pick and place task performed as a series of reaching operations on the Baxter robot involving obstacle avoidance. Trajectories (dotted yellow lines) were generated using the learned contracting model with real-time obstacle avoidance.

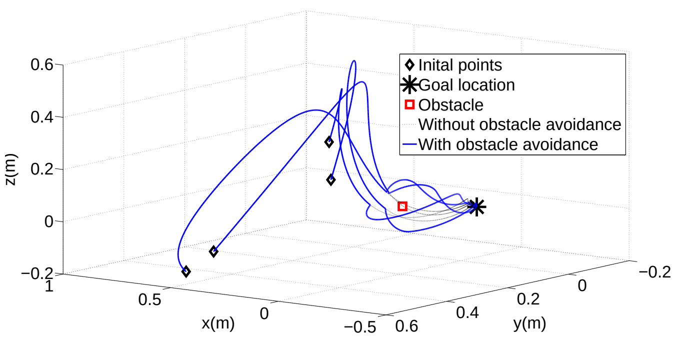

FIGURE 8. Trajectories generated by the contracting NN both with and without obstacle avoidance.

get location. Thus in order to validate the proposed algorithm, human arm motion dynamics were learned from demonstrations recorded using Microsoft Kinect. A constrained optimization problem was formulated to facilitate the learning of contracting dynamics. The non-convex constraints that result from contraction analysis were relaxed to obtain LMI constraints. A set of five experiments were conducted to demonstrate the contracting behavior of the system starting at arbitrary initial conditions. Experiments 1 and 2 showed that contracting models could be learned from a single demonstration and are robust to perturbations in the trajectory. Experiment 3 showed that the contracting NN is robust to perturbations in the trajectory. Experiment 4 showed that the learned model can be used to generate trajectories for robot arm motion. Finally, Experiment 5 illustrated the ability of the CDSP to handle obstacles placed on the preplanned trajectories

References

[1] Wang, C., Zhao, Y., Lin, C.-Y., and Tomizuka, M., 2014. "Fast planning of well conditioned trajectories for model learning". In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1460–1465.

[2] Billard, A., and Matarić, M. J., 2001. "Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture". *Robotics and Autonomous Systems,* ***37***(2), pp. 145–160.

[3] Khansari-Zadeh, S. M., and Billard, A., 2011. "Learning stable nonlinear dynamical systems with gaussian mixture models". *IEEE Transactions on Robotics,* ***27***(5), pp. 943–957.

[4] Gribovskaya, E., Khansari-Zadeh, S. M., and Billard, A., 2010. "Learning non-linear multivariate dynamics of motion in robotic manipulators". *The International Journal of Robotics Research.*

[5] Rossano, G. F., Martinez, C., Hedelind, M., Murphy, S., and Fuhlbrigge, T. A., 2013. "Easy robot programming concepts: An industrial perspective". In IEEE International Conference on Automation Science and Engineering (CASE), pp. 1119–1126.

[6] Ravichandar, H., and Dani, A. P., 2015. "Human intention inference and motion modeling using approximate e-m with online learning". In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[7] Ravichandar, H., and Dani, A. P., 2015. "Human intention inference through interacting multiple model filtering". In IEEE Conference on Multisensor Fusion and Integration (MFI).

[8] Tsai, C.-S., Hu, J.-S., and Tomizuka, M., 2014. "Ensuring safety in human-robot coexistence environment". In

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4191–4196.
[9] Morato, C., Kaipa, K. N., Zhao, B., and Gupta, S. K., 2014. "Toward safe human robot collaboration by using multiple kinects based real-time human tracking". *ASME Journal of Computing and Information Science in Engineering,* ***14***(1), p. 011006.
[10] Lasota, P. A., Rossano, G. F., and Shah, J. A., 2014. "Toward safe close-proximity human-robot interaction with standard industrial robots". In IEEE International Conference on Automation Science and Engineering (CASE), pp. 339–344.
[11] Fajen, B. R., and Warren, W. H., 2003. "Behavioral dynamics of steering, obstable avoidance, and route selection.". *Journal of Experimental Psychology: Human Perception and Performance,* ***29***(2), p. 343.
[12] Schaal, S., 1999. "Is imitation learning the route to humanoid robots?". *Trends in cognitive sciences,* ***3***(6), pp. 233–242.
[13] Lohmiller, W., and Slotine, J.-J. E., 1998. "On contraction analysis for nonlinear systems". *Automatica,* ***34***(6), pp. 683–696.
[14] Shor, N., 1987. "Class of global minimum bounds of polynomial functions". *Cybernetics and Systems Analysis,* ***23***(6), pp. 731–734.
[15] Khatib, O., 1986. "Real-time obstacle avoidance for manipulators and mobile robots". *The International Journal of Robotics Research,* ***5***(1), pp. 90–98.
[16] Spong, M. W., Hutchinson, S., and Vidyasagar, M., 2006. *Robot modeling and control*, Vol. 3. Wiley New York.
[17] Khansari-Zadeh, S. M., and Billard, A., 2010. "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming". In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2676–2683.
[18] Argall, B. D., Chernova, S., Veloso, M., and Browning, B., 2009. "A survey of robot learning from demonstration". *Robotics and autonomous systems,* ***57***(5), pp. 469–483.
[19] Ijspeert, A. J., Nakanishi, J., and Schaal, S., 2002. "Learning rhythmic movements by demonstration using nonlinear oscillators". In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 958–963.
[20] Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S., 2013. "Dynamical movement primitives: learning attractor models for motor behaviors". *Neural Computation,* ***25***(2), pp. 328–373.
[21] Rai, A., Meier, F., Ijspeert, A., and Schaal, S., 2014. "Learning coupling terms for obstacle avoidance". In International Conference on Humanoid Robotics, pp. 512–518.
[22] Kalakrishnan, M., Righetti, L., Pastor, P., and Schaal, S., 2012. "Learning force control policies for compliant robotic manipulation". In International Conference on Machine Learning.
[23] Chen, K., Zhang, Y., and Yi, J., 2014. "An integrated physical-learning model of physical human-robot interactions: A bikebot riding example". In ASME 2014 Dynamic Systems and Control Conference.
[24] Kaipa, K. N., Morato, C., Liu, J., and Gupta, S. K., 2014. "Human-robot collaboration for bin-picking tasks to support low-volume assemblies". In Human-Robot Collaboration for Industrial Manufacturing Workshop, held at Robotics: Science and Systems Conference (RSS).
[25] Liu, C., and Tomizuka, M., 2014. "Control in a safe set: Addressing safety in human-robot interactions". In ASME Dynamic Systems and Control Conference.
[26] Langsfeld, J. D., Kaipa, K. N., Gentili, R. J., Reggia, J. A., and Gupta, S. K., 2014. "Incorporating failure-to-success transitions in imitation learning for a dynamic pouring task". In IEEE/RSJ International Conference on Intelligent Robots and Systems.
[27] Grollman, D. H., and Billard, A., 2011. "Donut as I do: Learning from failed demonstrations". In IEEE International Conference on Robotics and Automation, pp. 3804–3809.
[28] Mainprice, J., and Berenson, D., 2013. "Human-robot collaborative manipulation planning using early prediction of human motion". In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 299–306.
[29] Abbeel, P., and Ng, A. Y., 2004. "Apprenticeship learning via inverse reinforcement learning". In ACM International Conference on Machine learning.
[30] Priess, M. C., Choi, J., and Radcliffe, C., 2014. "The inverse problem of continuous-time linear quadratic gaussian control with application to biological systems analysis". In ASME 2014 Dynamic Systems and Control Conference.
[31] Todorov, E., and Jordan, M. I., 2002. "Optimal feedback control as a theory of motor coordination". *Nature neuroscience,* ***5***(11), pp. 1226–1235.