Barrier-Lyapunov-Function-Based Backstepping Adaptive Hybrid Force/Position Control for Manipulator with Force and Position Constraints

Mohammad Adinehvand[1], Chow Yin Lai[2] and Reza Hoseinnezhad[3]

Abstract— **In this paper, we present a backstepping adaptive hybrid force/position control based on Barrier Lyapunov Function for a robotic manipulator to prevent constraint violation of applied force and position simultaneously. First, the task space is partitioned according to the constrained and unconstrained directions, and a new representation of dynamics is introduced. Next, force/position control is applied using the strict-feedback backstepping technique, in which a time-varying Barrier Lyapunov Function is employed to ensure that the force and position do not violate their constraints. Finally, to deal with uncertainty, disturbance and non-linearity of the system, an adaptive radial basis function neural network (RBFNN) is also implemented in the control algorithm. Stability proof of the proposed control method is presented, and simulation studies on a 2-link manipulator show the effectiveness as well as the performance of the proposed controller in preventing constraint violation.**

I. INTRODUCTION

For many contact tasks performed by robot manipulators (e.g. grinding, scribing, deburring and assembly-related tasks), it is desired to control the interaction force between end-effector and the environment while following a prescribed motion trajectory [1]. To achieve these tasks, several approaches have been introduced in the robotic literature which have been categorized as compliant motion control, pure force control and hybrid force/motion control [2].

Raibert and Craig [3] proposed a simple approach to control compliant motions of a robot manipulator. They divided the manipulator task configuration into position constraints along the normals to a generalized task surface and force constraints along the tangents of that surface. Slotine and Li [4] extended their joint-space adaptive control approach to free and constrained motions in Cartesian space and introduced adaptive strategies for hybrid force/motion control. Khatib [5] introduced a unified approach for motion and force control of robot manipulators in which he developed a framework for the analysis and control of manipulator systems with respect to the dynamic behaviour of end-effectors. He also defined the generalized task specification matrices which involved end-effector motion (both position and orientation) and contact forces (both forces and moments) described in the frame of reference. Lozano and Brogliato [2] introduced an approach to decompose representation of robot dynamic which enables one to clearly distinguish force, position and redundancy in joint and Cartesian space. C.M Kwan [6] also used the same approach to introduce a sliding-adaptive technique. Kouya et al. [1] presented a general scheme of an adaptive force/position control using the strict-feedback backstepping technique.

Intelligent control techniques such as neural network and fuzzy logic systems have been applied in control systems to approximate uncertainties and estimate the non-linearity of systems due to their learning capability. Kwan et al. [7] proposed a tuning scheme for a class of general nonlinear systems using neural networks (NNs) based on backstepping technique. Kwan [8] also applied the NN controller with online tuning algorithm to achieve force and motion control of constrained rigid robots.

One of the biggest challenges in robotic force control is the high impact force during the transition from non-contact phase to contact phase. A force overshoot would damage the workpiece, or result in chattering effect, where the tool keeps hitting the workpiece and bouncing back. One common approach to reduce the force overshoot is to let the robot approach the workpiece slowly. However, this increases the overall cycle time. To achieve both a minimal force overshoot and high speed response, Lai [9], [10] proposed a nonlinear damping control scheme. However, this method does not guarantee a limit of force within the constraints. This motivated us to propose a solution based on using a time-varying Asymmetric Barrier Lyapunov Function to prevent constraint violation as presented in this paper.

Although employing a suitable Lyapunov candidate function for investigating of the stability of control system is a common practice in the robotics, this approach would not guarantee to prevent constraint violation. However, the Barrier Lyapunov Function (BLF) has only recently received significant attention from the control community for its ability to prevent constraint violations. Tee et al [11] presented control designs for SISO non-linear systems in strict feedback form with an output constraint. Tee et al [12] also introduced an adaptive control which uses a time-varying BLF (TV-BLF). The TV-BLF can lead to a significant improvement because the set of feasible initial positions is expanded to the entire constrained task space. Liu et al [13] proposed an adaptive control-based neutral network to estimate the unknown function of a system and used TV-BLF to overcome

[1] Mohammad Adinehvand is a PhD student at the School of Engineering, RMIT University, Carlton, VIC 3053, Australia mohammad.adinehvand@student.rmit.edu.au

[2] Chow Yin Lai is with department of Electronic and Electrical Engineering, , University College London, United Kingdom c.lai@ucl.ac.uk

[3] Reza Hoseinnezhad is with the School of Engineering, RMIT University, Carlton, VIC 3053, Australia, reza.hoseinnezhad@rmit.edu.au.

the violation of constraints.

This paper takes advantage of TV-BLF to design force and position control systems that achieve a hybrid force/position control which does not violate the constraints. In order to design the controller, the task space is divided into two subspaces to introduce a new presentation of manipulator dynamics along the constrained and unconsidered direction. Since the dynamics can be written in a strict-feedback form, a backstepping controller incorporating the TV-BLF is used to stabilise the system and guarantee the boundedness of the position/force response. An adaptive radial basis neural network estimator is also employed to cope with uncertainties and nonlinearities of the manipulator. A stability analysis is performed to show the stability of the closed loop system, and simulation studies demonstrate the effectiveness of the proposed control algorithm.

The remainder of this paper is organized as follows. In section II, the joint and Cartesian representation of robot's dynamics, the model of the environment, and the problem formulation are presented. Section III is devoted to the development of hybrid force/motion control in the sense of TV-BLF. In section IV, a simple two-arm robot in contact with a wall is used to demonstrate the efficacy and performance of the proposed control scheme, followed by the conclusions in section V.

II. Preliminaries and Problem Formulation

A. Dynamics of Rigid Robots

Consider an n-link manipulator described in joint space:

$$M(q)\ddot{q}+V_m(q,\dot{q})\dot{q}+G(q)+F(\dot{q})+\tau_d=\tau_{act}-\tau_{int} \tag{1}$$

where $q,\ \dot{q},\ \ddot{q}\in\mathbb{R}^{n\times 1}$ are the vector of generalized displacement, velocity, and acceleration respectively in joint coordinate, $M(q)\in\mathbb{R}^{n\times n}$ is a symmetric positive definite inertia and mass matrix, $V_m(q,\dot{q})\in\mathbb{R}^{n\times n}$ is a matrix of the Coriolis and centrifugal forces, $F(\dot{q})\in\mathbb{R}^{n\times 1}$ represents the friction terms, $G(q)\in\mathbb{R}^{n\times 1}$ denotes the vector of gravity forces, $\tau_{act},\ \tau_{int},\tau_d\in\mathbb{R}^{n\times 1}$ are the joint input torque, the joint torque due to interaction force with the environment, and the disturbance torque respectively.

The Cartesian space is related to the joint space as follows:

$$X=\Omega(q),\dot{X}=\frac{\partial\Omega}{\partial q}\dot{q}\equiv J(q)\dot{q} \tag{2}$$

where $X=[X_1,X_2,\ \dots\ ,X_m]^\top\in\mathbb{R}^m$ $(m\leq 6)$ is the vector of Cartesian position and orientation of the manipulator end-effector in a frame of reference R_0 fixed on the robot base, and $\dot{X}\in\mathbb{R}^m$ is the derivatives of X with respect to time. $J(q)\in\mathbb{R}^{m\times n}$ is the Jacobian matrix.

In addition, based on the principle of conservation of energy [14], the relationship between the interaction force and the joint torque is:

$$\tau_{int}=J^\top(q)F_{int} \tag{3}$$

where $F_{int}\in\mathbb{R}^{m\times 1}$ is the interaction force with the environment in the frame of reference R_0. Therefore, the Cartesian representation of (1) can be written as follows:

$$M_x(X)\ddot{X}+V_{mx}(X,\dot{X})\dot{X}+G_x(X)+F_x(\dot{X})+F_d=F_{act}-F_{int} \tag{4}$$

B. Model of Environment

Let us to consider a transformation matrix $R\in\mathbb{R}^{m\times m}$ between compliance and operational coordinates and X,F and X^c,F^c as position and force parameters with respect to based frame and compliance frame respectively. Now, same as [15], the environment can be modelled as a frictionless and deformable plane. Thus, the elastic model of the environment leads to:

$$\begin{cases} F^c_{int}=K\left(X^c-X^c_{env}\right), & X^c\geq X^c_{env}, \\ F^c_{int}=0, & X^c<X^c_{env}. \end{cases} \tag{5}$$

where $X^c=R^\top(t)X,\ F^c=R^\top(t)F$ and $K\in\mathbb{R}^{m\times m}$ is a diagonal constant stiffness matrix and $X^c_{env}\in\mathbb{R}^{m\times 1}$ is the contact position when the robot starts to exert force on the environment.

C. Problem Formulation

In the presence of force and position constraints in operational and compliance frames, the control system is required to ensure that the robot exerts the desired force normal to a given surface without violating a given force constraint, while following a desired motion trajectory tangential to the surface, also without breaching a motion constraint. Let us assume that the force and motion outputs with respect to compliance frame are F^c and X^c, where both are required to remain in the following bounded set:

$$\begin{aligned} \underline{B}^c_{fi}(t)\leq F^c_i\leq\overline{B}^c_{fi}(t) \qquad & \forall t\geq 0\ \mathit{and}\ i\leq p \\ \underline{B}^c_{xi}(t)\leq X^c_i\leq\overline{B}^c_{xi}(t) \qquad & \forall t\geq 0\ \mathit{and}\ i>p \end{aligned} \tag{6}$$

By using (5), (6) becomes:

$$\begin{aligned} \frac{\underline{B}^c_{fi}(t)}{K_i}+X^c_{envi}\leq X^c_i\leq\frac{\overline{B}^c_{fi}(t)}{K_i}+X^c_{envi} \quad & \forall t\geq 0\ \mathit{and}\ i\leq p \\ \underline{B}^c_{xi}(t)\leq X^c_i\leq\overline{B}^c_{xi}(t) \qquad\qquad & \forall t\geq 0\ \mathit{and}\ i>p \end{aligned} \tag{7}$$

To express both force and position in term of position only, we can define:

$$\underline{B}^c(t)\leq X^c\leq\overline{B}^c(t) \tag{8}$$

where:

$$\begin{aligned} \underline{B}^c(t)=[&\frac{\underline{B}^c_{f1}(t)}{K_1}+X^c_{env1},\ \dots\ ,\frac{\underline{B}^c_{fp}(t)}{K_p}+X^c_{envp}, \\ &\underline{B}^c_{x(p+1)}(t) \qquad ,\ \dots\ ,\underline{B}^c_{xm}(t) \qquad] \end{aligned} \tag{9}$$

and

$$\begin{aligned} \overline{B}^c(t)=[&\frac{\overline{B}^c_{f1}(t)}{K_1}+X^c_{env1},\ \dots\ ,\frac{\overline{B}_{xfp}(t)}{K_p}+X^c_{envp}, \\ &\overline{B}^c_{x(p+1)}(t) \qquad ,\ \dots\ ,\overline{B}^c_{xm}(t) \qquad] \end{aligned} \tag{10}$$

By using $R^{-1}=R^\top$ and $X^c=R^\top(t)X$, (8) becomes:

$$\underline{B}(t)\leq X\leq\overline{B}(t) \tag{11}$$

where $\underline{B}(t) = R\underline{B}^c(t)$ and $\bar{B}(t) = R\bar{B}^c(t)$.

The control objective is to track the desired forces and position trajectories while ensuring that all closed loop signals are bounded and that the output constraint is not violated. The desired force and position trajectories are:

$$[F^c_{int_{d1}}, F^c_{int_{d2}}, \; \dots \; , F^c_{int_{dp}} |, X^c_{dp+1} \; \dots \; , X^c_{dm}]^\top \tag{12}$$

which can also be expressed in terms of positions only:

$$\begin{aligned} X^c_d = [&\frac{F^c_{int_{d1}}}{K_1} + X^c_{env1}, \; \dots \; , \frac{F^c_{int_{dp}}}{K_p} + X^c_{envp}, \\ &X^c_{dp+1} \; \dots \; , X^c_{dm}] \end{aligned} \tag{13}$$

by using $X^c = R^\top(t)X$ in (13), we obtain:

$$X_d = R^{-\top} X^c_d \tag{14}$$

Some assumptions are required to facilitate the control system design:

Assumption 1: There exist constants $\underline{\mathbb{B}}_{i1}$, $\underline{\mathbb{B}}_{i2}$, $\bar{\mathbb{B}}_{i1}$ and $\bar{\mathbb{B}}_{i2}$ such that $|\bar{B}_i| \leq \bar{\mathbb{B}}_{i1}$, $|\underline{B}_i| \geq \underline{\mathbb{B}}_{i1}$, $|\dot{\bar{B}}_i| \leq \bar{\mathbb{B}}_{i2}$, and $|\dot{\underline{B}}_i| \geq \underline{\mathbb{B}}_{i2}$ $\forall t > 0$.

Assumption 2: There exist functions $\bar{X}_{i0} : \mathbb{R}_+ \to \mathbb{R}_+$ and $\underline{X}_{i0} : \mathbb{R}_+ \to \mathbb{R}_+$ that satisfy $\bar{X}_{i0} < \bar{B}_i$ and $\underline{X}_{i0} > \underline{B}_i$ $\forall t$, and positive constants $\bar{X}_{i1}$ and $\underline{X}_{i1}$ such that the desired trajectory X_{id} and its time derivative satisfy $\underline{X}_{i0} < X_{id} < \bar{X}_{i0}$ and $\underline{X}_{i1} < \dot{X}_{id} < \bar{X}_{i1}$.

Several lemmas are also required to facilitate the control design. Due to the page limitation, reference for lemmas will be given without providing the details:

Lemma 1: Barrier Lyapunov Function theory in [12] will be used to establish constraint satisfaction and performance.

Lemma 2: Logarithm inequalities in [16] will be used in the stability analysis.

Lemma 3: Young's inequality in [17] will be also used in the stability analysis, especially in transforming matrices of the neural network weight to the norm of those matrices.

III. Time-varying BLF-Based Control

Let us define $\eta_1 = X$ and $\eta_2 = \dot{X}$, which lead to the strict feedback representation of equation (4) as follow:

$$\left\{ \begin{aligned} \dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= M_x^{-1} \big[F_{act} - V_{mx}\eta_2 - G_x - F_d \\ &\quad - RK \left(R^\top \eta_1 - X^c_{env} \right) \big] \end{aligned} \right. \tag{15}$$

where all the arguments of the functions are not shown for the sake of clarity. The controller can then be designed through the following steps:

<u>Step 1:</u> Design a fictitious control α_1 such that the state η_1 tracks the desired position X_d. To this end, only an upper bound is required for forces since an excessive force overshoot may damage the workpiece or result in chattering. Thus, the lower bound can be ignored. In this case, the asymmetric barrier Lyapunov Function is an excellent tool to deal with this requirement, as it allows different upper and lower limits. In this work, the asymmetric barrier Lyapunov function proposed in the literature is chosen as follows:

$$V_1(z_1) = \frac{1}{2r} \sum_{i=1}^{n} \log \frac{1}{1 - \xi_i^{2r}}. \tag{16}$$

where $z_1 = \eta_1 - X_d$ is the tracking error, and the time-varying barriers are given by

$$\begin{aligned} B_{ai} &:= X_{di} - \underline{B}_i(t) \\ B_{bi} &:= \bar{B}_i(t) - X_{di} \end{aligned} \tag{17}$$

$$h(z_{1i}) := \begin{cases} 1 & z_{1i} > 0 \\ 0. & z_{1i} \leq 0 \end{cases} \tag{18}$$

and

$$\xi_{ai} = \frac{z_{1i}}{B_{ai}}, \xi_{bi} = \frac{z_{1i}}{B_{bi}}, \; and \; \xi_i = h_i \xi_{bi} + (1 - h_i)\xi_{ai} \tag{19}$$

Note that the condition $|\xi_i(t)| < 1$ holds iff $-B_{ai} < z_{1i} < B_{bi}$ [16].

Throughout this paper, for ease of notation, we abbreviate each function by omitting its arguments, unless otherwise stated.

It is clear that V_1 is positive definite and continuously differentiable in the set $|\xi_i| < 1$. $\dot{V}_1$ is given by

$$\begin{aligned} \dot{V}_1 = \sum_{i=1}^{n} \frac{\dot{\xi}_i \xi_i^{2r-1}}{1 - \xi_i^{2r}} = \sum_{i=1}^{n} \frac{h_i \xi_{bi}^{2r-1}}{B_{bi}(1 - \xi_{bi}^{2r})} \left(\dot{z}_{1i} - z_{1i} \frac{\dot{B}_{bi}}{B_{bi}} \right) + \\ \sum_{i=1}^{n} \frac{(1 - h_i)\xi_{ai}^{2r-1}}{B_{ai}(1 - \xi_{ai}^{2r})} \left(\dot{z}_{1i} - z_{1i} \frac{\dot{B}_{ai}}{B_{ai}} \right) \end{aligned} \tag{20}$$

Define $z_2 = \eta_2 - \alpha_1$ which is another tracking error, where α_1 is a stabilizing function to be designed. Since $z_1 = \eta_1 - X_d$, $\dot{z}_1 = \dot{\eta}_1 - \dot{X}_d = \eta_2 - \dot{X}_d = z_2 + \alpha_1 - \dot{X}_d$. The stabilizing function α_1 is then designed as:

$$\alpha_1 = -\Lambda_1 z_1 + \dot{X}_d \tag{21}$$

where $\Lambda_1 = \text{diag}\left[\lambda_{11} + \bar{\lambda}_{11}, \lambda_{12} + \bar{\lambda}_{12}, \; \dots \; , \lambda_{1n} + \bar{\lambda}_{1n} \right] \in \mathbb{R}_+^{n \times n}$ and

$$\bar{\lambda}_{1i}(t) = \sqrt{\left(\frac{\dot{B}_{ai}}{B_{ai}} \right)^2 + \left(\frac{\dot{B}_{bi}}{B_{bi}} \right)^2 + \beta_i} \tag{22}$$

where λ_{1i} and β_i are positive design parameters which ensure that the time derivatives of α_1 are bounded even when both $\dot{B}_{ai}$ and $\dot{B}_{bi}$ are zero. By defining

$$\mu_i = \frac{h_{1i}}{B_{bi}^{2r} - z_{1i}^{2r}} + \frac{1 - h_{1i}}{B_{ai}^{2r} - z_{1i}^{2r}} \tag{23}$$

and the fact that

$$\bar{\lambda}_{1i} + h_{1i} \frac{\dot{B}_{bi}}{B_{bi}} + (1 - h_{1i}) \frac{\dot{B}_{ai}}{B_{ai}} \geq 0. \tag{24}$$

by substituting (21) in to the equation (20), we obtain

$$\dot{V}_1 \leq -\sum_{i=1}^{n} \frac{\lambda_{1i} \xi_i^{2r}}{1 - \xi_i^{2r}} + \sum_{i=1}^{n} \mu_i z_{1i}^{2r-1} z_{2i}. \tag{25}$$

As the first term of (25) is negative-definite in the $|\xi_i| < 1$ domain, it can be concluded that $\dot{V}_1$ will become negative-definite if the second terms is negative semi-definite. This will be ensured through the subsequent step, where the second term is cancelled.

Step 2: Design the controller F_{act} such that z_2 approaches zero, or equivalently η_2 tracks α_1 (which was designed in step 1). Since η_2 does not need to be constrained, consider a Lyapunov Function (LF) candidate by augmenting V_1 with a quadratic function [13] [11]:

$$V_2 = V_1 + \frac{1}{2} z_2^\top z_2 \tag{26}$$

Differentiating V_2 leads to

$$\dot{V}_2 = \dot{V}_1 + z_2^\top \ \dot{z}_2 \quad \Longrightarrow \dot{V}_2 = \dot{V}_1 + z_2^\top \ (\dot{\eta}_2 - \dot{\alpha}_1) \tag{27}$$

and by defining:

$$\begin{aligned} U(Z) &= [U_1(Z), \ \ldots \ , U_n(Z)]^\top \\ &= -M_x^{-1} [V_{mx}\eta_2 + G_x + F_d + \\ &\qquad RK \left(R^T \eta_1 - X_{env}^c \right) + M_x \dot{\alpha}_1] \end{aligned} \tag{28}$$

equation (27) becomes:

$$\dot{V}_2 = \dot{V}_1 + z_2^\top \left[M_x^{-1} F_{act} + U(Z) \right]. \tag{29}$$

By substituting (25) in (29), we obtain:

$$\dot{V}_2 \leq -\sum_{i=1}^{n} \frac{\lambda_{1i}\xi_i^{2r}}{1-\xi_i^{2r}} + \sum_{i=1}^{n} \mu_i z_{1i}^{2r-1} z_{2i} + z_2^\top \left[M_x^{-1} F_{act} + U(Z) \right]. \tag{30}$$

If $U(Z)$ were fully known, F_{act} could be designed as:

$$F_{act} = -M_x \left[\Lambda_2 z_2 + U(Z) + A \right] \tag{31}$$

where $\Lambda_2 = \operatorname{diag}[\lambda_{21}, \lambda_{22}, \ \ldots \ , \lambda_{2n}] \in \mathbb{R}_+^{n\times n}$ is the diagonal control gain matrix with $\lambda_{2i} > 0.5$ and A is $[\mu_1 z_{11}^{2r-1}, \mu_2 z_{12}^{2r-1}, \ \ldots \ , \mu_n z_{1n}^{2r-1}]^\top$.

Substituting (31) into (30) leads to

$$\dot{V}_2 \leq -\sum_{i=1}^{n} \frac{\lambda_{1i}\xi_i^{2r}}{1-\xi_i^{2r}} - \sum_{i=1}^{n} \lambda_{2i} z_{2i}^2 \tag{32}$$

which means the chosen ABLF (16) and LF (26) are stable and the tracking errors, z_1 and z_2, converge to a small neighborhood of zero.

In equation (31), F_{act} has been designed assuming that $U(Z)$ is known. However, in practical applications, this term is quite difficult to be obtained accurately as there are many model uncertainties and disturbances. Therefore, $U(Z)$ will be handled separately in the next step.

Step 3: Design an adaptive neural network to compensate for the unknown function $U(Z)$. In this work, the unknown function $U(Z)$ will be estimated via a radial basis function neural network (RBFNN), whose input is $Z = [X^\top, \ \dot{X}^\top, \ \alpha_1^\top, \ \dot{\alpha}_1^\top]^\top \in \Omega_Z$. The unknown function $U(Z)$ can be modeled as:

$$U_i(Z) = W_i^{*T} S_i(Z) + \varepsilon_i(Z) \tag{33}$$

where $S_i(Z) = [S_{i1}(Z), S_{i2}(Z), \ \ldots \ , S_{is_i}(Z)]^\top \in \mathbb{R}^{s_i}$, $s_i > 1$ is the number of nodes in the RBFNN and $S_{ij}(Z)$ is selected as a Gaussian function which is given by,

$$S_{ij}(Z) = \exp\left(\frac{-(Z - Z_{ij})^\top (Z - Z_{ij})}{\varphi_i^2} \right) \tag{34}$$

where Z_i is the center of the i^{th} input and φ_i is the width of the Gaussian function. $\varepsilon_i(Z)$ is the neural network approximation error, whereas W_i^* is the optimal neural network weight for which $\varepsilon_i(Z)$ would be the smallest.

Since $U(Z)$ is unknown, W^* is also unknown. Therefore, the RBFNN has to be implemented as

$$U_{i,NN} = \hat{W}_i^\top S_i(Z) \tag{35}$$

where $\hat{W}$ is an estimate for W^*. F_{act} in (31) is now implemented as:

$$F_{act} = -M_x \left[\Lambda_2 z_2 + U_{NN}(Z) + A \right] \tag{36}$$

Substituting (33), (35), (36) into (30) leads to:

$$\begin{aligned} \dot{V}_2 \leq &-\sum_{i=1}^{n} \frac{\lambda_{1i}\xi_i^{2r}}{1-\xi_i^{2r}} - \sum_{i=1}^{n} \lambda_{2i} z_{2i}^2 + \sum_{i=1}^{n} z_{2i}\varepsilon_i(Z) \\ &- \sum_{i=1}^{n} z_{2i} \tilde{W}_i^\top S_i(Z) \end{aligned} \tag{37}$$

where

$$\tilde{W}_i = \hat{W} - W_i^* \tag{38}$$

is the neural weight error. To reduce the $\tilde{W}_i$ while maintaining the stability of all states, consider the following Lyapunov function:

$$V_3 = V_2 + \frac{1}{2} \sum_{i=1}^{n} \tilde{W}_i^T \Gamma_i^{-1} \tilde{W}_i. \tag{39}$$

Differentiating (39) and using (37) gives:

$$\begin{aligned} \dot{V}_3 \leq &-\sum_{i=1}^{n} \frac{\lambda_{1i}\xi_i^{2r}}{1-\xi_i^{2r}} - \sum_{i=1}^{n} \lambda_{2i} z_{2i}^2 + \sum_{i=1}^{n} z_{2i}\varepsilon_i(Z) \\ &- \sum_{i=1}^{n} z_{2i} \tilde{W}_i^\top S_i(Z) + \sum_{i=1}^{n} \tilde{W}_i^\top \Gamma_i^{-1} \dot{\hat{W}}_i. \end{aligned} \tag{40}$$

The adaptive control law is designed as follows [13]:

$$\dot{\hat{W}}_i = \Gamma_i \left[z_{2i} S_i(Z) - \sigma_i \hat{W}_i \right] \tag{41}$$

where $\Gamma_i = \Gamma_i^\top > 0$ is the gain matrix, and $\sigma_i > 0, i = 1, 2, \ \ldots \ , n$ is a small constant.

By using the adaptive control law (41), (40) becomes:

$$\begin{aligned} \dot{V}_3 \leq &-\sum_{i=1}^{n} \frac{\lambda_{1i}\xi_i^{2r}}{1-\xi_i^{2r}} - \sum_{i=1}^{n} \lambda_{2i} z_{2i}^2 + \sum_{i=1}^{n} z_{2i}\varepsilon_i(Z) \\ &- \sum_{i=1}^{n} \sigma_i \tilde{W}_i^\top \hat{W}_i \end{aligned} \tag{42}$$

Based on lemma 2 and 3 and by substituting (38) into (42) we can obtain:

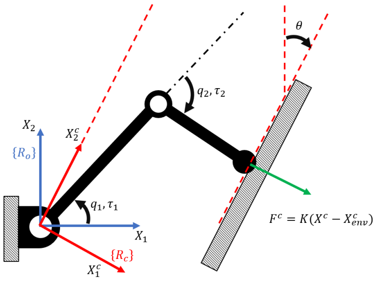

Fig. 1: Two link manipulator

$$\dot{V}_3 \leq -\sum_{i=1}^{n} \lambda_{1i} \, \log \frac{1}{1-\xi_i^{2r}} - \sum_{i=1}^{n} \left(\lambda_{2i} - \frac{1}{2}\right) z_{2i}^2 + \frac{1}{2}\sum_{i=1}^{n} \varepsilon_i^2(Z)$$
$$-\frac{1}{2}\sum_{i=1}^{n} \sigma_i \|\tilde{W}_i\|^2 + \frac{1}{2}\sum_{i=1}^{n} \sigma_i \|W_i^*\|^2. \tag{43}$$

By selecting parameters ρ and C as follows:

$$\rho = \min\left[2r\lambda_{1i}, 2\lambda_{2i} - 1, \sigma_i \lambda_{\min}(\Gamma_i)\right]$$
$$C = \frac{1}{2}\sum_{i=1}^{n} \sigma_i \|W_i^*\|^2 + \frac{1}{2}\sum_{i=1}^{n} \varepsilon_i^2(Z) \tag{44}$$

where $\lambda_{\min}(\Gamma_i)$ is the minimum eigenvalue of Γ_i, (43) becomes

$$\dot{V}_3 \leq -\rho V_3 + C. \tag{45}$$

Therefore, based on Lemma 1, z_1, z_2 and $\hat{W}$ in the close-loop system will remain uniformly ultimately bounded. By multiplying both side of (45) by $e^{\rho t}$ and integrating over $[0,t]$, we can indicate that V_3 is bounded as follows:

$$0 \leq V_3(t) \leq V_3(0)e^{-\rho t} + \frac{C}{\rho}. \tag{46}$$

Now, by assuming that $\xi \to 1$, then the Lyapunov function $V_3 \to \infty$ which contradicts the boundedness of the Lyapunov function. This means that ξ cannot grow to 1.

IV. Simulation

Consider a two link robot with two revolute joints as shown in Fig.1. The robot model and its parameters can be found in [18].

In this particular case, $R(\theta = 30°)$ is a constant matrix and the desired force and position are given as 10 N and $-0.4\sin(t)\exp(-t)$ m respectively. It should be noted that force and position are also required to remain in the set: $max(F^c - F_d^c) < 1N$ and $max(X^c - X_d^c) < 0.06m$.

The performance of the proposed controller in tracking the desired position and force trajectories are shown in figures 2 and 3 respectively. As can be seen, the proposed controller successfully prevented the violation of the force and position constraints. Figures 4 and 5 show that the force and position errors converge to small values around zero.

Finally, the corresponding neural network weights are given in figure 6. It can be seen that the weights slowly converges and will not grow unbounded.

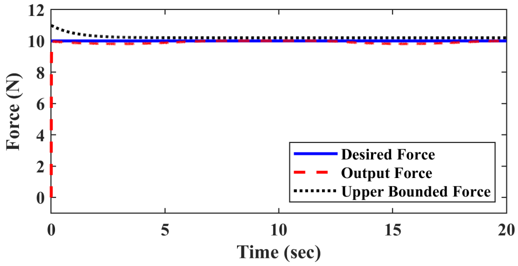

Fig. 2: Force Trajectory

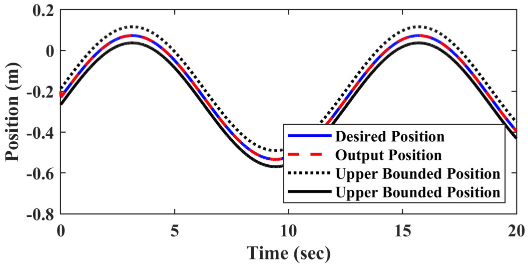

Fig. 3: Position Trajectory

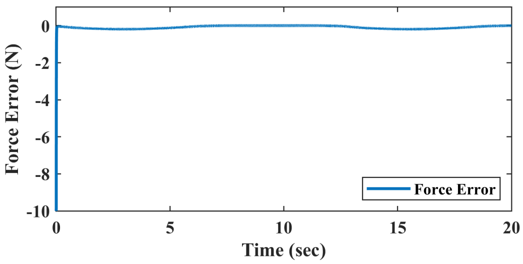

Fig. 4: Force Error

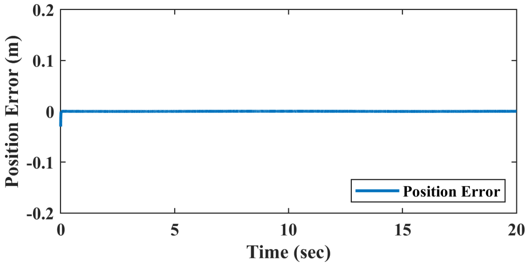

Fig. 5: Position Error

For comparison purpose, we also ran a simulation for a controller which is derived purely based on conventional Quadratic Lyapunov Function (QLF) which does not take constraints into consideration. The comparison result is shown in figure 7, in which we can see that the applied force exceeded the desired bounds using the QLF approach, whereas with the BLF approach the applied force remained in desired set.

V. Conclusion

In this paper, a backstepping adaptive hybrid force/position control which takes the manipulator dynamics into consideration has been introduced to prevent constraint violation of the applied force and position. Firstly, we introduced a new representation of dynamics based on the constrained and

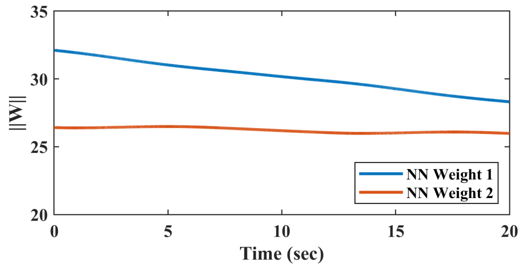

Fig. 6: Neural network weights

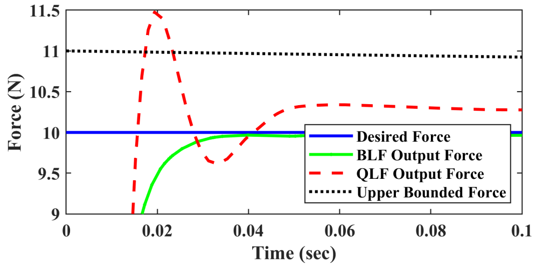

Fig. 7: BLF compared with QLF

unconstrained directions. Next, based on the strict-feedback backstepping technique, a force/position control was designed by incorporating a Time-Varying Barrier Lyapunov Function (BLF), which grows to infinity when its arguments approach some finite limits. By defining proper bounded set for both force and position, the Barrier-Lyapunov-Function-based controller in the closed loop ensures that these limits are not transgressed. Finally, an appropriate RBFNN weight update law were implemented to compensate uncertainty, disturbance and non-linearity of the system and improve the system robustness. A stability analysis has been provided, and finally, the effectiveness of the proposed controller was illustrated via simulations on a two-link manipulator. Simulations showed that the actual force and position do not exceed the limits by using the BLF controller, in contrast to the case of using a conventional quadratic Lyapunov Function.

VI. ACKNOWLEDGEMENT

The authors greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organisations and the Australian Federal Government's Business Cooperative Research Centres Program) through Project R2.7.10 – Miniature Crawling Robots for Rolling Stock Manufacture and Maintenance.

REFERENCES

[1] D Nganga-Kouya, Maarouf Saad, and Louis Lamarche. Backstepping adaptive hybrid force/position control for robotic manipulators. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4595–4600. IEEE, 2002.

[2] R Lozano and B Brogliato. Adaptive hybrid force-position control for redundant manipulators. *IEEE Transactions on Automatic Control*, 37(10):1501–1505, 1992.

[3] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[4] J-J Slotine and Weiping Li. Adaptive strategies in constrained manipulation. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 595–601. IEEE, 1987.

[5] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[6] CM Kwan. Hybrid force/position control for manipulators with motor dynamics using a sliding-adaptive approach. *IEEE Transactions on Automatic Control*, 40(5):963–968, 1995.

[7] Chiman Kwan and Frank L Lewis. Robust backstepping control of nonlinear systems using neural networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(6):753–766, 2000.

[8] CM Kwan, A Yesildirek, and FL Lewis. Robust force/motion control of constrained robots using neural net network. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 2, pages 1862–1867. IEEE, 1994.

[9] Chow Yin Lai, Yuan Ping Li, Ngoc Dung Vuong, Tao Ming Lim, Chong You Ma, and Chee Wang Lim. Nonlinear damping for improved transient performance in robotics force control. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 314–319. IEEE, 2012.

[10] Chow Yin Lai. Improving the transient performance in robotics force control using nonlinear damping. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 892–897. IEEE, 2014.

[11] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[12] Keng Peng Tee, Rui Yan, and Haizhou Li. Adaptive admittance control of a robot manipulator under task space constraint. In *2010 IEEE International Conference on Robotics and Automation*, pages 5181–5186. IEEE, 2010.

[13] Yan-Jun Liu, Shumin Lu, and Shaocheng Tong. Neural network controller design for an uncertain robot with time-varying output constraint. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(8):2060–2068, 2017.

[14] Darren M Dawson, Chaouki T Abdallah, and Frank L Lewis. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[15] R Carelli and R Kelly. Adaptive hybrid impedance/force controller for robot manipulators. *IFAC Proceedings Volumes*, 23(8):255–261, 1990.

[16] Wei He, Haifeng Huang, and Shuzhi Sam Ge. Adaptive neural network control of a robotic manipulator with time-varying output constraints. *IEEE transactions on cybernetics*, 47(10):3136–3147, 2017.

[17] Tingting Gao, Yan-Jun Liu, Lei Liu, and Dapeng Li. Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints. *IEEE/CAA Journal of Automatica Sinica*, 5(5):923–933, 2018.

[18] Wei He, Yuhao Chen, and Zhao Yin. Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE transactions on cybernetics*, 46(3):620–629, 2016.