NN Controller of the Constrained Robot under Unknown Constraint
Shenghai Hu, Marcelo H. Ang Jr.*, and H. Krishnan
Dept of Mechanical and Production Engineering
National University of Singapore
Singapore 119260
*mpeangh@nus.edu.sg

Abstract

In this paper, the problems faced in the constrained force control is studied (uncertainties in dynamic model and the unknown constraints). A neural network (NN) controller is proposed based on the derived dynamic model of robot in the task space. The feed-forward neural network is used to adaptively compensate for the uncertainties in the robot dynamics. Training signals are proposed for the feed-forward neural network controller. The NN weights are tuned on-line, with no off-line learning phase required. An on-line estimation algorithm is developed to estimate the local shape of the constraint surface by using measured data on the force and position of the end-effector. The suggested controller is simple in structure and can be implemented easily. Real-time experiments are conducted using the five-bar robot to demonstrate the effectiveness of the proposed controller.

1. Introduction

To apply the robot manipulator to a wider class of tasks, it will be necessary to control not only the position of a manipulator but also the force exerted by the end-effector on an object. To accomplish such tasks successfully, several approaches have been proposed such as impedance control [1], hybrid position/force control [2] and constrained motion control [3].

Constrained motion control has been extensively studied in recent years. In constrained motion control, the robot end-effector is assumed to be in contact with rigid frictionless surface. As a result kinematic constraints are imposed on the manipulator motion, which correspond to some algebraic constraints among the manipulator state variables. It is necessary to control both the motion on the constraint surfaces and the generalized constrained force.

A general theoretical framework of constrained motion control is rigorously developed in [3]. The controller is addressed by assuming exact knowledge of the robot dynamic model and the constraints. More recently, the research is extended to force/motion control of constrained robots with uncertainties using adaptive control, sliding mode control, robust control, etc.

The ability of neural network to approximate arbitrary non-linear functions and to learn through examples lends it to many useful applications in the control engineering. Many researchers have applied the NN in robot motion control with substantial success [4,5,6]. Some research works [7,8,9] have also dealt with NN controller design in robot force control.

The above constrained force controller are all based on the assumption that the constraint equations can be exactly described. This assumption is the key to obtain the needed nonlinear coordinate transformation so that the robot dynamic equation can be transformed into a reduced form, which enable position control and force control to be designed separately.

Due to task kinematic modelling inaccuracy, it is not possible to know the constraints exactly, either due to lack of information on the shape and size of the object or inaccurate positioning of the otherwise precisely known object. Furthermore, even when it is possible to give the constraint hypersurfaces, it would be much better to be able to perform the force control task without a very precise description of the constraints.

Take these points into account, in this paper a nonlinear transformation is used to derive the robot dynamics into the format suitable for the controller design. A NN control law with online constraints estimation is proposed based on the derived equations. Experimental results illustrate the effectiveness of the proposed controller in presence of the uncertainties and unknown constraints.

2. Dynamic Model of Constrained Robot Manipulator

Dynamic equation of a general rigid link manipulator having n degree of freedom in contact with the environment can be written as

\[ M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = f + u \]  

(1)

where \( q \in \mathbb{R}^n \) is the generalized joint vector, \( \tau \in \mathbb{R}^n \) is the applied joint torque; \( M(q) \in \mathbb{R}^{n \times n} \) denotes the generalized moments of inertia, the term \( C(q,\dot{q})\dot{q} \in \mathbb{R}^n \) is the vector characterizing Coriolis and Centrifugal forces, \( G(q) \in \mathbb{R}^n \) is the gravitational forces, \( f \in \mathbb{R}^n \) is the vector of constraint forces in joint space.

Let \( \mathbf{r} \in \mathbb{R}^m \) denote the generalized position vector of the end-effector in Cartesian space. Suppose that the environment is described by a set of \( m \) rigid hypersurfaces.
\( \Phi(r) = 0, \Phi(r) = \phi_1(r), \ldots, \phi_m(r) \) \( m \leq n \) which are mutually independent, and \( \phi_i(r) \) is assumed to be twice differentiable with respect to \( r \). When motion of the robot is constrained on the surfaces \( (2) \), only \( (n-m) \) coordinates of the position vector can be specified independently. As pointed out in \([3]\), control of all the position coordinates of the robot is unnecessary, and only \( (n-m) \) position coordinates need to be controlled in the constrained motion. Hence, \( \Phi(r) \) is assumed to be twice continuously differentiable and independent of \( \Phi(r) \) in the finite workspace \( \Omega \). Thus, once \( \Phi(r) \) is regulated to the workspace \( \Omega \), the position \( \Phi(r) \) is uniquely determined.

The relation between the end-effector position \( r \) and joint vector \( q \) is given by
\[
\begin{align*}
    r &= h(q), \\
h: R^n &\rightarrow R^m
\end{align*}
\] where the mapping \( h \) is invertible and twice continuously differentiable.

Here we select the nonlinear transformation as
\[
\begin{align*}
    r_c &= [r^T_c, r^T_d] \\
r_p &= [\psi_p(r), \ldots, \psi_{n-1}(r)] \\
r_f &= [\phi_f(r), \ldots, \phi_{n-1}(r)]
\end{align*}
\] Differentiate (4), we have
\[
\dot{r}_c = \begin{bmatrix} r_p \\ 0 \end{bmatrix} = E_p \dot{r}_p = E\dot{r}
\] where
\[
E_p = \begin{bmatrix} e_1 \ldots e_{n+1} \end{bmatrix} \quad E_f = \begin{bmatrix} \frac{\partial \phi_1(r)}{\partial r} \ldots \frac{\partial \phi_{n-1}(r)}{\partial r} \end{bmatrix}, j = 1, \ldots, n-m
\] Matrices \( E_p \) and \( E_f \) represent the position and force control directions, respectively. Conversely, by differentiating (3) we obtain
\[
\dot{r} = J\dot{q}
\] where \( J = \partial h(q)/\partial q \). Let \( f_c = [f_{c1}^T, f_{c2}^T] \) be the generalized force representation of the force exerted by the end-effector, corresponding to the generalized coordinate vector \( r_c = [r^T_c, r^T_d] \). According to the assumption of frictionless constraints, we can have \( f_{c1} = 0 \) and \( f_{c2} \) is the normal force exerted on the constraints. The force exerted on the constraints in the joint space can be represented as,
\[
f = \begin{bmatrix} \partial \Phi(r)/\partial q \end{bmatrix}^T f_r = \begin{bmatrix} \partial \Phi(r)/\partial r \end{bmatrix}^T \begin{bmatrix} \partial \Phi(r)/\partial q \end{bmatrix}^T f_r = J^T E_f f_r
\] Substitute (5), (8), (9) into (1), we can get the dynamic equation in the new coordinate
\[
M(q)J^{-1} \left\{ E^{-1} \begin{bmatrix} \tau_f \\ 0 \end{bmatrix} - \dot{\Theta} \right\} - \dot{\Theta} + C(q, \dot{q})\dot{q} + G = \tau + J^T E_f f_r
\] \( (10) \)

3 Proposed NN controller Scheme

Based on the derived equation (10) and consider the model uncertainties, a NN controller is proposed,
\[
\tau = \tilde{M}(q)J^{-1} \left\{ E^{-1} \begin{bmatrix} u_i \\ 0 \end{bmatrix} - \dot{\Theta} \right\} - \dot{\Theta} + \dot{C}(q, \dot{q})\dot{q}
\] \[ + \tilde{G} - J^T E_f^T u_z + v \] \( (11) \)
where \( u_i = \tilde{r}_p + K_{r_p}(\tilde{r}_p - \tilde{r}_p) + K_{r_d}(r_{d2} - r_p) \) \( (12) \)
\[ u_z = f_{z2} + K_{f_p} (f_{z2}(t) - f_p(t)) + K_{f_d} \left[ f_{z2}(r_{d2}) - f_p(r_p) \right] \] \( (13) \)
\( K_{r_p}, K_{r_d} \) are diagonal \( (n-m) \times (n-m) \) symmetric positive definite matrices. \( K_{r_p} \) and \( K_{r_d} \) are diagonal \( m \times m \) symmetric positive matrix. \( \tilde{M}, \tilde{C}, \tilde{G} \) are the estimates of matrix \( M, C, G \). \( v \) is the output of the neural network controller.

Let \( e_r = r_{d2} - r_p \) and \( e_z = f_{z2} - f_p \). \( (14) \)

Substituting the control law (14) into the (10), we can obtain the close loop error as
\[
\tilde{M}J^{-1} \left\{ E^{-1} \begin{bmatrix} \dot{\tilde{e}}_p + K_{r_p}\dot{\tilde{e}}_p + K_{r_d}\dot{e}_p \end{bmatrix} \right\} \\
- J^T E_f \left( K_{f_p} + I \right) \tilde{e}_p + K_{f_d} \left[ e_z \right] d\tau = \Theta - v 
\] \( (15) \)
where \( \Theta = \Delta M J^{-1} \left\{ E^{-1} \begin{bmatrix} \dot{r}_p \\ 0 \end{bmatrix} - \dot{\Theta} \right\} - \dot{\Theta} + \Delta C \dot{\Theta} + \Delta G \) \( (16) \), it represents the uncertainties in the dynamic model of the robot manipulator, and \( \Delta M = M - \tilde{M}, \Delta C = C - \tilde{C}, \Delta G = G - \tilde{G} \).

Define the partition matrix \( P \),
\[
P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} I_{n-m} & 0 \\ 0 & I_n \end{bmatrix} = I_{\nu_n}
\] \( (17) \)
Using this partition matrix \( P \), we can derive the position and force error equation as following
\[
\dot{e}_r + K_{r_p}\dot{e}_p + K_{r_d}\dot{e}_p = (P E^{-1} J^+ \tilde{M} J^{-1} \dot{E}_f )^{-1} (\Theta - v) \] \( (18) \)
\[
(K_{f_p} + I)\dot{e}_p + K_{f_d} \int e_z d\tau = P E^{-1} J^+ [\tilde{M} J^{-1} \dot{E}_f^{T} ] (\dot{\tilde{e}}_p + K_{r_p} \dot{\tilde{e}}_p + K_{r_d} \dot{e}_p) - \Theta + v \] \( (19) \)
In the ideal case, \( \Delta M = \Delta C = \Delta G = 0 \) from (16) we can get \( \Theta = 0 \). Thus from (18) and (19) we find that with the control law (11), the position error approaches zero exponentially and the force error is identically equal to zero even if the NN is not used.

Since there always exists uncertainties in the robot dynamics model, the ideal dynamic model assumption is not valid in general, that is \( \Theta \neq 0 \). Therefore the position and force error will not go zero.

Define an error signal \( \xi \) as
\[
\xi = \dot{e}_r + K_{r_p}\dot{e}_p + K_{r_d}\dot{e}_p
\] \( (20) \)
The NN control objective is to generate $v$ to reduce $\xi$ in (20) to zero. Therefore here we use $\xi$ as the training signal for the NN. The ideal value of $v$ at $\xi = 0$ is,

$$v = \Theta = \Delta MJ^{-1}(E^{-1}P_0^\top r_f + E^{-1}\dot{E}\ddot{q}) + \Delta C\dot{q} + \Delta G$$ (21)

Clearing minimizing the error signal $\xi$ allows us to achieve the required force and motion tracking control objectives.

4. Neural Network Compensator Design

The two-layer feedforward neural network (with $3(n-m)$ inputs and $n$ outputs) shown in Figure 1 is used as the compensator. It is composed of an input layer, one non-linear hidden layer, and a linear output layer. The mathematical representation of the network is given by

$$v = W^2 f(W^1 x + b^1) + b^2$$ (22)

where $W^1 \in \mathbb{R}^{n \times (n-m)}$, $W^2 \in \mathbb{R}^{n \times m}$ denote the interconnection weights for the hidden and output layers respectively, $b^1 \in \mathbb{R}^{n-m}$, $b^2 \in \mathbb{R}^n$ the bias terms to the nodes of the corresponding layer, $x \in \mathbb{R}^{(n-m)\times 1}$. The nonlinear function $f(\bullet)$ is a sigmoid function bounded in magnitude between 0 and 1:

$$f(\bullet) = \frac{1}{1 + \exp(-\bullet)}$$ (23)

![Figure 1. NN Controller Structure](image)

The weight updating law minimizes the objective function $J$, which is a quadratic function of the training signal $\xi$ i.e.

$$J = \frac{1}{2} \xi^T \xi$$ (24)

Differentiating (24) yields the gradient of $J$ as follows:

$$\frac{\partial J}{\partial w} = \frac{1}{2} (\frac{\partial^2 \xi^T \xi}{\partial w} + \frac{\partial \xi^T \xi}{\partial w} \frac{\partial \xi}{\partial w}) = \frac{\partial \xi^T \xi}{\partial w}$$ (25)

and $\xi$ is obtained from (20) and (18) as

$$\xi = (PE^{-1}J^{-1}\dot{M}J^{-1}E^{-1}P_0^\top)^{-1}(\Theta - v)$$ (26)

then, we have

$$\frac{\partial \xi^T \xi}{\partial w} = -\frac{\partial \xi^T \xi}{\partial w}(PE^{-1}J^{-1}\dot{M}J^{-1}E^{-1}P_0^\top)^{-1}$$ (27)

The back-propagation update rule for the weights with a momentum term is now chosen as

$$\Delta w = -\eta \frac{\partial J}{\partial w} + \eta \frac{\partial \xi^T \xi}{\partial w}(PE^{-1}J^{-1}\dot{M}J^{-1}E^{-1}P_0^\top)^{-1}$$

$$w(t) = w(t-1) - \eta \frac{\partial J}{\partial w}(PE^{-1}J^{-1}\dot{M}J^{-1}E^{-1}P_0^\top)^{-1}$$ (28)

where $\eta$ is the update rate and $\alpha$ is the momentum coefficient.

5 Online Estimation of The Unknown Constraint Surfaces

One major difficulty in implementing the NN controller described in the previous section is that a precise equation (2) of the constraint hypersurfaces is difficult to obtain.

Several research efforts related to estimation of the constraint surface for force control [10,11] have been proposed. Merlet [11] proposed using force measurement to determine the surface normal. T. Yoshikawa and A. Sudou [10] proposed the method combining the position and force measurement to treat the three-dimensional (3-D) situation directly and provided a mean for cancellation of frictional force. In this paper, a new estimation algorithm is proposed and provided a mean for cancellation of friction force.

First, we note that, for calculating $\tau$ of the NN control law (10) at any instant of time, we do not need to know the whole function $\phi(r)$ but only the current values of $E$ and $\dot{E}$ which represent a local property of the function $\phi(r)$. Hence, if we can estimate the current values of $E$ and $\dot{E}$ from available measurement, we would be able to perform NN constrained force controller for unknown constraint hypersurfaces. In the following, such an estimation algorithm will be given.

Similar to [10], the assumptions are made as following:

1) The end-effector position and the translational forces in 3-D space are of concern. The orientation need not be considered because either it is arbitrary or it is fixed. So the position vector $r$ and force vector $f$, are 3-D vectors $(n = 3)$.

2) The real trajectory of the end-effector position and force $f$, is measurable.

3) The number of constraint hypersurface is known to be one $(m=1)$ but its equation is not known. This means that the end-effector position is constrained on an unknown smooth 2-D curved surface.

4) The end-effector is required to track the intersection of an appropriately specified plane $Q$ and the constraint surface in a specified direction while applying a specified force in the normal direction of the constraint surface. This plane is represented by

$$W(r) = 0$$ (29)

and will be called the virtual constraint plane.

5) An estimate of matrix $E$ at the initial time $\dot{E}_0$ is given.

Suppose that the end-effector is moving on the constraint surface near the virtual constraint plane $Q$. Based on the measured force $f(t)$, which is exerted on
Therefore, the friction coefficient can be obtained from see Figure 2) of the constraint surface. If the constraint surface is frictionless, we can obtain nominal tangential and normal direction of the contact point. Here, there exists error in the normal and tangential component, then we can obtain the correct actual direction as shown in Figure 3. If we can know the force component, there exists error in the nominal and actual direction. Let us construct the force control input using (30-32). Let

$$E(t) = \left[ e_i(t), e_i(t), e_i(t) \right]$$

and

$$\dot{E} = \frac{E(t) - E(t - \Delta t)}{\Delta t}$$

Using these values, we can calculate the linearizing control input $\tau$.

Next, we consider the NN controller. Let the velocity and acceleration of the end-effector in $e_i$ and $e_i$ direction be $\dot{r}_i = \begin{bmatrix} \dot{r}_{i_1} & \dot{r}_{i_2} \end{bmatrix}$ and $\ddot{r}_i = \begin{bmatrix} \ddot{r}_{i_1} & \ddot{r}_{i_2} \end{bmatrix}$. In addition, let $u_i = \begin{bmatrix} u_{i_1} & u_{i_2} \end{bmatrix}$. We give the position control algorithm in the $e_i$ direction by

$$u_{i_1} = k_{p_i} \dot{r}_{i_1} + k_{d_i} (\dot{r}_{i_1} - \dot{r}_{i_1}) + k_{f_i} \Delta r_{i_1}$$

$$\Delta r_{i_1} = \int (\dot{r}_{i_1} - \dot{r}_{i_1}) dt'$$

where $\dot{r}_{i_1}$ and $\dot{r}_{i_1}$ are desired value for $\dot{r}_{i_1}$ and $\dot{r}_{i_1}$, and $k_{p_i}$, $k_{d_i}$, and $k_{f_i}$ are constant feedback gains. Since the desired velocity and acceleration in the $e_i$ direction is zero, we given the control algorithm in that direction by

$$u_{i_2} = k_{p_i} \dot{r}_{i_2} + k_{d_i} (\dot{r}_{i_2} - \dot{r}_{i_2}) + k_{f_i} \Delta r_{i_2}$$

$$\Delta r_{i_2} = \int (\dot{r}_{i_2} - \dot{r}_{i_2}) dt'$$

where $\Delta r_{i_2}$ is an approximated deviation of the end-effector position from the virtual plane measured in the direction of $e_i$ and $k_{p_i}$, $k_{d_i}$, and $k_{f_i}$ are constant feedback gains.

As for the force control algorithm in the $e_i$ direction, we construct the force control input $u_i$ by

$$u_i = f_{i_1} + k_{f_i} \Delta r_{i_1} + k_{f_i} \int (f_{i_1} - f_{i_1}) dt'$$

where $k_{f_i}$ and $k_{f_i}$ are constant feedback gains.

## 5 Experimental Results

Real-time implementation of the NN controller has been carried out using the five-bar linkage parallelogram robot with two degrees of freedom (DOF) as shown in Figure 4. The robot was designed and built at our laboratory for our experimental work. Experiments have been performed on this robot to evaluate the effectiveness of the proposed neural network controller.

The robot dynamic coefficient matrices in equation (1) and its forward kinematics are given by

$$M(q) = \begin{bmatrix} 1.747 & -0.467 \cos(q_2 - q_1) \\ -0.467 \cos(q_2 - q_1) & 1.439 \end{bmatrix}$$

$$C(q_1, \dot{q}_1, \ddot{q}_1, q_2, \dot{q}_2, \ddot{q}_2, \dot{q}_2) = \begin{bmatrix} 0 & 0.476 \sin(q_2 - q_1) \dot{q}_1 \\ -0.476 \sin(q_2 - q_1) \dot{q}_1 & 0 \end{bmatrix}$$
The robot is required to move along the rigid unknown constraints and exert 10 N force on it as shown in Figure 4.

The robot contact with the constraints initially $E_0 = [0.5, 0.6, 0]$ and $E_v = [0 - 1 0]$, the virtual plane is specified as a horizontal plane, that is, a plane including the above initial position and having $[0, 0, -1]^T$ as the normal vector. The desired trajectory is to remain at the initial position for 2s, then to move in the $e_i$ direction with a trapezoidal velocity curve (with constant acceleration of $0.01 \text{m/s}^2$ for the first 1s, with zero acceleration for the middle time and $-0.01 \text{m/s}^2$ for the last 1s). The desired force trajectory is 10N. The experimental results are shown in Figure 5-9. The velocity response in the $e_i$ direction is shown in Figure 5. Force response in the $e_3$ direction is plotted in Figure 6. The output of the neural network $v(t)$ is shown in Figure 7. Figure 8 and Figure 9 show x, y components of the estimated directions $e_i$ and $e_3$ of the constraints.
6. Conclusion

A simple and effective neural network controller, which uses the two-layer feed-forward neural network, is proposed for the constrained robot in presence of the unknown constraints. On-line estimation algorithm of the unknown constraints using the force and position measurement is also presented. Experiments have been carried out on a 2DOF direct-drive robot manipulator to evaluate the performance of the proposed controller. The experimental results show that good performance can be attained even in presence of uncertainties in the robot model and unknown constraints.

References


