Adaptive Hybrid Control for Omnidirectional Mobile Manipulators Using Neural-Network

Xiang-min Tan[1], Dongbin Zhao[1, 2], *Member, IEEE*, Jianqiang Yi[1], *Member, IEEE* and Dong Xu[1]

Abstract—**An omnidirectional mobile manipulator, due to its large-scale mobility and dexterous manipulability, has attracted lots of attention in the last decades. However, modeling and control of such a system are very challenging because of its complicated mechanism. In this paper, we achieve the kinematics of the mobile platform according to its mechanical structure firstly, and then deduce its unified dynamic model by Lagrangian formalism. By applying the unified model to calculate the coupling torque vector between the mobile platform and the robot arm, an adaptive hybrid controller is proposed subsequently. This controller consists of two parts: one is responsible for the tracking control of the mobile platform in kinematics. The other part is for the robot arm in dynamics. For further consideration of unmodeled dynamics and external disturbances, a Radial Basis Function Neural-Network (RBFNN) is adopted in the adaptive controller. Simulation results show the correctness of the presented model and the effectiveness of the control scheme.**

I. Introduction

A mobile manipulator, which generally consists of a mobile platform and a robot arm, provides a new direction in robot researches and applications due to its large-scale mobility and dexterous manipulability. The mobility of the mobile platform substantially increases the size of workspace, and enables the end-effector of the manipulator to reach a relatively better position to operate dexterously. Meanwhile, the manipulability of the robot arm greatly improves the functionality of the mobile manipulator. Because of these distinct advantages, mobile manipulators have been applied more and more extensively.

Compared to the differential-driven mobile platform, the omnidirectional mobile platform has all three degrees of freedom (DOFs) in the horizontal motion plane. Therefore, it can completely use the null space motions to improve the workspace and overall dynamic endpoint properties. A number of related works have been developed in this field in the last decades. In literatures [1, 2, 3], modeling and control of omnidirectional mobile robots were analyzed in details. The dynamic model and kinematic model of an omnidirectional mobile robot with three castor wheels were presented in [4, 5]. Tan and Xi [6] proposed a unified dynamic model for a mobile manipulator consisting of a Nomadic XR4000 mobile platform and a Puma 560 robot arm. Holmberg and Khatib [7, 8, 9] developed a holonomic mobile robot and presented a dynamic control scheme for a parallel redundant system. Liu and Lewis presented a decentralized robust controller for mobile manipulators in [10, 12], and Chung and Velinsky [11] also derived the dynamic model of the holonomic mobile platform and the manipulators separately.

On the whole, existing works can be approximately divided into two groups: One is modeling and controlling of the mobile platform and the robot arm separately [10, 11], i.e., the mobile manipulator is regarded as two subsystems. The other is the integrated modeling and control approach, that is, the mobile platform is regarded as a multiple-DOFs joint [6]. For the former strategy, although it is relatively easy to get the dynamic models of the mobile platform and the robot arm respectively, its difficulty lies in the calculation of the coupling torque vector between the mobile platform and the robot arm. For the latter strategy, the difficulty lies in the computational complexity. In addition, control of such systems is very challenging even if we have derived the overall dynamic model. Many intelligent control schemes are unable to be applied online due to the computational complexity of the model. Furthermore, the difference of the dynamic response between the mobile platform and the robot arm is significant in practical applications. This characteristic leads to some difficulties for designing the controller as a whole.

Computed Torque Control method (CTC) is applied widely in mechanical systems. However, requirements of precise model are very rigor in practical applications. Furthermore, this kind of controllers becomes unstable when the unmodeled dynamics or the external disturbance is significant. Song and Yi [13] adopted a fuzzy approach to compensate the uncertainties.

In order to overcome these difficulties referred above, an adaptive hybrid control scheme, combining the kinematics of the mobile platform and the unified dynamic model of the mobile manipulator, is proposed in this paper. The proposed controller consists of two parts: one is responsible for the tracking control of the mobile platform in kinematics. The other part, which is capable of compensating the coupling force between the mobile platform and the robot arm, is for the robot arm in dynamics. Because neural network provides a fast method of autonomously learning the relation between a set of output states and a set of input states, we introduce a

Manuscript received September 21, 2007. This work was partly supported by NSFC Projects under Grant No. 60475030 and No.60621001, the Joint Lab of Intelligent Sciences & Technology under Grant No.JL0605, and the National 863 Program No.2007AA04Z239, China.

1. Xiang-min Tan, Dongbin Zhao, Jianqiang Yi and Dong Xu are with lab of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences. 95 Zhongguancun East Road, Haidian District, Beijing 100080, P.R. China. (phone: 86-10-82615422; fax: 86-10-62658815; e-mail: xiangmin.tan@ia.ac.cn).

2. Dongbin Zhao is also with the University of Arizona as a visiting scholar funded by China Scholarship Council.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

RBFNN into the controller to approximate the unstructured or structured uncertainties of the proposed model.

In the following section, we apply Lagrangian formalism to obtain the unified model of an omnidirectional mobile manipulator. In addition, we also introduce the structure of the mobile manipulator and deduce the kinematics of its mobile platform. Section III is devoted to controller design based on the presented kinematic model and the dynamic model using the computed torque control method (CTC) and a RBFNN. Section IV includes simulation results to validate feasibility and efficiency of the proposed method. Some conclusions and remarks are finally included in Section V.

II. System Descriptions

A mobile manipulator generally consists of a mobile platform and a robot arm. Fig.1 shows the omnidirectional mobile manipulator and the mechanical structure of its mobile platform in our lab. In this experimental setup, we adopt an omnidirectional mobile platform driven by three identical castor wheels, and these three identical castor wheels are fixed symmetrically on the bottom of the mobile platform, as is shown in Fig.1 (right). The angle between two neighboring wheels is $2\pi/3$. ***Wheel 1, 2, 3*** are assigned clockwise to indicate the three wheels. The center of the mobile platform is indicated by ***C***. Each identical castor wheel consists of a rolling motor and a steering motor. Therefore, the mobile platform has all three DOFs (degrees of freedom) $[x, y, \theta_b]$ for moving on the horizontal plane, and isn't subject to nonholonomic constraint. The robot arm, mounted on the omnidirectional platform, is similar to a SCARA manipulator.

The side-view of the identical castor wheels is shown in Fig.2 (1). The radius of each wheel is *r*, and the horizontal distance between *C* and the center F_i of the vertical axis of each wheel is *D*. The offset *d* is the horizontal distance between the center of each wheel O_i and F_i. The angle displacements for wheel rolling and steering are φ_i and η_i, respectively. Fig.2 (2) shows the top view of a wheel, and $\beta_i = \eta_i + \alpha_i + \theta_b$ is the orientation of ***Wheel i*** (i=1,2,3)in the global frame OXY, and $\alpha_1 = 0$, $\alpha_2 = -2\pi/3$, $\alpha_3 = 2\pi/3$.

Based on the above description, we first define the following variables for easy reference:

- $q_1 = [x, y, \theta_b]^T$: The pose of the mobile platform.
- $q_2 = [\theta_1, \theta_2, \theta_3]^T$: The angle displacements of the robot arm.
- $\zeta = [\varphi_1, \eta_1, \varphi_2, \eta_2, \varphi_3, \eta_3]^T$: The drive variables of the platform.
- $I_{n\times n}$: The $n \times n$ identity matrix. $0_{m\times n}$: The $m \times n$ zero matrix.
- $q = [x, y, \theta_b, \theta_1, \theta_2, \theta_3]^T$: The pose of the mobile manipulator. Obviously, $q = [q_1^T, q_2^T]^T$.

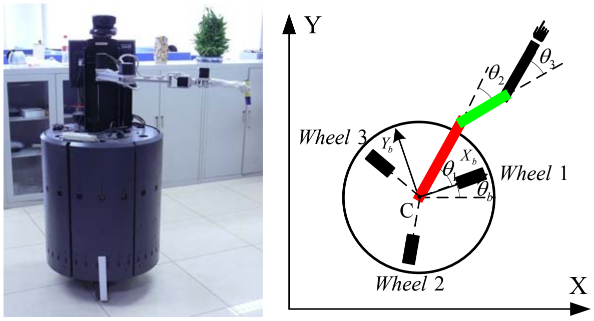
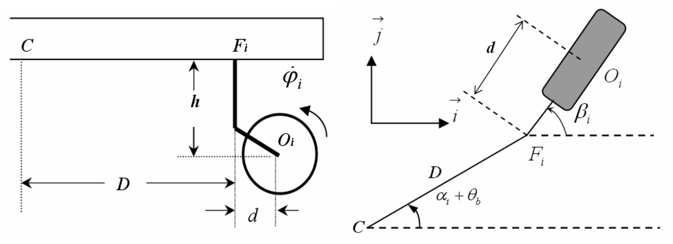


Fig.1 The omnidirectional manipulator and its top view



(1) (2)

Fig.2 Side view and top view of **Wheel *i***

As shown in Fig.4, we also define the relative coordinates.

- World frame $\sum(XOY)$: the inertial frame.
- Moving frame $\sum(X_b C Y_b)$: the frame attached on the mobile platform.

To facilitate modeling, it is assumed that the mobile manipulator has the following characteristics.

- The mobile platform is uniform, and its barycenter is the center of the platform, which is denoted as C .
- The joints between links are rigid and massless, and all links are uniform.

A. Kinematic model

According to the above description, the kinematics of the mobile platform can be described as follow:

$$\dot{\zeta} = J_0(\beta, \eta) \cdot \dot{q}_1 \tag{1}$$

where $J_0(\beta, \eta) = [J_1 \quad J_2 \quad J_3]^T$, and

$$J_i = \begin{bmatrix} -\frac{1}{r}\cos\beta_i & -\frac{1}{r}\sin\beta_i & -\frac{D}{r}\sin\eta_i \\ \frac{1}{d}\sin\beta_i & -\frac{1}{d}\cos\beta_i & -\frac{D}{d}\cos\eta_i - 1 \end{bmatrix} \quad (i = 1, 2, 3) \tag{2}$$

Property 1: For $J_0(\beta, \eta)$, if $\gamma = \frac{d}{D} \le \frac{1}{2}$, rank $J_0(\beta, \eta) = 3$ holds [14].

To avoid the wheels colliding each other, $\gamma \le 1/2$ can be always guaranteed in practical application. Based on the **Property 1,** the pseudoinverse matrix of the $J_0(\beta, \eta)$ always exists, and its Moore-Penrose pseudoinverse matrix can be depicted by

$$J_0^{++}(\beta, \eta) = [J_0^T(\beta, \eta) J_0(\beta, \eta)]^{-1} J_0^T(\beta, \eta) \tag{3}$$

Therefore, the kinematics of the mobile platform also can be written as

$$\dot{q}_1 = J_0^{++}(\beta, \eta) \cdot \dot{\zeta} \tag{4}$$

B. Dynamic model

For the sake of simplicity, we only consider six DOFs as shown in Fig.1（2）in this unified dynamic model. According to Lagrange Formalism, in the absence of friction and other external disturbance, the unified dynamic model can be written as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{5}$$

where $M(q) \in R^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in R^{n\times n}$ is the centripetal and Coriolis matrix, and $G(q) \in R^n$ is the gravitational vector. $\tau \in R^n$ is the input torques vector. Especially, n=6 and $G(q) = 0$ in this model.

Considering the external disturbance d and the unmodeled dynamics $\Delta(q,\dot{q})$, the dynamic model of the mobile manipulator can be depicted by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Delta(q,\dot{q}) + d = \tau \tag{6}$$

Obviously, the above dynamic models satisfy the following three properties.

Property 2: The inertia matrix $M(q)$ is symmetric and positive definite, and satisfies:

$$0 < \lambda_m \cdot I \le \|M(q)\| \le \lambda_M \cdot I \qquad \forall q \in R^n \tag{7}$$

where λ_m, λ_M are positive scalar constants, and $\|.\|$ denotes the Euclidean vector norm.

Property 3: The centripetal and Coriolis matrix $C(q,\dot{q})$ is bounded as a function of $\dot{q}$, i.e.,

$$\|C(q,\dot{q})\| \le k_c \|\dot{q}\|, \ \forall q,\dot{q} \in R^n \tag{8}$$

where k_c is a positive constant.

Property 4: $\dot{M}(q) - 2\cdot C(q,\dot{q})$ is a skew-symmetric matrix, i.e., satisfies the following relationship:

$$x^T\left[\dot{M}(q) - 2\cdot C(q,\dot{q})\right]x = 0, \ \forall x \in R^n \tag{9}$$

III. Controller design and stability analysis

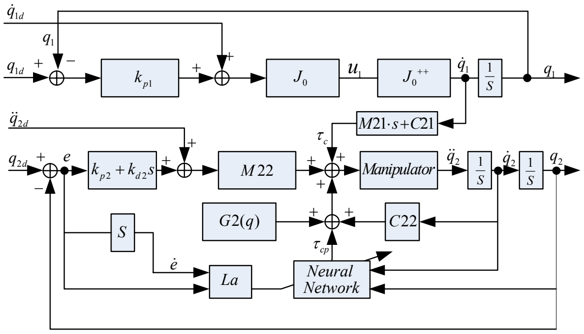


Fig.3 block diagram of the adaptive hybrid control system

In order to control the mobile manipulator effectively, an adaptive hybrid controller, based on the kinematics of the mobile platform and the unified dynamic model of the whole system, is proposed in this section. As shown in Fig.3, the upper part denotes a tracking controller based on the kinematics of the mobile platform, and the lower part is an adaptive controller for the robot arm in dynamics. In addition, q_{1d}, q_{2d} represents the desired trajectories of the mobile platform and the robot arm respectively, and La is the learning algorithm of the RBFNN.

A. Tracking controller for the omnidirectional mobile platform

As for the mobile platform, we assume the given trajectory is $q_{1d} = [x_d, y_d, \theta_{bd}]^T$. Based on the kinematics, we design the control law as

$$u_1 = J_0(\beta,\eta)(\dot{q}_{1d} + k_{p1}(q_{1d} - q_1)) \tag{10}$$

where $k_{p1} \in R^{3\times 3}$ is a symmetric and positive definite matrix. Define $e_1 = q_{1d} - q_1$. By submitting (10) into (4), we can obtain the following relation easily:

$$\dot{e}_1 + k_{p1}e_1 = 0 \tag{11}$$

From (11), we can conclude that if we choose a proper k_{p1} and $t \to \infty$, the tracking error e_1 will decay to zero.

Remark 1: Because this controller is based on the kinematics of the mobile platform, the reaction from the robot arm is also involved in this controller.

B. Adaptive controller for the robot arm

B.1. Problem formulation

To obtain the coupling torque vector, we can rewrite the unified dynamic model as

$$\begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G1 \\ G2 \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \tau 1 \\ \tau 2 \end{bmatrix} \tag{12}$$

where $M11$, $M12$, $M21$, $M22$, $C11$, $C12$, $C21$, $C22$ are 3×3 block matrixes, and $G1 = [G_1 \ G_2 \ G_3]^T$, $G2 = [G_4 \ G_5 \ G_6]^T$, $\tau 1 = \begin{bmatrix} F_x & F_y & \tau_{\theta b}\end{bmatrix}^T$ denotes the equivalent drive torque vector of the mobile platform, $\tau 2 = \begin{bmatrix}\tau_{\theta 1} & \tau_{\theta 2} & \tau_{\theta 3}\end{bmatrix}^T$ is the drive torque vector of the robot arm.

Choose the second row of (12). We can easily obtain the following relations:

$$M22\cdot\ddot{q}_2 + C22\cdot\dot{q}_2 + G2 + \Delta_2(q,\dot{q}) + d_2 + \tau_C = \tau 2 \tag{13}$$

where $\tau_C = M21\cdot\ddot{q}_1 + C21\cdot\dot{q}_1$ denotes the coupling torque vector due to the movement of the mobile platform. Because the velocity of the mobile platform $\dot{q}_1$ can be obtained by the kinematic controller mentioned above, then the acceleration $\ddot{q}_1$ could be derived by calculating the derivative of the $\dot{q}_1$.

Without the compensating torque of the unmodeled dynamics and external disturbance, we get the control law based on CTC as

$$u_2 = \tau_0 = M22(k_{p2}e_2 + k_{d2}\dot{e}_2 + \ddot{q}_{2d}) + C22\cdot\dot{q}_2 + G2 + \tau_C \tag{14}$$

where k_{p2} and k_{d2} are proportional and derivative matrixes, and $e_2 = q_{2d} - q_2$.

Substitute (14) into (13) yields

$$\ddot{e}_2 + k_{d2}\dot{e}_2 + k_{p2}e_2 = -\rho \tag{15}$$

where $\rho = -M22^{-1}(q)[\Delta_2(q,\dot{q}) + d_2]$.

It is obvious that errors will asymptotically converge to zero when ρ =0 and k_{p2}, k_{d2} are chosen appropriately. On the other hand, the existence of ρ influences the performance of CTC and makes the closed-loop system unstable. Therefore, RBFNN can be used to compensate the system uncertainties. Then, the overall control law becomes

$$u_2 = \tau_0 + \tau_{cp} \tag{16}$$

where τ_0 is the output torque of the CTC defined like (14), τ_{cp} is the compensating torque generated by RBFNN.

B.2. Radial basis function neural network

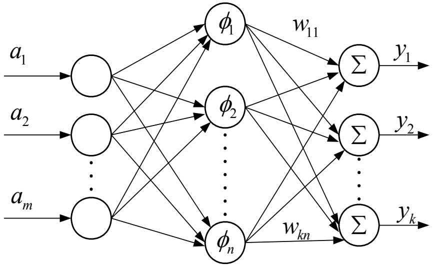


Fig.4 Three-Layer Neural Network

Fig.4 shows the structure of the three-layer NN, which includes an input layer, a hidden layer, and an output layer. Assume that there are m nodes in the input layer, n nodes in the hidden layer, k nodes in the output layer. u_i^l represents the input of the node i in layer l, and o_i^l represents the output of the node i in layer l. The relationships among these layers can be described as follows:

- ➢ Input Layer: signals input to the RBFNN via this layer

$$o_i^1 = u_i^1 = a_i \qquad (i = 1,2,3,\cdots,m) \tag{17}$$

- ➢ Hidden Layer: produce nonlinearity

$$u_j^2 = \sum_{i=1}^{m} o_i^1 \qquad (j = 1,2,\cdots,n) \tag{18}$$

$$o_j^2 = \phi_j = \exp[-\frac{\left\|u_j^2 - c_j\right\|^2}{2\sigma_j^{\,2}}] \qquad (j = 1,2,\cdots,n) \tag{19}$$

$Ce = [c_1, c_2, \cdots, c_n]^T$ denotes the centre vector of Gaussian function. $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_n]^T$ represents the width vector of Gaussian function.

- ➢ Output Layer

$$y_h = o_h^3 = u_h^3 = \sum_{j=1}^{n} w_{hj} \cdot o_j^2 \qquad (h = 1,2,\cdots,k) \tag{20}$$

Let

$$W = \begin{bmatrix} w_{11}, w_{12}, \cdots, w_{1n} \\ w_{21}, w_{22}, \cdots, w_{2n} \\ \vdots \quad \vdots \quad\quad \vdots \\ w_{k1}, w_{k2}, \cdots, w_{kn} \end{bmatrix},\ \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix},\ Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \tag{21}$$

In this way, the relationship of the output layer can be written as:

$$Y = W \cdot \phi \tag{22}$$

where W is the weight matrix of the RBFNN, ϕ is the excitation function vector .

B.3. RBFNN adaptive controller

As for these reasons referred above, we adopt a RBFNN to approximate ρ . Assume that the ideal output of the neural network is

$$\rho(z) = W^* \phi(z) + \varepsilon(z) \tag{23}$$

where $\phi(z)$ is the excitation function vector of the RBFNN, $\varepsilon(z)$ is the reconstruction error of the RBFNN, and $z = [q_2, \dot{q}_2, \ddot{q}_2]^T$ denotes the input of the RBFNN, $W^* = [w_{ij}] \in R^{k \times n}$ is the optimal weight matrix satisfying

$$W^* = \arg\min_{W} \left\{ \sup_{z \in D_z} \left| \hat{\rho}(z \mid \hat{W}) - \rho(z) \right| \right\} \tag{24}$$

D_Z is the bounds of the input z , which represents the limitation of position, velocity and acceleration.

$\hat{\rho}(z \mid \hat{W})$ is an estimation of $\rho(z)$, and we define it as

$$\hat{\rho}(z \mid \hat{W}) = \hat{W} \phi(z) \tag{25}$$

Let compensative control law τ_{cp} in (16) be

$$\tau_{cp} = -M22 \cdot \hat{\rho}(z \mid \hat{W}) \tag{26}$$

Combining (13), (14), (26), we write the closed-loop system as

$$\ddot{e}_2 + k_{d2}\dot{e}_2 + k_{p2}e_2 = \tilde{\rho}(z) \tag{27}$$

where $\tilde{\rho}(z)$ denotes

$$\tilde{\rho}(z) = \hat{\rho}(z \mid \hat{W}) - \rho(z) = \tilde{W}\phi(z) + \varepsilon(z) \tag{28}$$

with $\tilde{W} = W^* - \hat{W}$ representing the error between the adjustable weight matrix and the optimal matrix.

Define the state vector as $x = [e_2^T, \dot{e}_2^T]^T$. The state-space equation of (27) has the form as

$$\dot{x} = Ax + B\tilde{\rho}(z) \tag{29}$$

where $A = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ -k_{p2} & -k_{d2} \end{bmatrix}, B = \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix}$.

Then, the learning algorithm of W is designed as

$$\dot{\hat{W}} = \Lambda^{-1} B^T P x \phi^T \tag{30}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$, λ_i>0 $(i = 1,2,\cdots,n)$ is gain matrix and P is positive definite solution of the following Riccati equation:

$$A^T P + PA + P^T BB^T P + Q = 0 \tag{31}$$

where Q is a constant matrix with appropriate dimensions.

In order to prove the stability of the closed-loop system, the following assumptions are made:

Assumption 1: The reconstruction error $\varepsilon(z)$ is bounded, i.e., $\left\|\varepsilon\right\| \le b_\varepsilon$ for $\forall z \in D_z$.

Assumption 2: The norm of the optimal weight matrix is bounded, that is, $\left\|W^*\right\| \le b_w$.

Theorem: Consider the manipulator with the structured uncertainty involved by mobile platform and external disturbance; we apply the whole controller as (16) and the learning algorithm (30) for the compensating controller part. Based on the Riccati equation (31) and Assumption 1 and Assumption 2, we can get following result:

The state vector x for the manipulator is uniformly ultimately bounded. Then, if reconstruction error of the neural network $\varepsilon \in L_2$, i.e., $\int_0^\infty \varepsilon^2(\mathrm{t})\mathrm{dt} < \infty$, trajectory tracking errors of the manipulator tend to zero as time goes to infinity.

Proof: Consider the following Lyapunov function [13]:

$$V = x^T Px + Tr(\tilde{W}\Lambda\tilde{W}) \tag{32}$$

By applying the properties of matrix theory, we can obtain the time derivative of V as

$$\begin{aligned}
\dot{V} &= x^T(A^TP+PA)x + \tilde{\rho}^{\mathrm{T}}B^{\mathrm{T}}Px + x^{\mathrm{T}}PB\tilde{\rho} + 2Tr(\tilde{W}^{\mathrm{T}}\Lambda\dot{\hat{W}}) \\
&= x^{\mathrm{T}}(A^{\mathrm{T}}P+PA)x + 2x^{\mathrm{T}}PB\tilde{\rho} - 2Tr(\dot{\hat{W}}^{\mathrm{T}}\Lambda\tilde{W}) \\
&= -x^{\mathrm{T}}(P^{\mathrm{T}}BB^{\mathrm{T}}P+Q)x + 2x^{\mathrm{T}}PB\varepsilon + 2Tr(\phi^{\mathrm{T}}\tilde{W}^{\mathrm{T}}B^{\mathrm{T}}Px) \\
&\quad -2\mathrm{T}r(\tilde{W}^{\mathrm{T}}\Lambda\dot{\hat{W}}) \\
&= -x^{\mathrm{T}}(P^{\mathrm{T}}BB^{\mathrm{T}}P+Q)x + 2x^{\mathrm{T}}PB\varepsilon \\
&\quad +2Tr\left[\tilde{W}^{\mathrm{T}}(B^{\mathrm{T}}Px\phi^{\mathrm{T}} - \Lambda\dot{\hat{W}})\right] \\
&= -x^{\mathrm{T}}(P^{\mathrm{T}}BB^{\mathrm{T}}P+Q)x + 2x^{\mathrm{T}}PB\varepsilon \\
&= -x^{\mathrm{T}}Qx - (B^{\mathrm{T}}Px-\varepsilon)^{\mathrm{T}}(B^TPx-\varepsilon) + \varepsilon^{\mathrm{T}}\varepsilon \\
&\le -x^{\mathrm{T}}Qx + \varepsilon^{\mathrm{T}}\varepsilon
\end{aligned} \tag{33}$$

It is easy to obtain $\dot{V} \le -\lambda_{min}(Q)\|x\|^2 + \|\varepsilon\|^2$, and $V(x,\tilde{W})$ is negative outside the following compact set Σ_{x}:

$$\Sigma_x = \left\{ x \,\middle|\, 0 \le \|x\| \le \sqrt{\frac{1}{\lambda_{min}(Q)}}\|\varepsilon\| \right\} \tag{34}$$

Assumption 2 ensures that W is bounded, i.e., $\tilde{W}$ is bounded. Thus x is uniformly ultimately bounded. Integrating both sides of equation (33) from $t=0$ to $t=\infty$ gives

$$\int_0^\infty x^{\mathrm{T}}Qx\mathrm{dt} \le \int_0^\infty \varepsilon^{\mathrm{T}}\varepsilon\mathrm{dt} + V(0) - V(\infty) \tag{35}$$

Then, we can easily get $\int_0^\infty \|x\|^2\mathrm{dt} \le k/\lambda_{min}(Q)$, where $k = \int_0^\infty \varepsilon^{\mathrm{T}}\varepsilon\mathrm{dt} + V(0) - V(\infty)$. Noting that $V(t)$ is a non-increasing function of time and has low bounded, this implies $V(0) - V(\infty) < \infty$. if $\int_0^\infty \varepsilon^2(\mathrm{t})\mathrm{dt} < \infty$, we know $k < \infty$ and $x \in L_2$. In addition, the boundedness of x above denotes $x \in L_\infty$. From closed-loop dynamic equation (29) and boundedness of $x(t)$, $\tilde{W}(t)$ and $\varepsilon(t)$, we can get $\dot{x} \in L_\infty$. Then, $x \in L_2 \cap L_\infty$, $\dot{x} \in L_\infty$. Thus, $\lim_{t\to\infty} x(t) = 0$ is achieved according to Barbalat's lemma.

Therefore, the whole closed-loop system is asymptotically stable, i.e., the position tracking error e_2 and velocity tracking error $\dot{e}_2$ will asymptotically tend to zero as time goes to infinity.

IV. Simulation Results

To illustrate the validity of the proposed method, the proposed model is simulated in this section. In addition, a computed torque controller without compensating is also applied to compare with the proposed method. Simulation parameters are shown in TABLE I, and r = 0.1m, d = 0.05m and D = 0.2m. In TABLE I, NV denotes **n**ominal **v**alue and AV represents **a**ctual **v**alue.

TABLE I
SIMULATION PARAMETERS

	Mass(kg)		Length(m) or radius(m)		Inertial of moment(kgm^2)	
	NV	AV	NV	AV	NV	AV
Mobile Platform	60	100	0.32	0.32	3.0720	5.1200
Link1	5	5.5	0.25	0.25	0.1042	0.1146
Link2	3	3.5	0.35	0.35	0.1225	0.1429
Link3	2	2.5	0.21	0.21	0.0294	0.0367

The initial conditions are given as follow:

$$q(0) = \begin{bmatrix} 2 & 2 & 0.5 & 1 & 1 & 1 \end{bmatrix}^T$$

$$\dot{q}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Let the desired trajectory be

$$q_d(t) = \begin{bmatrix} 10\sin(0.1t) & 8\cos(0.05t) & \sin(0.1t) & 3\sin(0.1t) & 2\sin(0.1t) & \sin(0.1t) \end{bmatrix}^T$$

Parameters involved in the presented controller (11) and (15) are $k_{p1} = 5 \times I_{3\times3}$, $k_{p2} = 19 \times I_{3\times3}$, $k_{d2} = 8 \times I_{3\times3}$ and $Q = 50 \times I_{6\times6}$. The weights of the RBFNN are initialized to zero.

In order to simulate the unmodeled dynamics caused by friction and other factors, we define the unmodeled dynamics $\Delta_2(q,\dot{q})$ as follows:

$$\Delta_2(q,\dot{q}) = 5 \times \begin{bmatrix} 0.5sign(\dot{\theta}_1)[0.5 + \exp(-|\dot{\theta}_1|)] \\ 0.3sign(\dot{\theta}_2)[0.3 + \exp(-|\dot{\theta}_2|)] \\ 0.2sign(\dot{\theta}_3)[0.2 + \exp(-|\dot{\theta}_3|) \end{bmatrix}$$

The external disturbance is

$$d_2 = [3\sin(t)\ 2\sin(\mathrm{t})\ \sin(t)]^T \text{ Nm}$$

Fig.5 to Fig.8 show the position errors of x, y, θ_b, θ_1, θ_2 and θ_3 respectively. AHC represents the adaptive hybrid controller we presented, and CTC denotes computed torque controller.

Remark 2: In the practical application, the kinematics is more precise than the dynamics. Thus we control the mobile platform with regardless of its unmodeled errors. In AHC and CTC, the control scheme and parameters are similar.

Remark 3: When the model error and the external disturbance is relatively small compared to the drive torque (<5%), CTC is able to track the given trajectory. However, Fig.6 to Fig.8 show CTC is not stable when the model error or the external disturbance is significant. Obviously, the proposed controller is capable of tracking the given trajectory even if the interference is very serious.

From above, it is clear that the system quickly converges to the desired value, so the proposed controller is able to achieve trajectory tracking successfully.

V. Conclusions

In this paper, an adaptive hybrid controller, based on the kinematics of the mobile platform and the unified dynamic model of the whole system, is proposed. The stability analysis is proved by Lyapunov approach. From the discussion and simulation results, the following conclusions can be reached:

1. Using a RBFNN, the proposed controller is able to track the given trajectory even if the unmodeled error and the external disturbance are significant.

2. Compared to the method presented in literature [10], an easy way to compute the coupling torque between the mobile platform and the robot arm has been found based on the unified model.

Simulation results show the validity of the presented models and effectiveness of the developed controller, and this control scheme is also can be adopted in position and force control occasions in the future.

AKNOWLEDGEMENT

The authors gratefully acknowledge the support of K. C. Wong Education Foundation, Hong Kong.

References

[1] G. Campion, and G. Bastin, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation,* Vol.12, No.1, pp. 47-62, 1996.

[2] A. B´etourn´e, and G. Campion, “Dynamic model and control design of a class of omnidirectional mobile robots,” *Proceedings of the 1996 IEEE International Conference on Robotics and Automation,* pp.2810-2815, 1996.

[3] B.SicilianO, C. C. Wit, and G. Bastin, *Theory of robot control,* Springer-Verlag, 1996.

[4] J. H. Chung, B.J. Yi, W. K. Kim, and H. Lee, “The dynamic modeling and analysis for an omnidirectional mobile robot with three castor wheels,” *Proceedings of the 2003 IEEE International Conference on Robotics and Automation,* pp. 521-527, 2003.

[5] B. J. Yi, and W. K. Kim, “The dynamics for redundantly actuated omnidirectional mobile robots,” *IEEE International Conference on Robotics and Automation,* pp.2485-2492, 2001.

[6] J.D. Tan, and N. Xi, “Unified model approach for planning and control of mobile manipulators,” *IEEE International Conference on Robotics and Automation,* pp.3145-3152, 2001.

[7] R. Holmberg, and O. Khatib, “Development and control of a holonomic mobile robot for mobile manipulation tasks,” *International Journal of Robotics Research,* Vol.19, No.11, pp.1066-1074, 2000.

[8] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, “Coordination and decentralized cooperation of multiple mobile manipulators,” *International Journal of Robotic System,* Vol.13, No.11, pp.755–764, 1996.

[9] O. Khatib, “A unified approach to motion and force control of robot manipulators: the operational Space formulation,” *IEEE Journal on Robotics and Automation,* Vol. 3, No. 1, pp. 43-53, February 1987.

[10] K. Liu, and F. L. Lewis, “Decentralized continuous robust controller for mobile robots,” *Proceedings of IEEE International Conference on Robotics and Automation,* pp. 1822-1827,1990.

[11] J. H. Chung, S. A. Velinsky, and A. H. Ronald, “Interaction control of a redundant mobile manipulator,” *The International Journal of Robotics Research,* Vol. 17, No. 12, pp. 1302-1309, Dec. 1998.

[12] C. Kwan, F. L. Lewis, Darren M. Dawson, “Robust neural-network control of rigid-link electrically driven robots,” *IEEE Transactions on Neural Networks,* Vol.9, No.4, pp.581-589, July 1998.

[13] Z.S.Song, J.Q.Yi, D.B.Zhao, and X.C.Li, “A computed torque controller for uncertain robotic manipulator systems: fuzzy approach,” *fuzzy Sets and Systems.* Vol.154, No.2, pp.208-226, 2005.

[14] D.B.Zhao, J.Q.Yi and X.Y.Deng, “Motion Regulation of Redundantly Actuated Omni-directional Wheeled Mobile Robots with Internal Force Control,” *Proceedings of the 2007 IEEE International Conference on Intelligent Robots and Systems*, pp.3919-3924, 2007.

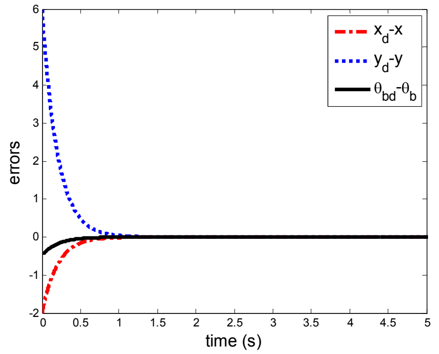


Fig.5 Position errors of the pose of the mobile platform in AHC

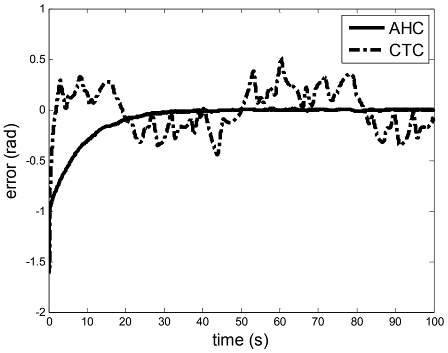


Fig.6 Position errors of θ_1

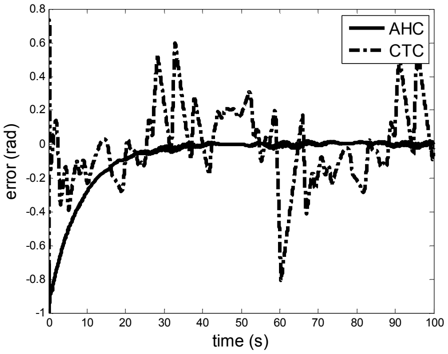


Fig.7 Position errors of θ_2

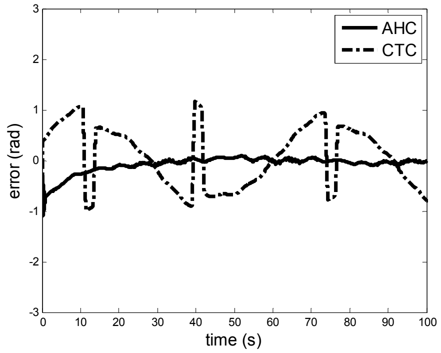


Fig.8 Position errors of θ_3