Tracking Control of Nonholonomic Mobile Robots *

Jindong Tan and Ning Xi
Department of Electrical and Computer Eng.
Michigan State University
East Lansing, MI 48824, U.S.A.

Wei Kang
Mathematics Department
Naval Postgraduate School
Monterey, CA 93943, U. S. A.

Abstract

This paper presents a novel design method for a non-time based mobile robot controller so that it can track an arbitrary twice differentiable path. The key to the non-time based control method is the introduction of a suitable motion reference other than time, which is directly related to the desired and measurable system output. It enables the construction of control systems with integrated planning capability, in which planning becomes a real-time closed- loop process. This new design method converts a controller designed with traditional time-based approaches to a non-time based controller using a state to reference(STR) projection. This significantly simplifies the design procedure for a non-time based controller. While designing a time-based controller, the curvature is introduced as a parameter to describe the forward velocity of a mobile robot. As a result, for any given differentiable curve, the same time-based controller can be projected to different non-time based controllers based on the selection of non- time based motion reference. The design procedure is therefore simplified significantly. The method is exemplified by a nonholonomic mobile robot tracking control problem. The controller has been implemented and tested for different curves such as sine waves, circles and ellipses. Experimental results demonstrate advantages of the proposed method.

1 Introduction

The objective of tracking control is to ensure the output of a system to track a given reference input or desired path. A common feature of many path tracking feedback design is a state trajectory following approach. For instance, the path tracking controller for a mobile robot is required to ensure the convergence of the its states to desired states which is a prescribed function of time. The time plays a role of action reference in the system. While mathematically elegant, the well developed time-based approaches may not be the best for some path-tracking problems.

*Research Partially supported under NSF Grant IIS-9796300 and IIS-9796287.

Non-time based controller has attracted researchers' attention of different fields. In [1], the event-based controller design was first introduced. Since then, it has been successfully applied to robot motion control [2], multi-robot coordination [3], force and impact control [4], robotic teleoperation [5] and manufacturing automation [6].

The basic idea of non-time reference is to introduce the concept of an action reference parameter which is directly relevant to the sensory measurement and the task. The event-based planning and control scheme and the traditional time-based planning and control scheme are compared in Figure 1.

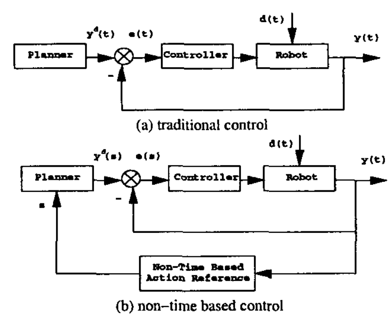

Figure 1: The comparison

In the event-based planning and control scheme, the function of the Action Reference block in Figure 1, (b) is to compute the action reference parameter s on-line, based on sensory measurements. The planner generates the desired value to the system, according to the on-line computed action reference parameter s. The action reference parameter is calculated near or at the same rate as the feedback control. In other words, the action plan is adjusted at a very high rate which enables the planner to handle unexpected or uncer-

0-7803-5806-6/99/$10.00

tain discrete and continuous events. For example, the event-based planning and control scheme has been successfully applied to deal with unexpected obstacle in a robot motion [2]without extra sensors which are necessary with traditional methods. Furthermore, once the obstacle is removed, the traditional methods require replanning to complete the task. But the event-based planning and control in this case does not require any extra force-torque or contact sensing, or replanning. The experimental results in [2] clearly demonstrate the feature.

In this paper, we develop a new method which is different from all the non-time based approaches mentioned above. We focus on the problem of finding controllers for the tracking of a desired path described as a function of non-time based action reference. We use the existing design methods for time-based controller, then the time variable in the feedback is substituted by a special transformation called state-to-reference(STR) projection. The resulting feedback becomes non-time based controller which also can drive the system asymptotically approaching the desired path. Our approach has the following four advantages. First, the method is applicable to a wide range of tracking control problems Second, a transformation is provided to transform a time dependent controller into a non-time based controller. The third advantage of our design algorithm is the flexibility in choosing motion references. The fourth advantage of our method is that the transformation from state space to the action reference is not necessarily the orthogonal projection.

In the third section, we develop a unified time based controller for the random differentiable paths. In our paper, nonlinear transformation is used to simplify the kinematic model, so the simplified model is independent of the specified paths such as straight line or circles. The derived controller based on this simplified model now becames applicable to more general pathes. Theoretically, any differentiable paths can be tracked by this unified controller. The locally asymptotically stability around the desired path can be proved without the restriction of constant forward velocity.

2 Motion Reference Projection

Dynamic systems are modeled by differential equations in which the free variable is the time variable t. A desired trajectory is often modeled as a function of time. We denote it by $x_d(t)$, where x represents the state of the system. Controllers can be designed so that the trajectory of the plan system $x(t)$ asymptotically approaches the desired trajectory $x_d(t)$. This is a typical tracking control problem. Since the design is based on the model driven by t, the controller is time dependent. However, in the control of mobile robots, we often prefer a feedback controller which is independent of time. Furthermore, the desired path is not necessarily defined as a function of time. It is defined by parametric equations. The parameter is called *motion reference* or *action reference*,which is denoted by s. For instance, a desired path can be defined by a parametric equation $x_d(s)$, where s is the arc length, or s is the projection of x_d to a coordinate axis. In this section, we prove the feasibility of transforming a time dependent feedback controller into a time independent controller for a desired path driven by motion reference.

A system is defined by the equation

$$\dot{x} = f(x,u), \quad x \in I\!R^n, u \in I\!R^m, y = h(x), \quad y \in I\!R^k \tag{2.1}$$

where u is the control input, and x is the state of the system. Suppose that $y = h(x)$ represents the output of the system, the desired path is given by a parametric equation $y = y_d(s)$, where s is the motion reference.

The first step in the controller design is to define a corresponding path in time domain. If s is not time, then we assign an strictly increasing function $s = v(t)$. How to pick the function $v(t)$ depends on the desired speed. For example, if the problem require that the system is operated so that s is increasing at a constant speed v_0, then $v(t) = v_0 t$.

The feedback satisfies

$$\lim_{t\to\infty} (h(x(t)) - y_d(v(t))) = 0.$$

Specifically, there exists an initial condition of the system x_0 such that $h(x(t)) = y_d(t)$. Denote this path by $x_d(s)$ or $x_d(v(t))$.

The third step is to find a suitable transformation, $s = \gamma(x)$, from the state space to the reference s. The transformation satisfies

$$\gamma(x_d(s)) = s \tag{2.2}$$

For example, given any state x_0, let $x_d(s_0)$ be the orthogonal projection from x_0 to $x_d(s)$. If we define $\gamma(x_0) = s_0$, then it satisfies (2.2). The transformation satisfying (2.2) is called a state-to-reference projection (STR projection).

The last step is to construct the feedback. Let

$$u(x) = \tilde{u}(x, v^{-1}(\gamma(x))) \tag{2.3}$$

where $\tilde{u}(x,t)$ is the feedback found in the second step, $\gamma(x)$ is a state-to-reference projection. The closed-loop system is

$$\dot{x} = f(x, u(x)). \tag{2.4}$$

In the following, we prove that the non-time based control law (2.4) drives the vehicle asymptotically approaching the desired path. Given any state x and any time t, the distance from x to the point $x_d(t)$ in the desired path is denoted by $d(x,t)$. So, $d(x,t) = \|x - x_d(t)\|$ is a function from $\mathbb{R}^n \times \mathbb{R}$ to $\mathbb{R}$. The open set $U(x_d, r)$ consists of $(x,t) \in \mathbb{R}^n \times \mathbb{R}$ such that $d(x,t)$ is less than r, i.e.

$$U(x_d, r) = \{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid d(x,t) < r\}.$$

The set $W(\delta)$ is the δ neighborhood of the curve x_d in the state space. In this paper, we assume that the vector field $F(x,u)$, the feedback $u(x,t)$, and their derivatives are bounded. Assume that the derivatives of $v(t)$ and $v^{-1}(s)$ are bounded. It can be seen that these assumptions are true for mobile robots.

Let $\tilde{\phi}(t, t_0, x_0)$ be the trajectory of the closed-loop system

$$\dot{x} = f(x, \tilde{u}(x,t)) \tag{2.5}$$

with the initial time t_0 and initial condition $x(t_0) = x_0$. We assume that there exists a neighborhood $U(x_d, r)$ so that $(x_0, t_0) \in U(x_d, r)$ implies that $\tilde{\phi}(t, t_0, x_0)$ approaches x_d exponentially. Furthermore, the trajectory satisfies

$$\begin{array}{c} me^{-\alpha_1(t-t_0)} d(x_0, t_0) \le d(\tilde{\phi}(t, t_0, x_0), t) \\ d(\tilde{\phi}(t, t_0, x_0), t) \le Me^{-\alpha_2(t-t_0)} d(x_0, t_0) \end{array} \tag{2.6}$$

for some $m > 0$, $M > 0$, $\alpha_1 > 0$ and $\alpha_2 > 0$. The following function is used as our Lyapunov function,

$$V(x_0, t_0) = \int_{t_0}^{\infty} d(\tilde{\phi}(t, t_0, x_0), t) dt. \tag{2.7}$$

We assume that all derivatives of $V(x,t)$ of order less than or equal to three are bounded in a neighborhood $U(x_d, r)$. From (2.6), it is easy to show that

$$md(x,t) \le V(x,t) \le Md(x,t). \tag{2.8}$$

Theorem 2.1 *There exists a neighborhood $W(\delta)$ of the desired trajectory $x_d(s)$ such that the trajectory $x(t)$ of (2.4) satisfies*

$$\lim_{t \to \infty} d(\ x(t), v^{-1}(\gamma(x(t)))\ ) = 0. \tag{2.9}$$

provided $x(0)$ is in $W(\delta)$. This implies that the trajectory of the closed-loop system with a non-time based feedback asymptotically approaches the desired path.

Proof. The trajectory of (2.5) with initial time t_0 and initial state $x(t_0) = x_0$ is denoted by $\tilde{\phi}(t, t_0, x_0)$. Similarly, a trajectory of (2.4) is denoted by $\phi(t, t_0, x_0)$. By the definition of $V(x,t)$, it is easy to check that the derivative of $V(x,t)$ along $\tilde{\phi}(t, t_0, x_0)$, a trajectory of (2.5), satisfies

$$\frac{d}{dt} V(x(t), t) = -d(x(t), t).$$

Therefore,

$$\frac{\partial V(x,t)}{\partial x} f(x, \tilde{u}(x,t)) + \frac{\partial V(x,t)}{\partial t} = -d(x,t). \tag{2.10}$$

The desired trajectory $x_d(v(t))$ is a solution to the equation (2.5). So, $\dot{x}_d(v(t)) = f(x_d, \tilde{u}(x_d, t))$. Since $s = \gamma(x_d(s))$, we have $t = v^{-1}(\gamma(x_d(v(t))))$. The derivative of this equation with respect to t yields

$$\frac{\partial v^{-1}}{\partial s} \frac{\partial \gamma}{\partial x} f(x_d, \tilde{u}(x_d, t)) = 1. \tag{2.11}$$

Now, let's consider the closed-loop system (2.4) with the feedback (2.3). The derivative of $V(x, v^{-1}(\gamma(x)))$ in the direction of (2.4) is

$$\begin{aligned} &\dot{V}(x, v^{-1}(\gamma(x))) \\ &= \frac{\partial V}{\partial x} f(x, u(x)) + \frac{\partial V}{\partial t} \frac{\partial v^{-1}}{\partial s} \frac{\partial \gamma}{\partial x} f(x, u(x)). \end{aligned}$$

From (2.3) and (2.10), we have

$$\begin{aligned} &\dot{V}(x, v^{-1}(\gamma(x))) = -d(x, v^{-1}(\gamma(x))) \\ &+ \left(\frac{\partial v^{-1}}{\partial s} \frac{\partial \gamma}{\partial x} f(x, u(x)) - 1\right) \frac{\partial V}{\partial t}(x, v^{-1}(\gamma(x))). \end{aligned} \tag{2.12}$$

By the definition of $V(x,t)$, it is easy to check that $V(x,t) > 0$ for all x in a neighborhood of $x_d(t)$ and $V(x_d(v(t)), t) = 0$. Therefore, $V(x,t)$ has a minimum value at $x_d(t)$. It can be proved that

$$V(x,t) = (x - x_d(t))^T Q(x,t)(x - x_d(t))$$

for some positive definite matrix $Q(x,t)$. Therefore, $\frac{\partial V}{\partial t}(x,t) = P(x,t)(x - x_d(t))$ for some row vector $P(x,t)$. In a neighborhood $U(x_d, r)$, $P(x,t)$ is bounded. So,

$$\left| \frac{\partial V(x,t)}{\partial t} \right| \le M_1 d(x,t) \tag{2.13}$$

in a neighborhood $U(x_d, r)$ for some $M_1 > 0$. Meanwhile, (2.11) implies

$$\frac{\partial v^{-1}}{\partial s} \frac{\partial \gamma}{\partial x} f(x, u(x)) - 1 = 0$$

if x is a point on $x_d(s)$. Therefore,

$$\left|\frac{\partial v^{-1}}{\partial s}\frac{\partial \gamma}{\partial x}f(x,u(x)) - 1\right| < \frac{1}{2M_1} \tag{2.14}$$

in a neighborhood $W(\delta)$. From (2.12), (2.13) and (2.14), we have

$$\dot{V}(x, v^{-1}(\gamma(x))) < -\frac{d(x, v^{-1}(\gamma(x)))}{2}.$$

From (2.8),

$$\dot{V}(x, v^{-1}(\gamma(x))) < -\frac{1}{M}V(x,, v^{-1}(\gamma(x)))$$

in a neighborhood of x_d. By Gronwall's inequality, $V(x, v^{-1}(\gamma(x)))$ approaches zero exponentially along a trajectory $\phi(t, t_0, x_0)$ if x_0 is in a neighborhood of x_d. From (2.8),

$$md(x,t) \leq V(x,t).$$

Therefore, $d(x, v^{-1}(\gamma(x)))$ approaches zero exponentially along a trajectory $\phi(t, t_0, x_0)$ if x_0 is in a neighborhood, $W(\delta)$, of $x_d(s)$. ◁

3 Path tracking control for mobile robots

Considering the mobile robot in Figure 2, the rear wheels are aligned with the vehicle while the front wheels are allowed to spin about the vertical axes. The constraints on the system arise by allowing the wheels to roll and spin, but not slip. Let (x, y, θ, ϕ) denote the configuration of the robot, parameterized by the location of the rear wheels. The dynamics model of the mobile robot can be represented as:

$$\begin{aligned}\dot{x} &= u_1 \cos\theta \\ \dot{y} &= u_1 \sin\theta \\ \dot{\theta} &= \frac{u_1}{l} tan\phi \\ \dot{\phi} &= u_2,\end{aligned} \tag{3.1}$$

where u_1 corresponds to the forward velocity of the rear wheels of the robot and u_2 corresponds to the angular velocity of the steering wheels, the angle of the robot body with respect to the horizontal is θ, the steering angle with respect to the car body is ϕ, (x, y) is the location of the rear wheels, l is the length between the front and the rear wheels. Let $X = (x, y, \theta, \phi), U = (u_1, u_2)$ denote the state variables and control of the robot respectively. $X_d = (x_d, y_d, \theta_d, \phi_d), U_d = (u_{1d}, u_{2d})$ denote the desired

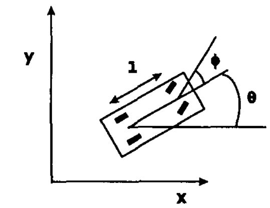

Figure 2: The configuration of a mobile robot.

states and control of the robot. We defined the following transform of coordinates to achieve linearization of the four equations in (3.1),

$$\begin{aligned}e_1(t) &= -[x(t) - x_d(t)]\sin(\theta_d(t)) \\ &\quad +[y(t) - y_d(t)]\cos(\theta_d(t)) \\ e_2(t) &= \sin(\theta - \theta_d) \\ e_3(t) &= \cos(\theta - \theta_d)(\tfrac{1}{l}\tan(\phi) - \tfrac{u_{1d}}{lu_1}\tan(\phi_d))\end{aligned} \tag{3.2}$$

From this definition, the state equations are transformed into

$$\begin{aligned}\dot{e}_1(t) &= u_1 e_2(t) \\ \dot{e}_2(t) &= u_1 e_3(t) \\ \dot{e}_3(t) &= u_1(\alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3)\end{aligned} \tag{3.3}$$

where

$$\begin{aligned}\beta_1 &= -\tfrac{1}{l}\sin(\theta - \theta_d)(\tan(\phi) - \tan(\phi_d))^2 \\ &\quad -\tfrac{1}{l^2}\cos(\theta - \theta_d)\sec(\phi_d)^2\dot{\phi}_d \\ \beta_2 &= \tfrac{1}{l}\cos(\theta - \theta_d)\sec^2(\phi) \\ \beta_1 &+ \beta_2 u_2 = u_1(\alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3)\end{aligned} \tag{3.4}$$

Theorem 3.1 *By selecting the values of $\alpha_i, i = 1, 2, 3$, so that all the eigenvalues of*

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}$$

are on the left half s plane, then system (3.1) is locally asymptotically stable.

Proof. (3.1) is simplified into (3.3),which is equivalent to (3.5).

$$\dot{e} = u_1 A e \tag{3.5}$$

The equilibrium point of (3.5) is $e = 0$. For matrix A which has negative real part eigenvalues and $u_1 > 0$, (3.5) is asymptotically stable, which means $\lim_{t\to\infty} e(t) = 0$. From the definition of $e_1(t), e_2(t), e_3(t)$,we get

$$\lim_{t\to\infty} X(t) = X_d(t) \tag{3.6}$$

Since $(x_d(t), y_d(t), \theta_d(t), \phi_d(t))$ is a solution of (3.1), (3.6) means that (3.1) is asymptotically stable.

Parameters $\alpha_i (i = 1, 2, 3)$ are determined by assigning all the eigenvalues at the left half s plane. From equation (3.4), we can get the feedback control.

$$u_2 = \frac{\alpha_2 - \alpha_1 + u_1(a_1 e_1 + a_2 e_2 + a_3 e_3)}{\beta_1 - \beta_2} \tag{3.7}$$

So, for selected $\alpha_i (i = 1, 2, 3)$, system (3.1) is asymptotically stable provided (3.7) is bounded. $\triangleleft$

Remark. Though u_1 can shifts the positions of the poles of A, it does not influence the stability of system (3.5). The restriction of constant velocity is therefore removed for the proposed controller.

From the definition of $e_1(t), e_2(t), e_3(t)$, we know $e(t)$ is actually a kind of offset of the robot from the desired trajectory. $e(t)$ tend to zero when X tends to X_d. $e(t)$ reflects the state error and it holds only when the offset is small enough, which means the controller based on the nonlinear transformation (3.2) is only locally asymptotically stable. $\triangleleft$

In some literatures, the forward velocity is constrained to be constant to track one kind of specific curve. However,to different curves or different curvature on the same curve, constant velocity is not necessary. If the forward velocity has a profile in practical use, u_1 can be any positive functions, while the stablility is garanteed. If the tracking accuracy is the more important criteria, the forward velocity could be a function of curvature. In the domain of a curve which has greater curvature, tracking error will increase if the forward velocity keeps constant. Therefore the curvature of a curve is considered in designing u_1 in this section. However, it is not necessary in practical use to change the forward velocity as long as the curvature is changed. We can divided the the curvature into several regions and assign a forward velocity for each region. In this case, the forward velocity would be piecewise continuous and be a constant at one region. We define

$$u_1 = u_{1d}[g(r)]$$

where

$$[g(r)] = w_k, r_k < r \leq r_{k+1}$$

and w_k is a constant.

To summarize, we have the time dependent feedback

$$\begin{aligned} u_1 &= u_{1d}[g(r)] \\ u_2 &= \frac{u_1(\alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3) - \beta_1}{\beta_2} \end{aligned} \tag{3.8}$$

Theorem 3.2 *Under the feedback (3.8), the solution $(x(t), y(t))$ of (3.1) asymptotically approaches a desired path $(x_d(t), y_d(t))$, provided that the offset is small enough.*

The feedback (3.8) depends on time. However, it can be converted into a time-indpendent feedback using state-to- reference projection as proved in § 2. In the following, we use the method developed in § 2 and the feedback (3.8) to design time-invariant feedbacks for the tracking of the desired curves. The key step is to find a motion reference for the task. The parametric equation for these curves could be arc length, angle corresponded to arc length, orthogonal projection from a point (x, y) to the curve, etc. As pointed out in § 2, the design method proposed in this paper is not based on a specific motion reference. It works for arbitrary choice of the reference. So, in the following section, we are going to select different reference according to different curves. Since the controller is based on the state variables, any state variable or the projection of state variables can be used as motion reference, which can be used to compute the control feedback.

4 Experiments and results

The new non-time based controller has been implemented on a Nomadic XR4000 mobile robot. The mobile robot has four wheels. Each wheel can be individually steered. However, in order to implement the above control scheme, only two front wheels are steered in the experiment. The other two wheels maintain straight forward configuration. The distance between front and rear wheels is 0.3053 m. The velocity along a given path in the following experiments is always 0.1 m/sec. In the controller the closed loop poles are assigned to the locations, -3, $-5 - i$ and $-5 + i$.

Tracking sine wave: In this example, the desired path is

$$y_d = sin(s); x_d = s \tag{4.1}$$

where s is the non-time based reference. coordinator x is used as the motion reference. It can be seen that the controller can rapidly overcome the 30% initial error. From Figure 3, It can be seen that that the non-time based controller is stable and has no tracking error after the robot is blocked for about 10 sec, while the time-based controller is not stable any more.

Tracking a circle: In this example, the desired path is

$$x_d^2 + y_d^2 = a^2; y_d = a\sin(s); x_d = a\cos(s) \tag{4.2}$$

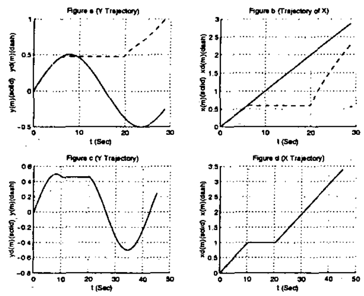

Figure 3: Tracking a sine wave.

where s is the non-time based reference. It is the angle corresponded to the tracked arc by the robot. The reference can be obtained directly by the location of the robot,(x, y). In many applications, desired velocity for tracking desired path is required. In this experiment, we set the desired velocity as $v_0 * (1 + 0.2 * \sin(3t))$. In Figure 4, it can be seen that both the velocity and position errors are very small. In the same experiment, when I suddenly changed the forward velocity to $1.2u_1$ at half of the circle,it keeps a very small tracking error and has a very short transition time. The system is stable at the large disturbance. It demonstrates the robustness of the system.

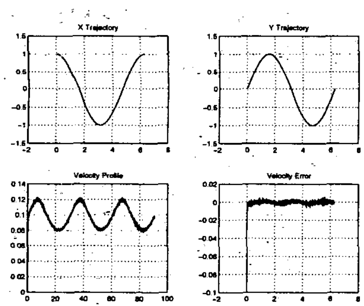

Figure 4: Tracking a circle with varying forward velocity.

Tracking a ellipse: An ellipse has also been tracked. Here the angle corresponded to the tracked arc length is used as the motion reference.

5 Conclusion

A new design method for non-time based tracking controller has been developed. It can convert a controller designed by traditional time-based approach to a new non-time based controller. It significantly simplifies the design procedure. The method has been applied to design a tracking controller of a unmanned vehicle. The experimental results have clearly demonstrated the advantage of non-time based controller. More importantly, the results and the unified controller have provided a efficient and systematic approach to design non-time based tracking controller for a general nonlinear dynamic system.

References

[1] Ning Xi, "Event-Based Planning and Control for Robotic Systems", Doctoral Dissertation, Washington University, December, 1993.

[2] T.J. Tarn, A.K. Bejczy and Ning Xi, "Intelligent Planning and Control for Robot Arms," Proceedings of the IFAC 1993 World Congress, Sydney, Australia, July 18-23, 1993.

[3] Ning Xi, T.J. Tarn and A.K. Bejczy, "Event-Based Planning and Control for Multi-Robot Coordination," Proceedings of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia, pp.251-258, May 2-6, 1993.

[4] Y. Wu, T.J. Tarn, Ning Xi, "Force and Transition Control with Environmental Uncertainties", The Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, May 1995.

[5] T.J. Tarn, Ning Xi, C. Guo, A.K. Bejczy, "Fusion of Human and Machine Intelligence for telerobotic Systems", The Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, May, 1995.

[6] Ning Xi, T.J. Tarn, "Integrated Task Scheduling and Action Planning/Control for Robotic Systems Based on a Max-Plus Algebra Model", the Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems, Sept. 1997, Grenoble, France.

[7] R. M. DeSantis, it Path-tracking for a tractor-trailer-like robot, The International Journal of Robotics Research, Vol. 13, No. 6, 533-544, 1994.

[8] A. Hemami, M. G. Mehrabi, and R. M. H. Cheng, *A synthesis of an optimal control law for path tracking in mobile robots*, Automatica, 8(2): 383-387, 1992.