Multilayer Perceptron Dual Adaptive Control for Mobile Robots

Marvin K. Bugeja and Simon G. Fabri
Department of Electrical Power and Control Engineering, University of Malta, Msida (MSD06), Malta

***Abstract*— This paper presents a novel dual adaptive dynamic controller for trajectory tracking of nonholonomic wheeled mobile robots. The controller is developed in discrete-time and the robot's nonlinear dynamic functions are assumed to be unknown. A sigmoidal multilayer perceptron neural network is employed for function approximation, and its weights are estimated stochastically in real-time. In contrast to adaptive certainty equivalence controllers hitherto published for mobile robots, the proposed control law takes into consideration the estimates' uncertainty, thereby leading to improved tracking performance. The proposed method is verified by realistic simulations and Monte Carlo analysis.**

I. Introduction

Motion control of nonholonomic mobile robots has been receiving considerable attention for the last fifteen years [1]. This activity is not only justified by the vast array of existing and potential practical applications, but also by some particularly interesting theoretical challenges. In particular most mobile configurations manifest restricted mobility, giving rise to nonholonomic constraints in the kinematics. Moreover the majority of mobile vehicles are underactuated, since they have more degrees of freedom than control inputs. Consequently the linearised kinematic model lacks controllability; full-state feedback linearisation is out of reach; and pure, smooth, time-invariant feedback stabilisation of the Cartesian model is unattainable [2].

Earlier research focused only on kinematic control of nonholonomic vehicles [1]–[3], assuming that the control signals instantaneously establish the desired robot velocities. This is commonly known as *perfect velocity tracking* [4]. Controllers based on a full dynamic model [4]–[6] capture better the behaviour of real robots because they account for dynamic effects such as mass, friction and inertia, which are otherwise neglected. However, the exact values of these dynamic parameters are often uncertain or even unknown, and may even vary over time. These factors call for the development of adaptive dynamic controllers to handle better unmodelled robot dynamics, as well as noise and external disturbances.

To address these advanced control issues, some researchers opt to use pre-trained function estimators, specifically artificial neural networks (ANNs), to render nonadaptive conventional controllers more robust [7]. These techniques require prior off-line training and remain blind to variations which take place after the training phase. To account for parametric variations, adaptive control and robust sliding mode control [6] have also been proposed. Another approach is that of online functional-adaptive control, where the uncertainty is not restricted to parametric terms, but covers the dynamic functions themselves [8]–[11].

Adaptive controllers which have hitherto been proposed for the control of mobile robots are based on the *heuristic certainty equivalence* (HCE) property [8]–[11]. In such cases, the estimated functions are used for control as if they were the true ones; ignoring completely their uncertainty. When the uncertainty is large, for instance during startup or when the unknown functions are changing, HCE often leads to large tracking errors and excessive control actions which can excite unmodelled dynamics or lead to instability. Consequently we opt to employ stochastic adaptive control, more specifically the *dual control* principle introduced by Fel'dbaum in [12]–[14]. Basically a dual adaptive control law is designed with two aims in mind: (i) to ensure that the output tracks the desired reference signal, with due consideration given to the estimates' uncertainty; (ii) to excite the plant sufficiently so as to accelerate estimation, thereby reducing quickly the uncertainty in future estimates. These two features are known as *caution* and *probing* respectively [15], [16].

In contrast to other work on mobile robot motion control, the novel contribution of this paper is to introduce a neuro-adaptive dynamic controller featuring these dual adaptive properties. Moreover, the control law is developed entirely in discrete-time and the resulting closed-loop dynamics are independent of the plant parameters. In this paper we focus on the trajectory tracking problem of wheeled mobile robots (WMRs). Nevertheless the employed framework is completely modular, and can easily be adopted for other robot control scenarios.

The presented method employs a sigmoidal multilayer perceptron (MLP) ANN to estimate the robot's nonlinear dynamic functions, which are assumed to be completely unknown. The ANN parameters are estimated stochastically in real-time and no preliminary off-line training is assumed. The estimated functions and a measure of their degree of uncertainty are *both* used in the suboptimal dual adaptive control law, which operates in cascade with a trajectory tracking kinematic controller. Section II of this paper develops the stochastic discrete-time dynamic model of the robot. This is then utilised in the dual adaptive control design outlined in Section III. The proposed method is verified and compared by realistic simulation and Monte Carlo analysis in Section IV, which is followed by a brief conclusion in Section V.

This work was supported by the National Grant, RTDI-2004-026.

II. Modelling

This paper considers the differentially driven wheeled mobile platform depicted in Fig. 1. We ignore the passive wheel and adopt the following notation:

P_o: midpoint between the two wheels
P_c: centre of mass of the platform without wheels
d: distance from P_o to P_c
b: distance from each wheel to P_o
r: radius of each wheel
m_c: mass of the platform without wheels
m_w: mass of each wheel
I_c: moment of inertia of the platform about P_c
I_w: moment of inertia of wheel about its axle
I_m: moment of inertia of wheel about its diameter

The robot dynamic state can be expressed as a five dimensional vector $\boldsymbol{q} \triangleq [x \quad y \quad \phi \quad \theta_r \quad \theta_l]^T$, where (x, y) is the coordinate of P_o, ϕ is the robot orientation angle with reference to the xy frame, θ_r and θ_l are the angular displacements of the right and left driving wheels respectively. The *pose* of the robot refers to the three-dimensional vector $\boldsymbol{p} \triangleq [x \quad y \quad \phi]$.

A. Kinematics

Assuming rolling without slipping, the mobile platform is subject to three kinematic constraints, two of which are nonholonomic [5]. The three kinematic constraints can be written in the form $\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$, where

$$\boldsymbol{A}(\boldsymbol{q}) = \begin{bmatrix} -\sin\phi & \cos\phi & 0 & 0 & 0 \\ \cos\phi & \sin\phi & b & -r & 0 \\ \cos\phi & \sin\phi & -b & 0 & -r \end{bmatrix}.$$

Furthermore, one can verify that $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{0}$, where

$$\boldsymbol{S} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ \frac{r}{2b} & -\frac{r}{2b} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The kinematic state-space model of the WMR in Fig. 1 can now be expressed as

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q})\boldsymbol{\nu}, \tag{1}$$

where $\boldsymbol{\nu}$ represents a column vector composed of the angular velocities of the two driving wheels, specifically $\boldsymbol{\nu} \triangleq [\nu_r \quad \nu_l]^T \triangleq \left[\dot{\theta}_r \quad \dot{\theta}_l\right]^T$.

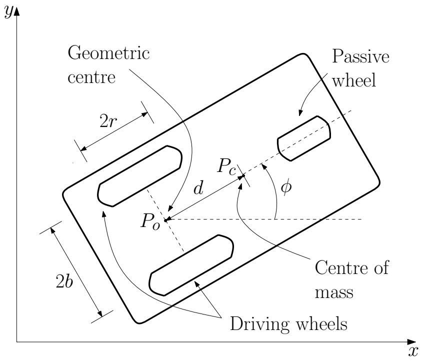

Fig. 1. Differentially driven wheeled mobile robot

B. Dynamics

The dynamic equations of motion of the WMR under consideration can be written in matrix form as [5]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}) = \boldsymbol{E}(\boldsymbol{q})\boldsymbol{\tau} - \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda}, \tag{2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{V}(\dot{\boldsymbol{q}}, \boldsymbol{q})$ is the centripetal and Coriolis matrix, $\boldsymbol{F}(\dot{\boldsymbol{q}})$ is a vector of frictional forces, $\boldsymbol{E}(\boldsymbol{q})$ is the input transformation matrix, $\boldsymbol{\tau}$ is the torque vector and $\boldsymbol{\lambda}$ is the vector of constraint forces.

Differentiating (1) with respect to time, substituting the expression for $\ddot{\boldsymbol{q}}$ in (2), premultiplying the resulting expression by $\boldsymbol{S}^T(\boldsymbol{q})$, and noting that $\boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{A}^T(\boldsymbol{q}) = \boldsymbol{0}$ it can be shown that

$$\bar{\boldsymbol{M}}\dot{\boldsymbol{\nu}} + \bar{\boldsymbol{V}}(\dot{\boldsymbol{q}})\boldsymbol{\nu} + \bar{\boldsymbol{F}}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{3}$$

where:

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \frac{r^2}{4b^2}(mb^2 + I) + I_w & \frac{r^2}{4b^2}(mb^2 - I) \\ \frac{r^2}{4b^2}(mb^2 - I) & \frac{r^2}{4b^2}(mb^2 + I) + I_w \end{bmatrix},$$

$$\bar{\boldsymbol{V}}(\dot{\boldsymbol{q}}) = \begin{bmatrix} 0 & \frac{m_c r^2 d\dot{\phi}}{2b} \\ \frac{m_c r^2 d\dot{\phi}}{2b} & 0 \end{bmatrix},$$

$\bar{\boldsymbol{F}}(\dot{\boldsymbol{q}}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{F}(\dot{\boldsymbol{q}})$, $I = (I_c + m_c d^2) + 2(I_m + m_w b^2)$ and $m = m_c + 2m_w$. It is important to note that:

Remark 2.1: $\bar{\boldsymbol{M}}$ is symmetric and positive definite.

Remark 2.2: The nonlinearities in the WMR dynamics can be totally attributed to $\bar{\boldsymbol{V}}(\dot{\boldsymbol{q}})$ and $\bar{\boldsymbol{F}}(\dot{\boldsymbol{q}})$ since $\bar{\boldsymbol{M}}$ is independent of the state vector and/or its derivatives.

Remark 2.3: $\bar{\boldsymbol{V}}(\dot{\boldsymbol{q}})$ is effectively a function of $\boldsymbol{\nu}$ only, since $\dot{\phi} = \frac{r}{2b}(\nu_r - \nu_l)$ as can be seen in (1).

We will now discretise the continuous-time dynamics (3) to account for the fact that the controller is implemented on a digital computer. Using a first order forward Euler approximation with sampling interval T seconds the following discrete-time dynamic model is obtained:

$$\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1} = \boldsymbol{f}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{\tau}_{k-1}, \tag{4}$$

where the subscript integer k denotes that the corresponding variable is evaluated at time kT seconds and vector $\boldsymbol{f}_{k-1}$ and matrix $\boldsymbol{G}_{k-1}$ are given by

$$\begin{aligned} \boldsymbol{f}_{k-1} &= -T\bar{\boldsymbol{M}}_{k-1}^{-1}\left(\bar{\boldsymbol{V}}_{k-1}\boldsymbol{\nu}_{k-1} + \bar{\boldsymbol{F}}_{k-1}\right), \\ \boldsymbol{G}_{k-1} &= T\bar{\boldsymbol{M}}_{k-1}^{-1}. \end{aligned}$$

The following conditions are assumed to hold:

Assumption 2.1: The control input vector $\boldsymbol{\tau}$ remains constant over each sampling interval.

Assumption 2.2: The sampling interval is chosen low enough for the Euler approximation to hold.

To account for noise, uncertainty and disturbances we introduce an additive discrete random vector $\boldsymbol{\epsilon}_k$. The deterministic model (4) is hence converted to the following nonlinear, stochastic, discrete-time dynamic model

$$\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1} = \boldsymbol{f}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{\tau}_{k-1} + \boldsymbol{\epsilon}_k, \tag{5}$$

under the following condition

Assumption 2.3: $\boldsymbol{\epsilon}_k$ is an independent, zero-mean, white, Gaussian process, with covariance matrix $\boldsymbol{R}_{\boldsymbol{\epsilon}}$.

III. Control Design

A very simple, yet useful representation of the trajectory tracking problem, is through the concept of the *virtual vehicle* [3]. Basically, the time dependent reference trajectory is designated by a *non-stationary* virtual vehicle having the same nonholonomic constraints as the real robot. The controller aims for the real WMR to track the virtual vehicle at all times, in both pose and velocity.

A. Kinematic Control

The discrete-time tracking error $\boldsymbol{e}_k$ is defined as:

$$\boldsymbol{e}_k \triangleq \begin{bmatrix} e_{1_k} \\ e_{2_k} \\ e_{3_k} \end{bmatrix} \triangleq \begin{bmatrix} \cos\phi_k & \sin\phi_k & 0 \\ -\sin\phi_k & \cos\phi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\boldsymbol{p}_{\boldsymbol{r}_k} - \boldsymbol{p}_k\right),$$

where $\boldsymbol{p}_{\boldsymbol{r}_k} \triangleq \begin{bmatrix} x_{r_k} & y_{r_k} & \phi_{r_k} \end{bmatrix}^T$ denotes the virtual vehicle sampled pose vector. In trajectory tracking the objective is to make $\boldsymbol{e}$ converge to zero so that $\boldsymbol{p}$ converges to $\boldsymbol{p}_{\boldsymbol{r}}$. For this task we propose a discrete-time version of the continuous-time kinematic controller proposed in [3]:

$$\boldsymbol{\nu}_{\boldsymbol{c}_k} = \boldsymbol{C} \begin{bmatrix} v_{r_k}\cos e_{3_k} + k_1 e_{1_k} \\ \omega_{r_k} + k_2 v_{r_k} e_{2_k} + k_3 v_{r_k} \sin e_{3_k} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{\nu}_{\boldsymbol{c}_k}$ is the wheel velocity command vector issued by the kinematic controller, $(k_1, k_2, k_3) > 0$ are design parameters, v_{r_k} and ω_{r_k} are the translational and angular virtual vehicle velocities respectively, and $\boldsymbol{C}$ is a velocity conversion matrix given by:

$$\boldsymbol{C} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix}.$$

If we consider only the kinematic model (1) of the WMR and assume perfect velocity tracking (i.e. $\boldsymbol{\nu}_k = \boldsymbol{\nu}_{\boldsymbol{c}_k} \ \forall \ k$), then (6) solves the trajectory tracking problem. However mere kinematic control rarely suffices and often leads to performance degradation, since it ignores all dynamic effects [8].

B. Nonadaptive Dynamic Control

If the nonlinear dynamic functions $\boldsymbol{f}_k$ and $\boldsymbol{G}_k$ are assumed *perfectly known*, the control law

$$\boldsymbol{\tau}_k = \boldsymbol{G}_k^{-1}\left(\boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} - \boldsymbol{\nu}_k - \boldsymbol{f}_k + k_d\left(\boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k\right)\right) \tag{7}$$

with the design parameter $-1 < k_d < 1$, yields the following plant-independent, closed-loop dynamics

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} + k_d\left(\boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k\right) + \boldsymbol{\epsilon}_{k+1}. \tag{8}$$

This solves the velocity tracking problem since (8) and the choice of k_d, clearly indicate that $|\boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k| \rightarrow \boldsymbol{\epsilon}_k$ as $k \rightarrow \infty$. It is important to note that:

Remark 3.1: The control law (7) requires the velocity command vector $\boldsymbol{\nu}_{\boldsymbol{c}}$ to be known one sampling interval ahead. For this reason it is required to advance the kinematic law (6) by one sampling interval. This is achieved by generating the reference trajectory vectors corresponding to the $(k+1)$ instant, and using a first order hold to estimate $\boldsymbol{p}_{k+1}$ from $\boldsymbol{p}_k$. The latter is a valid approximation in the light of Assumption 2.2.

Remark 3.2: The case with $k_d = 0$ in (7), corresponds to *deadbeat control* associated with digital control.

C. Dual Adaptive Dynamic Control

The dynamic control law (7) driven by the kinematic law (6), solves the trajectory tracking problem when the WMR dynamic functions $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ in (5) are completely known. As emphasised in Section I this is rarely the case in real-life robotic applications, commonly manifesting: unmodelled dynamics, unknown/time-varying parameters and imperfect/noisy measurement. Consequently, we consider $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ to be completely unknown.

1) Neuro-Stochastic Function Estimator: A sigmoidal MLP ANN is used for the approximation of the vector of discrete nonlinear functions $\boldsymbol{f}_{k-1}$. Unlike the activation functions synonymous with other classes of ANNs, such as Gaussian radial basis function (RBF) ANNs [15], the sigmoidal functions in MLPs are not localised, implying that typically MLP networks require less neurons than RBF ANNs for the same degree of accuracy. This implies that MLPs are less computationally demanding, making them attractive for high-order systems, since the number of neurons need not rise exponentially with the number of states, as with other types of ANNs that exhibit the *curse of dimensionality* [17].

The MLP ANN used to approximate $\boldsymbol{f}_{k-1}$, which estimate is denoted as $\hat{\boldsymbol{f}}_{k-1}$, is given by

$$\hat{\boldsymbol{f}}_{k-1} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_{k-1}, \hat{\boldsymbol{a}}_k)\hat{\boldsymbol{w}}_{\boldsymbol{1}_k} \\ \boldsymbol{\phi}^T(\boldsymbol{x}_{k-1}, \hat{\boldsymbol{a}}_k)\hat{\boldsymbol{w}}_{\boldsymbol{2}_k} \end{bmatrix}, \tag{9}$$

in the light of the following definitions and assumption:

Definition 3.1: $\boldsymbol{x}_{k-1} \triangleq \begin{bmatrix} \boldsymbol{\nu}_{k-1} & 1 \end{bmatrix}$ denotes the ANN input and the augmented constant serves as a bias input.

Definition 3.2: $\boldsymbol{\phi}(\cdot,\cdot)$ is the vector of activation functions, whose ith element is given by $\phi_i = 1/\left(1+\exp\left(-\hat{\boldsymbol{s}}_{\boldsymbol{i}}^T\boldsymbol{x}\right)\right)$, where $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ is ith vector element in the group vector $\hat{\boldsymbol{a}}$; i.e. $\hat{\boldsymbol{a}} \triangleq \begin{bmatrix} \hat{\boldsymbol{s}}_{\boldsymbol{1}}^T & \cdots & \hat{\boldsymbol{s}}_{\boldsymbol{L}}^T \end{bmatrix}^T$ where L denotes the number of neurons. In practice $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ characterises the shape of the ith neuron.

In this definition the time index has been dropped for clarity, and throughout the paper the ˆ notation is used to indicate that the operand is undergoing estimation.

Definition 3.3: $\hat{\boldsymbol{w}}_{\boldsymbol{i}_k}$ represents the synaptic weight estimate vector of the connection between the neuron hidden layer and the ith output element of the ANN.

Assumption 3.1: The input vector $\boldsymbol{x}_{k-1}$ is contained within a known, arbitrarily large compact set $\boldsymbol{\chi} \subset \mathbb{R}^2$.

It is known that $\boldsymbol{G}_{k-1}$ is a state-independent matrix with unknown elements (refer to Remark 2.2). Hence, its estimation does not require the use of an ANN. Moreover it is a symmetric matrix (refer to Remark 2.1), a property which is exploited to construct its estimate as follows:

$$\hat{\boldsymbol{G}}_{k-1} = \begin{bmatrix} \hat{g}_{1_{k-1}} & \hat{g}_{2_{k-1}} \\ \hat{g}_{2_{k-1}} & \hat{g}_{1_{k-1}} \end{bmatrix}, \tag{10}$$

where $\hat{g}_{1_{k-1}}$ and $\hat{g}_{2_{k-1}}$ represent the unknown elements.

The ANN weight-tuning algorithm is developed next. The following formulation is required in order to proceed.

Definition 3.4: The unknown parameters requiring estimation are grouped in a single vector $\hat{\boldsymbol{z}}_k \triangleq \left[\hat{\boldsymbol{v}}_k^T \;\; \hat{\boldsymbol{g}}_k^T\right]^T$, where $\hat{\boldsymbol{v}}_k \triangleq \left[\hat{\boldsymbol{w}}_{\boldsymbol{1}_k}^T \;\; \hat{\boldsymbol{w}}_{\boldsymbol{2}_k}^T \;\; \hat{\boldsymbol{a}}_k^T\right]^T$ and $\hat{\boldsymbol{g}}_k \triangleq \left[\hat{g}_{1_{k-1}} \;\; \hat{g}_{2_{k-1}}\right]$.

Definition 3.5: The *measured output* in the identification model (5) is denoted by $\boldsymbol{y}_k \triangleq \boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1}$.

Assumption 3.2: Inside the compact set $\boldsymbol{\chi}$ the neural network approximation error is negligibly small when the ANN parameter vector $\hat{\boldsymbol{v}}_k$ is equal to some unknown optimal vector denoted by $\boldsymbol{v}_k^*$.
This * notation is used throughout to refer to the optimal value of the operand. Assumption 3.2 is justified due to the *Universal Approximation Theorem* of ANN [15].

By (9), (10), Definitions 3.1 to 3.5 and Assumptions 3.1, 3.2; it follows that the stochastic dynamic model (5) can be represented in the following state-space form:

$$\begin{aligned} \boldsymbol{z}_{k+1}^* &= \boldsymbol{z}_k^* \\ \boldsymbol{y}_k &= \boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right) + \boldsymbol{\epsilon}_k, \end{aligned} \tag{11}$$

where $\boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right)$ is the nonlinear function of the unknown optimal parameters in $\boldsymbol{z}_k^*$ given by

$$\begin{aligned} \boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right) &\triangleq \hat{\boldsymbol{f}}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{\boldsymbol{1}_k}^*, \boldsymbol{w}_{\boldsymbol{2}_k}^*, \boldsymbol{a}_k^*) \\ &+ \hat{\boldsymbol{G}}_{k-1}(\boldsymbol{g}_k^*)\boldsymbol{\tau}_{k-1}. \end{aligned} \tag{12}$$

It is proper to note that:

Remark 3.3: The unknown optimal parameter vector $\boldsymbol{z}_k^*$, required for the estimation of $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ in (9) and (10) respectively, does not appear linearly in the system model (11). Consequently, nonlinear estimation techniques have to be used.
In this paper we opt to employ the well known Extended Kalman Filter (EKF) in predictive mode, for the estimation of $\boldsymbol{z}_{k+1}^*$, as detailed right after the following set of necessary preliminaries.

Definition 3.6: $\boldsymbol{\nabla}_{\boldsymbol{h}_k}$ denotes the Jacobian matrix of $\boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right)$ with respect to $\boldsymbol{z}_k^*$ evaluated at $\hat{\boldsymbol{z}}_k$.
By (9), (10) and (12) it can be shown that:

$$\boldsymbol{\nabla}_{\boldsymbol{h}_k} = \left[\boldsymbol{\nabla}_{\boldsymbol{f}_k} \;\; \boldsymbol{\nabla}_{\boldsymbol{\Gamma}_k}\right] \triangleq \left[\begin{array}{cc} \dfrac{\partial(\hat{\boldsymbol{f}}_{k-1})}{\partial(\hat{\boldsymbol{v}}_k)} & \dfrac{\partial(\hat{\boldsymbol{G}}_{k-1}\boldsymbol{\tau}_{k-1})}{\partial(\hat{\boldsymbol{g}}_k)} \end{array}\right]$$

where:

$$\frac{\partial(\hat{\boldsymbol{f}}_{k-1})}{\partial(\hat{\boldsymbol{v}}_k)} = \left[\begin{array}{ccc} \boldsymbol{\phi}_{k-1}^T & \boldsymbol{0}^T & \cdots \hat{w}_{1,i}(\phi_i - \phi_i^2)\boldsymbol{x}^T \cdots \\ \boldsymbol{0}^T & \boldsymbol{\phi}_{k-1}^T & \cdots \hat{w}_{2,i}(\phi_i - \phi_i^2)\boldsymbol{x}^T \cdots \end{array}\right]$$

where $i = 1, \; \ldots \; , L$ and $\hat{w}_{j,i}$ denotes the ith element of the jth output weight vector $\hat{\boldsymbol{w}}_{\boldsymbol{j}_k}$, notation-wise $\boldsymbol{\phi}_{k-1}$ implies that the activation function is evaluated for $\boldsymbol{x}_{k-1}$ and $\hat{\boldsymbol{a}}_k$, $\boldsymbol{0}$ denotes a zero vector having the same length as $\boldsymbol{\phi}_{k-1}$, and in this equation both ϕ_i and $\boldsymbol{x}$ correspond to time instant $(k-1)$;

$$\frac{\partial(\hat{\boldsymbol{G}}_{k-1}\boldsymbol{\tau}_{k-1})}{\partial(\hat{\boldsymbol{g}}_k)} = \left[\begin{array}{cc} \tau_{r_{k-1}} & \tau_{l_{k-1}} \\ \tau_{l_{k-1}} & \tau_{r_{k-1}} \end{array}\right],$$

where $\tau_{r_{k-1}}$ and $\tau_{l_{k-1}}$ are the first and second elements of the input torque vector $\boldsymbol{\tau}_{k-1}$ respectively.

Definition 3.7: The *information state*, I^k [15] consists of all the output measurements up to instant k and all the previous inputs, denoted by Y^k and U^{k-1} respectively.

Assumption 3.3: The density $p(\boldsymbol{z}_0^*) \sim \mathcal{N}(\bar{\boldsymbol{z}}_0, \boldsymbol{R}_{\boldsymbol{z}_0})$.

Assumption 3.4: $\boldsymbol{z}_0^*$ and $\boldsymbol{\epsilon}_k$ are mutually independent.
In the light of Definitions 3.6, 3.7 and Assumptions 3.3, 3.4, the EKF is applied to the nonlinear stochastic model (11). Inherently, it introduces the approximation

$$p(\boldsymbol{z}_{k+1}^* | I^k) \approx \mathcal{N}(\hat{\boldsymbol{z}}_{k+1}, \boldsymbol{P}_{k+1}), \tag{13}$$

where $\hat{\boldsymbol{z}}_{k+1}$ and $\boldsymbol{P}_{k+1}$ satisfy these recursive equations

$$\begin{aligned} \hat{\boldsymbol{z}}_{k+1} &= \hat{\boldsymbol{z}}_k + \boldsymbol{K}_k \boldsymbol{i}_k \\ \boldsymbol{P}_{k+1} &= \boldsymbol{P}_k - \boldsymbol{K}_k \boldsymbol{\nabla}_{\boldsymbol{h}_k} \boldsymbol{P}_k, \end{aligned} \tag{14}$$

where the EKF gain matrix and the innovations vector are given by $\boldsymbol{K}_k = \boldsymbol{P}_k \boldsymbol{\nabla}_{\boldsymbol{h}_k}^T \left(\boldsymbol{\nabla}_{\boldsymbol{h}_k} \boldsymbol{P}_k \boldsymbol{\nabla}_{\boldsymbol{h}_k}^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}\right)^{-1}$ and $\boldsymbol{i}_k = \boldsymbol{y}_k - \boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \hat{\boldsymbol{z}}_k\right)$ respectively, with initial conditions $\hat{\boldsymbol{z}}_0 = \bar{\boldsymbol{z}}_0$ and $\boldsymbol{P}_0 = \boldsymbol{R}_{\boldsymbol{z}_0}$. These initial conditions reflect the prior estimate of the unknown optimal vector and its uncertainty respectively.

Expressing $\boldsymbol{y}_{k+1}$ as a first order Taylor series around $\boldsymbol{z}_{k+1}^* = \hat{\boldsymbol{z}}_{k+1}$, yields the following approximation

$$\begin{aligned} \boldsymbol{y}_{k+1} \approx \; & \boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right) + \\ & \boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}} \left(\boldsymbol{z}_{k+1}^* - \hat{\boldsymbol{z}}_{k+1}\right) + \boldsymbol{\epsilon}_{k+1}, \end{aligned} \tag{15}$$

which leads the following lemma.

Lemma 3.1: On the basis of approximations (13) and (15) it follows that $p(\boldsymbol{y}_{k+1} | I^k)$ is *approximately* Gaussian with mean $\boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right)$ and covariance $\boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}} \boldsymbol{P}_{k+1} \boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}}^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}$.

Proof: The proof follows directly from the linearity of (15), the *approximate* conditional distribution of $\boldsymbol{z}_{k+1}^*$ in (13), and the Gaussian distribution of $\boldsymbol{\epsilon}_{k+1}$ as specified in Assumption 2.3. ■

The EKF formulation (14) constitutes the adaptation law for the proposed dual adaptive scheme. Additionally, it provides a real-time update of the density $p(\boldsymbol{y}_{k+1} | I^k)$ as detailed in Lemma 3.1. This information is essential in the dual control law detailed next.

2) The Control Law: The proposed control law is based on an explicit-type, suboptimal dual performance index based on the innovations dual method originally proposed by Milito et. al. [18] for single-input single-output (SISO) linear systems. This approach was later extended by Fabri and Kadirkamanathan [19] for the dual adaptive neural control of nonlinear SISO systems. In contrast to these works, our control law caters for the *nonlinear*, *multiple-input multiple-output (MIMO)* nature of the relatively more complex system, namely the WMR.

The innovation-based performance index J_{inn}, adopted from [19] and modified to fit the MIMO scenario at hand, is given by

$$\begin{aligned} J_{inn} \;\; = \;\; & E\Big\{ \left(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}_{k+1}}\right)^T \boldsymbol{Q}_{\boldsymbol{1}} \left(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}_{k+1}}\right) \\ & + \left(\boldsymbol{\tau}_k^T \boldsymbol{Q}_{\boldsymbol{2}} \boldsymbol{\tau}_k\right) + \left(\boldsymbol{i}_{k+1}^T \boldsymbol{Q}_{\boldsymbol{3}} \boldsymbol{i}_{k+1}\right) \Big| I^k \Big\}, \end{aligned} \tag{16}$$

where $E\left\{\cdot | I^k\right\}$ denotes the mathematical expectation conditioned on I^k, and the following definitions apply.

Definition 3.8: $\boldsymbol{y}_{\boldsymbol{d}_{k+1}}$ is the reference vector of $\boldsymbol{y}_{k+1}$ and is given by $\boldsymbol{y}_{\boldsymbol{d}_{k+1}} \triangleq \boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} - \boldsymbol{\nu}_{\boldsymbol{c}_k}$ (by Definition 3.5).

Definition 3.9: Design parameters $\boldsymbol{Q_1}$, $\boldsymbol{Q_2}$ and $\boldsymbol{Q_3}$ are *diagonal* and $\in \mathbb{R}^{2\times 2}$. Additionally $\boldsymbol{Q_1}$ is positive definite, $\boldsymbol{Q_2}$ is positive semi-definite and each element of $\boldsymbol{Q_3}$ is ≤ 0 and $\geq$ the corresponding element of $-\boldsymbol{Q_1}$.

It should be noted that:

Remark 3.4: The design parameter $\boldsymbol{Q_1}$ is introduced to penalise high deviations in the output, $\boldsymbol{Q_2}$ induces a penalty on large control signals and prevents ill-conditioning, and $\boldsymbol{Q_3}$ affects the innovation vector so as to induce the *dual* feature characterising our scheme.

It is now possible to present the dual adaptive control law.

Theorem 3.1: The control law minimising performance index J_{inn} (16), subject to the WMR dynamic model (5) and all the previous definitions, assumptions and Lemma 3.1, is given by

$$\begin{aligned} \boldsymbol{\tau}_k &= \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \hat{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N}_{k+1}\right)^{-1} \\ &\quad \times \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \left(\boldsymbol{y}_{\boldsymbol{d}_{k+1}} - \hat{\boldsymbol{f}}_k\right) - \boldsymbol{\kappa}_{k+1}\right), \end{aligned} \tag{17}$$

where the following definitions apply.

Definition 3.10: $\boldsymbol{Q_4} \triangleq \boldsymbol{Q_1} + \boldsymbol{Q_3}$, and the i^{th} row, j^{th} column term of any matrix $\boldsymbol{A_S}$ be denoted by $a_S(i,j)$.

Definition 3.11: $\boldsymbol{P}_{k+1}$ is repartitioned as

$$\boldsymbol{P}_{k+1} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{ff}_{k+1}} & \boldsymbol{P}_{\boldsymbol{Gf}_{k+1}}^T \\ \boldsymbol{P}_{\boldsymbol{Gf}_{k+1}} & \boldsymbol{P}_{\boldsymbol{GG}_{k+1}} \end{bmatrix},$$

where: $\boldsymbol{P}_{\boldsymbol{ff}_{k+1}} \in \mathbb{R}^{5L\times 5L}$ and $\boldsymbol{P}_{\boldsymbol{GG}_{k+1}} \in \mathbb{R}^{2\times 2}$.

Definition 3.12: Matrix $\boldsymbol{B} \triangleq \boldsymbol{P}_{\boldsymbol{Gf}_{k+1}} \boldsymbol{\nabla}_{\boldsymbol{f}_k}^T \boldsymbol{Q_4}$, so that $\boldsymbol{\kappa}_{k+1} \triangleq \begin{bmatrix} b(1,1) + b(2,2) & b(1,2) + b(2,1) \end{bmatrix}^T$.

Definition 3.13: The elements of $\boldsymbol{N}_{k+1}$ are given by:

$$\begin{aligned} n(1,1) &= q_4(1,1)p_{GG}(1,1) + q_4(2,2)p_{GG}(2,2) \\ n(2,2) &= q_4(1,1)p_{GG}(2,2) + q_4(2,2)p_{GG}(1,1) \\ n(1,1) &= 0.5 \times (q_4(1,1) + q_4(2,2)) \\ &\quad \times (p_{GG}(1,2) + p_{GG}(2,1)) \\ n(2,1) &= n(1,2). \end{aligned}$$

Note that the time index in $\boldsymbol{N}_{k+1}$ indicates that each individual element $p_{GG}(\cdot,\cdot)$ corresponds to $\boldsymbol{P}_{\boldsymbol{GG}_{k+1}}$.

Proof: Given the approximately Gaussian distribution $p(\boldsymbol{y}_{k+1}|I^k)$ in Lemma 3.1, and several general results from multivariate probability theory, it follows that

$$\begin{aligned} J_{inn} = \left(\boldsymbol{h}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}_{k+1}}\right)^T \boldsymbol{Q_1} \left(\boldsymbol{h}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}_{k+1}}\right) + \boldsymbol{\tau}_k^T \boldsymbol{Q_2} \boldsymbol{\tau}_k \\ + \ trace\left\{\boldsymbol{Q_4}\left(\boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}} \boldsymbol{P}_{k+1} \boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}}^T + \boldsymbol{R}_\epsilon\right)\right\}, \end{aligned}$$

where $\boldsymbol{h}_{k+1}$ denotes $\boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right)$. Replacing $\boldsymbol{h}_{k+1}$ by $\hat{\boldsymbol{f}}_k + \hat{\boldsymbol{G}}_k \boldsymbol{\tau}_k$, and employing the formulations in Definitions 3.6 and 3.11 to factorise completely in terms of $\boldsymbol{\tau}_k$; it is possible to differentiate the resulting cost function with respect to $\boldsymbol{\tau}_k$, and equate to zero to get the dual control law (17). The resulting second order partial derivative of J_{inn} with respect to $\boldsymbol{\tau}_k$ (the Hessian matrix), is given by $2 \times \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \hat{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N}_{k+1}\right)$. By Definitions 3.9, 3.13 it is clear that this Hessian matrix is positive definite, meaning that (17) *minimises* the dual performance index (16) *uniquely*. Moreover, the latter implies that the inverse term in (17) exists *without exceptions*. ■

Referring to control law (17) it is important to note that:

Remark 3.5: $\boldsymbol{Q_3}$ acts as a weighting factor where at one extreme, with $\boldsymbol{Q_3} = -\boldsymbol{Q_1}$, the controller completely ignores the estimates' uncertainty resulting in *HCE control*, and at the other extreme, with $\boldsymbol{Q_3} = \boldsymbol{0}$, it gives maximum attention to them, which leads to *cautious control*. For intermediate settings of $\boldsymbol{Q_3}$, the controller operates in a dual adaptive mode. It is well known that HCE control leads to large tracking errors and excessive control actions when the estimates' uncertainty is relatively high. On the other hand, cautious control is known for its slowness of response and *control turn-off* [15]. Consequently, dual control exhibits superior performance by striking a balance between the two extremes.

IV. Simulation Results

The WMR was simulated via the continuous-time, full model given by (1) and (2) with the following nominal parameter values: $b = 0.5$m, $r = 0.15$m, $d = 0.2$m, $m_c = 30$kg, $m_w = 2$kg, $I_c = 15$kgm^2, $I_w = 0.005$kgm^2, $I_m = 0.0025$kgm^2. Sampling interval $T = 50$ms and the sampled data was corrupted with noise $\boldsymbol{\epsilon}_k$. To render the simulations more realistic, a number of model parameters (such as masses, frictions and inertias) were allowed to vary randomly (within realistic limits) about their nominal values, from one simulation trial to another. The MLP ANN contained 10 neurons ($L = 10$) and $\hat{\boldsymbol{z}}_0$ was generated randomly. It took a standard desktop computer with no code optimisation merely 8s to simulate 30s of real-time. Clearly, this indicates that the proposed dual control algorithm is not computationally demanding.

For comparison purposes, trials were conducted using the three modes of operation in (17) namely: HCE ($\boldsymbol{Q_3} = -\boldsymbol{Q_1}$), cautious ($\boldsymbol{Q_3} = \boldsymbol{0}$) and dual ($\boldsymbol{Q_3} = -0.8\boldsymbol{Q_1}$). Another control mode, referred to as tuned non-adaptive (TNA) control, was also included for comparison. The TNA controller is implemented via (7) assuming the model parameter nominal values specified above. In contrast, the HCE, cautious and dual controllers, assume no preliminary information about the model. In Fig. 2: Plot (a) depicts the WMR (dual control) tracking the reference trajectory (reaching 2m/s). It clearly verifies the good tracking performance of the proposed scheme, even with non-zero initial conditions. Plots (c) and (d) compare the Euclidian vector norm of the pose error during the transient and steady-state performance (respectively), for the four controllers under test. Plot (c) clearly indicates that dual control exhibits the best transient initial performance among the adaptive modes (in accordance to Remark 3.5). It is not surprising that the TNA controller leads to better initial transient response, since it requires no learning process and is pre-tuned to the nominal parameters of the actual model. However this superiority is quickly lost in the steady-state phase, depicted in Plot (d), since by that time, the initially random estimates used by the adaptive controllers would have converged to better

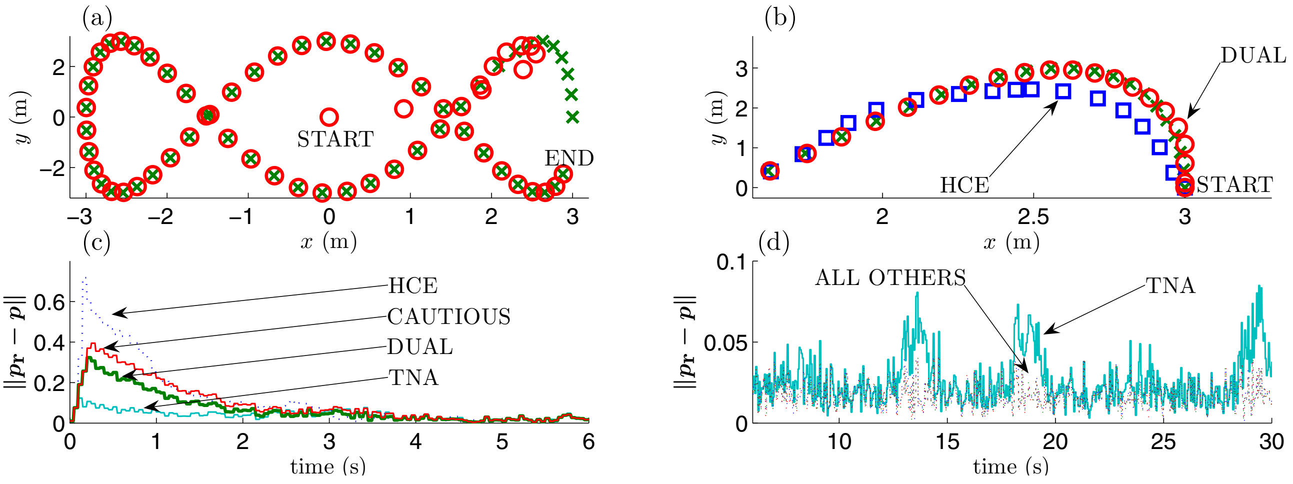

Fig. 2. (a): reference (×) & dual WMR (◯); (b): same as (a) & HCE WMR (□); (c): transient performance; (d): steady-state performance.

approximations of the real functions, while the TNA would still be assuming the far less accurate nominal parameters that it was originally tuned with. Plot (b) also verifies the superiority of dual control over the more crude HCE controller. To quantify the performance objectively, a Monte Carlo analysis involving 500 trials was performed on all four controllers. The accumulated Euclidian norm of the pose error was calculated over the whole three minute simulation interval after each trial. The mean and variance of the accumulated cost are tabulated in Table I, with the dual control case leading to the best performance, as stated in Remark 3.5.

V. Conclusions

The novelty in this paper comprises the introduction of dual neuro-adaptive control for the discrete-time, dynamic control of mobile robots using MLPs. The proposed controller exhibits great improvements in steady-state and transient performance over non-adaptive and non-dual adaptive schemes respectively. This was confirmed by simulations and Monte Carlo analysis. Future research will investigate the addition of fault-tolerant schemes for the control of mobile robots. We also anticipate to get satisfactory experimental results once the proposed algorithm is tested on a real mobile robot.

References

[1] I. Kolmanovsky and N. H. McClamroch, "Developments in nonholonomic control problems," *IEEE Control Systems Magazine*, vol. 15, no. 6, pp. 20–36, 1995.

[2] C. Canudas de Wit, H. Khennoul, C. Samson, and O. J. Sordalen, "Nonlinear control design for mobile robots," in *Recent Trends in Mobile Robots*, ser. Robotics and Automated Systems, Y. F. Zheng, Ed. World Scientific, 1993, ch. 5, pp. 121–156.

[3] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE International Conference of Robotics and Automation*, Cincinnati, OH, May 1990, pp. 384–389.

[4] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," in *Proc. IEEE 34th Conference on Decision and Control (CDC'95)*, New Orleans, LA, Dec. 1995, pp. 3805–3810.

[5] N. Sarkar, X. Yun, and V. Kumar, "Control of mechanical systems with rolling constraints: Applications to dynamic control of mobile robots," *International Journal of Robotics Research*, vol. 13, no. 1, pp. 55–69, Feb. 1994.

[6] M. L. Corradini and G. Orlando, "Robust tracking control of mobile robots in the presence of uncertainties in the dynamic model," *Journal of Robotic Systems*, vol. 18, no. 6, pp. 317–323, 2001.

[7] M. L. Corradini, G. Ippoliti, and S. Longhi, "Neural networks based control of mobile robots: Development and experimental validation," *Journal of Robotic Systems*, vol. 20, no. 10, pp. 587–600, 2003.

[8] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Networks*, vol. 9, no. 4, pp. 589–600, July 1998.

[9] M. K. Bugeja and S. G. Fabri, "Multilayer perceptron functional adaptive control for trajectory tracking of wheeled mobile robots," in *Proc. 2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO'05)*, vol. 3, Barcelona, Spain, Sept. 2005, pp. 66–72.

[10] ——, "Neuro-adaptive dynamic control for trajectory tracking of mobile robots," in *Proc. 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO'06)*, Setúbal, Portugal, Aug. 2006, pp. 404–411.

[11] ——, "Multilayer perceptron adaptive dynamic control for trajectory tracking of mobile robots," in *Proc. 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON'06)*, Paris, France, Nov. 2006, pp. 3798–3803.

[12] A. A. Fel'dbaum, "Dual control theory I-II," *Automation and Remote Control*, vol. 21, pp. 874–880, 1033–1039, 1960.

[13] ——, "Dual control theory III-IV," *Automation and Remote Control*, vol. 22, pp. 1–12, 109–121, 1961.

[14] ——, *Optimal Control Systems*. New York, NY: Academic Press, 1965.

[15] S. G. Fabri and V. Kadirkamanathan, *Functional Adaptive Control: An Intelligent Systems Approach*. London, UK: Springer-Verlag, 2001.

[16] N. M. Filatov and H. Unbehauen, "Survey of adaptive dual control methods," in *Proc. IEE Control Theory Applications*, vol. 147, no. 1, Jan. 2000, pp. 118–128.

[17] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ: Princeton University Press, 1961.

[18] R. Milito, C. S. Padilla, R. A. Padilla, and D. Cadorin, "An innovations approach to dual control," *IEEE Transactions on Automatic Control*, vol. 27, no. 1, pp. 133–137, Feb. 1982.

[19] S. G. Fabri and V. Kadirkamanathan, "Dual adaptive control of nonlinear stochastic systems using neural networks," *Automatica*, vol. 34, no. 2, pp. 245–253, 1998.

TABLE I

Monte Carlo analysis results

	HCE	CAUTIOUS	DUAL	TNA
Average cost	501	372	352	399
Variance	58,902	87	23	1259