2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Dual Adaptive Control for Trajectory Tracking of Mobile Robots

Marvin K. Bugeja† and Simon G. Fabri‡
Department of Electrical Power and Control Engineering
University of Malta
Msida (MSD06), Malta
† merv@ieee.org ‡ sgfabr@eng.um.edu.mt

Abstract— **This paper presents a novel dual adaptive dynamic controller for trajectory tracking of nonholonomic wheeled mobile robots. The controller is developed entirely in discrete-time and the and the robot's nonlinear dynamic functions are assumed to be unknown. A Gaussian radial basis function 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.**

Index Terms— **Nonholonomic mobile robots, trajectory tracking, dual adaptive control, neural networks.**

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 [2]; and pure, smooth, time-invariant feedback stabilisation of the Cartesian model is unattainable [3].

Originally researchers focused only on kinematic control of nonholonomic vehicles [1], [2], [4], assuming that the control signals instantaneously establish the desired robot velocities. This is commonly known as *perfect velocity tracking* [5]. This approach may be reasonably valid for non-critical applications. However it stands to reason that controllers based on a full dynamic model [5], [6] capture better the behaviour of real robots because they account for dynamic effects such as mass, friction and inertia, which are otherwise neglected by a mere kinematic controller. On the other hand, the exact values of the parameters in the dynamic model 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.

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

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 in the face of uncertainty [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 in the kinematic/dynamic model, robust sliding mode control and parametric adaptive control [8] 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 [9]–[11].

Adaptive controllers which have hitherto been proposed for the control of mobile robots [9]–[11] are based on the *heuristic certainty equivalence* (HCE) property [12]. In other words, the estimated functions are used in the control law 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 poor control performance. The latter is usually exhibited as large tracking errors and excessive control actions which if not limited can excite unmodelled dynamics or pull the system outside the ANN approximation region.

To account for the estimates' uncertainty in the control design we opt to employ stochastic adaptive control techniques, more specifically the so-called *dual control* principle introduced by Fel'dbaum [13]. Basically a dual adaptive control law is designed with two aims in mind: (i) to ensure that the system output tracks the desired reference signal, with due consideration given to the estimates' uncertainty; (ii) to excite the plant sufficiently so as to accelerate the estimation process, thereby reducing quickly the uncertainty in future estimates. These two features are known as *caution* and *probing* respectively [12].

In contrast to other work on mobile robots, the novel contribution of this paper is to introduce a neural adaptive dynamic controller that features these dual adaptive properties. Moreover the control law is developed in discrete-time, and in contrast to [9] has the desirable property of yielding closed loop dynamics which are completely independent of the robot parameters. In this paper we focus on trajectory tracking of wheeled mobile robots (WMRs). Nevertheless the employed framework is completely modular, and the dynamic controller can easily be adopted for other robot control scenarios, such as posture stabilisation or path tracking [2].

The presented method employs a Gaussian radial basis function (RBF) ANN to estimate the unknown 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 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.

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 that the robot wheels roll without slipping, the mobile platform is subject to three kinematic constraints, two of which are nonholonomic [6]. The three 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, it is straightforward to 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 $\boldsymbol{\nu} \triangleq [\nu_r \quad \nu_l]^T \triangleq \left[\dot{\theta}_r \quad \dot{\theta}_l\right]^T$.

B. Dynamics

The dynamic equations of motion of this WMR can be written in matrix form as

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

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 the frictional forces vector, $\boldsymbol{E}(\boldsymbol{q})$ is the input transformation matrix, $\boldsymbol{\tau}$ is the torque vector and $\boldsymbol{\lambda}$ is the constraint forces vector [6]. 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:

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

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

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

Remark 2.1: $\bar{\boldsymbol{M}}$ is symmetric, positive definite, and independent of the state vector and/or its derivatives.

Remark 2.2: $\bar{\boldsymbol{V}}(\dot{\boldsymbol{q}})$ and $\bar{\boldsymbol{F}}(\dot{\boldsymbol{q}})$ constitute the nonlinear terms in the WMR dynamics.

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 explicit forward Euler approximation with sampling interval T seconds the following discrete-time dynamic model is obtained

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

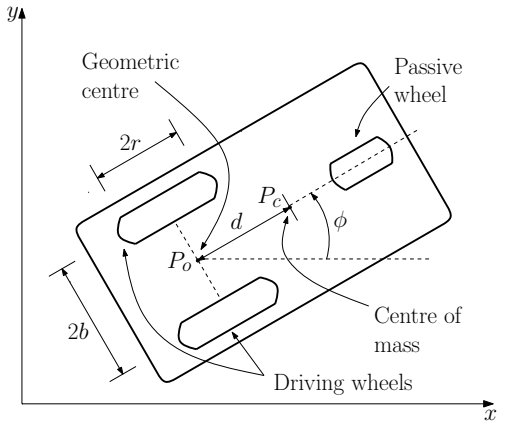

Fig. 1. Differentially driven wheeled mobile robot

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 assumption

Assumption 2.3: $\boldsymbol{\epsilon}_k$ is an independent, zero-mean, white, Gaussian process, with known 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* [4]. Basically, the time dependent trajectory to be tracked by the WMR 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 vector $\boldsymbol{e}_k$ is commonly defined as

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

where $\boldsymbol{p}_{\boldsymbol{r}k} \triangleq [x_{rk} \;\; y_{rk} \;\; \phi_{rk}]^T$ denotes the virtual vehicle sampled pose. 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 controller originally presented in [4] given by

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

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_{rk} and ω_{rk} 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) completely solves the trajectory tracking problem. However as mentioned earlier, mere kinematic control rarely suffices and often leads to performance degradation in demanding, practical control situations [9].

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 closed-loop dynamics

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

Note that in contrast to [9], the proposed control law in (7) has the desirable property of yielding closed loop dynamics which are completely independent of the robot parameters. Moreover, this control law solves the velocity tracking problem since (8) and the choice of k_d, clearly indicate that $|\boldsymbol{\nu}_{\boldsymbol{c}k} - \boldsymbol{\nu}_k| \to \boldsymbol{\epsilon}_k$ as $k \to \infty$. 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 (assumed available from an external path-planning algorithm) 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 reasonable since a typical sampling interval (in the order of milliseconds) is high enough compared to the relatively slow dynamics of a WMR.

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

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 sensor measurements. Consequently throughout the following, we consider $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ to be completely unknown. In previous publications we addressed the issue of adaptivity for WMR via an HCE approach [10], [11]. This is improved upon in the following, where we develop a control scheme featuring dual adaptive properties.

1) Neuro-Stochastic Function Estimator: A Gaussian RBF ANN is used to approximate the vector of discrete nonlinear functions $\boldsymbol{f}_{k-1}$. The advantage of using Gaussian RBF ANN [12] comes from the fact that the RBF network weights appear linearly in the final state-space output equation. This detail enables the use of a standard Kalman filter for weight estimation, leading to the least-squares-sense optimal tuning of the neural network weights.

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

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

in the light of the following definitions and assumptions:

Definition 3.1: $\boldsymbol{x}_{\boldsymbol{f}_{k-1}}$ represents the neural network input vector, which in this case is set to $\boldsymbol{\nu}_{k-1}$.

Definition 3.2: $\boldsymbol{\phi}_{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{f}})$ is the Gaussian RBF vector whose ith element is given by

$$\phi_{f_i} = \exp\left(-0.5 \times \left(\boldsymbol{x}_{\boldsymbol{f}} - \boldsymbol{m}_{\boldsymbol{f}_i}\right)^T \boldsymbol{R}_{\boldsymbol{f}}{}^{-1} \left(\boldsymbol{x}_{\boldsymbol{f}} - \boldsymbol{m}_{\boldsymbol{f}_i}\right)\right),$$

where $\boldsymbol{m}_{\boldsymbol{f}_i}$ is the coordinate vector of the centre of the ith basis function, $\boldsymbol{R}_{\boldsymbol{f}}$ is the corresponding covariance matrix, and the time index has been dropped for clarity.

Definition 3.3: $\hat{\boldsymbol{w}}_{\boldsymbol{i}k}$ represents the weight vector of the connection between the RBFs and the ith output element of the network.

Definition 3.4: L_f denotes the number of basis functions.

Assumption 3.1: The ANN input vector $\boldsymbol{x}_{\boldsymbol{f}_{k-1}}$ is assumed to be contained within an arbitrarily large compact set $\boldsymbol{\chi}_{\boldsymbol{f}} \subset \mathbb{R}^2$ fixed by the designer.

Assumption 3.2: The basis functions are shaped and placed within the compact set $\boldsymbol{\chi}_{\boldsymbol{f}}$ by setting $\boldsymbol{m}_{\boldsymbol{f}_i}$ and $\boldsymbol{R}_{\boldsymbol{f}}$ accordingly.

Sanner and Slotine in [14] show that with knowledge of the bounds on the frequency characteristics of the function being estimated, the number of basis functions and their corresponding means and covariance matrices can be appropriately selected. Moreover, simulation results indicate that the control performance is not overly sensitive to the placement and covariance of the RBFs.

It is known that $\boldsymbol{G}_{k-1}$ is a symmetric, state-independent matrix with unknown elements (refer to Remark 2.1). Consequently its estimation does not require the use of an ANN. These properties are exploited to construct its estimate at instant $(k-1)$ as follows

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

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

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

Definition 3.5:

$$\boldsymbol{\Phi}_{k-1} \triangleq \begin{bmatrix} \boldsymbol{\phi}_{\boldsymbol{f}}^T & \mathbf{0}_{\boldsymbol{f}}^T \\ \mathbf{0}_{\boldsymbol{f}}^T & \boldsymbol{\phi}_{\boldsymbol{f}}^T \end{bmatrix}, \quad \boldsymbol{\Gamma}_{k-1} \triangleq \begin{bmatrix} \tau_{rk-1} & \tau_{lk-1} \\ \tau_{lk-1} & \tau_{rk-1} \end{bmatrix},$$

and $\boldsymbol{H}_{k-1} \triangleq [\boldsymbol{\Phi}_{k-1} \ \vdots \ \boldsymbol{\Gamma}_{k-1}]$, where $\mathbf{0}_{\boldsymbol{f}}$ is a zero vector having the same length as $\boldsymbol{\phi}_{\boldsymbol{f}}$, τ_{rk-1} and τ_{lk-1} are the first and second elements of the input torque vector $\boldsymbol{\tau}_{k-1}$ respectively, and the time index in $\boldsymbol{\Phi}_{k-1}$ indicates that $\boldsymbol{\phi}_{\boldsymbol{f}}$ is evaluated for $\boldsymbol{x}_{\boldsymbol{f}_{k-1}}$.

Definition 3.6: The individual weight vectors are given by $\hat{\boldsymbol{w}}_{\boldsymbol{f}_k} \triangleq \left[\hat{\boldsymbol{w}}_{\mathbf{1}k}^T \ \ \hat{\boldsymbol{w}}_{\mathbf{2}k}^T\right]^T$ and $\hat{\boldsymbol{w}}_{\boldsymbol{G}k} \triangleq \left[\hat{g}_{1k-1} \ \ \hat{g}_{2k-1}\right]^T$; and grouped into a single vector $\hat{\boldsymbol{w}}_k \triangleq [\hat{\boldsymbol{w}}_{\boldsymbol{f}k}^T \ \ \hat{\boldsymbol{w}}_{\boldsymbol{G}k}^T]^T$.

Definition 3.7: The *measured output* in the identification model (5) is denoted by $\boldsymbol{y}_k \triangleq \boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1}$.

Definition 3.8: The *information state* [12] denoted by I^k 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: Assume that inside the compact set $\boldsymbol{\chi}_{\boldsymbol{f}}$, the neural network approximation error is negligibly small when the weight vector $\hat{\boldsymbol{w}}_{\boldsymbol{f}_k}$is equal to some optimal vector denoted by $\boldsymbol{w}^*_{\boldsymbol{f}_k}$.

This assumption is justified in the light of the *Universal Approximation Theorem* of neural networks [12]. Similarly, let $\boldsymbol{w}^*_{\boldsymbol{G}k}$ and $\boldsymbol{w}^*_k$ denote the optimal estimates of $\hat{\boldsymbol{w}}_{\boldsymbol{G}k}$ and $\hat{\boldsymbol{w}}_k$ respectively.

Assumption 3.4: The density $p(\boldsymbol{w}^*_0) \sim \mathcal{N}(\bar{\boldsymbol{w}}_0, \boldsymbol{R}_{\boldsymbol{w}0})$.

In practice $\boldsymbol{R}_{\boldsymbol{w}0}$ can be used to reflect the extent of prior knowledge of the weight vector; larger values indicating less confidence in the initial weight vector $\bar{\boldsymbol{w}}_0$ [15].

Assumption 3.5: $\boldsymbol{w}^*_0$ and $\boldsymbol{\epsilon}_k$ are mutually independent $\forall k$.

By (9), (10), Definitions 3.1 to 3.7, and Assumptions 3.1 to 3.3; it follows that the WMR stochastic dynamic model (5) can be represented in the following state-space form

$$\begin{aligned} \boldsymbol{w}^*_{k+1} &= \boldsymbol{w}^*_k \\ \boldsymbol{y}_k &= \boldsymbol{H}_{k-1}\boldsymbol{w}^*_k + \boldsymbol{\epsilon}_k. \end{aligned} \tag{11}$$

It is proper to note that:

Remark 3.3: The optimal weight vector $\boldsymbol{w}^*_k$ is the only unknown parameter in (11), and it needs to be estimated in order to determine the values of $\hat{\boldsymbol{f}}_{k-1}$ and $\hat{\boldsymbol{G}}_{k-1}$ in (9) and (10) respectively.

Remark 3.4: $\boldsymbol{w}^*_k$ appears linearly in (11).

The latter (referred to earlier to justify the use of RBF ANN) is exploited by employing the well established Kalman filter in predictive mode for the optimal (least-square sense) stochastic estimation of $\boldsymbol{w}^*_{k+1}$, as detailed in the following.

Lemma 3.1: In the light of all previous definitions, Assumptions 2.3, 3.1 to 3.5 and Remark 3.4, it follows that $p(\boldsymbol{w}^*_{k+1}|I^k) \sim \mathcal{N}(\hat{\boldsymbol{w}}_{k+1}, \boldsymbol{P}_{k+1})$, and so $\hat{\boldsymbol{w}}_{k+1}$ is the optimal estimate of $\boldsymbol{w}^*_{k+1}$ conditioned on I^k given that $\hat{\boldsymbol{w}}_{k+1}$ and $\boldsymbol{P}_{k+1}$ satisfy the following Kalman filter equations [12]:

$$\hat{\boldsymbol{w}}_{k+1} = \hat{\boldsymbol{w}}_k + \boldsymbol{K}_k\boldsymbol{i}_k \text{ and } \boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{H}_{k-1}\boldsymbol{P}_k, \tag{12}$$

where the Kalman gain matrix and the innovations vector are given by $\boldsymbol{K}_k = \boldsymbol{P}_k\boldsymbol{H}_{k-1}^T\left(\boldsymbol{H}_{k-1}\boldsymbol{P}_k\boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}\right)^{-1}$ and $\boldsymbol{i}_k = \boldsymbol{y}_k - \boldsymbol{H}_{k-1}\hat{\boldsymbol{w}}_k$ respectively. Additionally $p(\boldsymbol{y}_{k+1}|I^k) \sim \mathcal{N}(\boldsymbol{H}_{\boldsymbol{k}}\hat{\boldsymbol{w}}_{k+1}, \boldsymbol{H}_k\boldsymbol{P}_{k+1}\boldsymbol{H}_k^T + \boldsymbol{R}_{\boldsymbol{\epsilon}})$.

Proof: The proof follows directly that of the standard predictive type Kalman filter, when applied to the state-space stochastic model (11). ■

The Kalman filter formulation (12) constitutes the adaptation law for the proposed dual adaptive scheme. Additionally, it provides a real-time update of the density $p(\boldsymbol{y}_{k+1}|I^k)$. This information is essential in dual control since the uncertainty of the estimates is not ignored.

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.* [16] for single-input single-output (SISO) linear systems. This approach was later adopted by Fabri and Kadirkamanathan [15] 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 [15] and modified to fit the MIMO scenario at hand, is given by

$$\begin{aligned} J_{inn} &= E\Big\{ \left(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}k+1}\right)^T \boldsymbol{Q_1} \left(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}k+1}\right) \\ &\quad + \left(\boldsymbol{\tau}_k^T \boldsymbol{Q_2} \boldsymbol{\tau}_k\right) + \left(\boldsymbol{i}_{k+1}^T \boldsymbol{Q_3} \boldsymbol{i}_{k+1}\right) \Big| I^k \Big\}, \end{aligned} \quad (13)$$

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

Definition 3.9: $\boldsymbol{y}_{\boldsymbol{d}k+1}$ is the reference vector of $\boldsymbol{y}_{k+1}$ and is given by $\boldsymbol{y}_{\boldsymbol{d}k+1} \triangleq \boldsymbol{\nu}_{\boldsymbol{c}k+1} - \boldsymbol{\nu}_{\boldsymbol{c}k}$ (refer to Definition 3.7).

Definition 3.10: 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 3.5: 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, proposed in this work.

Theorem 3.1: The control law minimising J_{inn} (13), subject to the WMR dynamic model (5) and all the previous definitions, assumptions and Lemma 3.1, is given by

$$\begin{aligned} \boldsymbol{\tau}_k &= \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \hat{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N}_{k+1}\right)^{-1} \\ &\quad \times \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \left(\boldsymbol{y}_{\boldsymbol{d}k+1} - \hat{\boldsymbol{f}}_k\right) - \boldsymbol{\kappa}_{k+1}\right), \end{aligned} \quad (14)$$

where the following definitions apply:

Definition 3.11: Let $\boldsymbol{Q_4} \triangleq \boldsymbol{Q_1} + \boldsymbol{Q_3}$, and the i^{th} row, j^{th} column element of any matrix $\boldsymbol{A_S}$ be denoted by $a_S(i,j)$.

Definition 3.12: Note that $\boldsymbol{P}_{k+1}$ is repartitioned as

$$\boldsymbol{P}_{k+1} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{f}\boldsymbol{f}k+1} & \boldsymbol{P}_{\boldsymbol{G}\boldsymbol{f}k+1}^T \\ \boldsymbol{P}_{\boldsymbol{G}\boldsymbol{f}k+1} & \boldsymbol{P}_{\boldsymbol{G}\boldsymbol{G}k+1} \end{bmatrix},$$

where: $\boldsymbol{P}_{\boldsymbol{f}\boldsymbol{f}k+1} \in \mathbb{R}^{2L_f \times 2L_f}$ and $\boldsymbol{P}_{\boldsymbol{G}\boldsymbol{G}k+1} \in \mathbb{R}^{2\times 2}$.

Definition 3.13: If auxiliary matrix $\boldsymbol{B} \triangleq \boldsymbol{P}_{\boldsymbol{G}\boldsymbol{f}k+1} \boldsymbol{\Phi}_k^T \boldsymbol{Q_4}$, then $\boldsymbol{\kappa}_{k+1} \triangleq \begin{bmatrix} b(1,1)+b(2,2) & b(1,2)+b(2,1) \end{bmatrix}^T$.

Definition 3.14: The elements of $\boldsymbol{N}_{k+1}$ are given by:

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

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

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

$$\begin{aligned} J_{inn} &= \left(\boldsymbol{H}_k \hat{\boldsymbol{w}}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}k+1}\right)^T \boldsymbol{Q_1} \left(\boldsymbol{H}_k \hat{\boldsymbol{w}}_{k+1} - \boldsymbol{y}_{\boldsymbol{d}k+1}\right) \\ &\quad + trace\left\{\boldsymbol{Q_4}\left(\boldsymbol{H}_k \boldsymbol{P}_{k+1} \boldsymbol{H}_k^T + \boldsymbol{R_\epsilon}\right)\right\} + \boldsymbol{\tau}_k^T \boldsymbol{Q_2} \boldsymbol{\tau}_k. \end{aligned}$$

By replacing $\boldsymbol{H}_k \hat{\boldsymbol{w}}_{k+1}$ by $\hat{\boldsymbol{f}}_k + \hat{\boldsymbol{G}}_k \boldsymbol{\tau}_k$, and employing the formulations in Definitions 3.5 and 3.12 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 (14). 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 3.10, 3.14 it is clear that the Hessian matrix is positive definite, meaning that $\boldsymbol{\tau}_k$ (14) minimises the dual performance index (13) *uniquely*. Moreover, the latter implies that the inverse term in (14) exists without exceptions. ■

Referring to control law (14) it is important to note that:

Remark 3.6: $\boldsymbol{Q_3}$ 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* [12]. 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 robot parameters (such as masses, frictions and inertias), were allowed to vary realistically about a set of nominal values, from one simulation trial to another. The ANN contained 49 RBFs with $\boldsymbol{R_f} = 100\boldsymbol{I_2}$, where $\boldsymbol{I_i}$ denotes an $(i \times i)$ identity matrix. $\hat{\boldsymbol{w}}_0$ was generated randomly. Using MATLAB®, it took a standard desktop computer (Pentium®4 @ 3GHz, 512MB RAM) with no code optimisation merely 5s 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 (14) 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. This was implemented via (7)

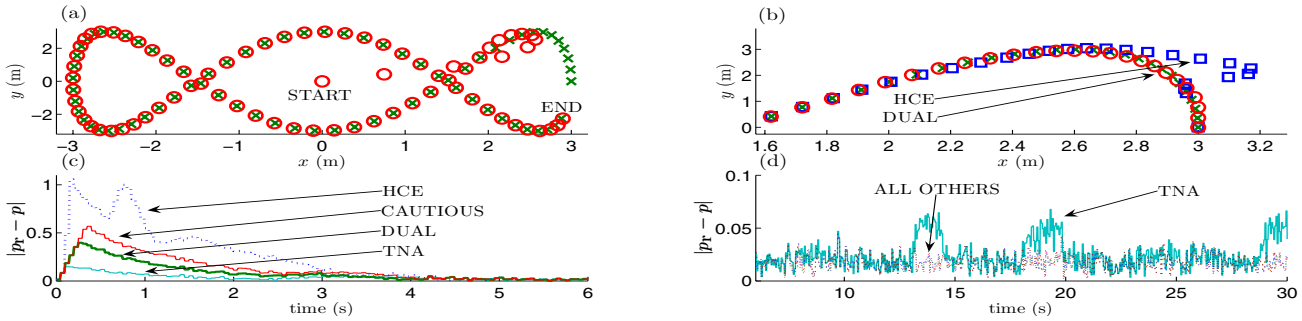

Fig. 2. (a): reference (×) & dual WMR (◯); (b): same as (a) & HCE WMR (□); (c): transient performance; (d): steady state performance.

assuming the nominal values for the model parameters. 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 1.8m/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 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 3.6). 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 approximates, 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 100 trials was performed. The accumulated Euclidian norm of the pose error was calculated over the whole simulation interval (3 minutes) after each trial. The mean and variance of the accumulated cost over 100 trials are tabulated in Table I, where the lower values of the mean and variance in the dual control case, substantiate the arguments in Remark 3.6.

V. CONCLUSIONS

The novelty in this paper comprises the introduction of dual adaptive control for the discrete-time, dynamic control of mobile robots. 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 use of *multiple model approaches* [12] to introduce fault-tolerant schemes for the control of mobile robots. We also anticipate to get satisfactory experimental results once the proposed algorithm is implemented on a real mobile robot in the near future.

TABLE I

MONTE CARLO ANALYSIS RESULTS

	HCE	CAUTIOUS	DUAL	TNA
Mean cost	1602	398	362	408
Variance	6.2×10^7	186	33	1381

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] R. W. Brockett, *Asymptotic Stability and Feedback Stabilisation*, ser. Differential Geometric Control Theory, R. S. Millman and H. J. Sussman, Eds. Boston, MA: Birkhäuser, 1983.

[4] 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.

[5] 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.

[6] 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.

[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] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, Oct. 2000.

[9] 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.

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

[13] A. A. Fel'dbaum, *Optimal Control Systems*. New York, NY: Academic Press, 1965.

[14] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[15] 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.

[16] 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.