Unscented Transform-based Dual Adaptive Control for Mobile Robots: Comparative Analysis and Experimental Validation

Marvin K. Bugeja and Simon G. Fabri
Department of Systems and Control Engineering
University of Malta
Msida, Malta
Email: marvin.bugeja@um.edu.mt, simon.fabri@um.edu.mt

***Abstract*—Adaptive control involves both estimation and control, which are generally interdependent and partly in conflict. Yet, the majority of adaptive controllers separate the two by assuming that certainty equivalence holds, even if this is not the case. In contrast a *dual adaptive* controller, based on the idea postulated by A. Fel'dbaum in the early 1960s, aims to strike a balance between estimation and control at all times. In this manner, the control law is a function of the estimates' uncertainty, besides the estimates themselves, thereby leading to improved control performance. Few such controllers have ever been implemented and tested in practice, especially within the context of intelligent control, and to the best of our knowledge none on mobile robots. This paper present two novel dual adaptive neural control schemes for the dynamic control of mobile robots in the presence of functional uncertainty. Furthermore, by means of realistic Monte Carlo simulations and real-life experiments, a thorough comparative analysis is performed. A notable novel contribution of this work is the use of the unscented transform within the context of dual adaptive control, aimed at improving further the performance of the system.**

***Index Terms*—Dual adaptive control; nonlinear stochastic control; neural networks; unscented transform; mobile robots.**

I. Introduction

A major motive for adaptive control is the need to have automatic systems that operate satisfactorily in the ambience of uncertainty. The uncertainty is typically due to unknown and/or time-varying structure or parameters pertaining to the system or process under control. Hence, in addition to keeping the controlled variable tracking its reference, an adaptive controller needs to simultaneously estimate the unknown system functions or parameters. These two objectives, termed *control* and *estimation* respectively, are generally interdependent and partly in conflict, in that typically estimation improves with perturbing (persistently exciting) input signals, while tracking performance does not. On the other hand, good tracking performance still requires good estimates.

Most of the adaptive controllers proposed over the past fifty-five years, including the well-established model-reference adaptive systems (MRAS) and self-tuning regulators (STRs), artificially separate estimation and control via the heuristic certainty equivalence (HCE) assumption. In this manner the parameter estimates are used in the control law as if they were the true values of the unknown parameters, without any due consideration to their inherent uncertainty. Though simple to implement, and adequately applied in many applications, HCE adaptive control can lead to large tracking errors and excessive control actions, which can excite unmodelled dynamics or even lead to instability and possibly physical damage [1]. These effects are more pronounced in situations characterized by high uncertainty, short control horizon and/or time-varying system parameters [2], [3].

The issue of simultaneous estimation and control is best addressed via stochastic adaptive control theory. Unlike deterministic approaches, in stochastic adaptive control the uncertainty in the system; be it due to unknown process parameters, noisy measurements, or both; is characterized by probability distributions and their associated statistical measures. Consequently, the whole system is described via a stochastic dynamic model, and the simultaneous estimation and control problem boils down to the minimization of the expected value of a pre-specified cost function. However, this task is rarely straightforward and the general conditions guaranteeing the existence of an optimal control scheme are yet unknown [2].

A major contribution in the field of stochastic adaptive control was made by A. A. Fel'dbaum in his seminal work on optimal control [4]–[6]. Fel'dbaum postulated that the control signal of an optimal adaptive system should have dual goals, namely: (i) to ensure that the controlled variable tracks the desired reference signal, with due consideration given to the estimates' uncertainty, and (ii) to perturb the plant sufficiently so as to accelerate estimation, thereby reducing quickly the uncertainty in future estimates. These two properties are commonly known as *caution* and *probing* respectively, or in Fel'dbaum's own terminology *directing* and *investigating*. Controllers exhibiting these features are named *dual adaptive*. In contrast to an HCE controller, a dual adaptive control law is dependent on the estimates' uncertainty, besides the estimates themselves, and aims to strike a balance between estimation and control at all times. Fel'dbaum also showed that the exact solution to the optimal adaptive dual control problem can be derived using *dynamic programming*, specifically by solving

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

the Bellman equation. However, in almost all practical situations, with the exception of a few very simple examples [7], the Bellman equation is impossible to solve, both analytically or numerically, due to the very large dimensions of the underlying space [2], [3], [8], [9].

The difficulty in finding the optimal dual adaptive control law in almost every practical case, led to the development of a number of simplified approaches, that though suboptimal, still exhibit the dual properties of caution and probing featured by the optimal dual solution. These suboptimal dual adaptive control schemes can be coherently divided into two groups, namely *implicit* and *explicit* methods. Implicit solutions try to introduce approximations to render the Bellman equation tractable [10], while explicit solutions reformulate the problem via modified cost functions that explicitly include a term related to parameter estimation, in order to induce a form of probing [8], [9], [11]. As pointed out on several occasions [8], [9], implicit solutions are typically more complex and more computationally intensive.

Dual adaptive control has been applied successfully in a number of practical applications [12]–[15]. However none of these applications involve mobile robots. Motion control of mobile robots has captured the interest of numerous researchers over the past three decades [1], [16]–[26]. This interest stems from a vast array of existing and potential practical applications [27]–[31], as well as from a number of particularly interesting theoretical challenges enriching this field of study. In particular, due to their mechanical configuration most wheeled mobile robots (WMRs) manifest restricted mobility, giving rise to nonholonomic constraints in their kinematics. Moreover, many of these WMRs are also underactuated since they exhibit less control inputs than degrees of freedom. Consequently, the linearized kinematic model of these robots lacks controllability, full-state feedback linearization is out of reach [18], and pure smooth time-invariant feedback stabilization of the Cartesian model is unattainable [32].

Most of the early contributions in the field of WMR motion control focus solely on the kinematic/steering control problem [16]–[18], [33]. In other words they base their designs on a robot model with velocity control inputs, rather than the more realistic model with torque control inputs. In doing so the controller is completely ignoring the vehicle dynamics due to mass, inertia and friction. This is known as the *perfect velocity tracking* assumption [19]. When it comes to the practical implementation of these kinematic controllers, this approach assumes that there is an independent low-level velocity control loop (usually implemented via a proportional-integral-derivative (PID) controller), that ascertains that the actual wheel velocities track precisely those requested by the kinematic control law [34]. However, while the use of independent PID velocity control loops is convenient and leads to acceptable performance in many applications involving slow-moving robots tracking low-acceleration trajectories, it can lead to high tracking errors, possibly resulting in total mission failure, in the face of more challenging tasks characterized by high reference velocities and accelerations [19]. In such situations, the robot nonlinear dynamics are no longer negligible and a better approach would be to replace the PID controller by a superior, though generally more complex, velocity controller whose design is based on a model relating the wheel velocities to the input torques. Such a controller would explicitly account for the vehicle's dynamic effects due to mass, friction and inertia. One such example is the well-established *computed-torque* approach [19], [34].

However, the dynamic model of a mobile robot is not only nonlinear but includes parameters or functions; such as mass, frictional terms and inertia; that are highly uncertain, time-varying or even unknown. Consequently, a number of adaptive control methods for the dynamic control of mobile robots have been proposed. These include both parametric adaptive control [20] and functional adaptive control [22], [25], [35]–[37]. The latter differs from the former in that the uncertainty is not restricted to parametric terms, but covers the dynamic functions themselves. We consider functional adaptive control to be more general and superior in handling higher degrees of uncertainty and unmodelled dynamics. Yet, all the mentioned adaptive robot controllers rely on the HCE assumption and so are prone to suffer from the aforementioned ill effects. In contrast, in our recent works [1], [26], we propose dual adaptive control techniques, rooted in computational intelligence, to address the problem of mobile robot control with uncertain/unknown dynamics.

In [26] we propose two novel dual control schemes employing two different kinds of artificial neural networks (ANNs), namely Gaussian radial basis functions (GaRBFs) and multilayer perceptrons (MLPs) [38], to estimate the WMR dynamic functions in real-time. The advantage of GaRBFs over MLPs lies in the fact that with GaRBFs the unknown ANN weights appear linearly in the stochastic state-space model formulated for estimation. This permits the use of the Kalman filter (KF) [39] for the recursive optimal ANN weight-tuning. However in the MLP case, this desirable property of linearity in the network parameters is not preserved, and the KF weight-tuning algorithm has to be replaced by a suboptimal nonlinear stochastic estimator, such as the extended Kalman filter (EKF) [40], which not only complicates the derivation of the control law, but introduces several approximations. On the other hand, unlike the activation functions employed in GaRBF ANNs, the sigmoidal functions in MLPs do not have localized receptive fields. This implies that typically MLP networks require less neurons than GaRBF ANNs to achieve the same degree of accuracy. Consequently, MLPs tend to be less computationally demanding, especially in the case of high-order systems, since the number of neurons need not rise exponentially with the number of states as in the case of GaRBF ANNs. The latter effect is known as the *curse of dimensionality* [41].

In the light of these arguments, the MLP dual adaptive scheme we proposed in [26] uses the EKF to estimate the nonlinearly-appearing ANN optimal parameters in real-time. The EKF approximates the state (in this case parameter) distribution by a Gaussian random variable (GRV) and propagates it analytically through the first-order linearization of the nonlinear stochastic model. Moreover, the dual adaptive control law proposed for that scheme, is based on another first-order Taylor approximation of the measurement equation in the stochastic model. This adds further to the suboptimality of the proposed approach.

To lessen the extent of these approximations, in this paper, which extends on our recent preliminary work [1], we propose a novel MLP dual adaptive control scheme that uses a specifically devised form of the unscented Kalman filter (UKF) [42], [43] as a recursive weight-tuning algorithm, instead of the EKF employed in [26]. In addition, we propose a new dual adaptive control law that employs the unscented transform (UT) [42] to improve on the first-order Taylor approximation used in deriving the EKF-based controller in [26].

It should be pointed out that the convergence and stability analysis of dual adaptive control schemes presents a very difficult challenge, mainly due to the stochastic and adaptive nature of the problem. The few works that address these issues consider only linear systems of a particular form and are characterized by a number of nontrivial assumptions [9], [44]. Consequently, in contrast to the case of deterministic approaches, to prove convergence and stability for a dual adaptive nonlinear controller, is still considered to be an open problem within the research community. Hence in practice, as argued in [9], the stability of dual adaptive controllers is commonly demonstrated by computer simulations and real-life experiments.

The contribution of this paper comprises a detailed treatment of the two dual adaptive MLP control schemes mentioned previously and a set of verifying and comparative results, comprising realistic Mont Carlo simulations backed by rigourous statistical analysis and real-life experiments. In particular, we show that the proposed UT-based dual adaptive controller brings about significant improvements in tracking performance over the EKF-based dual adaptive scheme recently proposed in [26], while still employing the same computationally-friendly MLP architecture. To the best of our knowledge this is the first work in which the UT is being used in the context of dual adaptive control. In addition, one should note that very few adaptive controllers have ever been implemented and tested on a physical WMR, amongst which one finds [45], [46]. However, none of these address fully the uncertainty in the WMR dynamic functions nor take a dual adaptive control approach.

The rest of the paper is organized as follows. Section II contains preliminary material, including the development of the discrete-time dynamic model of the WMR considered in this work, and a formulation of the WMR trajectory tracking control problem. Section III presents the design of both the EKF-based and the proposed UT-based dual adaptive MLP control schemes. Monte Carlo simulation results supported by statistical hypothesis comparative tests and real-life experiments are then presented in Section IV, which is followed by a brief conclusion in Section V.

II. Preliminaries

In this work we address the trajectory tracking problem of the differentially driven WMR depicted in Figure 1. However, the framework we adopt in our design is completely modular. Consequently, the dual adaptive dynamic control scheme proposed in this paper can be easily adopted to address different navigation problems such as posture stabilization and path following [34], possibly even for different types of robotic configurations. In this section we outline the development of the dynamic model of the differentially driven WMR and formulate the trajectory tracking problem considered in this work.

A. Modelling

With reference to the WMR configuration in Figure 1, we ignore the passive caster wheels and adopt the following notation throughout the article:

$\mathrm{P_o}$:	axle midpoint between the two wheels
$\mathrm{P_c}$:	centre of mass of the platform without wheels
d:	distance between $\mathrm{P_o}$ to $\mathrm{P_c}$
b:	distance from the centre of each wheel to $\mathrm{P_o}$
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 $\mathrm{P_c}$
I_w:	moment of inertia of each wheel about the axle
I_m:	moment of inertia of each wheel about its diameter

The robot coordinate vector is denoted by $\boldsymbol{q} = [x \;\; y \;\; \phi \;\; \theta_r \;\; \theta_l]^T$, where (x, y) is the Cartesian coordinate of $\mathrm{P_o}$, ϕ is the robot's orientation with reference to the x-axis, and θ_r, θ_l are the angular displacements about the axle of the right and left motorized wheels respectively. The *pose* of the robot refers to the vector $\boldsymbol{p} = [x \;\; y \;\; \phi]$.

1) Kinematic Model: The differential configuration of this WMR is subject to three kinematic constraints, stemming from

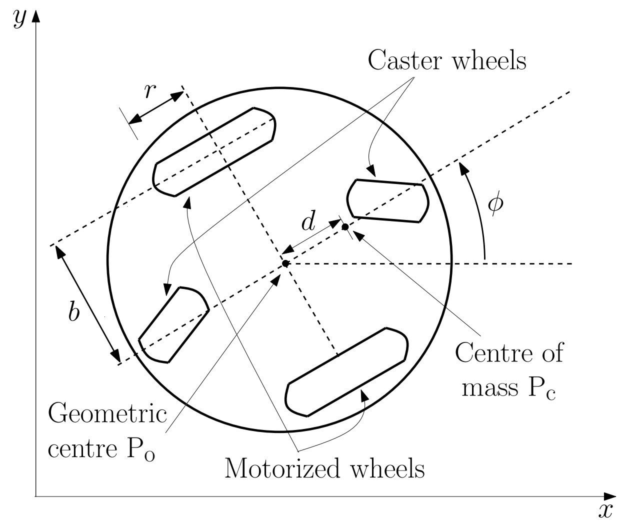
y
Caster wheels
r
ϕ
d
b
Centre of mass $\mathrm{P_c}$
Geometric centre $\mathrm{P_o}$
Motorized wheels
x

Fig. 1. Differentially driven wheeled mobile robot.

the fact that the translational velocity of the geometric centre $\mathrm{P_o}$ is always in the direction perpendicular to the driving axle, and the two driving wheels roll without slipping. The former leads to a holonomic constraint while the latter leads to two nonholonomic constrains [47]. Mathematically this is described by $\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \mathbf{0}$, where

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

These three kinematic constraints, along with a few other relationships arising from the geometry of the WMR depicted in Figure 1, can be used to show that the kinematic model of this differentially driven WMR is given by

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

where

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

and $\boldsymbol{\nu}$ denotes a vector composed of the angular velocities of the two motorized wheels, that is, $\boldsymbol{\nu} = [\nu_r \;\; \nu_l]^T = \left[\dot{\theta}_r \;\; \dot{\theta}_l\right]^T$. It is important to note that:

Remark II.1. *The two independent columns of $\boldsymbol{S}(\boldsymbol{q})$ are in the null space of $\boldsymbol{A}(\boldsymbol{q})$, that is, $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) = \mathbf{0}$.*

2) Dynamic Model: The equations of motion of this WMR can be derived using Lagrangian mechanics. The Euler-Lagrange equation for the nonholonomic WMR considered in this paper is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} = Q_i - \sum_{c=1}^{3} a_{ci}\lambda_c, \quad (i = 1, 2, \ldots, 5), \quad (2)$$

where $K(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the total kinetic energy of the WMR, q_i is the i^{th} element of the coordinate vector $\boldsymbol{q}$, Q_i is the i^{th} Lagrange force, a_{ci} is the $(c, i)^{\text{th}}$ element of the constraints matrix $\boldsymbol{A}(\boldsymbol{q})$ and λ_c is the c^{th} element of the vector of Lagrange multipliers $\boldsymbol{\lambda}$. It can be shown that the total kinetic energy of the WMR is given by

$$\begin{aligned} K(\boldsymbol{q}, \dot{\boldsymbol{q}}) \;\; &= \;\; \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2\right) + m_c d\dot{\phi}\left(\dot{y}\cos\phi - \dot{x}\sin\phi\right) \\ &\quad + \frac{I}{2}\dot{\phi}^2 + \frac{I_w}{2}\left(\dot{\theta}_r{}^2 + \dot{\theta}_l{}^2\right), \qquad (3) \end{aligned}$$

where $m = m_c + 2m_w, \;\; I = (I_c + m_c d^2) + 2(I_m + m_w b^2)$. Equation (3) can then be used to work out the derivative terms in (2). This leads to the equations of motion of the WMR, given by:

$$\boldsymbol{M}(q)\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{E}\boldsymbol{\tau} - \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda}, \qquad (4)$$

where:

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} m & 0 & -m_c d\sin\phi & 0 & 0 \\ 0 & m & m_c d\cos\phi & 0 & 0 \\ -m_c d\sin\phi & m_c d\cos\phi & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix},$$

$$\boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -m_c d\dot{\phi}^2\cos\phi \\ -m_c d\dot{\phi}^2\sin\phi \\ 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and $\boldsymbol{\tau} = [\tau_r \;\; \tau_l]^T$ is the torque vector, with τ_r and τ_l denoting the torques applied to the right and left wheels respectively.

The kinematic model in (1) and the equations of motion in (4) can be used to determine the WMR dynamics relating the wheels acceleration $\dot{\boldsymbol{\nu}}$ to the wheels torque $\boldsymbol{\tau}$ as follows [47]. We differentiate (1) with respect to time, substitute the expression for $\ddot{\boldsymbol{q}}$ in (4), and finally pre-multiply the resulting expression by $\boldsymbol{S}^T$. Noting that: $\boldsymbol{S}^T\boldsymbol{A}^T = \mathbf{0}$ (by Remark II.1), $\boldsymbol{S}^T\boldsymbol{E} = \mathbf{I_2}$ (where throughout the paper $\mathbf{I}_{\boldsymbol{i}}$ denotes an $(i \times i)$ identity matrix), and $\dot{\phi} = \frac{r}{2b}(\nu_r - \nu_l)$ (by (1)); the resulting dynamic model can be expressed by

$$\bar{\boldsymbol{M}}\dot{\boldsymbol{\nu}} + \bar{\boldsymbol{V}}(\boldsymbol{\nu}) + \bar{\boldsymbol{F}}(\boldsymbol{\nu}) = \boldsymbol{\tau}, \qquad (5)$$

where:

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

$$\bar{\boldsymbol{V}}(\boldsymbol{\nu}) = \boldsymbol{S^T M \dot{S} \nu} + \boldsymbol{S^T V} = \frac{m_c d r^3}{4b^2}\begin{bmatrix} \nu_r\nu_l - \nu_l^2 \\ \nu_r\nu_l - \nu_r^2 \end{bmatrix},$$

and $\bar{\boldsymbol{F}}(\boldsymbol{\nu})$ is introduced to account for any wheel velocity-dependent frictional terms.

Remark II.2. *$\bar{\boldsymbol{M}}$ is symmetric, positive definite, and is independent of the coordinate vector and/or its derivatives.*

Remark II.3. *In general, $\bar{\boldsymbol{V}}(\boldsymbol{\nu})$ and $\bar{\boldsymbol{F}}(\boldsymbol{\nu})$ are the two terms that render the WMR dynamics nonlinear.*

To account for the fact that the controller is to be implemented on a digital computer, the continuous-time dynamics (5) are discretized through a first-order forward Euler approximation with a sampling interval of T seconds. The resulting nonlinear discrete-time dynamic model is given by

$$\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1} = \boldsymbol{f}_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{\tau}_{k-1}, \qquad (6)$$

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

$$\begin{aligned} \boldsymbol{f}_{k-1} &= -T\bar{\boldsymbol{M}}_{k-1}^{-1}\left(\bar{\boldsymbol{V}}_{k-1} + \bar{\boldsymbol{F}}_{k-1}\right), \\ \boldsymbol{G}_{k-1} &= T\bar{\boldsymbol{M}}_{k-1}^{-1}. \qquad (7) \end{aligned}$$

The following conditions are assumed to hold:

Assumption II.1. *The control input vector $\boldsymbol{\tau}$ remains constant over a sampling interval of T seconds (zero-order hold).*

Assumption II.2. *The sampling interval T is chosen low enough for the Euler approximation error to be negligible.*

B. Trajectory Tracking

The trajectory tracking task of WMRs is commonly defined via the concept of the *virtual vehicle* [17]. In this formulation, the time-dependent reference trajectory is designated by a nonstationary virtual vehicle, *kinematically identical* to the real vehicle. The control task is for the real vehicle to track the virtual vehicle at all times, *in both pose and velocity*. It is important to note that this problem is different and generally more challenging than path-following. This stems from the fact that in trajectory tracking the reference path is time-indexed (hence dictating speed as well as position), while in path-following the reference contains no temporal information and the vehicle speed is typically fixed and predetermined [34].

The trajectory tracking error in discrete-time is commonly defined by a tracking error vector $\boldsymbol{e}_k = [e_{1k}\ e_{2k}\ e_{3k}]^T$, expressed pictorially in Figure 2, and mathematically defined by

$$\boldsymbol{e}_k = \begin{bmatrix} \cos\phi_k & \sin\phi_k & 0 \\ -\sin\phi_k & \cos\phi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\boldsymbol{p_r}_k - \boldsymbol{p}_k\right), \qquad (8)$$

where $\boldsymbol{p_r}_k = [x_{rk}\ \ y_{rk}\ \ \phi_{rk}]^T$ denotes the virtual vehicle pose vector. Hence, in trajectory tracking the objective is to make $\boldsymbol{e}_k$ converge to zero, so that $\boldsymbol{p}_k$ converges to $\boldsymbol{p_r}_k$.

III. Control Design

As argued in Section I, the motion control of WMRs is commonly addressed as two separate tasks, namely kinematic and dynamic control [19], [34], [37]. Kinematic control is concerned solely with the steering system (1). Specifically its aim is to devise a control law for the robot wheel velocities, so as to stabilize the pose of the robot as required by the navigation task at hand; be it trajectory tracking, path-following or posture stabilization. In the case of trajectory tracking, the aim of the kinematic controller is to compute the wheel velocities required to minimize the robot tracking error $\boldsymbol{e}_k$. On the other hand, the aim of the dynamic controller is to compute the wheel torques required in order to ensure that the robot accurately tracks the velocities computed by the kinematic controller. Hence, the two control loops operate in cascade; with the kinematic controller's output (a velocity command) serving as the reference input of the cascaded dynamic controller, which computes the torque required to drive the WMR wheels at the specified velocities. This approach renders the overall control architecture modular, since the kinematic controller, which is specific to the navigation problem at hand, can be easily replaced while still retaining the same dynamic controller. In our work we adopt this modular architecture, depicted in Figure 3, and design the dynamic controller to be dual adaptive as detailed in the rest of this section.

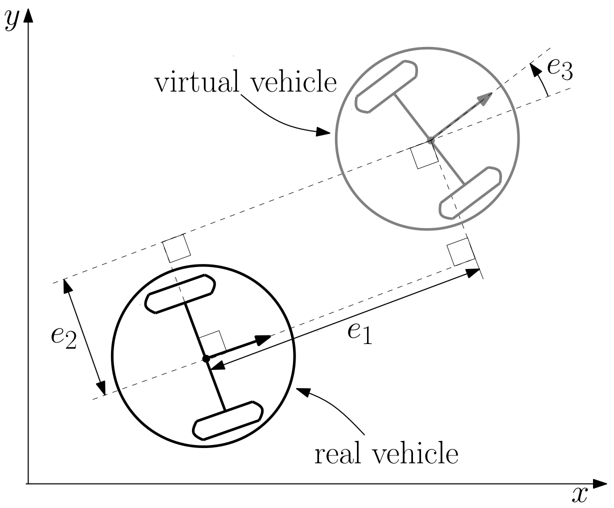

Fig. 2. Trajectory tracking via the concept of the virtual vehicle.

A. The Kinematic Controller

As argued earlier, the role of the kinematic controller in trajectory tracking is to make $\boldsymbol{e}_k$ converge to zero, so that $\boldsymbol{p}_k$ converges to $\boldsymbol{p_r}_k$. To address this well-researched problem we opt to adopt an established kinematic controller, originally presented in [17], and convert it to discreet-time so as to integrate it in our formulation. The resulting kinematic control law is given by

$$\boldsymbol{\nu_c}_k = \boldsymbol{C} \begin{bmatrix} v_{rk}\cos e_{3k} + k_1 e_{1k} \\ \omega_{rk} + k_2 v_{rk} e_{2k} + k_3 v_{rk}\sin e_{3k} \end{bmatrix}, \qquad (9)$$

where $\boldsymbol{\nu_c}_k$ is the wheel velocity command vector issued by the kinematic controller, k_1, k_2, and k_3 are positive design parameters, $v_{rk} > 0$ and ω_{rk} are the translational and angular *virtual vehicle* velocities respectively (assumed to be *continuous* functions, at least know one sampling interval ahead), and $\boldsymbol{C}$ is a velocity conversion matrix given by

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

Stability analysis and the corresponding necessary conditions of this controller in continuous-time are detailed in [17].

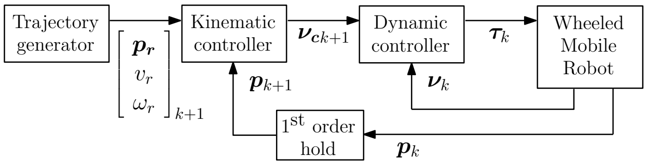

Fig. 3. Dynamic control architecture.

B. Nonadaptive Dynamic Control

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

$$\boldsymbol{\tau}_k = \boldsymbol{G}_k^{-1}\left(\boldsymbol{\nu_c}_{k+1} - \boldsymbol{\nu}_k - \boldsymbol{f}_k + k_d\left(\boldsymbol{\nu_c}_k - \boldsymbol{\nu}_k\right)\right), \qquad (10)$$

with the design parameter $-1 < k_d < 1$, yields the following closed-loop stable linear dynamics

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu_c}_{k+1} + k_d\left(\boldsymbol{\nu_c}_k - \boldsymbol{\nu}_k\right),$$

when substituted in the dynamic model in (6). This ensures that $|\boldsymbol{\nu_c}_k - \boldsymbol{\nu}_k| \to 0$ as $k \to \infty$. It is important to note that:

Remark III.1. *Control law (10) requires the velocity command vector $\boldsymbol{\nu_c}_{k+1}$ to be available at instant k. For this reason, the kinematic control law (9) is advanced by one sampling interval. This means that at instant k, the values of $v_{r\,k+1}$, $\omega_{r\,k+1}$ and $\boldsymbol{e}_{k+1}$ need to be known. Additionally, from (8) it is clear that $\boldsymbol{p_r}_{k+1}$ and $\boldsymbol{p}_{k+1}$ are needed to determine $\boldsymbol{e}_{k+1}$. Having the values of reference signal $\boldsymbol{p_r}_{k+1}$, $v_{r\,k+1}$ and $\omega_{r\,k+1}$ available at instant k is easy, since it simply means that the path-planning algorithm is required to generate the reference trajectory one sampling interval ahead. On the other hand, for the non-reference signal $\boldsymbol{p}_{k+1}$, we propose to estimate its value via the first-order approximation $\boldsymbol{p}_{k+1} \approx 2\boldsymbol{p}_k - \boldsymbol{p}_{k-1}$. This is justified in the light of Assumption II.2.*

Remark III.2. *The case with $k_d = 0$ in (10), corresponds to* deadbeat control *associated with digital control systems [48].*

C. Dual Adaptive Dynamic Control using MLPs

The computed-torque dynamic control law (10) driven by the kinematic law (9), is a solution to the trajectory tracking problem *only if* the WMR dynamic functions $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ in (6) are perfectly known. As emphasized in Section I, this is rarely the case in real-life robotic applications commonly exhibiting: unmodelled dynamics, unknown/time-varying parameters, and imperfect/noisy sensor measurements. Most works, address these issues via some form of HCE adaptive control. In contrast, the two schemes presented in this paper not only consider $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}_{k-1}$ to be completely unknown to the controller, but also feature dual adaptive properties to handle the issue of uncertainty as explained in Section I. The two dual adaptive schemes, detailed in this section, both employ a stochastically-trained ANN-based algorithm to approximate these functions recursively in real-time.

Specifically, a sigmoidal MLP ANN with one hidden layer is used to approximate the nonlinear vector-valued function $\boldsymbol{f}_{k-1}$, as depicted in Figure 4. Its output is given by

$$\tilde{\boldsymbol{f}}_{k-1} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_{k-1}, \hat{\boldsymbol{a}}_k)\hat{\boldsymbol{w}}_{\boldsymbol{1}k} \\ \boldsymbol{\phi}^T(\boldsymbol{x}_{k-1}, \hat{\boldsymbol{a}}_k)\hat{\boldsymbol{w}}_{\boldsymbol{2}k} \end{bmatrix}, \qquad (11)$$

in the light of the following statements:

Definition III.1. $\boldsymbol{x}_{k-1} = [\boldsymbol{\nu}_{k-1} \quad 1]$ *denotes the ANN input. The augmented constant serves as a bias input. This selection of the ANN input stems from the fact that $\boldsymbol{f}_{k-1}$ is effectively a function of $\boldsymbol{\nu}_{k-1}$.*

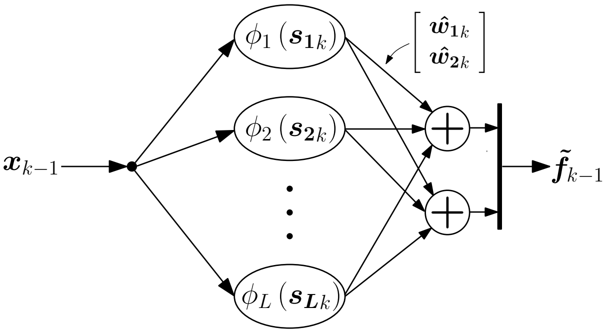
$\phi_1(\boldsymbol{s_1}_k)$
$\begin{bmatrix}\hat{\boldsymbol{w}}_{\boldsymbol{1}k} \\ \hat{\boldsymbol{w}}_{\boldsymbol{2}k}\end{bmatrix}$
$\phi_2(\boldsymbol{s_2}_k)$
$\boldsymbol{x}_{k-1}$
$\tilde{\boldsymbol{f}}_{k-1}$
$\phi_L(\boldsymbol{s_L}_k)$

Fig. 4. Sigmoidal Multilayer Perceptron neural network.

Definition III.2. *$\boldsymbol{\phi}(\cdot,\cdot)$ is the vector of sigmoidal activation functions, whose i^{th} 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 i^{th} vector element in the group vector $\hat{\boldsymbol{a}}$;* i.e. *$\hat{\boldsymbol{a}} = \left[\hat{\boldsymbol{s}}_{\boldsymbol{1}}^T \quad \cdots \quad \hat{\boldsymbol{s}}_{\boldsymbol{L}}^T\right]^T$ where L denotes the number of neurons. The time index has been dropped for clarity, and throughout the paper the ˆ notation indicates that the operand is undergoing estimation.*

Definition III.3. *$\hat{\boldsymbol{w}}_{\boldsymbol{i}k}$ represents the synaptic weight estimate vector of the connection between the neuron hidden layer and the i^{th} output element of the ANN.*

Assumption III.1. *The input vector $\boldsymbol{x}_{k-1}$ is contained within a known, arbitrarily large compact set $\boldsymbol{\chi} \subset \mathbb{R}^2$. This is justified since the wheel velocities are inherently bounded.*

Moreover, it is known that $\boldsymbol{G}_{k-1}$ is a state-independent matrix with unknown elements (refer to (7)). Hence, its estimation does not require the use of an ANN. In addition it is a symmetric matrix, a property which is exploited to construct its estimate as follows

$$\tilde{\boldsymbol{G}}_{k-1} = \begin{bmatrix} \hat{g}_{1\,k-1} & \hat{g}_{2\,k-1} \\ \hat{g}_{2\,k-1} & \hat{g}_{1\,k-1} \end{bmatrix}, \qquad (12)$$

where $\hat{g}_{1\,k-1}$ and $\hat{g}_{2\,k-1}$ represent the estimates of the unknown elements in $\boldsymbol{G}_{k-1}$.

We formulate the ANN weight-tuning task as a stochastic nonlinear estimation problem. The following preliminaries are necessary in order to proceed.

Definition III.4. *The unknown parameters requiring estimation are grouped in a single vector $\hat{\boldsymbol{z}}_k = \left[\hat{\boldsymbol{r}}_k^T \quad \hat{\boldsymbol{g}}_k^T\right]^T$, where $\hat{\boldsymbol{r}}_k = \left[\hat{\boldsymbol{w}}_{\boldsymbol{1}k}^T \quad \hat{\boldsymbol{w}}_{\boldsymbol{2}k}^T \quad \hat{\boldsymbol{a}}_k^T\right]^T$ and $\hat{\boldsymbol{g}}_k = \left[\hat{g}_{1\,k-1} \quad \hat{g}_{2\,k-1}\right]^T$.*

Definition III.5. *The measured output in the dynamic model (6) is denoted by $\boldsymbol{y}_k = \boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1}$. In our practical implementation $\boldsymbol{\nu}_k$ is acquired from the wheel encoders.*

Assumption III.2. *By the* Universal Approximation Theorem *of ANN, inside the compact set $\boldsymbol{\chi}$, the ANN approximation error is negligibly small when the estimate $\hat{\boldsymbol{r}}_k$ is equal to some unknown optimal vector $\boldsymbol{r}_k^*$. The * notation denotes optimality.*

In view of the stochastic adaptive approach taken in this work, the unknown optimal parameter vector $\boldsymbol{z}_k^*$ is treated as a random variable, with the initial condition $p(\boldsymbol{z}_0^*) \sim \mathcal{N}(\hat{\boldsymbol{z}}_0, \boldsymbol{P}_0)$, meaning that $\boldsymbol{z}_0^*$ is normally distributed with mean $\hat{\boldsymbol{z}}_0$ and covariance $\boldsymbol{P}_0$. This notation is adopted throughout the article. Effectively, the covariance value $\boldsymbol{P}_0$ reflects the confidence in the initial guess $\hat{\boldsymbol{z}}_0$.

By (11), (12), all previous definitions and assumptions, it follows that the model in (6) can be represented in the following stochastic state-space form

$$\begin{aligned} \boldsymbol{z}_{k+1}^* &= \boldsymbol{z}_k^* + \boldsymbol{\rho}_k \\ \boldsymbol{y}_k &= \boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right) + \boldsymbol{\epsilon}_k, \end{aligned} \tag{13}$$

where the vector-valued function $\boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \boldsymbol{z}_k^*\right)$ is non-linear in $\boldsymbol{z}_k^*$, and is given by

$$\boldsymbol{h}\left(\cdot\right) = \tilde{\boldsymbol{f}}(\boldsymbol{x}_{k-1}, \boldsymbol{r}_k^*) + \tilde{\boldsymbol{G}}(\boldsymbol{g}_k^*)\boldsymbol{\tau}_{k-1}. \tag{14}$$

In this model, the unknown optimal parameter vector $\boldsymbol{z}_k^*$ is characterized as a stationary process corrupted by an artificial process noise $\boldsymbol{\rho}_k$, which aids convergence and tracking during estimation. In addition, observation uncertainty is catered for by augmenting a random measurement noise $\boldsymbol{\epsilon}_k$ to $\boldsymbol{y}_k$.

It is evident, from (14), that the use of the MLP ANN, which brings about certain practical advantages over GaRBF as argued in Section I, results in a *nonlinear* measurement equation in the stochastic state-space model (13) formulated for estimation. In order to address this issue in a stochastic framework, we have to employ a nonlinear recursive estimator.

The two dual adaptive schemes presented in this paper depart from this point in our formulation and proceed to tackle the estimation and control problems in different ways, as detailed next.

1) EKF-based Dual Adaptive Scheme: For the sake of clarity and completeness, the MLP dual adaptive scheme proposed in [26] and used for comparisons in this paper is revisited in this section. In this scheme, we employ the EKF in prediction mode for the recursive real-time estimation of $\boldsymbol{z}_{k+1}^*$ as follows.

Definition III.6. *The* information state *denoted by I^k, consists of all measurements up to instant k and all previous inputs.*

Assumption III.3. *$\boldsymbol{\epsilon}_k$ and $\boldsymbol{\rho}_k$ are both zero-mean white Gaussian processes with covariances $\boldsymbol{R}_{\boldsymbol{\epsilon}}$ and $\boldsymbol{Q}_{\boldsymbol{\rho}}$ respectively. Moreover $\boldsymbol{\epsilon}_k$, $\boldsymbol{\rho}_k$ and $\boldsymbol{z}_0^*$ are mutually independent $\forall k$.*

Lemma III.1. *In the light of (13), Definition III.6, and Assumption III.3, it follows that $p(\boldsymbol{z}_{k+1}^*|I^k) \approx \mathcal{N}(\hat{\boldsymbol{z}}_{k+1}, \boldsymbol{P}_{k+1})$, where $\hat{\boldsymbol{z}}_{k+1}$ and $\boldsymbol{P}_{k+1}$ are computed at each control step according to the EKF Algorithm III.1. Consequently, $\hat{\boldsymbol{z}}_{k+1}$ is considered to be the estimate of $\boldsymbol{z}_{k+1}^*$ conditioned on I^k, and $\boldsymbol{P}_{k+1}$ can be viewed as a measure of this estimate's uncertainty.*

Proof: The proof of this lemma follows directly that of the EKF in prediction mode, when applied to the nonlinear stochastic state-space model in (13). ■

Given the previous prediction $\left(\hat{\boldsymbol{z}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\right)$; denoted in short-form by $(\hat{\boldsymbol{z}}_k, \boldsymbol{P}_k)$; the following EKF (prediction mode) algorithm generates the new prediction $(\hat{\boldsymbol{z}}_{k+1}, \boldsymbol{P}_{k+1})$.

1) Evaluating $\boldsymbol{\nabla_h}_k$, 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 $\hat{\boldsymbol{z}}_k$:

$$\boldsymbol{\nabla_h}_k \triangleq \left[\boldsymbol{\nabla_f}_k \;\; \boldsymbol{\nabla_\Gamma}_k\right] = \left[\frac{\partial(\tilde{\boldsymbol{f}}_{k-1})}{\partial(\boldsymbol{r}_k^*)}\bigg|_{\hat{\boldsymbol{r}}_k} \quad \frac{\partial(\tilde{\boldsymbol{G}}_{k-1}\boldsymbol{\tau}_{k-1})}{\partial(\boldsymbol{g}_k^*)}\bigg|_{\hat{\boldsymbol{g}}_k}\right],$$

where by (11), (12) and (14), it can be shown that:

$$\begin{aligned} \boldsymbol{\nabla_f}_k &= \begin{bmatrix} \boldsymbol{\phi}_{k-1}^T & \mathbf{0}^T & \cdots \hat{w_{1,i}}(\phi_i - {\phi_i}^2)\boldsymbol{x}^T \cdots \\ \mathbf{0}^T & \boldsymbol{\phi}_{k-1}^T & \cdots \hat{w_{2,i}}(\phi_i - {\phi_i}^2)\boldsymbol{x}^T \cdots \end{bmatrix}, \\ \boldsymbol{\nabla_\Gamma}_k &= \begin{bmatrix} \tau_{r\,k-1} & \tau_{l\,k-1} \\ \tau_{l\,k-1} & \tau_{r\,k-1} \end{bmatrix}, \end{aligned} \tag{15}$$

where: $i = 1, \;\ldots\;, L$, $\hat{w_{j,i}}$ denotes the i^{th} element of the j^{th} 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 $\hat{\boldsymbol{a}}_k$, $\mathbf{0}$ denotes a zero-vector of the same length as $\boldsymbol{\phi}_{k-1}$, and ϕ_i and $\boldsymbol{x}$ both correspond to time instant $(k-1)$.

2) Performing the prediction step:

$$\begin{aligned} \hat{\boldsymbol{z}}_{k+1} &= \hat{\boldsymbol{z}}_k + \boldsymbol{K}_k\boldsymbol{i}_k \\ \boldsymbol{P}_{k+1} &= \boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{\nabla_h}_k\boldsymbol{P}_k + \boldsymbol{Q}_{\boldsymbol{\rho}} \end{aligned} \tag{16}$$

where the Kalman gain and the innovation vector are respectively given by:

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_k\boldsymbol{\nabla_h}_k^T\left(\boldsymbol{\nabla_h}_k\boldsymbol{P}_k\boldsymbol{\nabla_h}_k^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}\right)^{-1} \\ \boldsymbol{i}_k &= \boldsymbol{y}_k - \boldsymbol{h}\left(\boldsymbol{x}_{k-1}, \boldsymbol{\tau}_{k-1}, \hat{\boldsymbol{z}}_k\right). \end{aligned}$$

Algorithm III.1: The EKF parameter-prediction algorithm.

Lemma III.2. *On the basis of Lemma III.1, 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 $\boldsymbol{\nabla_h}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{\nabla_h}_{k+1}^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}$.*

Proof: 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

$$\boldsymbol{y}_{k+1} \approx \boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right) + \boldsymbol{\nabla_h}_{k+1}\left(\boldsymbol{z}_{k+1}^* - \hat{\boldsymbol{z}}_{k+1}\right) + \boldsymbol{\epsilon}_{k+1}.$$

Noting that $\boldsymbol{z}_{k+1}^*$ and $\boldsymbol{\epsilon}_{k+1}$ are the only probabilistic terms on the right-hand side of this approximation, the expected value of $\boldsymbol{y}_{k+1}$ conditioned on I^k, denoted by $E\left\{\boldsymbol{y}_{k+1}|I^k\right\}$, can be expressed as a sum of three terms:

$$\boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right) + \boldsymbol{\nabla_h}_{k+1}\left(E\left\{\boldsymbol{z}_{k+1}^*\right\} - \hat{\boldsymbol{z}}_{k+1}\right) + E\left\{\boldsymbol{\epsilon}_{k+1}\right\}.$$

Since $E\left\{\boldsymbol{z}_{k+1}^*\right\} = \hat{\boldsymbol{z}}_{k+1}$, by Lemma III.1, and $E\left\{\boldsymbol{\epsilon}_{k+1}\right\} = 0$, by Assumption III.3, the second and third term are both equal to zero, leaving the first term as the mean value of $p(\boldsymbol{y}_{k+1}|I^k)$. Using the same Taylor series approximation, we note that the covariance of the right-hand side can be written as

Cov $\left(\boldsymbol{\nabla}_{\boldsymbol{h}k+1}\boldsymbol{z}^{*}_{k+1}\right) + \text{Cov}\left(\boldsymbol{\epsilon}_{k+1}\right)$ which by Lemma III.1 and Assumption III.3 reduces to $\boldsymbol{\nabla}_{\boldsymbol{h}k+1}\boldsymbol{P}_{k+1}\boldsymbol{\nabla}_{\boldsymbol{h}k+1}^{T} + \boldsymbol{R}_{\boldsymbol{\epsilon}}$. ■

Algorithm III.1, in view of Lemma III.1, constitutes the adaptation law for the EKF-based dual adaptive scheme. Moreover, by Lemma III.2, this algorithm provides a real-time update of the probability density function $p(\boldsymbol{y}_{k+1}|I^{k})$, which is used to develop the dual adaptive control law as follows.

The explicit-type suboptimal innovation-based performance index J_{inn}, adopted from [8], and modified to fit our multiple-input multiple-output (MIMO) nonlinear problem, is given by

$$\begin{aligned} J_{inn} \;&=\; E\Big\{\left(\boldsymbol{y}_{k+1} - \boldsymbol{y_d}_{k+1}\right)^{T}\boldsymbol{Q_1}\left(\boldsymbol{y}_{k+1} - \boldsymbol{y_d}_{k+1}\right) \\ &\quad + \left(\boldsymbol{\tau}_k^{T}\boldsymbol{Q_2}\boldsymbol{\tau}_k\right) + \left(\boldsymbol{i}_{k+1}^{T}\boldsymbol{Q_3}\boldsymbol{i}_{k+1}\right)\Big|I^{k}\Big\}, \end{aligned} \tag{17}$$

in view of the following definitions:

Definition III.7. *$\boldsymbol{y_d}_{k+1}$ is the reference vector of $\boldsymbol{y}_{k+1}$ and is given by $\boldsymbol{y_d}_{k+1} = \boldsymbol{\nu_c}_{k+1} - \boldsymbol{\nu}_k$.*

Definition III.8. *Design parameters $\boldsymbol{Q_1}$, $\boldsymbol{Q_2}$ and $\boldsymbol{Q_3}$ are diagonal and $\in \mathbb{R}^{2\times2}$. Additionally: $\boldsymbol{Q_1}$ is positive definite, $\boldsymbol{Q_2}$ is positive semi-definite, and $-\boldsymbol{Q_1} \leq \boldsymbol{Q_3} \leq \boldsymbol{0}$ (element-wise).*

Remark III.3. *The design parameter $\boldsymbol{Q_1}$ is introduced to penalize tracking errors, $\boldsymbol{Q_2}$ induces a penalty on large control inputs, and $\boldsymbol{Q_3}$ affects the innovation vector so as to induce the dual adaptive feature characterizing this stochastic control law.*

The EKF-based dual adaptive control law is given by:

Theorem III.1. *The control law minimizing performance index J_{inn} in (17), subject to the WMR dynamic model (5) and all the previous definitions, assumptions and lemmas in this formulation, is given by*

$$\begin{aligned} \boldsymbol{\tau}_k \;&=\; \left(\tilde{\boldsymbol{G}}_k^{T}\boldsymbol{Q_1}\tilde{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N}_{k+1}\right)^{-1} \\ &\quad \times \left(\tilde{\boldsymbol{G}}_k^{T}\boldsymbol{Q_1}\left(\boldsymbol{y_d}_{k+1} - \tilde{\boldsymbol{f}}_k\right) - \boldsymbol{\kappa}_{k+1}\right), \end{aligned} \tag{18}$$

where $\tilde{\boldsymbol{f}}_k$ and $\tilde{\boldsymbol{G}}_k$ are computed via (11) and (12) using the latest estimate vector $\hat{\boldsymbol{z}}_{k+1}$ given by Algorithm III.1, and $\boldsymbol{\kappa}_{k+1}$ and $\boldsymbol{N}_{k+1}$ are computed as follows.

Definition III.9. *Let: $\boldsymbol{Q_4} \triangleq \boldsymbol{Q_1}+\boldsymbol{Q_3}$, $\boldsymbol{B} \triangleq \boldsymbol{P_{Gf}}_{k+1}\boldsymbol{\nabla}_{\boldsymbol{f}k}^{T}\boldsymbol{Q_4}$, $a_S(i,j)$ be used to denote the $(i,j)^{th}$ element of any matrix $\boldsymbol{A_S}$ and the covariance matrix $\boldsymbol{P}_{k+1}$ in (16) be repartitioned as*

$$\boldsymbol{P}_{k+1} = \begin{bmatrix} \boldsymbol{P_{ff}}_{k+1} & \boldsymbol{P_{Gf}}_{k+1}^{T} \\ \boldsymbol{P_{Gf}}_{k+1} & \boldsymbol{P_{GG}}_{k+1} \end{bmatrix}, \tag{19}$$

where $\boldsymbol{P_{ff}}_{k+1} \in \mathbb{R}^{5L\times5L}$ and $\boldsymbol{P_{GG}}_{k+1} \in \mathbb{R}^{2\times2}$. Then

$$\boldsymbol{\kappa}_{k+1} = \begin{bmatrix} b(1,1) + b(2,2) \\ b(1,2) + b(2,1) \end{bmatrix},$$

and the elements of $\boldsymbol{N}_{k+1}$ are given by:

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

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

Proof: By the approximate Gaussian distribution $p(\boldsymbol{y}_{k+1}|I^{k})$ in Lemma III.2, and standard results from linear algebra involving matrices [49], it follows that within this scheme, (17) can be written as

$$\begin{aligned} J_{inn} = \left(\boldsymbol{h}_{k+1} - \boldsymbol{y_d}_{k+1}\right)^{T}\boldsymbol{Q_1}\left(\boldsymbol{h}_{k+1} - \boldsymbol{y_d}_{k+1}\right) + \boldsymbol{\tau}_k^{T}\boldsymbol{Q_2}\boldsymbol{\tau}_k \\ +\;\; \text{tr}\left(\boldsymbol{Q_4}\left(\boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}}\boldsymbol{P}_{k+1}\boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}}^{T} + \boldsymbol{R}_{\epsilon}\right)\right), \end{aligned} \tag{20}$$

where $\boldsymbol{h}_{k+1}$ denotes $\boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{\tau}_k, \hat{\boldsymbol{z}}_{k+1}\right)$. By employing the relations in (14), (15) and (19) to expand $\boldsymbol{h}_{k+1}$, $\boldsymbol{\nabla}_{\boldsymbol{h}_{k+1}}$ and $\boldsymbol{P}_{k+1}$ respectively in (20), one is able to factorize J_{inn} completely in terms of $\boldsymbol{\tau}_k$. The resulting quadratic expression is differentiated with respect to $\boldsymbol{\tau}_k$ and then equated to zero in order to determine its stationary point. This leads to (18). Moreover, the resulting Hessian matrix is given by $2\left(\tilde{\boldsymbol{G}}_k^{T}\boldsymbol{Q_1}\tilde{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N}_{k+1}\right)$, which by the statements in Definitions III.8 and III.9 can be shown to be positive definite. This means that the dual adaptive control law specified in Theorem III.1, minimizes the selected cost function J_{inn} uniquely, and the inverse term in (18) exists without exceptions. ■

Remark III.4. *$\boldsymbol{Q_3}$ which appears in (18) via $\boldsymbol{\kappa}_{k+1}$ 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 uncertainty, which leads to cautious control. For intermediate settings of $\boldsymbol{Q_3}$, the controller strikes a compromise and operates in 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 notorious for sluggish response and* control turn-off *[8], [50]. Consequently, dual control exhibits superior performance by striking a balance between the two extremes.*

Remark III.5. *It is interesting to note that in the HCE case,* i.e. *when $\boldsymbol{Q_3} = -\boldsymbol{Q_1}$, if one sets $\boldsymbol{Q_1} = \mathbf{I_2}$ and $\boldsymbol{Q_2} = \boldsymbol{0}$, the control law in (18) is identical to the computed-torque law in (10), with $k_d = 0$ and the dynamic functions $\boldsymbol{f}_k$ and $\boldsymbol{G}_k$ replaced by their estimates $\tilde{\boldsymbol{f}}_k$ and $\tilde{\boldsymbol{G}}_k$ respectively. This clearly confirms that the HCE approach, which characterizes the majority of adaptive controllers, treats the estimates as if they were exact, which is never the case in real-life situations as argued in Section I.*

2) UT-based Dual Adaptive Scheme: The EKF-based dual adaptive scheme just presented employs the EKF algorithm to address the ANN weight-tuning task. Moreover, the corresponding dual adaptive control law in (18) relies on a first-order Taylor approximation of $p(\boldsymbol{y}_{k+1}|I^k)$, as detailed in Lemma III.2. In contrast, the novel UT-based dual adaptive scheme detailed in the following paragraphs uses a specifically devised form of the UKF [42], [43] as a recursive weight-tuning algorithm, to replace the less accurate EKF algorithm of the previous scheme, and in addition employs a novel dual adaptive law that uses the UT to improve on the first-order Taylor in Lemma III.2 which leads to the EKF-based control law in (18).

As argued in [43] the UKF, originally proposed by Julier *et. al.* in [42], provides a better alternative to the well established EKF to address the problem of stochastic nonlinear estimation. Both the EKF and UKF approximate the state (or parameter) distribution by a GRV. However, while the EKF propagates the mean and covariance of this GRV through the first-order linearization of the nonlinear system, the UKF uses a minimal set of deterministically chosen sample points, termed *sigma points*, that capture completely the true mean and covariance of the GRV, and propagates them through the true nonlinear system, yielding a posterior mean and covariance that are accurate up to the second order Taylor series expansion for any nonlinearity. In contrast, the EKF is accurate only up to the first-order Taylor series expansion [43]. Moreover, the UKF is a derivative-free algorithm and as shown later in Section IV-B, it is still computationally efficient enough to be implemented on available hardware in real-time practical applications.

Starting from the MLP ANN formulation of Section III-C leading to (14), we now proceed to propose the use of an UKF algorithm in prediction mode for the real-time estimation of $\boldsymbol{z}^*_{k+1}$ as follows.

Lemma III.3. *In the light of (13), Definition III.6, and Assumption III.3, it follows that* $p(\boldsymbol{z}^*_{k+1}|I^k) \approx \mathcal{N}(\hat{\boldsymbol{z}}_{k+1}, \boldsymbol{P}_{k+1})$, *where* $\hat{\boldsymbol{z}}_{k+1}$ *and* $\boldsymbol{P}_{k+1}$ *are computed at each control step according to the UKF Algorithm III.2. Consequently,* $\hat{\boldsymbol{z}}_{k+1}$ *is considered to be the estimate of* $\boldsymbol{z}^*_{k+1}$ *conditioned on* I^k, *and* $\boldsymbol{P}_{k+1}$ *can be viewed as a measure of this estimate's uncertainty.*

Proof: The UKF algorithm in prediction mode, presented in Algorithm III.2, is effectively the standard UKF algorithm as stated in [43] for parameter estimation, with the difference that the measurement-update step precedes that for time-update. In addition, the time-update step is advanced by one sample to obtain $\hat{\boldsymbol{z}}_{k+1|k}$ at instant k. Hence, the proof of Lemma III.3 follows directly that of the UKF (additive noise version) when applied to the nonlinear stochastic state-space model in (13). ■

Given the previous prediction $\left(\hat{\boldsymbol{z}}_{k|k-1}, \boldsymbol{P}_{k|k-1}\right)$, denoted in short-form by $(\hat{\boldsymbol{z}}_k, \boldsymbol{P}_k)$; the following UKF algorithm (prediction mode) generates the new prediction $(\hat{\boldsymbol{z}}_{k+1}, \boldsymbol{P}_{k+1})$:

1) Sigma-points sampling and propagation:

$$\begin{aligned}
\mathcal{Z}_{k|k-1} &= \left[\hat{\boldsymbol{z}}_k \quad \hat{\boldsymbol{z}}_k + \left(\gamma\sqrt{\boldsymbol{P}_k}\right) \quad \hat{\boldsymbol{z}}_k - \left(\gamma\sqrt{\boldsymbol{P}_k}\right)\right] \\
\mathbb{F}_{k|k-1} &= \tilde{\boldsymbol{f}}(\boldsymbol{x}_{k-1}, \mathcal{R}_{k|k-1}),\ \mathbb{G}_{k|k-1} = \tilde{\boldsymbol{G}}(\mathcal{G}_{k|k-1}) \\
\mathbb{Y}_{k|k-1} &= \mathbb{F}_{k|k-1} + \mathbb{G}_{k|k-1}\boldsymbol{\tau}_{k-1} \\
\hat{\boldsymbol{y}}_k &= \sum_{i=0}^{2N} W_{m\,i}\mathbb{Y}_{i,k|k-1} \qquad (21)
\end{aligned}$$

2) Measurement update and estimate prediction:

$$\boldsymbol{P}_{\boldsymbol{yy}\,k} = \sum_{i=0}^{2N} W_{c\,i}\left[\mathbb{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_k\right]\left[\mathbb{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_k\right]^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}$$

$$\begin{aligned}
\boldsymbol{P}_{\boldsymbol{zy}\,k} &= \sum_{i=0}^{2N} W_{c\,i}\left[\mathcal{Z}_{i,k|k-1} - \hat{\boldsymbol{z}}_k\right]\left[\mathbb{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_k\right]^T \\
\boldsymbol{K}_k &= \boldsymbol{P}_{\boldsymbol{zy}\,k}\boldsymbol{P}_{\boldsymbol{yy}\,k}^{-1}, \quad \boldsymbol{i}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k \\
\hat{\boldsymbol{z}}_{k+1} &= \hat{\boldsymbol{z}}_k + \boldsymbol{K}_k\boldsymbol{i}_k \\
\boldsymbol{P}_{k+1} &= \boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{P}_{\boldsymbol{yy}\,k}\boldsymbol{K}_k^T + \boldsymbol{Q}_{\boldsymbol{\rho}}
\end{aligned}$$

where: $\mathcal{Z}^T = \left[\mathcal{R}^T \quad \mathcal{G}^T\right]^T$, $\gamma = \sqrt{N+\lambda}$, N is the length of $\hat{\boldsymbol{z}}_k$, the scaling parameter $\lambda = \alpha^2\left(N+\kappa\right) - N$, constant α determines the spread of the sigma-points, constant κ is a secondary scaling parameter, the UT weights are given by: $W_{m0} = \frac{\lambda}{N+\lambda}$, $W_{c0} = W_{m0} + 1 - \alpha^2 + \beta$, and $W_{mi} = W_{ci} = \frac{1}{2(N+\lambda)}$ $(i = 1, \ldots, 2N)$, and β includes prior knowledge of the estimate's distribution.
Moreover, in the UKF framework the linear algebra operation of adding a column vector to a matrix is defined as the addition of the vector to each column of the matrix. For further details, including guidelines for selecting the UKF scaling parameters, one is referred to [43].

Algorithm III.2: The UKF parameter-prediction algorithm.

Lemma III.4. *On the basis of Lemma III.3, it follows that* $p(\boldsymbol{y}_{k+1}|I^k)$ *is approximately Gaussian with mean* $\hat{\boldsymbol{y}}_{k+1}$ *and covariance* $\boldsymbol{P}_{\boldsymbol{yy}\,k+1}$ *given by:*

$$\hat{\boldsymbol{y}}_{k+1} = \hat{\boldsymbol{f}}_k + \hat{\boldsymbol{G}}_k\boldsymbol{\tau}_k, \qquad (22)$$

$$\textit{where,}\quad \hat{\boldsymbol{f}}_k = \sum_{i=0}^{2N} W_{m\,i}\mathbb{F}_{i,k+1|k}, \quad \hat{\boldsymbol{G}}_k = \tilde{\boldsymbol{G}}\left(\hat{\boldsymbol{g}}_{k+1}\right) \qquad (23)$$

and the covariance

$$\boldsymbol{P}_{\boldsymbol{yy}\,k+1} = \qquad (24)$$
$$\sum_{i=0}^{2N} W_{c\,i}\left[\boldsymbol{D}_{\boldsymbol{f}\,i} + \boldsymbol{D}_{\boldsymbol{G}\,i}\boldsymbol{\tau}_k\right]\left[\boldsymbol{D}_{\boldsymbol{f}\,i} + \boldsymbol{D}_{\boldsymbol{G}\,i}\boldsymbol{\tau}_k\right]^T + \boldsymbol{R}_{\boldsymbol{\epsilon}}$$

$$\textit{where,}\quad \boldsymbol{D}_{\boldsymbol{f}\,i} = \mathbb{F}_{i,k+1|k} - \hat{\boldsymbol{f}}_k\ , \quad \boldsymbol{D}_{\boldsymbol{G}\,i} = \mathbb{G}_{i,k+1|k} - \hat{\boldsymbol{G}}_k.$$

Proof: The equation of $\hat{\boldsymbol{f}}_k$ in (23) is derived by applying the UT to estimate the mean of $p\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{r}^*_{k+1})|I^k\right)$. The equation of $\hat{\boldsymbol{G}}_k$ in (23) is more straightforward since $\tilde{\boldsymbol{G}}_k$ is linear in the unknown parameters. Hence we simply employ the fact that $E\left\{\boldsymbol{g}^*_{k+1}\right\} = \hat{\boldsymbol{g}}_{k+1}$. To derive the equation of

$\boldsymbol{P_{yy}}_{k+1}$ in (24) one needs to advance the equation for $\boldsymbol{P_{yy}}_k$ in Algorithm III.2 by one sampling instant, and substitute for $\mathbb{Y}_{i,k+1|k}$ and $\hat{\boldsymbol{y}}_{k+1}$, using the relations leading to (21) in the same algorithm. ■

Remark III.6. *One should particularly note that in Lemma III.4, the evaluation of the approximate mean and covariance of $p(\boldsymbol{y}_{k+1}|I^k)$ are not based on a first-order Taylor approximation, as in the case of the EKF-based scheme specifically in Lemma III.2, but are generated through the more accurate method for approximating the statistics of random variables which undergo a nonlinear transformation, namely the UT [42].*

Algorithm III.2, in the light of Lemma III.3 constitutes the weight adaptation law for the novel UT-based MLP dual adaptive scheme. In addition, Lemma III.4 provides a real-time update of the probability density function $p(\boldsymbol{y}_{k+1}|I^k)$. This information is employed by the UT-based dual adaptive control law stated in the theorem below.

Theorem III.2. *The control law minimizing performance index J_{inn} in (17), subject to the WMR dynamic model (6), Definitions III.7 and III.8, Remark III.3 and Lemmas III.3 and III.4, is given by*

$$\begin{aligned}\boldsymbol{\tau}_k \quad = \quad & \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \hat{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N_{GG}}_{k+1}\right)^{-1} \\ & \times \left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} (\boldsymbol{y_d}_{k+1} - \hat{\boldsymbol{f}}_k) - \boldsymbol{n_{Gf}}_{k+1}\right),\end{aligned} \tag{25}$$

where

$$\boldsymbol{N_{GG}}_{k+1} = \sum_{i=0}^{2N} W_{ci} \boldsymbol{D_G}_i^T \boldsymbol{Q_4} \boldsymbol{D_G}_i \tag{26}$$

$$\boldsymbol{n_{Gf}}_{k+1} = \sum_{i=0}^{2N} W_{ci} \boldsymbol{D_G}_i^T \boldsymbol{Q_4} \boldsymbol{D_f}_i \quad \textit{and} \quad \boldsymbol{Q_4} = \boldsymbol{Q_1} + \boldsymbol{Q_3}.$$

Proof: Given the approximate Gaussian distribution of $p(\boldsymbol{y}_{k+1}|I^k)$ specified in Lemma III.4, and standard results from linear algebra involving matrices [49], it follows that within this scheme, (17) can be rewritten as

$$\begin{aligned}J_{inn} \quad = \quad & \left(\hat{\boldsymbol{y}}_{k+1} - \boldsymbol{y_d}_{k+1}\right)^T \boldsymbol{Q_1} \left(\hat{\boldsymbol{y}}_{k+1} - \boldsymbol{y_d}_{k+1}\right) \\ & + \quad \boldsymbol{\tau}_k^T \boldsymbol{Q_2} \boldsymbol{\tau}_k \quad + \quad \mathrm{tr}\left(\boldsymbol{Q_4} \boldsymbol{P_{yy}}_{k+1}\right).\end{aligned} \tag{27}$$

By substituting for $\hat{\boldsymbol{y}}_{k+1}$ and $\boldsymbol{P_{yy}}_{k+1}$ in (27), using the relations in (22) and (24) respectively, it is possible to factorize J_{inn} completely in terms of $\boldsymbol{\tau}_k$. The resulting quadratic expression is differentiated with respect to $\boldsymbol{\tau}_k$ and then equated to zero in order to determine its stationary point. This leads to (25). Moreover, the resulting Hessian matrix is given by $2\left(\hat{\boldsymbol{G}}_k^T \boldsymbol{Q_1} \hat{\boldsymbol{G}}_k + \boldsymbol{Q_2} + \boldsymbol{N_{GG}}_{k+1}\right)$, which by Definition III.8 and (26) can be shown to be positive definite. This means that the UT-based dual adaptive control law specified in (25) minimizes (17) uniquely, and the inverse term in (25) exists without exceptions. ■

Remark III.4 and III.5; with (18) replaced by (25) and $\boldsymbol{\kappa}_{k+1}$ replaced by $\boldsymbol{n_{Gf}}_{k+1}$ in the former, and with $\tilde{\boldsymbol{f}}_k$ and $\tilde{\boldsymbol{G}}_k$ replaced by $\hat{\boldsymbol{f}}_k$ and $\hat{\boldsymbol{G}}_k$ in the latter; also apply in the context of this scheme.

IV. Simulation and Experimental Results

As pointed out in Section I, the performance of dual adaptive controllers is typically tested by computer simulations and real-life experiments. In this section we present a number of *both* simulation and experimental results to demonstrate the effectiveness of the novel UT-based adaptive control scheme and to compare it with the EKF-based scheme originally proposed in [26] and briefly revisited in this paper.

A. Simulation Results

Some of the parameters in our simulations namely; the measurement noise and the robot mass, inertia and friction; are programmed to change arbitrarily from one simulation trial to the other. This renders the simulations more realistic but also nondeterministic. For this reason we do not base our controller validations and comparisons on a single simulation trial, but opt to perform a Monte Carlo exercise that involves 500 simulation trials instead. To strengthen our analysis even further, we employ a statistical hypothesis test using the data acquired from the Monte Carlo simulations as detailed later in this section.

The differential WMR under study is simulated via the continuous-time dynamic model given by (1) and (5). As indicated previously, a number of parameters in this model namely d, m_c, I_c and $\bar{\boldsymbol{F}}(\boldsymbol{\nu})$, are programmed to vary from one simulation trial to the other. These variations adhere to the physics of arbitrarily but realistically generated scenarios, comprising various robot load configurations and frictional conditions. Specifically, in the initialization stage of each simulation trial the modelled WMR is virtually loaded with a point mass, ranging from 0 to 10 kg, placed on the axis perpendicular to the driving axle at a distance, ranging from -0.5 to 0.5 m, away from $\mathrm{P_o}$. Effectively this yields a new set of arbitrary but realistic values for d, m_c and I_c. Moreover, wheel viscous friction is included in the simulation by setting $\bar{\boldsymbol{F}}(\boldsymbol{\nu}) = \boldsymbol{F_c}\boldsymbol{\nu}$, where $\boldsymbol{F_c}$ is a diagonal matrix of coefficients whose values are randomly generated afresh from a uniform distribution ranging from 0.001 to 0.5, prior to each simulation trial. All the other WMR parameters are held constant for all simulations and are tabulated in Table I, along with the values for d, m_c and I_c that correspond to the specific case of the unloaded WMR. These parameters are based on actual measurements taken from Neurobot, the experimental WMR designed and built by the authors for the purpose of this research.

Each simulation trial consists of eight consecutive controller simulations. The first six of these correspond to the three modes of operation; *i.e.* HCE mode ($\boldsymbol{Q_3} = -\boldsymbol{Q_1}$), cautious mode ($\boldsymbol{Q_3} = \boldsymbol{0}$) and dual mode ($\boldsymbol{Q_3} = -0.8\boldsymbol{Q_1}$); for each of the two adaptive schemes being compared. The remaining two correspond to: (1) a nominally-tuned nonadaptive (NTNA) controller, which is effectively the computed-torque controller in (10) with $k_d = 0$, pre-tuned with the mean values of the

TABLE I
WMR PHYSICAL PARAMETERS (NEUROBOT WITH NO LOAD).

Parameter	Value
d	0 m
b	22.95 cm
r	6.25 cm
m_c	21.0 kg
m_w	1.5 kg
I_c	0.55 kgm^2
I_w	0.0006 kgm^2
I_m	0.01 kgm^2

robot dynamic parameters, specifically: $\bar{d} = 0$ m, $\bar{m_c} = 26$ kg, $\bar{I_c} = 0.87$ kgm^2 and the diagonal values of $\bar{\boldsymbol{F_c}}$ both set to 0.25. It is important to appreciate that this is the best a *nonadaptive* controller can do when the exact robot parameters are unknown to the controller, as in the case of these simulations and typical real-life applications; (2) a perfectly-tuned nonadaptive (PTNA) controller, which is effectively the computed-torque control law (10) with $k_d = 0$, pre-tuned with the exact values of the robot parameters. The latter is the best theoretical controller since it perfectly cancels the nonlinearities and yields *deadbeat* control. Naturally this controller is unrealistic since the exact robot parameter values are never known in practice and are generally prone to change. Hence we use this controller solely to provide an ideal reference for quantitative comparisons. In contrast, the HCE, cautious and dual adaptive controllers assume no preliminary information about the robot dynamics whatsoever, since closed-loop control is activated immediately with the initial parameter estimate vector $\hat{\boldsymbol{z}}_0$ generated randomly from a zero-mean, Gaussian distribution with variance 0.025.

For the sake of fair comparison the same control sampling interval ($T = 50$ ms), velocity measurement noise sequence $p(\boldsymbol{\epsilon}_k) \sim \mathcal{N}(0, 0.0001\mathbf{I_2})$, reference trajectory, initial conditions, initial filter covariance matrix ($\boldsymbol{P}_0 = 0.5\mathbf{I_{27}}$), artificial process noise covariance ($\boldsymbol{Q_\rho} = 10^{-8}\mathbf{I_{27}}$), tracking error penalty ($\boldsymbol{Q_1} = \mathbf{I_2}$), and control input penalty ($\boldsymbol{Q_2} = \mathbf{0}$), are used in each controller simulation in a particular simulation trial. In addition, the sigmoidal MLP ANN used in each of the two schemes under test contains five neurons ($L = 5 \Rightarrow N = 27$). Our experiments indicated that adding more neurons did not improve the control performance significantly. In the UT-based scheme, the UKF scaling parameters are set to $\alpha = 1, \kappa = 0$ and $\beta = 2$.

1) Single Trial Analysis: A number of simulation results typifying the performance of the three control modes of the proposed UT-based adaptive scheme as well as the EKF-based adaptive scheme revisited in this paper are depicted in Figure 5. It should be emphasized at the outset that these results are only included to depict the typical performance of each adaptive control mode (HCE, cautious and dual) of each scheme, and not to be used to compare the two schemes (the UT-based and the EKF-based) themselves. The reason for this is that the results shown in Figure 5 correspond to single simulation trials, and since the nature of the simulation is stochastic, it is inappropriate to draw general conclusions based solely on the result of one or two simulation trials. The Monte Carlo analysis that follows later in this section is designed to address this issue and leads to a more fair and scientifically sound comparison of the proposed schemes. However, the single trial results presented in Figure 5 do give a number of important indications on the relative performance of the HCE, cautious and dual adaptive control modes, which we have found to be highly consistent and independent on the number of trials and even the scheme itself.

In Figure 5, the plots labelled (.i) correspond to the proposed UT-based scheme while those marked (.ii) correspond to the EKF-based scheme. The following comments and observations apply to both schemes. Plots (a.i) and (a.ii) depict the WMR, controlled by the respective adaptive controller in dual mode, tracking a demanding reference trajectory with nonzero initial tracking error. It is clear that the robot converges quickly to the reference trajectory and keeps tracking it with high precision, even when it reaches high speeds of around 1 m/s. Plots (b) to (e) focus on the transient of another simulation trial that uses the same reference trajectory, but purposely initiated with zero tracking error conditions. In this manner, any transient errors can be attributed to the capability of the respective controller to cope with the initially high levels of uncertainty in the estimates. Plots (b.i) and (b.ii) compare the Euclidean norm (denoted by $\|\cdot\|$ throughout the paper) of the $x - y$ position error vector. This is computed via $\|xy_{\text{error}}\| = \sqrt{(x_r - x)^2 + (y_r - y)^2}$. Plots (c.i) and (c.ii) show the magnitude of the WMR orientation error for the three control modes. Plots (d.i) and (d.ii) show the error in the robot pose while Plots (e.i) and (e.ii) compare the corresponding control inputs, more specifically the Euclidian norm of the torque vector. As can be seen in Plots (e.i) and (e.ii), the HCE controller leads to very high transient control inputs. This is a direct results of its aggressive and incautious nature, stemming from the fact that it completely ignores the high uncertainty in the initial estimates. Plots (b) to (d) clearly indicate that this leads to relatively high transient errors in both position and orientation. The cautious mode, which leads to lower transient errors relative to the HCE, is slightly more sluggish than the dual mode. This can be seen in Plots (e), where the initial control input issued by the cautious controller is the lowest. This leads to a slower (relative to the dual mode) decay of the pose error as indicated in Plots (d.i) and (d.ii). It is clear that the dual mode manages to strike a balance between these two extremes and leads to the best transient performance in both schemes. All these observations are in accordance with the anticipations of Remark III.4. In addition, the three adaptive modes in each scheme converge to the same performance at steady-state. This is not unexpected due to the fact that by the time steady-state is reached the parameter estimates would have practically converged to the actual parameters, meaning that the robot would have adapted well to its own current dynamics.

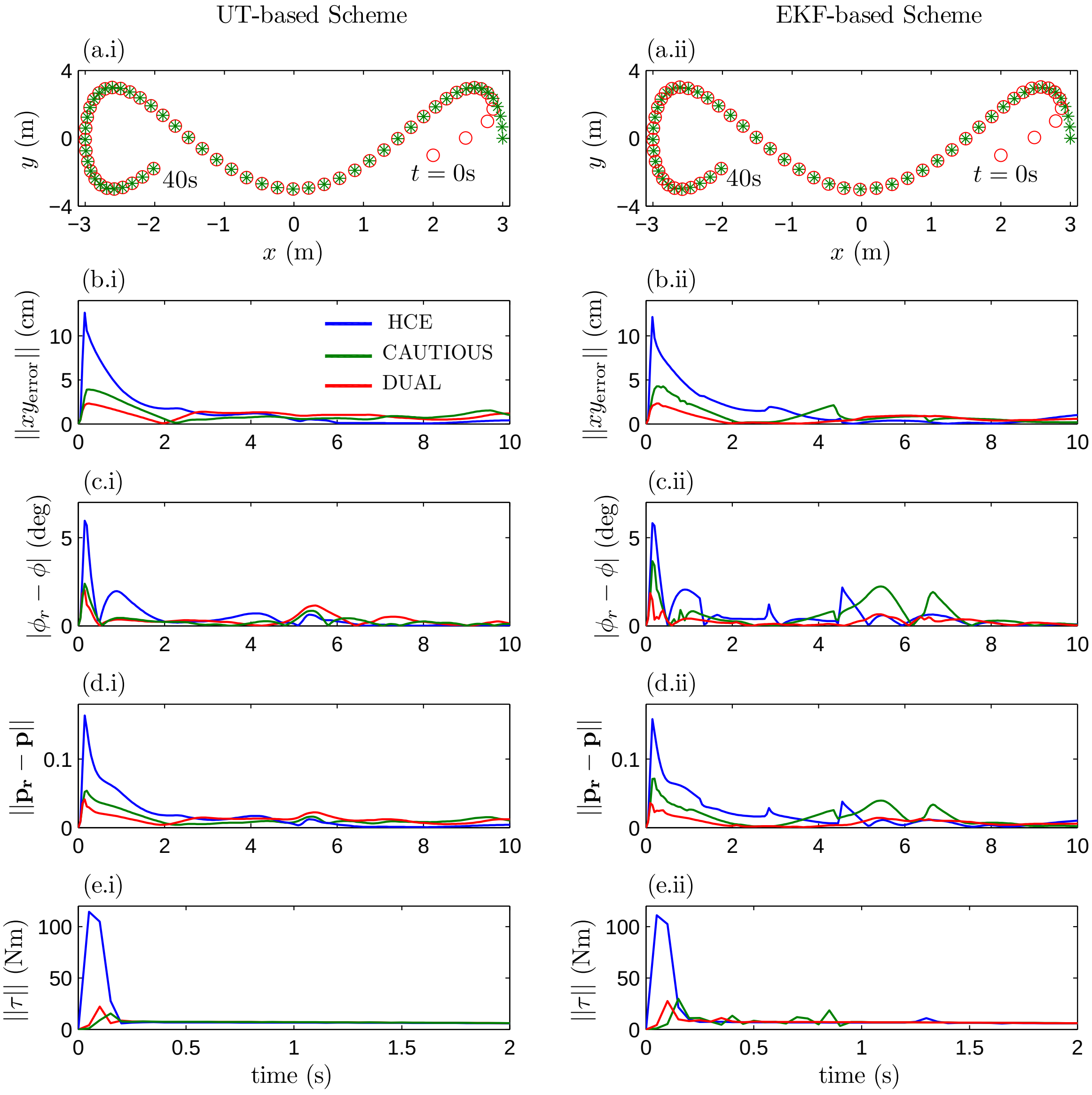

Fig. 5. Simulation results for the (i) UT-based and (ii) EKF-based schemes: (a) reference (green ×) and actual (red ◯) trajectories, (b) position error $\|xy_{\text{error}}\| = \sqrt{(x_r - x)^2 + (y_r - y)^2}$, (c) orientation error, (d) pose error, (e) control input. N.B. (a) controller in dual mode with non-zero initial error, (b) to (e) transients for zero initial error.

2) Monte Carlo Analysis: To quantify the controllers' performance objectively, a Monte Carlo simulation involving 500 simulation trials was performed. For each of the eight controller simulations in a trial, the reference trajectory depicted in Figure 5a, but with zero initial tracking error, is used and the simulation settings and conditions specified earlier apply. At the end of each trial, the following accumulated cost function $\mathcal{C}(k_{end})$ is calculated:

$$\mathcal{C}(k_{end}) = \sum_{k=1}^{k_{end}} \|\boldsymbol{p_r}_k - \boldsymbol{p}_k\|^2.$$

This cost function, based on the robot pose error over the whole time horizon (k_{end} sampling instants), serves as a performance measure for each of the eight controllers operating under the same conditions, where lower values of $\mathcal{C}(k_{end})$ are naturally preferred.

The salient statistical features of the resulting eight cost distributions resulting from this Monte Carlo simulation, are depicted in the boxplot of Figure 6. Additionally, the median, interquartile range (IQR), mean and variance of each of these distributions are given in Table II. Due to the skewness of these distributions and the high number of outliers in some of the cases, the median is preferred over the mean as a measure of central tendency while the IQR is preferred over the variance as a measure of dispersion (spread). The results in Figure 6 and Table II provide the first indications how one would rank the general performance of the controllers

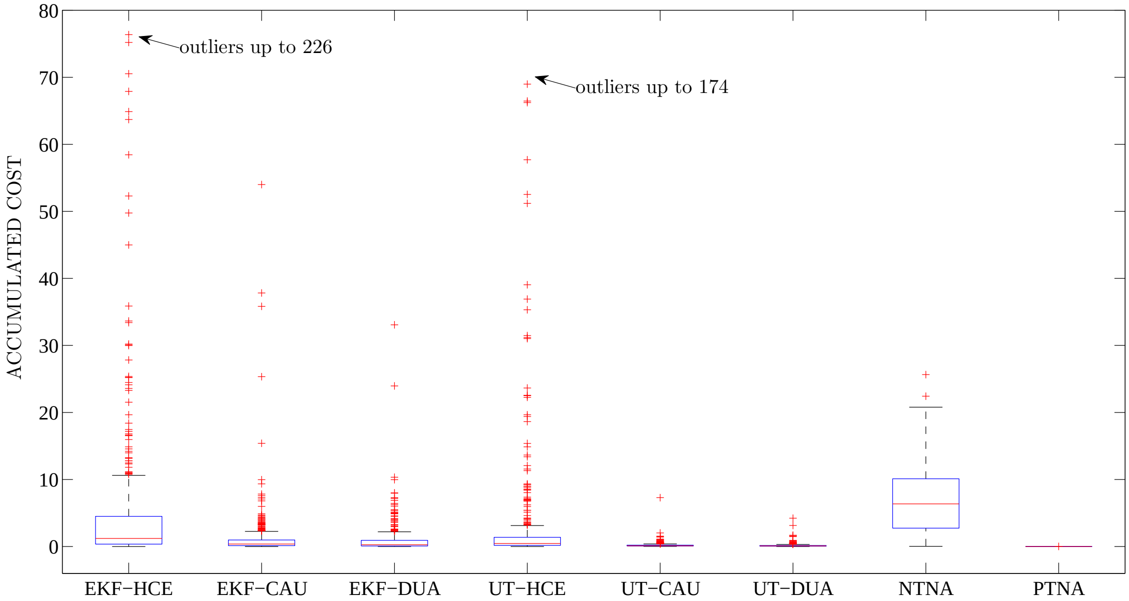

Fig. 6. Boxplot of the cost distributions.

TABLE II
STATISTICAL MEASURES OF THE COST DISTRIBUTIONS.

Controller	Median	IQR	Mean	Variance	Rank
EKF-HCE	1.20	4.16	7.24	471.12	6
EKF-CAU	0.37	0.87	1.18	14.07	4
EKF-DUA	0.27	0.85	0.91	5.06	3
UT-HCE	0.44	1.22	4.65	303.79	5
UT-CAU	0.12	0.13	0.19	0.14	2
UT-DUA	0.09	0.11	0.15	0.08	1
NTNA	6.36	7.38	6.86	21.06	-na-
PTNA	0.003	0.004	0.004	<0.000	-na-

under investigation, where lower values of the median and IQR are obviously preferred. From the outset one can easily notice that the NTNA controller yields the highest median and IQR, implying that in general it leads to the highest pose error and deviation in performance. This is not unexpected since this controller is not adaptive and so unable to cope well with the robot parameters that are constantly changing from one simulation trial to another. In fact, its performance could be much worse if the nominal parameters, to which it is tuned, are unknown or the model variations are higher. For this reason there is no scope in comparing it further to the other adaptive controllers, and so it is withdrawn from the following comparative analysis. Consequently in the following comparative treatment we focus solely on the remaining six adaptive controllers since the PTNA results are included only for reference.

Focusing back on the six adaptive controllers, one notices that the two HCE controllers yielded a relatively high number of extreme outliers (refer to Figure 6). This is the reason why in Table II the mean and variance corresponding to these controllers are exceptionally high. This implies that in a number of trials the HCE mode led to very high transient errors. Again, this implies that the complete lack of sensitivity exhibited by HCE adaptive controllers in the face of the highly uncertain estimates characterizing the startup phase, can lead to excessively high control inputs and tracking errors which can potentially result in mission failure and possibly hardware damage in a practical situation. This strengthens our previous results in Section IV-A1 and again consolidates the arguments in Remark III.4. The results in Table II also indicate that within each scheme the dual mode outperforms the cautious and HCE modes. In addition, it is evident that each mode in the UT-based scheme outperforms its counterpart in the EKF-based scheme. The latter implies that the proposed UT-based scheme brings by a considerable improvement over the EKF-based scheme, originally proposed in [26]. However, in order to strengthen these claims further and to ascertain that the observed differences in the performance of each controller, indicated by the results in Table II, are statistically significant and cannot be attributed to chance, we employed a statistical inference procedure via the following hypothesis test.

The One-Way Analysis of Variance (ANOVA) is a powerful statistical procedure used to make inferences on the population means of several independent samples. Like all other parametric tests it relies on a number of assumptions [51]. Most importantly, the samples should be independent, normally distributed and exhibit fairly similar variances. It is also known that ANOVA is quite robust in the face of violations to its assumptions, mostly so when the sample sizes are large and equal. However, the cost distributions corresponding to the six adaptive controllers left for investigation are all positively skewed, and therefore cannot be closely approximated to normal distributions. Hence, the original cost observations were all transformed using the natural logarithm function. The transformed samples were found to be fairly Gaussian (skewness and kurtosis in the range of ± 1). This was verified by investigating the histogram and the normal quantile-quantile (Q-Q) plots of each transformed sample [51]. However, the Levene's test for homogeneity of variance [51] indicated that equal variances among the six transformed samples still could not be assumed. In such cases it is suggested that the Brown-Forsythe F statistic or the Welch's F statistic are used instead of the standard F statistic in the ANOVA test [51].

Based on these results, the log transformed cost values were used in the ANOVA test, aimed to compare the population means of the six cost distributions. The null and alternative hypotheses for this two-tailed test are:

H_0 : In general the six adaptive controllers perform equally well. In other words: in an infinite number of Monte Carlo simulation trials the six controllers would yield the same mean cost.

H_1 : Some controllers perform better than the others. In other words: in an infinite number of Monte Carlo simulation trials two or more controllers would yield a different mean cost.

The resulting p-values [51], corresponding to the Brown-Forsythe and the Welch tests, were both approximately zero. Hence, since the p-value is smaller than the chosen level of significance $\alpha = 0.05$, the null hypothesis H_0 is rejected. This implies that *at least* one of the six controllers is significantly better (cost-wise) than the others. In order to investigate the underlying differences further and be able to rank the controllers according to their performance we employed the Games-Howell *post-hoc* test, which is highly recommended in the case of unequal variances [51]. The result was conspicuous, since all the p-values resulting from all pair-wise combinations were much lower than the chosen level of significance α. This implies that the means of the transformed samples, depicted in Figure 7, are *all significantly different* and can be used to rank the general performance of the six adaptive controllers as given in the last column of Table II. In addition, a non-parametric test using the original cost distributions instead of the transformed distributions, namely the Kruskal-Wallis test [51] was also employed to test the set hypothesis. The final result of this analysis fully confirms that of the ANOVA.

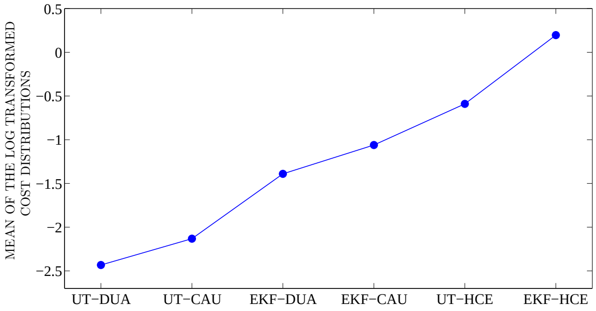

Fig. 7. Means plot of the log transformed cost distributions.

The results from this Monte Carlo comparative analysis fully support those derived from Table II and Figure 5. Hence, we can confidently claim that:

Remark IV.1. *The proposed UT-based scheme brings about a significant improvement in tracking performance over the EKF-based scheme, independent of the controller mode (HCE, cautious or dual). We associate this to the better estimations of the UKF over those of the EKF in the ANN training algorithm, and to the better (second-order) approximations of the UT-based control law as opposed to the first-order approximations inherent in the EKF-based control algorithm. Moreover, within each scheme the dual mode is superior to both the cautious and HCE modes. This complies with the dual control philosophy that a balance between caution and probing yields the best performance in adaptive control. It is also not surprising that the performance of the adaptive controllers is generally better than that of the computed-torque nonadaptive controller which assumes nominal values for the robot dynamic parameters, when these are prone to change.*

B. Experimental Results

The UT-based and EKF-based dual adaptive neuro-controllers presented in this article were both implemented successfully on a physical WMR, named Neurobot, which was designed and built by the authors as an experimental research testbed. This section introduces Neurobot and reports a number of experimental results that compliment those acquired by simulation and reported in the previous section.

Neurobot, pictured in Figure 8, is a differentially driven WMR. Each of the two 125 mm diameter, solid-rubber, motorized wheels, is independently driven by a 70 W, 24 V permanent magnet dc motor (from maxon motor [52]), equipped with a 113:1 planetary reduction gearbox and a 500 pulses per revolution incremental optical encoder. Each of the two motors is driven via the LMD18200 H-Bridge IC which is controlled by a 20 kHz pulse-width modulation (PWM) reference signal. The instantaneous current in each motor is measured using the LEM HX-03-P/SP2 Hall effect current transducer, and filtered by a 4th-order continuous-time Bessel low-pass anti-aliasing filter, tuned for a corner frequency of 2 kHz, and implemented via the MAX275 filter IC. Neurobot is powered by four 12 V, 9 Ah sealed lead acid (SLA) batteries.

The algorithms controlling Neurobot were all implemented on a *MicroAutoBox* embedded computer system from *dSPACE* [53]. The *MicroAutoBox* is a compact stand-alone prototyping unit designed specifically for rapid-prototyping of computationally demanding real-time control systems, typically requiring a number of general and specialized analogue/digital input and output channels. A digital pole-placement torque controller with integral action, was designed and implemented completely in software to account for the motor electrical dynamics. This inner torque control loop uses the motor current

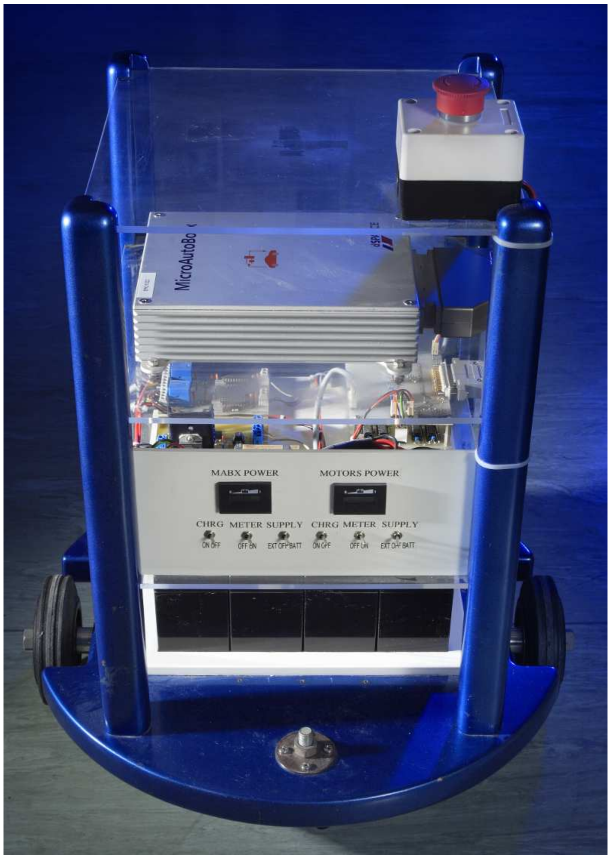

Fig. 8. Neurobot: the WMR built for the purpose of this research.

measurement as feedback and issues voltage commands to the motors. This ascertains that the actual torques at the wheels track those issued by the outer loop control law (the robot dynamic controller) and that motor current never exceeds a predefined safe value. This cascade approach imposes that the inner loop operates at a much faster rate than the outer loop. The sampling rates for the inner and outer loops were chosen to be 10 kHz and 200 Hz respectively.

A desktop computer was used to implement the control algorithms in *Simulink*® using the system blocks provided by the *dSpace Real-Time Interface*. *Real-Time Workshop*® is then used to automatically generate the required code which is then downloaded to the *MicroAutoBox* via the *dSpace Link Board* installed in the desktop computer. The system states and parameters along with other information about the real-time execution of each task running on the *MicroAutoBox*, such as sampling times, priorities and execution times, could also be monitored in real time via *ControlDesk*, also from *dSPACE*.

The initial network parameter vector $\hat{\boldsymbol{z}}_0$ was generated randomly. In addition, the MLP ANN contained five neurons ($L = 5 \Rightarrow N = 27$) and the UKF scaling parameters were set to $\alpha = 10^{-3}, \kappa = 3 - N$ and $\beta = 2$. The initial covariance matrix $\boldsymbol{P}_0 = 0.5\mathbf{I_{27}}$ and the process and measurement noise covariance matrices were set to $10^{-8}\mathbf{I_{27}}$ and $10^{-4}\mathbf{I_2}$ respectively. In addition, $\boldsymbol{Q_1}$ and $\boldsymbol{Q_2}$ were fixed to $\mathbf{I_2}$ and $\mathbf{0}$ respectively in all cases.

A number of experimental results, validating the proposed schemes and confirming the simulation results of this section, are presented in Figure 9. Plots (a) and (b) correspond to a challenging trajectory tracking experiment that tests the overall performance of the UT-based and EKF-based dual adaptive controllers in a real-life application. Plots (a.i) and (a.ii) show that in both cases Neurobot swiftly adapts to its own dynamics (with no preliminary offline training) and simultaneously converges smoothly to the reference trajectory, which it keeps tracking at very high precision for the rest of the experiment. Plots (b.i) and (b.ii) focus on the pose error vector norm $\|\boldsymbol{p}_{\boldsymbol{r}k} - \boldsymbol{p}_k\|$ measured during this experiment. In each case, the red trace corresponds to the dual adaptive controller while the black trace corresponds to a nonadaptive computed-torque controller subjected to the same experiment. This nonadaptive controller employs the control law in (10) with $k_d = 0$ and is tuned for Neurobot's physical parameters reported earlier in Table I. It is clear that the two dual adaptive schemes performed much better than the nonadaptive controller in steady-state. We attribute this results to the fact that the nonadaptive controller is based on a theoretical dynamic model (6), which like any other of its sort, is imperfect and relies on several physical parameters, such as friction and inertia, which are very difficult to measure precisely in practice. On the other hand, the adaptive controllers assume no preliminary information about the robot dynamics but acquire this knowledge autonomously in real-time. In addition, if one compares the pose error of the UT-based dual adaptive controller in Plot (b.i) to that of its EKF-based counterpart in Plot (b.ii), it is easy to notice that the steady-state performance of the former is relatively better than that of the latter. This result is in accordance to Remark IV.1 derived from our simulation results.

Plots (c) and (d) correspond to a different experiment with Neurobot. This experiment was designed specifically to test and compare the transient performance of the two adaptive schemes and their HCE, cautious and dual modes on a real WMR. In this experiment the reference trajectory follows a straight line along the x-axis, with a speed of 0.1 m/s. At $t = 5$ s, well after the robot has reached steady-state operation, the estimate vector $\hat{\boldsymbol{z}}_{k+1}$ is instantaneously reset to some randomly generated values, hence erasing all the knowledge acquired by the ANN estimator up to that point in time. In addition, the covariance matrix $\boldsymbol{P}_{k+1}$ is reset to its initial relatively high value, to reflect the high uncertainty in the new set of random network parameters. In this manner one can objectively compare the transient performance of the three control modes when faced with extremely high uncertainty in the robot dynamics. In practice, similar scenarios may arise during faults and jump variations in the robot dynamics. The question in these cases is not simply whether or not the robot adapts to the new situation, but also how smoothly and quickly it will do so. In Plots (c.i) and (c.ii), it is evident that the HCE mode (blue trace) by far yields the highest transient pose error, as a result of the sudden estimator disturbance at $t = 5$ s. As argued previously, this is clearly the result of

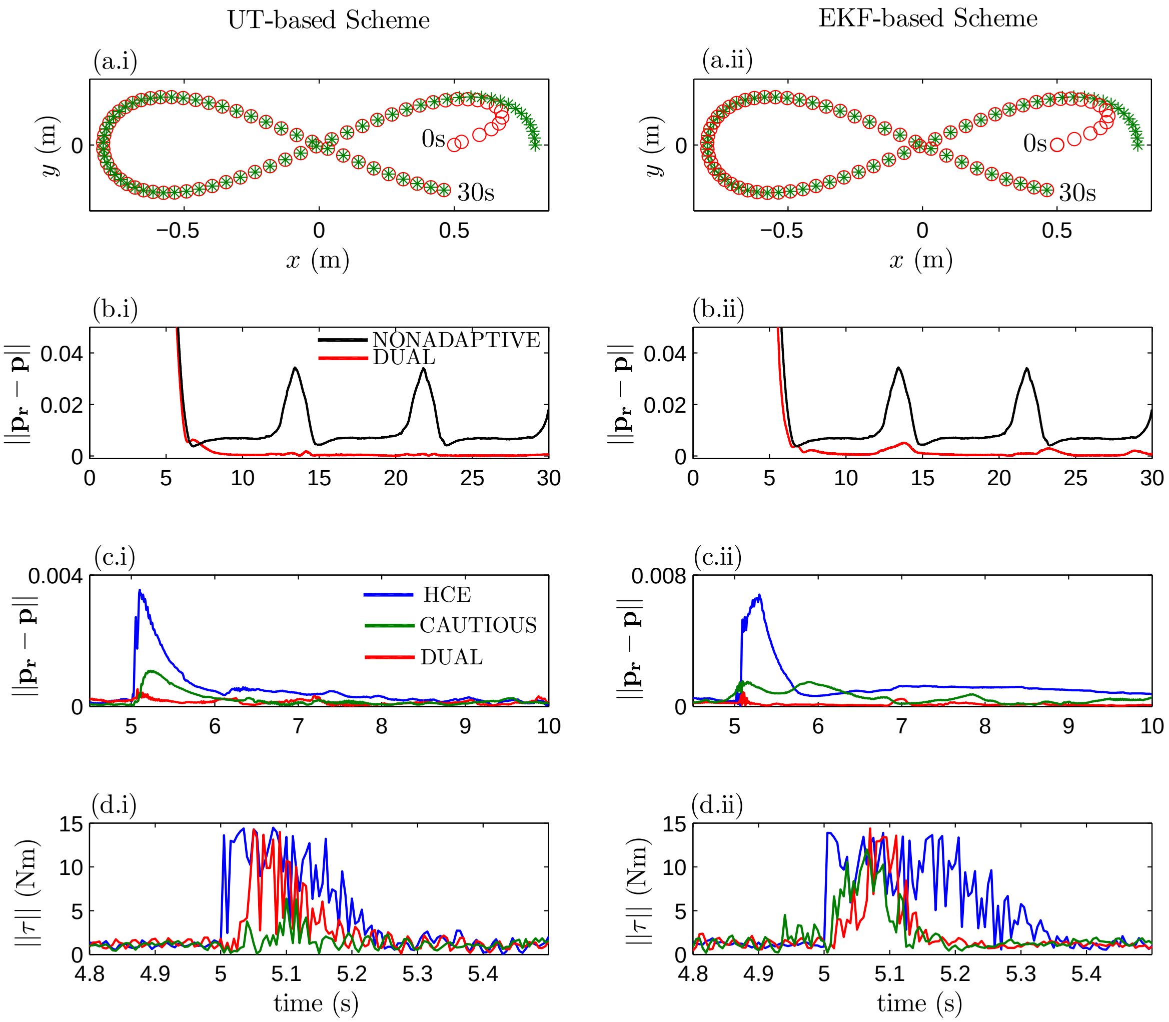

Fig. 9. Experimental results for the (i) UT-based and (ii) EKF-based schemes: (a) reference (green ×) and actual (red ◯) trajectories, (b) pose error (corresponding to the trajectory in (a)), (c) pose error (the line test), (d) control input (the line test). N.B. In (a) the controller is in dual mode (red trace) with non-zero initial error, (c) and (d) depict the line test results.

the relatively persistently aggressive and sudden control input issued by the HCE mode, which can be seen in Plots (d.i) and (d.ii) (blue trace). Specifically, these two plots depict the Euclidean norm of the *actual* torque vector developed by the motors and not that requested by the adaptive controller. In theory these are equal, but in our physical implementation we had to limit the requested torque via a saturation function so as not to damage the electronic circuitry driving the motors. If it were not for this safety feature, the situation would be closer to that depicted in Plots (e.i) and (e.ii) of Figure 5. These results also indicate that out of the three adaptive modes in each scheme, the dual mode (red traces) by far exhibits the best transient performance, due to the very low transient errors and the very quick recovery exhibited in this experiment. Moreover, it is also evident that the three controller modes in the UT-based scheme yielded lower pose errors, and hence better performance than their EKF-based counterparts. This can be clearly seen when one compares the magnitude of the pose errors depicted in Plot (c.i) with that of the errors in Plot (c.ii). One should particularly note the different scales used for the y-axes.

The experimental results presented in this section strongly endorse the simulation results, including those from the Monte Carlo analysis, reported previously in Section IV-A. Consequently they extend the arguments expressed in Remark IV.1 to the case of a real-life robotic application.

V. Conclusion

In this paper we have presented a novel MLP dual adaptive control scheme for the dynamic control of WMRs. The design employs the UKF and the UT to improve on the EKF-based MLP dual adaptive scheme we recently proposed in [26]. The presented designs are validated and compared extensively via both realistic Mont-Carlo simulations, backed by rigorous statistical analysis and real-life experiments with Neurobot, the WMR designed and built by the authors for the purpose of this research. All the results conspicuously show that:

1) The proposed UT-based scheme outperforms the EKF-based scheme.
2) In both schemes, the dual mode is superior (in transient performance) to both the cautious and the HCE controller modes.
3) The steady-state performance of the adaptive controllers is generally better than that of the computed-torque nonadaptive controller.

To the best of our knowledge this is the first time that the UT is being used in the context of dual adaptive control and where a dual adaptive controller is implemented and tested on a real mobile robot.

References

[1] M. K. Bugeja and S. G. Fabri, "Dual-adaptive computer control of a mobile robot based on the unscented transform," in *Proc. of the 3rd Int. Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP' 09)*, Sliema, Malta, Oct. 2009, pp. 136–141.

[2] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Reading, MA: Addison-Wesley, 1995.

[3] B. Wittenmark, "Adaptive dual control methods: An overview," in *5th IFAC Symp. on Adaptive Systems in Control and Signal Processing*, Budapest, Hungary, Jan. 1995, pp. 67–72.

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

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

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

[7] J. Sternby, "A simple dual control problem with an analytical solution," *IEEE Trans. Autom. Control*, vol. 21, no. 6, pp. 840–844, 1976.

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

[9] N. M. Filatov and H. Unbehauen, *Adaptive Dual Control: Theory and Applications*. London, UK: Springer-Verlag, 2004.

[10] Y. Bar-Shalom and E. Tse, *Concept and Methods in Stochastic Control*, ser. Control and Dynamic Systems, C. T. Leondes, Ed. New York, NY: Academic Press, 1976.

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

[12] G. A. Dumont and K. J. Åström, "Wood chip refiner control," *IEEE Control Syst. Mag.*, vol. 8, no. 2, pp. 38–43, 1988.

[13] N. M. Filatov, U. Keuchel, and H. Unbehauen, "Dual control for an unstable mechanical plant," *IEEE Control Syst. Mag.*, vol. 16, no. 4, pp. 31–37, 1996.

[14] B. J. Allison, J. E. Ciarniello, P. J.-C. Tessier, and G. A. Dumont, "Dual adaptive control of chip refiner motor load," *Automatica*, vol. 31, no. 8, pp. 1169–1184, 1995.

[15] A. Ismail, G. A. Dumont, and J. Backstrom, "Dual adaptive control of paper coating," *IEEE Trans. Contr. Syst. Technol.*, vol. 11, no. 3, pp. 289–309, May 2003.

[16] T. Tsumura, N. Fujiwara, T. Shirakawa, and M. Hashimoto, "An experimental system for automatic guidance of robot vehicle, following the route stored in memory," in *Proc. 11th Int. Symp. on Industrial Robots*, Oct. 1981, pp. 187–193.

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

[18] 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.

[19] 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.

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

[21] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *RAMSETE - Articulated and Mobile Robotics for Services and Technologies*, ser. Lecture Notes in Control and Information Sciences, S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi, Eds. Springer-Verlag, 2001, vol. 270, pp. 181–223.

[22] C. de Sousa, E. M. Hemerly, and R. K. H. Galvao, "Adaptive control for mobile robot using wavelet networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 32, no. 4, pp. 493–504, 2002.

[23] 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.

[24] M. Oubbati, M. Schanz, and P. Levi, "Kinematic and dynamic adaptive control of a nonholonomic mobile robot using RNN," in *Proc. IEEE Symp. on Computational Intelligence in Robotics and Automation (CIRA'05)*, Helsinki, Finland, Jun. 2005.

[25] T. Das and I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, no. 3, pp. 501–510, 2006.

[26] M. K. Bugeja and S. G. Fabri, "Dual adaptive dynamic control of mobile robots using neural networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 1, pp. 129–141, 2009.

[27] F. Lamiraux, J. P. Laumond, C. VanGeem, D. Boutonnet, and G. Raust, "Trailer truck trajectory optimization: the transportation of components for the Airbus A380," *IEEE Robot. Autom. Mag.*, vol. 12, no. 1, pp. 14–21, 2005.

[28] R. Murphy, "Rescue robotics for homeland security," *Communications of the ACM*, vol. 27, no. 3, pp. 66–69, 2004.

[29] P. Debanne, J. V. Herve, and P. Cohen, "Global self-localization of a robot in underground mines," in *Proc. Systems, Man, and Cybernetics - Computational Cybernetics and Simulation*, Orlando, FL, Dec. 1997.

[30] M. Long, A. Gage, R. Murphy, and K. Valavanis, "Application of the distributed field robot architecture to a simulated demining task," in *Proc. IEEE Int. Conference on Robotics and Automation (ICRA'05)*, Barcelona, Spain, Apr. 2005.

[31] D. Ding and R. A. Cooper, "Electric-powered wheelchairs," *IEEE Control Syst. Mag.*, vol. 25, no. 2, pp. 22–34, 2005.

[32] 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.

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

[34] P. Morin and C. Samson, "Motion control of wheeled mobile robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin Heidelberg: Spinger-Verlag, 2008, ch. 34.

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

[36] M. K. Bugeja and S. G. Fabri, "Neuro-adaptive dynamic control for trajectory tracking of mobile robots," in *Proc. 3rd Int. Conference on Informatics in Control, Automation and Robotics (ICINCO'06)*, Setúbal, Portugal, Aug. 2006, pp. 404–411.

[37] Z.-G. Hou, A.-M. Zou, L. Cheng, and M. Tan, "Adaptive control of an electrically driven nonholonomic mobile robot via backstepping and fuzzy approach," *IEEE Trans. Contr. Syst. Technol.*, vol. 17, no. 4, pp. 803–815, 2009.

[38] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. London, UK: Prentice Hall, 1999.

[39] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, 1960.

[40] P. S. Maybeck, *Stochastic Models, Estimation and Control*, ser. Mathematics in Science and Engineering, R. Bellman, Ed. London, UK: Academic Press Inc., 1979, vol. 141-1.

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

[42] S. J. Julier and J. K. Uhlmann, "A new extention of the Kalman filter to nonlinear systems," in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[43] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, ser. Adaptive and Learning Systems for Signal Processing, Communications, and Control, S. Haykin, Ed. John Wiley & Sons, Inc., 2001, ch. 7, pp. 221–280.

[44] M. S. Radenkovic, "Convergence of the generalised dual control algorithm," *Int. J. Control*, vol. 47, no. 5, pp. 1419–1441, 1988.

[45] A. D'Amico, G. Ippoliti, and S. Longhi, "A radial basis function networks apporach for the tracking problem of mobile robots," in *Proc. IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics*, Como, Italy, 2001, pp. 498–503.

[46] T.-Y. Wang and C.-C. Tsai, "Adaptive trajectory tracking control of a wheeled mobile robot via Lyapunov techniques," in *Proc. 30th Annual Conference of the IEEE Industrial Electronics Society*, Busan, Korea, Nov. 2004, pp. 389–394.

[47] Y. Yamamoto, "Control and coordination of locomotion and manipulation of a wheeled mobile manipulator," Ph.D. dissertation, Univ. of Pennsylvania, Philadelphia, USA, Aug. 1994.

[48] K. J. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 1997.

[49] K. B. Petersen and M. S. Pedersen. (2008, oct) The matrix cookbook. Version 20081110. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274

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

[51] A. Field, *Discovering Statistics using SPSS*, 2nd ed. London, UK: Sage Publications Ltd., 2005.

[52] (2010) maxon motor. [Online]. Available: http://www.maxonmotor.com

[53] (2010) dSPACE. [Online]. Available: http://www.dspaceinc.com