# NEURO-ADAPTIVE DYNAMIC CONTROL FOR TRAJECTORY TRACKING OF MOBILE ROBOTS

Marvin K. Bugeja† and Simon G. Fabri‡
*Department of Electrical Power and Control Engineering, University of Malta*
*Msida MSD06, Malta*

Keywords: Nonholonomic mobile robots, trajectory tracking, adaptive control, neural networks, stochastic estimation.

Abstract: This paper presents a novel functional-adaptive dynamic controller for trajectory tracking of nonholonomic wheeled mobile robots. The controller is developed in discrete-time and employs a Gaussian radial basis function neural network for the estimation of the robot's nonlinear dynamic functions, which are assumed to be completely unknown. Optimal on-line weight tuning is achieved by employing the Kalman filter algorithm, based on a specifically formulated stochastic inverse dynamic identification model of the mobile base. A discrete-time dynamic control law employing the estimated functions is proposed and cascaded with a trajectory tracking kinematic controller. The performance of the complete system is analysed and compared by realistic simulations.

## 1 INTRODUCTION

Motion control of nonholonomic mobile robots has been receiving considerable attention for the last fifteen years (Kolmanovsky and McClamroch, 1995). This activity is not only justified by the vast array of existing and potential practical applications (Lamiraux et al., 2005; Ding and Cooper, 2005), but also by some particularly interesting theoretical challenges. In particular, most mobile configurations manifest restricted mobility; giving rise to nonholonomic constraints in the kinematics. Moreover the majority of mobile vehicles are underactuated, since they have more degrees of freedom than control inputs. Consequently the linearised kinematic model lacks controllability, full-state feedback linearisation is out of reach (Canudas de Wit et al., 1993), and pure, smooth, time-invariant feedback stabilisation of the Cartesian kinematic model is unattainable (Brockett, 1983).

Originally researchers focused only on kinematic control of nonholonomic vehicles (Crowley, 1989; Kanayama et al., 1990; Canudas de Wit et al., 1993; Kolmanovsky and McClamroch, 1995), assuming that the control signals instantaneously establish the desired robot velocities. This is commonly known as *perfect velocity tracking*(Fierro and Lewis, 1995). This approach may be reasonably valid for light robots following low acceleration trajectories. However it stands to reason that controllers based on a full dynamic model (Sarkar et al., 1994; Fierro and Lewis, 1995; Corradini and Orlando, 2001) capture better the behaviour of real robots because they account for dynamic effects such as mass, friction and inertia, which are otherwise neglected by a mere kinematic controller. On the other hand, the exact values of the parameters in the dynamic model are often uncertain or even unknown, and may even vary over time. A typical case is that of a mobile robot carrying unspecified loads, and/or moving on several surfaces with different frictional conditions.

These factors call for the development of adaptive dynamic controllers to better handle unmodelled robot dynamics, as well as noise and external disturbances. Addressing of these advanced control issues of robustness and adaptation for the dynamic/kinematic control of nonholonomic mobile robots is a recent development. Some researchers opt to use pre-trained function estimators, specifically artificial neural networks (ANNs), to render nonadaptive conventional controllers more robust in the face of uncertainty (Corradini et al., 2003; Oubbati et al., 2005). These techniques require prior off-line training and remain blind to variations and situations which take place after the training phase. In contrast to the claim in (Oubbati et al., 2005) such techniques cannot be considered as adaptive because this requires real-time



Bugeja M. and Fabri S. (2006).
NEURO-ADAPTIVE DYNAMIC CONTROL FOR TRAJECTORY TRACKING OF MOBILE ROBOTS.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 404-411
DOI: 10.5220/0001218904040411

and continuous online training. To account for parametric variations in the kinematic/dynamic model, adaptive control (Fukao et al., 2000) and robust sliding mode control (Corradini and Orlando, 2001) has also been proposed. Yet another approach is that of online functional-adaptive control, where the uncertainty is not restricted to parametric terms, but covers also the dynamic functions themselves. We consider this to be more general and superior in handling higher degrees of uncertainty and unmodelled dynamics (Fierro and Lewis, 1998; de Sousa et al., 2002; Bugeja and Fabri, 2005).

In all these works, only Corradini et al. [2003] and Bugeja and Fabri [2005] develop the corresponding control algorithms in discrete-time. This is a relevant contribution because in practice robot controllers are ultimately implemented digitally on computer hardware. The control algorithm proposed by Corradini et al. [2003] is however not adaptive since it requires off-line training. It also considers solely the kinematic aspect of nonholonomic mobile robots, as does the adaptive method proposed by Bugeja and Fabri [2005]. The control scheme being proposed in this paper conforms with the discrete-time philosophy of the previous two papers, but addresses their limitations by proposing online functional adaptive control of the full dynamic model of the nonholonomic mobile robot.

This paper proposes a Gaussian radial basis function (RBF) ANN to estimate the robot's nonlinear dynamic functions, which are assumed to be completely unknown. The ANN parameters are estimated stochastically in real-time with no preliminary off-line training by employing the Kalman filter algorithm (Kalman, 1960). The estimated functions are used on a *certainty equivalence* (van de Water and Willems, 1981) basis in a discrete-time dynamic control law cascaded with a trajectory tracking kinematic controller. The stochastic nature of the ANN weight estimator is attractive for two main reasons: it provides a direct way to account for practically inevitable process and measurement noise; and it provides a direct measure of the estimation uncertainty through the Kalman covariance matrix. We plan to use the latter in the development of stochastic controllers in the nearby future to avoid certainty equivalence assumptions.

Section 2 of this paper will develop the stochastic discrete-time inverse dynamic model of the robot. This is then utilised for identification in the online stochastic estimator based on a Gaussian RBF ANN in Section 3. A discrete-time adaptive control law is proposed and incorporated with the recursive weight tuning algorithm in Section 4. The performance of the complete system is analysed and compared by realistic simulation results in Section 5, and conclusions are provided in Section 6.

# 2 PRELIMINARIES

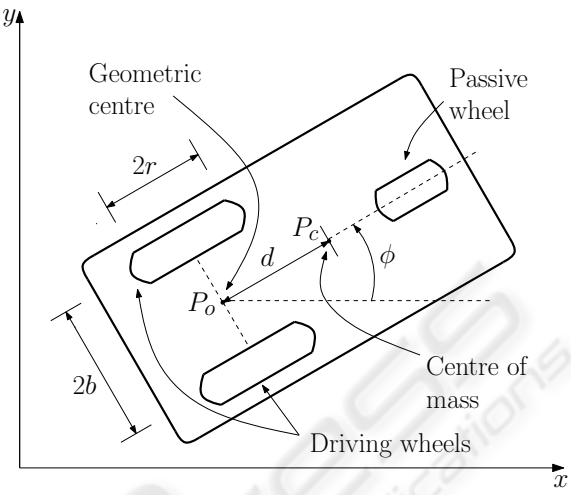


Figure 1: Differentially driven wheeled mobile robot.

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

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

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

## 2.1 Kinematics

Assuming that the driving wheels roll without slipping, the mobile platform is subject to three kinematic constraints, two of which are nonholonomic (Sarkar et al., 1994). The three kinematic constraints can be written in the form

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -\sin\phi & \cos\phi & 0 & 0 & 0 \\ \cos\phi & \sin\phi & b & -r & 0 \\ \cos\phi & \sin\phi & -b & 0 & -r \end{bmatrix}. \tag{2}$$

Furthermore, it is straightforward to verify that $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{0}$ where

$$\boldsymbol{S} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ \frac{r}{2b} & -\frac{r}{2b} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (3)$$

The kinematic state-space model of the wheeled mobile robot (WMR) in Figure 1 can now be expressed as

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q})\boldsymbol{\nu}, \quad (4)$$

where $\boldsymbol{\nu}$ represents a column vector composed of the angular velocities of the two driving wheels, specifically $\boldsymbol{\nu} = [\nu_r \;\; \nu_l]^T = \begin{bmatrix} \dot{\theta}_r & \dot{\theta}_l \end{bmatrix}^T$.

## 2.2 Dynamics

The WMR under consideration is a mechanical system with the kinematic constraints given in (1). Its dynamic equations of motion can be written in matrix form as (Sarkar et al., 1994):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}) = \boldsymbol{E}(\boldsymbol{q})\boldsymbol{\tau} - \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda}, \quad (5)$$

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

Differentiating (4) with respect to time, substituting the expression for $\ddot{\boldsymbol{q}}$ in (5), premultiplying the resulting expression by $\boldsymbol{S}^T(\boldsymbol{q})$, and noting that $\boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{A}^T(\boldsymbol{q}) = \boldsymbol{0}$ we get

$$\overline{\boldsymbol{M}}\dot{\boldsymbol{\nu}} + \overline{\boldsymbol{V}}(\dot{\boldsymbol{q}})\boldsymbol{\nu} + \overline{\boldsymbol{F}}(\dot{\boldsymbol{q}}) = \overline{\boldsymbol{E}}\boldsymbol{\tau}. \quad (6)$$

It can be shown that:

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

$$\begin{aligned} \overline{\boldsymbol{V}}(\dot{\boldsymbol{q}}) &= \begin{bmatrix} 0 & \frac{m_c r^2 d\dot{\phi}}{2b} \\ \frac{m_c r^2 d\dot{\phi}}{2b} & 0 \end{bmatrix} \quad (7) \\ \overline{\boldsymbol{F}}(\dot{\boldsymbol{q}}) &= \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{F}(\dot{\boldsymbol{q}}) \\ \overline{\boldsymbol{E}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

where $I = (I_c + m_c d^2) + 2(I_m + m_w b^2)$ and $m = m_c + 2m_w$. From (6) and (7) it is straightforward to note that the nonlinearities in the WMR dynamics can be totally attributed to $\overline{\boldsymbol{V}}(\dot{\boldsymbol{q}})$ and $\overline{\boldsymbol{F}}(\dot{\boldsymbol{q}})$, since $\overline{\boldsymbol{M}}$ is a constant matrix. Moreover, it is interesting to note that $\overline{\boldsymbol{M}}$ is invertible for all possible values of the WMR parameters and that $\overline{\boldsymbol{V}}(\dot{\boldsymbol{q}})$ is effectively a function of $\boldsymbol{\nu}$ only, since $\dot{\phi} = \frac{r}{2b}(\nu_r - \nu_l)$ as can be seen in (3) and (4).

We will now discretise the continuous-time dynamics in (6) to account for the fact that the controller is implemented on a digital computer system. Using a first order explicit forward Euler approximation with sampling interval $T$ seconds and assuming that the control input vector $\boldsymbol{\tau}$ remains constant over each sampling interval, the following discrete-time inverse dynamic model is obtained:

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{f}_{k-1} + \boldsymbol{G}\left(\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1}\right), \quad (8)$$

where the subscript integer $k$ denotes that the corresponding variable is evaluated at sample time instant $kT$ seconds. We assume that the sampling interval is chosen low enough for the Euler approximation to hold. Vector $\boldsymbol{f}_{k-1}$ and matrix $\boldsymbol{G}$ are given as:

$$\begin{aligned} \boldsymbol{f}_{k-1} &= \overline{\boldsymbol{V}}_{k-1}\boldsymbol{\nu}_{k-1} + \overline{\boldsymbol{F}}_{k-1} \\ \boldsymbol{G} &= \frac{1}{T}\overline{\boldsymbol{M}}. \quad (9) \end{aligned}$$

To account for noise, uncertainty and disturbances we introduce additively a vector of discrete white noise $\boldsymbol{\epsilon}_k$. The deterministic model in (8) is hence converted to the following nonlinear, stochastic, discrete-time inverse dynamic model of the WMR:

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{f}_{k-1} + \boldsymbol{G}\left(\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1}\right) + \boldsymbol{\epsilon}_k, \quad (10)$$

where $\boldsymbol{\epsilon}_k$ is assumed to be an independent, zero-mean, white, Gaussian-distributed process, with covariance matrix $\boldsymbol{R}$. Identification of this model is described in the next section.

# 3 GAUSSIAN RBF ESTIMATOR

Theoretically speaking, at any time instant $kT$ the vector of discrete functions $\boldsymbol{f}_k$ and the constant matrix $\boldsymbol{G}$, composing the WMR inverse dynamic model in (10), are both known and given in (9). This is true if one assumes that: there are no unmodelled dynamics, the robot parameters such as masses, inertias and frictions are perfectly known and do not change over time, and that perfect sensor measurements are available. To conform with these stringent conditions in a real life scenario is practically impossible. To address these issues we opt to use functional-adaptive control via a RBF ANN for the recursive estimation of the vector of discrete nonlinear functions $\boldsymbol{f}_k$, as if it was completely unknown. On the other hand, it is known that $\boldsymbol{G}$ is a constant and symmetrical matrix with unknown elements. Consequently its estimation does not require the use of an ANN, and so we integrate the estimation of the elements of $\boldsymbol{G}$ within the same Kalman filter algorithm used for the ANN weight training, as detailed later in Subsection 3.1.

The selection of a Gaussian RBF network (Poggio and Girosi, 1990) for the estimation of $\boldsymbol{f}_k$, was motivated by the fact that the RBF network weights appear linearly in the final state-space output equation (Fabri and Kadirkamanathan, 1998). This detail enables the use of a standard Kalman filter for weight estimation, leading to the least-squares-sense optimal tuning of the network weights. In our previous work (Bugeja and Fabri, 2005), which dealt with adaptive kinematic control of WMR, multilayer perceptron (MLP) ANN were used for the estimation process. Unlike the Gaussian basis functions in RBFs, the sigmoidal activation functions used in MLPs are not localised, implying that typically MLP networks require less neurons than RBF networks for the same degree of accuracy. However MLP networks do not posses the possible advantage of linearity, as in the case of RBF networks. As a result the former Kalman filter has to be replaced by a sub-optimal, nonlinear stochastic estimator, like the extended Kalman filter (EKF) (Maybeck, 1979), which complicates the algorithm and introduces significant approximations.

## 3.1 Formulation

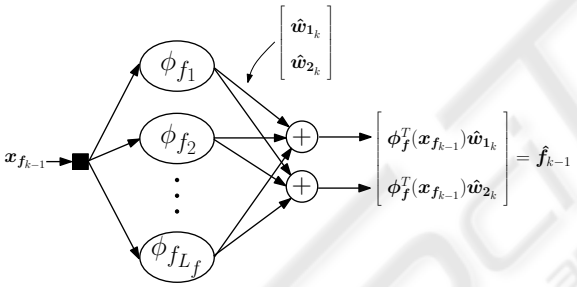


Figure 2: RBF neural network.

Consider the following definitions, in the light of the employed RBF ANN depicted in Figure 2.

- $\boldsymbol{x}_{\boldsymbol{f}_{k-1}}$ represents the neural network input vector defined as:
$$\boldsymbol{x}_{\boldsymbol{f}_{k-1}} \triangleq \boldsymbol{\nu}_{k-1}. \tag{11}$$
This state vector is assumed to be contained within the compact set $\boldsymbol{\chi}_{\boldsymbol{f}} \subset \mathbb{R}^2$.
- $\boldsymbol{\phi}_{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{f}})$ is the Gaussian RBF vector, whose $i$th element is given by
$$\phi_{f_i} = \exp\left\{ \frac{(\boldsymbol{x}_{\boldsymbol{f}} - \boldsymbol{m}_{\boldsymbol{f}_i})^T \boldsymbol{R}_{\boldsymbol{f}}^{-1} (\boldsymbol{x}_{\boldsymbol{f}} - \boldsymbol{m}_{\boldsymbol{f}_i})}{-2} \right\} \tag{12}$$
where $\boldsymbol{m}_{\boldsymbol{f}_i}$ is the coordinate vector of the center of the $i$th basis function and $\boldsymbol{R}_{\boldsymbol{f}}$ is the corresponding covariance matrix. The basis functions are placed on a regular grid within the corresponding compact set by setting the vector $\boldsymbol{m}_{\boldsymbol{f}_i}$ accordingly. Similarly the RBF covariance matrix is fixed a priori to some value corresponding to the required width of the basis functions. Sanner and Slotine [1992] show that with knowledge of the bounds on the frequency characteristics of the functions being estimated, the number of basis functions and their corresponding means and covariance matrices can be selected. Simulation results indicate that the placement and covariance of the RBFs is not critical for the overall performance of the controller.
- $L_f$ denotes the number of basis functions, and $\hat{\boldsymbol{f}}_{k-1}$ denotes the neural network output, corresponding to the estimate of $\boldsymbol{f}_{k-1}$.
- The neural network architecture depicted in Figure 2 yields the following input-output relation for the the ANN:
$$\hat{\boldsymbol{f}}_{k-1} = \begin{bmatrix} \boldsymbol{\phi}_{\boldsymbol{f}}^T(\boldsymbol{x}_{\boldsymbol{f}_{k-1}})\hat{\boldsymbol{w}}_{\mathbf{1}_k} \\ \boldsymbol{\phi}_{\boldsymbol{f}}^T(\boldsymbol{x}_{\boldsymbol{f}_{k-1}})\hat{\boldsymbol{w}}_{\mathbf{2}_k} \end{bmatrix}, \tag{13}$$
where $\hat{\boldsymbol{w}}_{\boldsymbol{i}_k}$ represents the weight vector of the connection between the RBFs and the $i$th output element of the neural network.
- As mentioned previously the estimation of $\boldsymbol{G}$ does not require the use of an ANN, since it is made up of constant elements. Additionally, it is a symmetrical matrix and this property is exploited to construct its estimate at time instant $(k-1)$ as follows:
$$\hat{\boldsymbol{G}}_{k-1} = \begin{bmatrix} \hat{g}_{1_{k-1}} & \hat{g}_{2_{k-1}} \\ \hat{g}_{2_{k-1}} & \hat{g}_{1_{k-1}} \end{bmatrix}, \tag{14}$$
where $\hat{g}_{1_{k-1}}$ and $\hat{g}_{2_{k-1}}$ represent the unknown elements of $\hat{\boldsymbol{G}}_{k-1}$.
- $\boldsymbol{\Phi}_{k-1}$, $\boldsymbol{\Gamma}_{k,k-1}$, $\hat{\boldsymbol{w}}_{\boldsymbol{f}_k}$ and $\hat{\boldsymbol{w}}_{\boldsymbol{G}_k}$ are defined as follows:
$$\boldsymbol{\Phi}_{k-1} \triangleq \begin{bmatrix} \boldsymbol{\phi}_{\boldsymbol{f}}^T & \mathbf{0}_{\boldsymbol{f}}^T \\ \mathbf{0}_{\boldsymbol{f}}^T & \boldsymbol{\phi}_{\boldsymbol{f}}^T \end{bmatrix}, \tag{15}$$
where $\mathbf{0}_{\boldsymbol{f}}$ is a zero vector having the same length as $\boldsymbol{\phi}_{\boldsymbol{f}}$, and the time index $(k-1)$ indicates that $\boldsymbol{\phi}_{\boldsymbol{f}}$ is evaluated for $\boldsymbol{x}_{\boldsymbol{f}_{k-1}}$,
$$\boldsymbol{\Gamma}_{k,k-1} \triangleq \begin{bmatrix} (\nu_{r_k} - \nu_{r_{k-1}}) & (\nu_{l_k} - \nu_{l_{k-1}}) \\ (\nu_{l_k} - \nu_{l_{k-1}}) & (\nu_{r_k} - \nu_{r_{k-1}}) \end{bmatrix} \tag{16}$$
$$\hat{\boldsymbol{w}}_{\boldsymbol{f}_k} \triangleq \begin{bmatrix} \hat{\boldsymbol{w}}_{\mathbf{1}_k}^T & \hat{\boldsymbol{w}}_{\mathbf{2}_k}^T \end{bmatrix}^T \tag{17}$$
and
$$\hat{\boldsymbol{w}}_{\boldsymbol{G}_k} \triangleq \begin{bmatrix} \hat{g}_{1_{k-1}}^T & \hat{g}_{2_{k-1}}^T \end{bmatrix}^T. \tag{18}$$

Assume that inside the compact set $\boldsymbol{\chi_f}$, the neural network approximation error is negligibly small when the weight vector $\hat{\boldsymbol{w}}_{\boldsymbol{f}_k}$ is equal to some optimal value denoted by $\boldsymbol{w}^*_{\boldsymbol{f}_k}$. This is justified in the light of the *Universal Approximation Theorem* of neural networks (Fabri and Kadirkamanathan, 2001). Moreover, let $\boldsymbol{w}^*_{\boldsymbol{G}_k}$ represent the optimal estimate for $\hat{\boldsymbol{w}}_{\boldsymbol{G}_k}$. It follows that the WMR stochastic inverse dynamic model in (10) can be rewritten, using the optimally weighted approximations $\hat{\boldsymbol{f}}_{k-1}$ and $\hat{\boldsymbol{G}}_{k-1}$ to replace $\boldsymbol{f}_{k-1}$ and $\boldsymbol{G}$ respectively. The resulting model is given by:

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{\Phi}_{k-1}\boldsymbol{w}^*_{\boldsymbol{f}_k} + \boldsymbol{\Gamma}_{k,k-1}\boldsymbol{w}^*_{\boldsymbol{G}_k} + \boldsymbol{\epsilon}_k. \quad (19)$$

Regrouping (19) in matrix form we get:

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{H}_{k,k-1}\boldsymbol{w}^*_k + \boldsymbol{\epsilon}_k, \quad (20)$$

where

$$\boldsymbol{H}_{k,k-1} \triangleq [\boldsymbol{\Phi}_{k-1} \vdots \boldsymbol{\Gamma}_{k,k-1}] \quad (21)$$

and

$$\boldsymbol{w}^*_k \triangleq [{\boldsymbol{w}^*_{\boldsymbol{f}_k}}^T \vdots {\boldsymbol{w}^*_{\boldsymbol{G}_k}}^T]^T. \quad (22)$$

It is proper to note that $\boldsymbol{w}^*_k$ is completely unknown and need to be estimated. However (20) clearly indicates that the optimal weight vector $\boldsymbol{w}^*_k$ appears linearly in the output equation. This feature is exploited in the recursive weight tuning process detailed in the following section.

# 4 ADAPTIVE CONTROL

## 4.1 Online Training

Having derived the stochastic model (20), for the WMR inverse dynamics that is dependent on the RBF network through $\boldsymbol{H}_{k,k-1}$, and the optimal weight vector $\boldsymbol{w}^*_k$, it is straightforward to rewrite it in the following linear state-space form to be used within the Kalman filter algorithm, in *predictive mode*, to enable the recursive estimation of the optimal weight vector:

$$\begin{aligned} \boldsymbol{w}^*_{k+1} &= \boldsymbol{w}^*_k \\ \boldsymbol{\tau}_{k-1} &= \boldsymbol{H}_{k,k-1}\boldsymbol{w}^*_k + \boldsymbol{\epsilon}_k, \end{aligned} \quad (23)$$

It is assumed that:

- The initial optimal parameter vector $\boldsymbol{w}^*_0$ is Gaussian with mean $\bar{\boldsymbol{w}}$ and covariance $\boldsymbol{V}$,
- $\boldsymbol{\epsilon}_k$ and $\boldsymbol{w}^*_0$ are independent.

From Kalman filter theory (Kalman, 1960; Maybeck, 1979), it follows that the following recursive weight adjustment rules can be employed for the estimation of the optimal weight vector as follows:

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{H}_{k,k-1}\boldsymbol{P}_k \quad (24)$$

$$\hat{\boldsymbol{w}}_{k+1} = \hat{\boldsymbol{w}}_k + \boldsymbol{K}_k\boldsymbol{i}_k \quad (25)$$

$$\hat{\boldsymbol{w}}_k \triangleq [\hat{\boldsymbol{w}}^T_{\boldsymbol{f}_k} \vdots \hat{\boldsymbol{w}}^T_{\boldsymbol{G}_k}]^T, \quad (26)$$

where $\hat{\boldsymbol{w}}_k$ is the recursive estimate to the optimal weight vector $\boldsymbol{w}^*_k$, $\boldsymbol{K}_k$ is the Kalman gain matrix given by:

$$\boldsymbol{K}_k = \boldsymbol{P}_k\boldsymbol{H}^T_{k,k-1}[\boldsymbol{H}_{k,k-1}\boldsymbol{P}_k\boldsymbol{H}^T_{k,k-1} + \boldsymbol{R}]^{-1}, \quad (27)$$

and $\boldsymbol{i}_k$ is the innovations vector given by:

$$\boldsymbol{i}_k = \boldsymbol{\tau}_{k-1} - \boldsymbol{H}_{k,k-1}\hat{\boldsymbol{w}}_k. \quad (28)$$

$\boldsymbol{P}_k$ denotes the prediction covariance matrix and represents a measure of belief in the estimate $\hat{\boldsymbol{w}}_k$.

The initial estimates for the weight vector $\hat{\boldsymbol{w}}_0$ and the prediction covariance $\boldsymbol{P}_0$ are set to some arbitrarily values. Note that high initial covariance values are the most appropriate since we assume no preliminary knowledge of $\hat{\boldsymbol{w}}_0$. For each estimate of $\hat{\boldsymbol{w}}_{k+1}$, the corresponding estimates $\hat{\boldsymbol{f}}_k$ and $\hat{\boldsymbol{G}}_k$ are computed using (13), (14) and the associated relations formulated in Subsection 3.1. These estimates are then used recursively in the discrete-time dynamic control law detailed in the following subsection. In contrast to (Oubbati et al., 2005) this feature renders the overall control scheme truly adaptive, since the dynamic WMR functions used for control, are estimated in real-time and require no preliminary off-line training.

Note that the extra computational burden that comes along with the Kalman filter pays off in several ways. Primarily, it renders the recursive weight tuning algorithm *optimal* under the specified assumptions. Secondly, the conditional probability density of the weight vector $\hat{\boldsymbol{w}}_{k+1}$ is being updated in real-time. The resulting information, most importantly the covariance matrix $\boldsymbol{P}_{k+1}$, is essential in the design of stochastic control laws (Fabri and Kadirkamanathan, 2001), which are part of our plan for future work.

## 4.2 Control Design

The complete adaptive control system proposed in this paper for the trajectory tracking of WMR is depicted in Figure 3. Some variables in this figure are defined later in the following paragraphs. At this point one should particularly note the cascade approach which divides the control problem in two main tasks namely, kinematic control and dynamic control. The kinematic controller ensures that the robot tracks a reference trajectory, which can be generated in real-time (one time-step ahead) by a path-planning algorithm. To achieve this the kinematic controller issues a velocity command that serves as the reference signal for the adaptive dynamic controller. The latter computes a control torque by employing the estimated dynamic functions, obtained from the ANN stochastic estimator, in order to ensure that the robot velocity

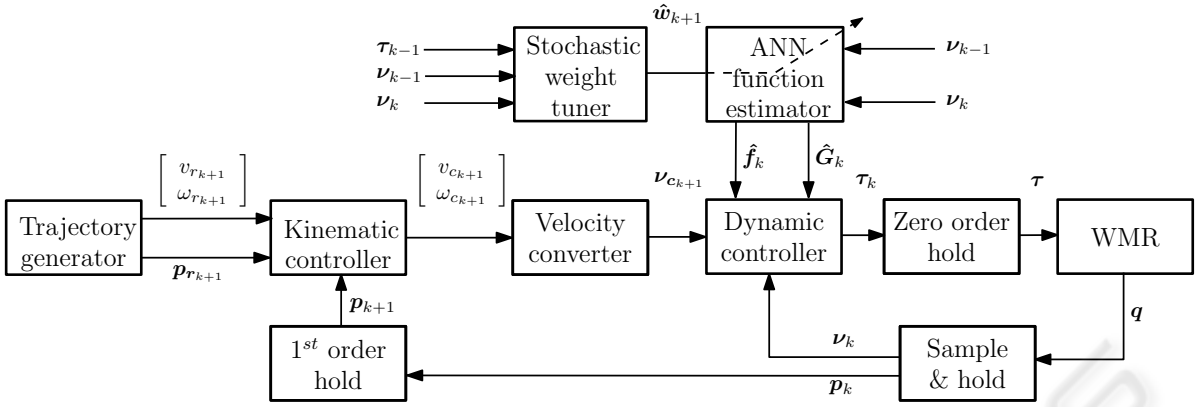


Figure 3: Complete adaptive scheme.

vector really tracks the velocity desired by the kinematic controller. These two sub-controllers are detailed next.

A very simple, yet useful representation of the trajectory tracking problem, is through the concept of the *virtual vehicle* (Crowley, 1989). Basically, the time dependent trajectory to be tracked by the WMR is designated by a *non-stationary* virtual vehicle having the same nonholonomic constraints as the real robot. The controller aims for the real vehicle to track the virtual vehicle at all times, in both posture and velocity. The discrete-time tracking error vector $\boldsymbol{e}_k = [e_{1_k} \;\; e_{2_k} \;\; e_{3_k}]^T$ is defined as:

$$\boldsymbol{e}_k \triangleq \begin{bmatrix} \cos\phi_k & \sin\phi_k & 0 \\ -\sin\phi_k & \cos\phi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} (\boldsymbol{p}_{\boldsymbol{r}_k} - \boldsymbol{p}_k), \qquad (29)$$

where $\boldsymbol{p}_{\boldsymbol{r}_k} = [x_{r_k} \;\; y_{r_k} \;\; \phi_{r_k}]^T$ denotes the virtual vehicle sampled pose vector. In trajectory tracking the aim is to make $\boldsymbol{e}$ converge to zero so that $\boldsymbol{p}$ converges to $\boldsymbol{p}_{\boldsymbol{r}}$. For this task we propose a discrete-time version of the continuous-time kinematic controller originally presented in (Kanayama et al., 1990) given by:

$$\boldsymbol{\nu}_{\boldsymbol{c}_k} = \begin{bmatrix} v_{r_k}\cos e_{3_k} + k_1 e_{1_k} \\ \omega_{r_k} + k_2 v_{r_k} e_{2_k} + k_3 v_{r_k} \sin e_{3_k} \end{bmatrix}, \qquad (30)$$

where $\boldsymbol{\nu}_{\boldsymbol{c}_k}$ is the wheel velocity command vector requested by the kinematic controller, $(k_1, k_2, k_3) > 0$ are design parameters, and $v_{r_k}$, $\omega_{r_k}$ are the translational and angular velocities respectively corresponding to the virtual vehicle. The latter are directly related to the virtual vehicle wheel velocity vector $\boldsymbol{\nu}_{\boldsymbol{r}_k}$ as follows

$$\boldsymbol{\nu}_{\boldsymbol{r}_k} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v_{r_k} \\ \omega_{r_k} \end{bmatrix}. \qquad (31)$$

If we ignore the dynamic effects and consider only the kinematic model (4) of the WMR and assume perfect velocity tracking, then (30) completely solves the trajectory tracking problem. However if a dynamic controller is to be implemented and assuming no uncertainty in $\boldsymbol{f}_k$ and $\boldsymbol{G}$, one can use the following dynamic control law to ensure that the WMR velocity vector $\boldsymbol{\nu}_k$ really tracks $\boldsymbol{\nu}_{\boldsymbol{c}_k}$

$$\boldsymbol{\tau}_k = \boldsymbol{f}_k + \boldsymbol{G}_k \left( \boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} - \boldsymbol{\nu}_k + k_d \left\{ \boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k \right\} \right), \qquad (32)$$

where $-1 < k_d < 1$ is a design parameter. Note that the control law requires the velocity command vector $\boldsymbol{\nu}_{\boldsymbol{c}}$ to be known one sampling interval ahead. For this reason we advance the kinematic control stage by one sampling interval. This is achieved by: generating the reference trajectory vectors corresponding to the $(k+1)$ instant, and using a first order hold to estimate $\boldsymbol{p}_{k+1}$ from $\boldsymbol{p}_k$. The latter is a valid approximation since any decent sampling rate (in the order of milli seconds) is still high enough compared to the relatively slow dynamics of a mobile robot. Substituting this control law in the stochastic dynamic model in (10) yields the following closed loop dynamics:

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} + k_d \left( \boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k \right) - \boldsymbol{\epsilon}_{k+1}. \qquad (33)$$

Equation (33) clearly indicates that $|\boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k| \rightarrow \boldsymbol{\epsilon}_k$ as $k \rightarrow \infty$. It is interesting to note that the case with $k_d = 0$, corresponds to the *deadbeat* control concept associated with digital control systems. When the dynamics are completely unknown and/or prone to change, we propose to replace the control law (32) with its heuristic certainty equivalent, utilising the estimates $\hat{\boldsymbol{f}}_k$ and $\hat{\boldsymbol{G}}_k$ from the ANN, leading to the adaptive control law:

$$\boldsymbol{\tau}_k = \hat{\boldsymbol{f}}_k + \hat{\boldsymbol{G}}_k \left( \boldsymbol{\nu}_{\boldsymbol{c}_{k+1}} - \boldsymbol{\nu}_k + k_d \left\{ \boldsymbol{\nu}_{\boldsymbol{c}_k} - \boldsymbol{\nu}_k \right\} \right). \qquad (34)$$

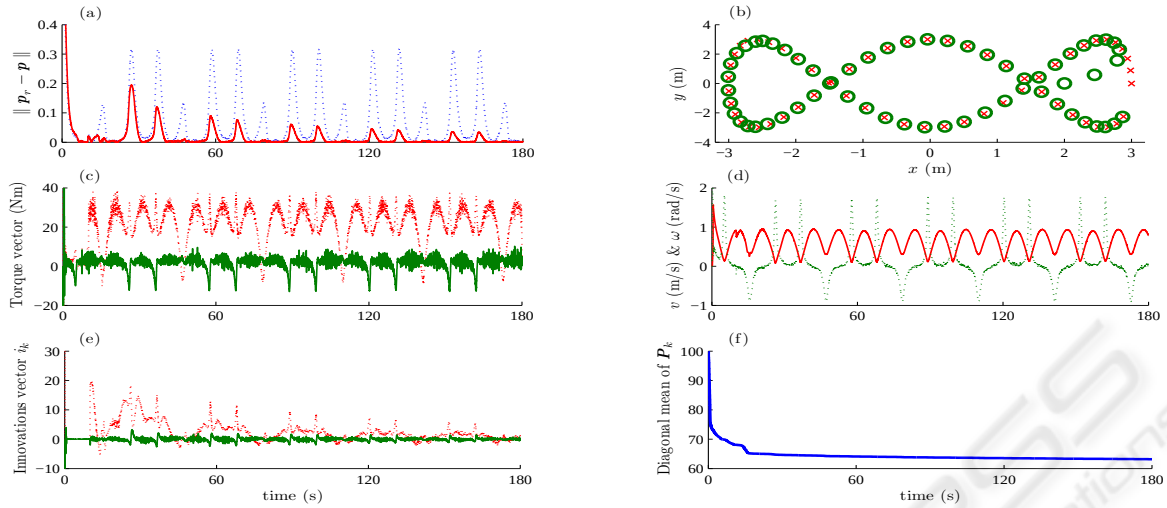


Figure 4: (a): adaptive (solid) & nonadaptive (dashed); (b): reference ($\times$) & actual ($\circ$) trajectories (first 60 seconds); (c): right wheel (dashed) & left wheel (solid) torques; (d): linear (solid) & angular (dashed) velocities; (e): innovations vector $\boldsymbol{i}_k$; (f): diagonal mean of $\boldsymbol{P}_k$. Plots (b) to (f) correspond to the proposed adaptive controller.

## 5 SIMULATION RESULTS

The proposed neuro-adaptive dynamic controller (34) was verified and compared to its non-adaptive version (32) by highly realistic simulations. The mobile robot was simulated through the continuous-time, full kinematic/dynamic model given in (4) and (6) with the following parameters: $b$=0.5m, $r$=0.15m, $d$=0.2m, $m_c$=30kg, $m_w$=1kg, $I_c$=15.625kgm$^2$, $I_w$=0.005kgm$^2$, $I_m$=0.0025kgm$^2$. We included viscous friction by setting $\boldsymbol{F}(\dot{\boldsymbol{q}}) = \boldsymbol{F_c}\dot{\boldsymbol{q}}$ in (7), where $\boldsymbol{F_c}$ is a diagonal matrix of coefficients whose diagonal was set to $[2, 2, 5, 0.5, 0.5]$. Sampling interval $T = 50ms$ and the sampled data was corrupted with noise $\boldsymbol{\epsilon}_k$. To make the simulations more realistic we imposed hard limits on the wheel torques ($\pm$50Nm) to simulate actuator saturation. Neural network pre-training is not used and to demonstrate further the adaptive feature of the proposed controller, the model used for simulations is abruptly modified at $t = 10$s by introducing an additional load of 15kg placed 0.4m away from $P_o$. Moreover, at the same instant we simulate a fault in the right wheel by increasing its viscous friction coefficient by a factor of ten. The reference trajectory used for simulations is very demanding since it is characterised by high velocities and accelerations. The neural network contained 49 RBFs with covariance matrix $\boldsymbol{R_f} = 100\boldsymbol{I_2}$, where $\boldsymbol{I_i}$ denotes an $(i \times i)$ identity matrix. The Kalman filter initial covariance $\boldsymbol{P}_0 = 100\boldsymbol{I_\alpha}$, where $\boldsymbol{\alpha}$ is the total number of weights. The initial weight vector $\hat{\boldsymbol{w}}_0$ was generated from a normal distribution with zero mean. It took a standard desktop computer with no code optimisation merely 1 minute to simulate 3 minutes of real-time. Clearly, this indicates that the proposed algorithm is not computationally demanding.

A number of results for this particular simulation are presented in Figure 4. Plot (a) shows clearly the difference in performance between the adaptive and nonadaptive controllers. In both cases the Euclidian vector norm of the pose error diminishes quickly from some non-zero initial value. However from $t = 10$s onwards, corresponding to the aforementioned model disruption, the adaptive controller reacts to the change and after a transitional period manages to reduce the pose error vector back to normal; the ANN has adapted to the new dynamic functions. As expected this does not happen in the nonadaptive case and the pose error remains relatively higher $\forall t > 10$s. The actual trajectory in the $xy$ plane is superimposed on the reference trajectory in plot (b), which also verifies the good tracking performance of the proposed scheme. Plot (c) depicts the wheel torque vector $\boldsymbol{\tau}$ and in (d) one should particularly note the high values of the translational $v$ and angular $\omega$ velocities of the WMR. Plot (e) shows how the Kalman innovation vector abruptly increases at 10s due to the model disruption and diminishes back with time as the ANN weights adapt to the new situation. Plot (f) depicts the mean of the diagonal of the covariance matrix $\boldsymbol{P}_k$ in time. The latter provides a measure of the uncertainty in the estimated weights, as generated recursively by the Kalman filter. As expected, the uncertainty decreases with time, indicating that the Kalman filter is stable and the neural network estimations are continuously improving. The slope of this curve is related to the learning rate of the ANN adaptive scheme.

# 6 CONCLUSIONS

In this paper an ANN functional-adaptive dynamic control scheme for the trajectory tracking problem of mobile robots is proposed. The resulting algorithm requires no preliminary information about the nonlinear functions in the dynamics and uses a RBF neural network, trained online in consideration of noise, uncertainty and disturbances by using a Kalman filter. The designed scheme was tested successfully by realistic simulations for several noise conditions and parameter variations. The adaptive controller showed improved performance when compared to the non-adaptive case in the face of parameter uncertainty. Future research will investigate the use of parameter resetting in the estimator and the development of dual stochastic nonlinear control laws (Fabri and Kadirkamanathan, 2001) which would take into account the inaccuracy of the ANN approximations, giving rise to better transient performance. We anticipate to get satisfactory experimental results once the proposed algorithm is implemented on a real mobile robot in the near future.

# REFERENCES


Brockett, R. W. (1983). *Asymptotic Stability and Feedback Stabilisation*. Differential Geometric Control Theory. Birkhäuser, Boston, MA.

Bugeja, M. K. and Fabri, S. G. (2005). Multilayer perceptron functional adaptive control for trajectory tracking of wheeled mobile robots. In *Proc. 2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO2005)*, volume 3, pages 66–72, Barcelona, Spain.

Canudas de Wit, C., Khennoul, H., Samson, C., and Sordalen, O. J. (1993). Nonlinear control design for mobile robots. In Zheng, Y. F., editor, *Recent Trends in Mobile Robots*, Robotics and Automated Systems, chapter 5, pages 121–156. World Scientific.

Corradini, M. L., Ippoliti, G., and Longhi, S. (2003). Neural networks based control of mobile robots: Development and experimental validation. *Journal of Robotic Systems*, 20(10):587–600.

Corradini, M. L. and Orlando, G. (2001). Robust tracking control of mobile robots in the presence of uncertainties in the dynamic model. *Journal of Robotic Systems*, 18(6):317–323.

Crowley, J. L. (1989). Asynchronous control of orientation and displacement in a robot vehicle. In *Proc. of the 1989 IEEE International Conference on Robotics and Automation (Vol. 3)*, pages 1277–1282, Scottsdale, AZ.

de Sousa, C., Hemerly, E. M., and Galvao, R. K. H. (2002). Adaptive control for mobile robot using wavelet networks. *IEEE Transactions on Systems, Man and Cybernetics*, 32(4):493–504.

Ding, D. and Cooper, R. A. (2005). Electric-powered wheelchairs. *IEEE Control Systems Magazine*, 25(2):22–34.

Fabri, S. G. and Kadirkamanathan, V. (1998). Dual adaptive control of nonlinear stochastic systems using neural networks. *Automatica*, 34(2):245–253.

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

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

Fierro, R. and Lewis, F. L. (1998). Control of a nonholonomic mobile robot using neural networks. *IEEE Trans. Neural Networks*, 9(4):589–600.

Fukao, T., Nakagawa, H., and Adachi, N. (2000). Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 16(5):609–615.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.*, 82:34–45.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. In *Proc. IEEE International Conference of Robotics and Automation*, pages 384–389, Cincinnati, OH.

Kolmanovsky, I. and McClamroch, N. H. (1995). Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 15(6):20–36.

Lamiraux, F., Laumond, J. P., VanGeem, C., Boutonnet, D., and Raust, G. (2005). Trailer truck trajectory optimization: the transportation of components for the Airbus A380. *IEEE Robotics and Automation Magazine*, 12(1):14–21.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, volume 141-1 of *Mathematics in Science and Engineering*. Academic Press Inc., London, UK.

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

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proc. IEEE*, 78(9):1481–1497.

Sarkar, N., Yun, X., and Kumar, V. (1994). Control of mechanical systems with rolling constraints: Applications to dynamic control of mobile robots. *International Journal of Robotics Research*, 13(1):55–69.

van de Water, H. and Willems, J. C. (1981). The certainty equivalence property in stochastic control theory. *IEEE Transactions on Automatic Control*, AC-26(5):1080–1087.