NONHOLONOMIC WHEELED MOBILE ROBOT CONTROL

Karam, Carlo B. **Khairallah, Nadim F.** **Rayess, Ralph C.**

ABSTRACT: The Kobuki Base is a fully embedded, research-oriented platform, which allows implementing and testing various control schemes on a nonholonomic differential drive configuration. After a comprehensive review of the previous promising control methods implemented on similar configurations, we develop a complete model of our specific system (both kinematics and dynamics), and propose a design for an adaptive controller. This controller aims to stabilize the robot along a pre-defined trajectory under dynamical parametric uncertainty. The controller is simulated Simulink and exhibits a good tracking performance.

1. INTRODUCTION

The Kobuki mobile base is a low-cost research platform designed to advance control theory related research, in particular nonlinear control. It is a wheeled mobile robot equipped with two differentially driven wheels, and two castor wheels for support. With continuous, fully integrated operation in mind, the platform provides external power output for a main computer and other possible actuators, and sports a fully embedded system equipped with a coded motherboard and factory calibrated odometry [1].

As with most commercially available mobile robots, the Kobuki platform does not allow direct motor voltage control, but is instead equipped with a low level PID which tracks input reference velocities. This allows for a higher-level control through ROS (Robot Operating System). As students of an adaptive control class, we look to apply the concepts of MRAC to a linearized model of the robot dynamics, combining that with a kinematic controller benefiting from gain scheduling, in the goal of successfully tracking a predefined trajectory.

Many other trajectory tracking control methods have been previously considered, achieving good performance, and will be discussed in the following section.

2. Literature Review

With the integration of robotics and autonomy into the different aspects of life, the motion control of wheeled mobile robot has gained a lot of traction in the academic field. As such, trajectory tracking of mobile robots has been the subject of a significant number of research papers and projects, which present different control methods ranging from simple output feedback to adaptive and intelligent control. We are able to split the available literature into two broad categories: that presenting control schemes solely based on the kinematic model of the robot (and thus assuming perfect tracking) [1,2], and that presenting control methods which allow for dynamic compensation [3,4,5]. In both cases, the most widely used method was the dynamic state feedback linearization [2,3,4,5].

As wheeled mobile robots are subjected to both holonomic and nonholonomic constraints and exhibit highly nonlinear dynamics, it has been shown that these platforms are not properly stabilizable by standard linear controllers [2], and are neither

input-state feedback linearizable nor static state input-output linearizable [7]. One instance of the use of dynamic state feedback linearization for a kinematic controller is in [2], where a kinematic controller based on the widely used unicycle kinematics model was developed. Through dynamic state feedback, the implemented controller was able to successfully stabilize a mobile robot along a trajectory and a setpoint, as opposed to other less involved methods such as linear and direct nonlinear control. In [3], the kinematic model of a TurtleBot2 (equipped with a Kobuki base) is developed in polar coordinates, and a nonlinear kinematic controller which provides stability according to a Lyupanov function is designed. The controller is then implemented both in a simulation environment using Gazebo, and on the physical robot, and its trajectory tracking performance is evaluated. Through odometry output feedback, the controller successfully stabilizes the platform along the predefined trajectories with minimal error.

One paper that offers a dynamic compensation technique is [4], where a complete dynamic model of a unicycle-like (widely used model) mobile robot was developed. That model was then subjected to a linear parametrization, and eventually input-output linearized. The robot and model were then integrated in a multi-robot formation, and further centralized control techniques were applied. The developed controller successfully drives the formation tracking error to 0. Another work [5] presented a Model Reference Adaptive Control (MRAC) scheme for mobile robot, combining a neural network-based adaptive controller and a dynamic feedback linearization method based on a nominal model. After deriving the adaptive dynamics controller through radial basis function neural networks, its parameters were adjusted via a Lyupanov-based stability law. The resulting controller achieved good tracking performance with a minimal computational effort. Similarly, [6] presents an adaptive nonlinear controller for a nonholonomic robot, based on input-output linearization technique. The controller performance was then simulated and compared to that of a nonadaptive controller, proving adequate trajectory tracking under parametric uncertainty. Similar methods are also employed in [8,9], where a nonlinear kinematic controller with a dynamic adaptive controller is developed with respect to motor voltage and torque input, a method which is not suitable for our application as we are required to directly control the robot velocities through ROS.

3. Dynamic Model

Kinematic Model

The simplest available model for nonholonomic wheeled mobile robots (WMR) is the unicycle model (figure 1), representing an upright wheel rolling on a plane), with generalized coordinate vector defined by $q = [x \quad y \quad \varphi]^T$.

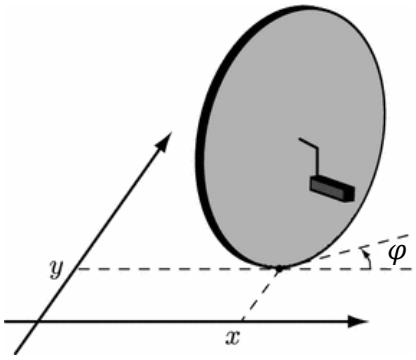

Figure 1. Unicycle Kinematic Diagram

These coordinates are in a 3 degrees of freedom space, but the wheel is constrained with respect to the lateral direction, and therefore cannot slip. This gives the constraint known as the rolling without slipping constraint given by:

$$A(q)\,\dot{q} = \dot{x}\,\sin\varphi - \dot{y}\cos\varphi \qquad (3.1)$$

The above equation represents a Pfaffian nonholonomic system with $n - m = 1$ nonintegrable constraint.

We can express the set of feasible velocities as a linear combination of the vectors spanning the null space of the matrix $A(q)$ as follows:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = g_1(q)v + g_2(q)\omega = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (3.2)$$

This is known as the 1st order kinematic model, and we can notice that the WMR is underactuated in the sense that its controllable states are less than its degrees of freedom (the velocity direction is restricted).

Dynamic Model

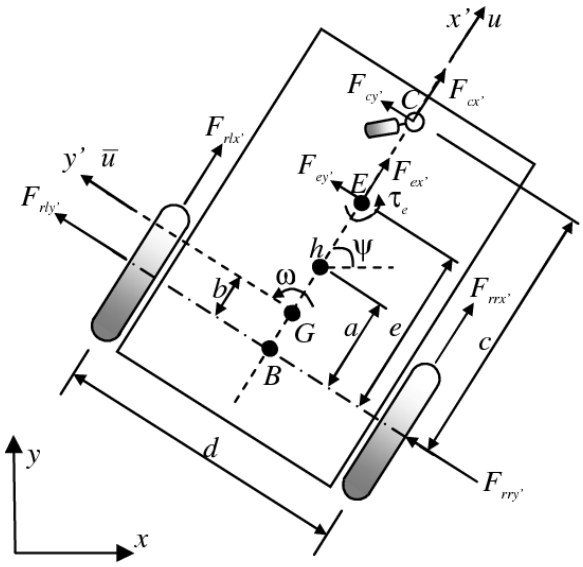

Figure 2. Dynamic Model Parameters of WMR

In figure 2, *G* is the center of mass of the WMR, *B* is the center of the wheel baseline, *h* is the point that is required to track the trajectory, *C* is the location of the castor wheel and *E* is the location of the tool. *x'Gy'* is the local coordinate axis attached to the center of mass *G* with u and $\bar{u}$ being the longitudinal and lateral velocities of *G* respectively. ω is the angular velocity of the WMR and Ψ is its heading angle.

a, *b*, *c*, *d* and *e* are distances between *B* and *h*, *B* and *G*, *B* and *C*, between the two wheels and between *B* and *E* respectively. The lateral and longitudinal forces acting on the right, left and castor wheels in addition to the forces and torque at the tool location *E* are also represented in figure 2.

To derive the dynamic model of the WMR, the forces in the *x'* and *y'* directions and the moments about the z-axis were summed and equated to the corresponding accelerations in the following equations of motion:

$$\sum F_{x'} = F_{rlx'} + F_{rrx'} + F_{ex'} + F_{cx'} = m(\dot{u} - \bar{u}\omega) \quad (3.3)$$

$$\sum F_{y'} = F_{rly'} + F_{rry'} + F_{ey'} + F_{cy'} = m(\dot{\bar{u}} + u\omega) \quad (3.4)$$

$$\sum M_z = \frac{d}{2}(F_{rrx'} - F_{rlx'}) - b(F_{rly'} + F_{rry'}) + (e - b)F_{ey'} + (c - b)F_{cy'} + \tau_e = I_z\dot{\omega} \quad (3.5)$$

Where *m* is the robot mass and I_z is the robot inertia about the z-axis

Then, neglecting the motor inductances, the right and left motor torques multiplied by the gear ratio between the motors and the wheels (τ_r and τ_l respectively) are related to the right and left wheels angular velocities (ω_r and ω_l respectively) through:

$$\tau_r = \frac{k_a(v_r - k_b\omega_r)}{R_a} \quad (3.6)$$

$$\tau_l = \frac{k_a(v_l - k_b\omega_l)}{R_a} \quad (3.7)$$

Where k_a and k_b are the motor torque constant multiplied by the gear ratio and the motor voltage constant multiplied by the gear ratio respectively, v_r and v_l are the right and left motor input voltages respectively, and R_a is the electric resistance of the motor.

The following motor to wheel dynamics are obtained from the equations of motion about the right and left wheels axis:

$$I_e \dot{\omega}_r + B_e w_r = \tau_r - F_{rrx} R_t \quad (3.8)$$

$$I_e \dot{\omega}_l + B_e w_l = \tau_l - F_{rlx} R_t \quad (3.9)$$

Where I_e is the moment of inertia of the wheel (about its axis), R_t is the nominal radius of the tire and B_e is the viscous friction coefficient of the motor.

The control of commercial WMR being usually done by sending reference linear and angular velocities since motor voltages are not accessible, a PID controller ensures these reference velocities are being tracked. This controller is modelled below while neglecting the integral action because of the small value of the integral gain:

$$\begin{bmatrix} V_T \\ V_R \end{bmatrix} = \begin{bmatrix} k_{PT}(u_{ref} - u) - k_{DT}(\dot{u}_{ref} - \dot{u}) \\ k_{PR}(\omega_{ref} - \omega) - k_{DR}(\dot{\omega}_{ref} - \dot{\omega}) \end{bmatrix} \quad (3.10)$$

Where V_T and V_R are the translational and rotational voltages respectively, k_{PT} and k_{PR} are the proportional control gains for the translation and rotation respectively, and k_{DT} and k_{DR} are derivative control gains for the translation and rotation respectively.

The model dynamics relating the linear and angular acceleration of the WMR to the reference linear and angular velocities is then obtained by combining 3.10 while neglecting $\dot{u}_{ref}$ and $\dot{\omega}_{ref}$ to simplify the model:

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\theta_3^0}{\theta_1^0}\omega^2 - \frac{\theta_4^0}{\theta_1^0}v \\ -\frac{\theta_5^0}{\theta_2^0}v\omega - \frac{\theta_6^0}{\theta_2^0}\omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\theta_1^0} & 0 \\ 0 & \frac{1}{\theta_2^0} \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix}$$

Where the model parameters θ^0 are given by the following formulas:

$$\theta_1^0 = \left(\frac{R_a}{k_a}(mR_t r + 2I_e) + 2rk_{DT}\right)/(2rk_{PT}) \quad (3.11)$$

$$\theta_2^0 = \left(\frac{R_a}{k_a}(I_e d^2 + 2R_t r(I_z + mb^2)) + 2rdk_{DR}\right)/(2rdk_{PR}) \quad (3.12)$$

$$\theta_3^0 = \frac{R_a}{k_a} mbR_t/(2k_{PT}) \quad (3.13)$$

$$\theta_4^0 = \frac{\frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right)}{rk_{PT}} + 1 \quad (3.14)$$

$$\theta_5^0 = \frac{R_a}{k_a} mbR_t/(dk_{PR}) \quad (3.15)$$

$$\theta_6^0 = \frac{\frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right)d}{2rk_{PR}} + 1 \quad (3.16)$$

Where r is the wheel radius

The terms defining the above coefficient represent the various phyiscal parameters of the robot as such:

The dynamical model is linearized by computing and evaluating the Jacobian at the equilibrium points in a first step demonstrated below:

$$\dot{x} = f(x, \theta^0) + g(\theta^0)\, u_{ref} \quad (3.17)$$

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\theta_3^0}{\theta_1^0}\omega^2 - \frac{\theta_4^0}{\theta_1^0}v \\ -\frac{\theta_5^0}{\theta_2^0}v\omega - \frac{\theta_6^0}{\theta_2^0}\omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\theta_1^0} & 0 \\ 0 & \frac{1}{\theta_2^0} \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix}$$

Solving $\dot{x} = f(x, \theta^0) = 0$, we find two pairs of equilibrium points:

$$x_1 = \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} -\frac{\theta_6^0}{\theta_2^0} \\ \sqrt{-\frac{\theta_6^0 \theta_4^0}{\theta_5^0 \theta_3^0}} \end{bmatrix}$$

$$x_2 = \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} -\frac{\theta_6^0}{\theta_2^0} \\ -\sqrt{-\frac{\theta_6^0 \theta_4^0}{\theta_5^0 \theta_3^0}} \end{bmatrix}$$

The Jacobian $J(x, \theta^0)$ of $f(x, \theta^0)$ is given by:

$$J(x, \theta^0) = \begin{bmatrix} -\frac{\theta_4^0}{\theta_1^0} & 2\frac{\theta_3^0}{\theta_1^0}\omega \\ -\frac{\theta_5^0}{\theta_2^0}\omega & -\frac{\theta_5^0}{\theta_2^0}v - \frac{\theta_6^0}{\theta_2^0} \end{bmatrix} \quad (3.18)$$

Replacing the equilibrium points in the Jacobian matrix, we get the following linearized dynamics:

$$\dot{x} = Ax + Bu_{ref},$$

where $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$, and $B = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix}$

with $a_1, a_2, a_3, a_4, b_1 \; and \; b_2$ being functions of $J(x, \theta^0)$ and equilibrium points found previously.

B is taken to be a diagonal matrix since v_{ref} acts as input for $\dot{v}$ and ω_{ref} acts as input for $\dot{\omega}$ only (i.e. $g(\theta^0)$ is diagonal as there is no cross-coupling between inputs and derivatives of states).

4. Controller Design

Being agile, maneuverable and low-cost platforms, WMRs are used to perform a wide variety of tasks, most of the time in environments with changing conditions: change of robot mass during load transportation, change of center of gravity and inertia of platform when tool installed on robot operates, … These varying robot parameters influence the WMR's dynamics and consequently result in the violation of the perfect velocity tracking assumption.

To address this critical issue, a dynamic controller based on adaptive control schemes is designed to guarantee kinematic velocity tracking and ultimately path following by compensating for the varying dynamics.

The proposed controller block diagram is shown in figure 3.

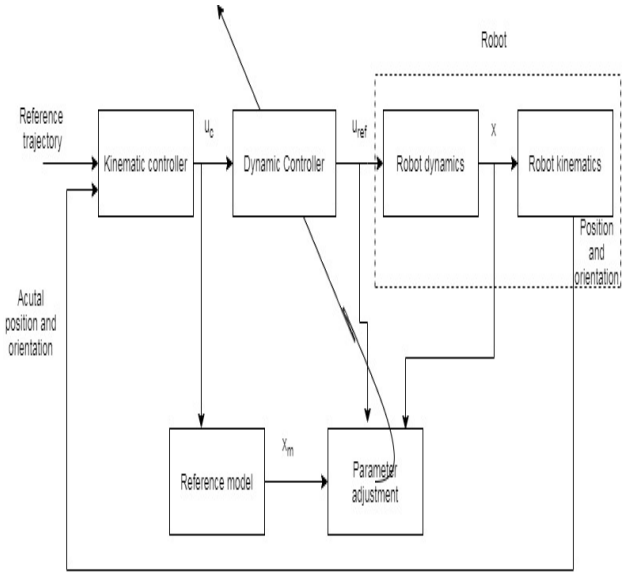

Figure 3. Block Diagram of Proposed Controller

Kinematic Controller

Given a cartesian motion for the unicycle, we are required to generate a reference trajectory defined by $q_r = [x_r(t) \quad y_r(t) \quad \varphi_r(t)]^T$ and control inputs $[u_{r1} \quad u_{r2}] = [v_r(t) \quad \omega_r(t)]$. Naturally, this trajectory needs to be compliant with the nonholonomic constraints of the robot, which is why a heading angle φ_r is imposed.

A simple trajectory tracking controller can be developed based on a tangent linearization along our reference trajectory. Following [10], we can define the state tracking error e as follows

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Rotation Matrix } R(\varphi)} \begin{bmatrix} x_r - x \\ y_r - y \\ \varphi_r - \varphi \end{bmatrix} \quad (4.1)$$

Using the nonlinear velocity input transformation defined by

$$\begin{aligned} v &= v_r \cos e_3 - u_{r1} \\ \omega &= \omega_r - u_{r2} \end{aligned} \quad (4.2)$$

we obtain the below error dynamics:

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e3 \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

We can then linearize those dynamics about the reference trajectory, to obtain

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.3)$$

Looking at the above linear time variant system, we notice that its controllability matrix defined by

$$\begin{aligned} C &= [B \quad AB \quad A^2B] \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_r^2 & v_r\omega_r \\ 0 & 0 & -\omega_r & v_r & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

is of full rank (3) for constant nonzero reference velocities v_r and ω_r, which also makes the system linear time invariant, and thus stabilizable along a trajectory using linear feedback. We can define a "linear" feedback law:

$$\begin{aligned} u_1 &= -k_1 e_1 \\ u_2 &= -k_2 u_{r1} e_2 - k_3 e_3 \end{aligned}$$

The gains are to be set so that the closed loop system's roots follow the characteristic polynomial defined by

$$(s + 2\zeta a)(s^2 + 2\zeta a s + a^2)$$

As such, the corresponding gains are given by

$$k_1 = k_3 = 2\zeta a \qquad \text{and} \qquad k_2 = \frac{a^2 - w_r^2}{|v_r|}$$

However, it is clear that for small reference velocities tending to 0, the gain k_2 tends to infinity and thus our controller will have to exhibit an infinite control effort to achieve the desired tracking performance. We can thus let the closed loop poles depend on the values of the references v_r and ω_r, by letting $a = \sqrt{\omega_r^2 + b v_r^2}$ where $b > 0$ and executing the below gain scheduling scheme:

$$k_1 = k_3 = 2\zeta\sqrt{\omega_r^2 + b v_r^2}$$

$$k_2 = b|v_r|$$

It is clear that the above controller generates valid gains for all reference velocities, and outputs no control effort when these velocities are zero, which makes sense intuitevely. For our particular controller, we have chosen $\zeta = 0.707$ and $b = 10$.

Dynamic Controller

The kinematic controller designed in the previous section guarantees asymptotic trajectory tracking when the perfect velocity tracking assumption is valid (i.e. $v = v_r$ and $\omega = \omega_r$).

This dynamic controller consists of a full state feedback MRAC algorithm designed based on the linearized dynamic model presented in section 3. We thus want our system,

$$\dot{x} = Ax + Bu_{ref}$$

to track the reference model

$$\dot{x}_m = A_m x_m + B_m u_c,$$

where,

$$x_m = \begin{bmatrix} v_m \\ \omega_m \end{bmatrix}, \qquad A_m = \begin{bmatrix} -a_{m1} & 0 \\ 0 & -a_{m2} \end{bmatrix},$$

$$u_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \qquad B_m = \begin{bmatrix} b_{m1} & 0 \\ 0 & b_{m2} \end{bmatrix}$$

with $a_{m1} > 0 \; and \; a_{m2} > 0$ so that A_m is Hurwitz

Again B_m is chosen to be diagonal for the same reason in section 3 above for B.

Also, A_m is chosen to be diagonal so that each state follows a first order differential equation:

$$\begin{aligned} \dot{v}_m &= -a_{m1} v_m + b_{m1} v_c \\ \dot{\omega}_m &= -a_{m2} \omega_m + b_{m2} \omega_c \end{aligned} \tag{4.4}$$

We start off by defining the controller structure, a 2-DOF controller with feedforward and state feedback terms

$$u_{ref} = \mathrm{M} u_c - \mathrm{L}x \,,$$

where $M = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix}$ and $L = \begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix}$ are the controller parameters to be adapted.

We then define θ_c to be the controller parameters to be updated by the MRAC algorithm as

$$\theta_c = [m_1 \; m_2 \; m_3 \; m_4 \; l_1 \; l_2 \; l_3 \; l_4 \;]^T$$

The closed loop system thus becomes:

$$\dot{x} = (A - BL)x + BMu_c,$$

where $A_{\mathrm{CL}} = \mathrm{A} - \mathrm{BL}$ and $B_{CL} = \mathrm{BM}$.

For perfect model following, we require that $A_{\mathrm{CL}} = A_m$ and $B_{CL} = B_m$, and obtain the following ideal controller parameters ${\theta_c}^0$:

$${\theta_c}^0 = \left[{m_1}^0 \; {m_2}^0 \; {m_3}^0 \; {m_4}^0 \; {l_1}^0 \; {l_2}^0 \; {l_3}^0 \; {l_4}^0 \right]^T$$

The parameters are defined as:

$${m_1}^0 = \frac{b_{m1}}{b_1}, \qquad {m_2}^0 = 0,$$

$${m_3}^0 = 0, \qquad {m_4}^0 = \frac{b_{m2}}{b_2},$$

$${l_1}^0 = \frac{a_1 + a_{m1}}{b_1}, \qquad {l_2}^0 = \frac{a_2}{b_1},$$

$${l_3}^0 = \frac{a_3}{b_2}, \qquad {l_4}^0 = \frac{a_4 + a_{m2}}{b_2}$$

We define the controller parameters error as

$$\tilde{\theta}_c = \theta_c - {\theta^0}_c \tag{4.5}$$

and the reference state tracking error as

$$e = x - x_m \tag{4.6}$$

yielding the following error dynamics

$$\dot{e} = \dot{x} - \dot{x}_m \tag{4.7}$$

After some manipulation of equations 4.5 through 4.7, we get:

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_m \\ &= A_m e + (A_{CL} - A_m)x \\ &\qquad + (B_{CL} - B_m)u_c \\ &= A_m e + \Psi \tilde{\theta}_c \end{aligned} \tag{4.8}$$

with Ψ defined as

$$\Psi^{\mathrm{T}} = \begin{bmatrix} b_1 v_c & 0 \\ b_1 \omega_c & 0 \\ 0 & b_2 v_c \\ 0 & b_2 \omega_c \\ -b_1 v & 0 \\ -b_1 \omega & 0 \\ 0 & -b_2 v \\ 0 & -b_2 \omega \end{bmatrix}$$

We can now define a Lyupanov function

$$V = \frac{1}{2}(e^T P e + {\tilde{\theta}_c}^T \Gamma^{-1} \tilde{\theta}_c),$$

where $\Gamma \succ 0$ and where P is the solution to the Lyapunov equation ${A_m}^T P + P A_m = -Q$ ($\mathrm{Q} \succ 0$), to obtain the following adaptation

law to keep the time derivation of V negative semi-definite:

$$\dot{\tilde{\theta}}_c = -\Gamma \Psi^T P e$$

This adaptation law ensures that $\dot{V} = -e^T Q e \leq 0$ since $Q \succ 0$

By applying Barbalat's lemma to the fact that $\dot{V} \leq 0$ with e, $\tilde{\theta}_c$ and $\ddot{V}$ bounded (which implies that $\dot{V}$ is uniformly continuous), we guarantee that $e \to 0 \; as \; t \to \infty$:

We have achieved perfect state tracking with $x \to x_m \; as \; t \to \infty$.

In the above adaptation law, the regressor Ψ contains unknown model parameters b_1 and b_2 with true values $b_1 = \frac{1}{\theta_1^0}$ and $b_2 = \frac{1}{\theta_2^0}$. So, in order to form this regressor and update the controller parameters according to the above adaptation law, a model parameter estimation scheme is designed to estimate θ_1 and θ_2 (inverse of b_1 and b_2 respectively) via a Recursive Least Square (RLS) algorithm.

To perform RLS, the dynamics equations are transformed to a linear regression model for model parameter estimation as such:

$$\begin{bmatrix} \dot{v} & 0 & -\omega^2 & v & 0 & 0 \\ 0 & \dot{\omega} & 0 & 0 & v\omega & \omega \end{bmatrix} \theta_m = \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix}$$

Or in a more compact notation as

$$\phi^T \theta_m = u_{ref}$$

Moreover, the elements of the regressor matrix ϕ^T and u_{ref} were filtered by a low-pass filter to prevent noise resulting from derivatives. With the both estimation algorithms designed above, successful model parameters estimation results in successful controller parameters estimation which in turn leads to the computation of the control law:

$$u_{ref} = \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix} = \mathrm{M}u_c - \mathrm{L}x$$

$$= \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} - \begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

This control law guarantees perfect velocity tracking (i.e. $v = v_c \; \& \; \omega = \omega_c$) and thus path following through an adaptive dynamic compensation.

5. Simulation Results

The controller was implemented and simulated on Simulink in order to evaluate its tracking performance. Choosing a circular reference trajectory defined by $x_r(t) = R\cos t$ and $y_r(t) = R\sin t$, where the radius R was set to be 2, we ran two different simulations. The first run was a basic, standard simulation, while the second was done for the purpose of evaluating the controller's disturbance rejection capabilities.

In the first simulation, the robot trajectory perfectly tracks the desired circular trajectory (figure 4). However, although the robot and the reference model trajectories have the same shape, both trajectories do not coincide with each other. This can be explained by the fact that the robot velocities track the reference model velocities through the full state feedback MRAC. This results in the same trajectory shape but does not necessarily translate in superposed trajectories because of the initial transient velocities tracking phase which results in different positions before both trajectories repeat the desired circular shape. The heading angle of the robot is delayed in comparison to that of the reference (figure 5). This delay is relatively small and is not a problem since the robot trajectory tracks the desire path.The linear and angular velocities tracking between the actual robot and the reference model is achieved through the full state feedback MRAC controller after an initial transient phase (for time less than 3 seconds) (figures 6 and 7). Moreover, since the reference model tracks the desired reference

trajectory velocities, and the robot velocities track the reference model velocities, the robot velocities will converge to the trajectory velocities which will ensure perfect path following. The MRAC algorithm is designed to ensure this goal even in the case of varying dynamics which are dealt with by adaptive compensation. The control inputs (linear and angular velocities) are very smooth after the initial bump needed to redirect the robot on the desired trajectory (figures 8 and 9). Linear velocity reference input attains 2 m/s at steady state while angular velocity reference input reaches 1 rad/s at steady state. These robot velocities input indeed converge to the reference velocities needed to track the circular trajectory

The model parameters' evolution is smooth in time and they converge quickly to a set of values different than their actual values (figure 10). This is explained by the fact that the input signals are not persistently exciting (PE) and are not rich and informative enough to excite the dynamics of the system in a way that requires the estimates model parameters to converge to their true values. The controller parameters' evolution is also smooth in time (figure 11), and these parameters converge quickly to their steady state values which are not their exact values derived for perfect model following. This is explained by the fact that the model parameters do not converge to their true values (input not PE) and in consequence the controller parameters do not converge to their true values since model parameters appear in the controller parameter adaptation algorithm. Nevertheless, the model parameters converge to a set of admissible values (minimizing the least square error) which in turn leads the controller parameter to converge to a set of values that ensures perfect path following.

In the second run, the disturbances consist of step disturbances added at the input channels (i.e. on v_{ref} and ω_{ref}) of magnitude 0.5 (25% of v_{ref} and 50% of ω_{ref} steady state values) introduced at 15 seconds and 10 seconds for v_{ref} and ω_{ref} respectively. With the introduction of disturbances, the reference model trajectory (figure 12) is not exactly the same for every iteration around the defined circular trajectory (green circles not superposed). However, the robot trajectory successfully tracks the desired circular shape and the path following performance is not deteriorated by the disturbances, which indicates robustness and successful rejection. Velocity tracking between the actual robot and the reference model is still achieved in the case of disturbances due to the MRAC full state feedback (figures 14 and 15). The disturbances, visualized through small bumps in the curves, are rejected quickly. Additionally, since the reference model tracks the reference velocities required for path following, the robot velocities converge to the required values for path following (2 m/s and 1 rad/s). The control efforts (figures 16 and 17) do not change significantly compared to the case without disturbances. Velocity inputs are smooth and converge to the same steady state values needed to ensure path following but with little bumps at the time of introduction of disturbances.

Model parameters evolution in time shows big deviations at the time of introduction of disturbances (figure 18). This can be explained by the variation of the inputs which excite the system and lead to these abruptly changing parameters. Model parameters do not converge to their true values in spite of perfect trajectory tracking due to the fact that the input is not PE. The controller parameters' evolution in time is still smooth (similar to that of the case without disturbances) but with small bumps at 10 and 15 seconds caused by the introduction of disturbances (figure 19). Also, the controller parameters' steady state values are comparable to those without disturbances.

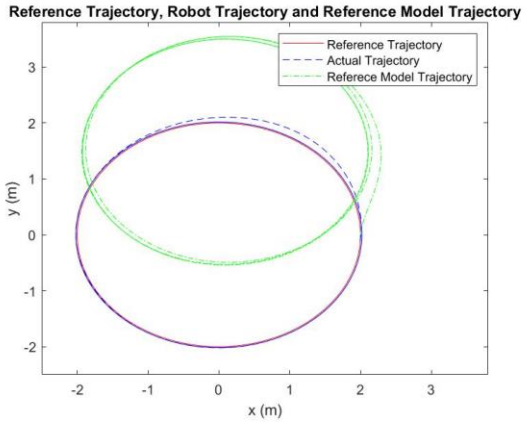

Figure 4. X and Y coordinate Tracking Performance

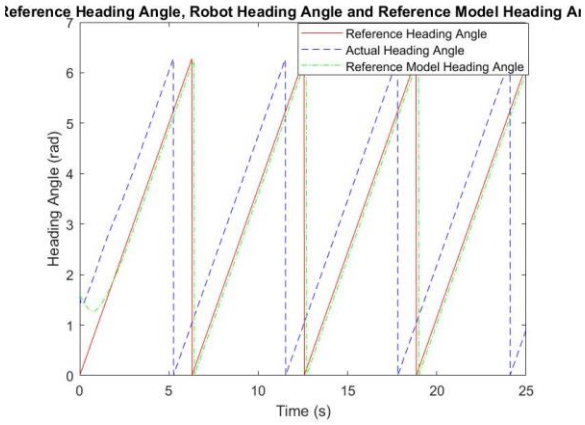

Figure 5. Heading Angle Tracking Performance

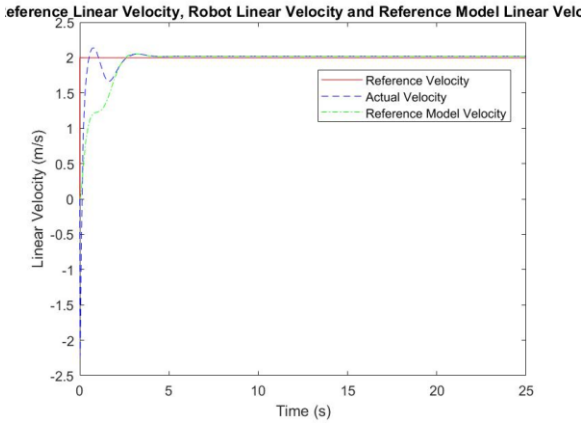

Figure 6. Comparison of reference, actual and model velocities

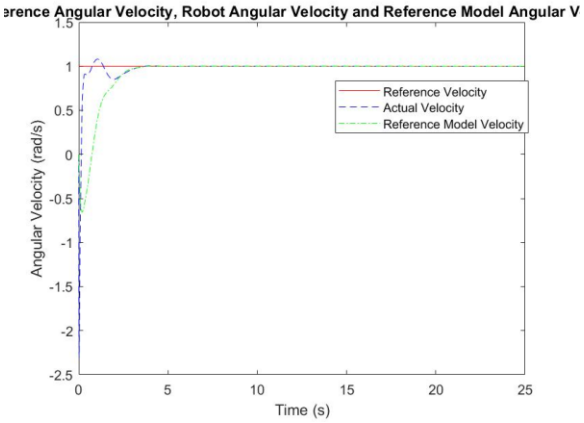

Figure 7. Comparison of reference, actual, and model angular velocities

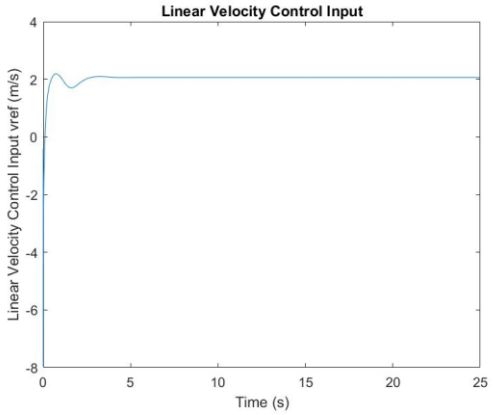

Figure 8. Linear Velocity Control Effort

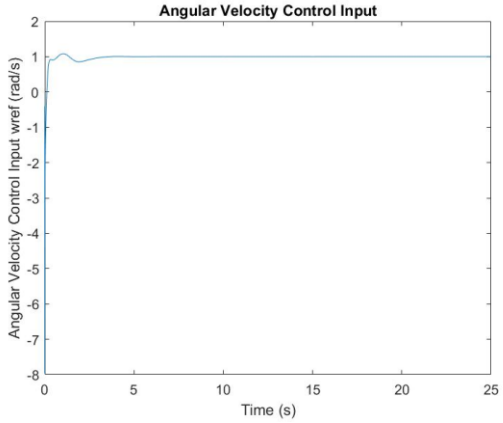

Figure 9. Angular Velocity Control Effort

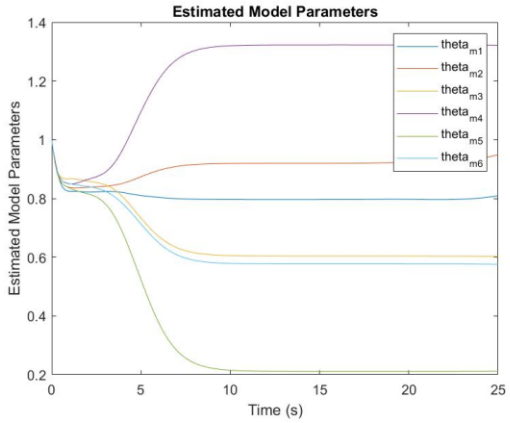

Figure 10. Model Parameter Estimation

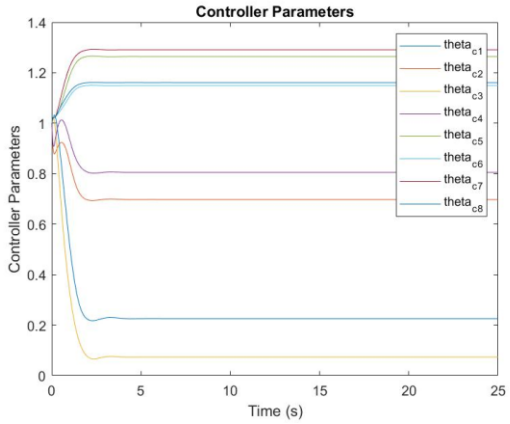

Figure 11. Controller Parameter Estimation

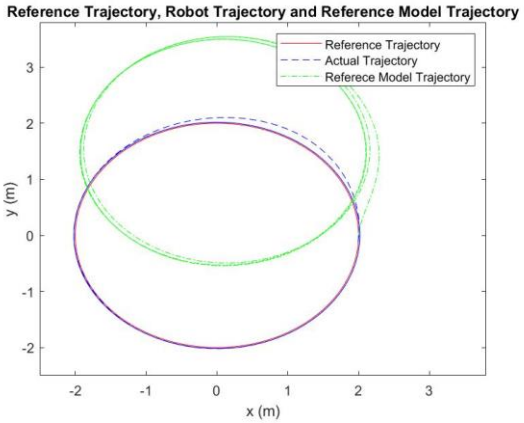

Figure 12. X and Y Coordinate Tracking Performance with disturbance

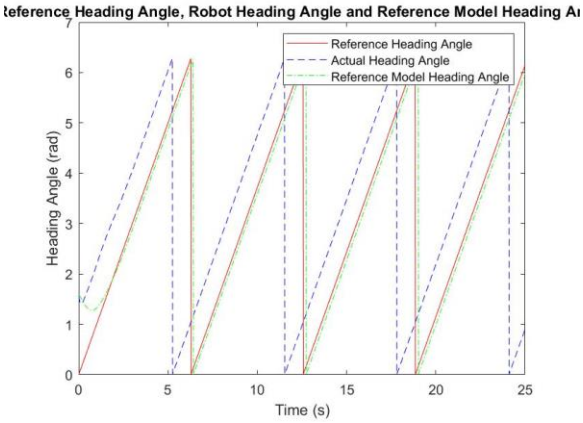

Figure 13. Heading Angle Tracking Performance with disturbance

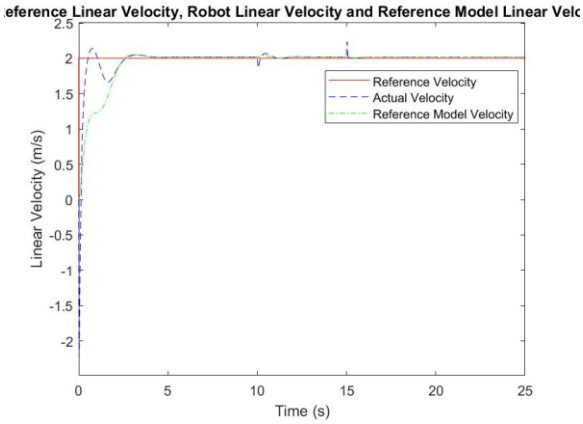

Figure 14. Comparison of reference, actual, and model linear velocities (with disturbance)

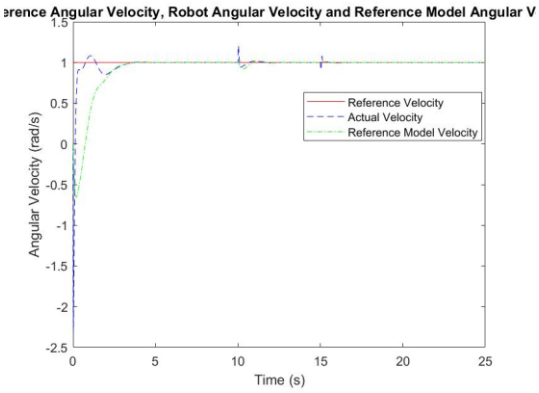

Figure 15. Comparison of reference, actual, and model angular velocities (with disturbance)

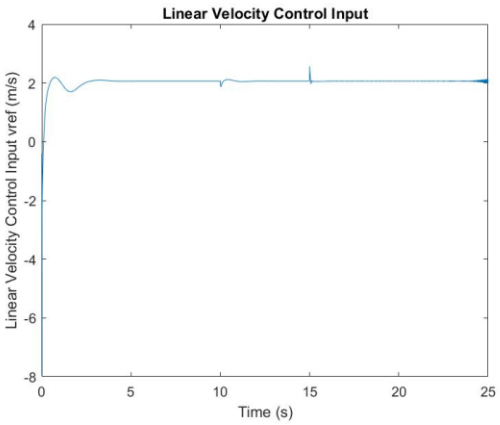

Figure 16. Linear Velocity Control Effort (with disturbance)

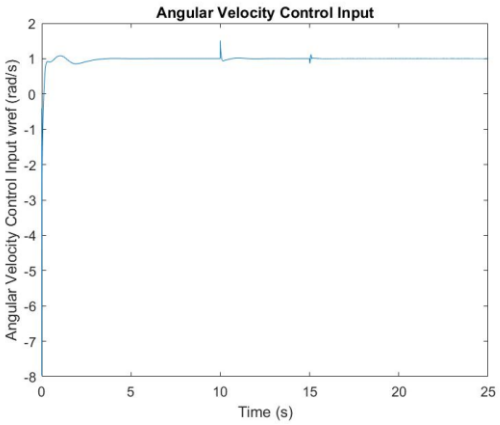

Figure 17. Angular Velocity Control Effort (with disturbance)

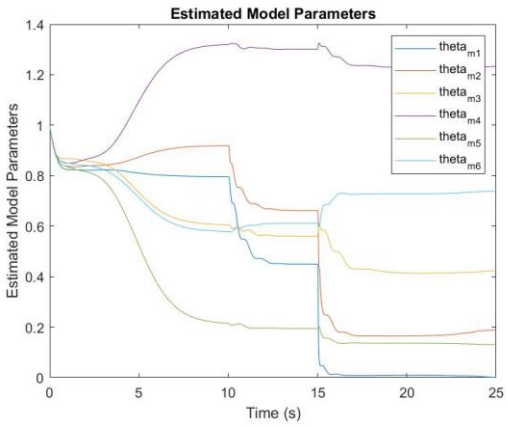

Figure 18. Model Parameter Estimation (with disturbance)

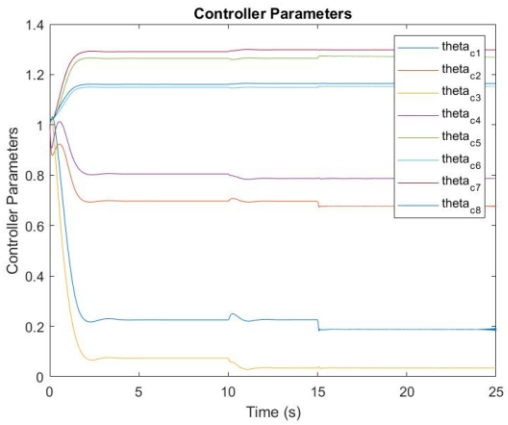

Figure 19. Controller Parameter Estimation (with disturbance)

6. Conclusion

We have then shown, through simulations, that the adaptive controller successfully tracks our reference trajectory with minimal error, while also rejecting possibl disturbances. Such a controller can then be implemented on WMR with dynamical parametric uncertainty (due to lack of available accurate information), and where it might be subject to parameter variation (in an industrial context for example where a robot is expected to carry and deliver cargo).

Due to time constraints, we were not able to benchmark our controller with a nonadaptive one to show the advantages of adaptive control techniques on dynamical systems. We also explored the dynamic linearizaing feedback technique widely used in nonlinear control, but, once again due to time constraints and our unfamiliarity with the concepts, we were not able to successfully complete the design process and simulate its performance.

7. References

[1] Dabit Industries, “Kobuki Documentation Release 2.0,” Oct. 2017

[2] G. Oriolo, A. De Luca, and M. Vendiletti, “WMR control via dynamic feedback linearization: design, implementation, and experimental validation,” *IEEE Transactions on Control Systems Technology*, November 2002, vol. 10, no. 6, pp. 835-852

[3] C. Bensaci, Y. Zennir, D. Pomorski, “Nonlinear Control of a differential wheeled mobile robot in real time-Turtlebot2”, *International Conference on Advanced Technologies and Electrical Engineering*, Skikda: 2018, pp.12 - 14.

[4] C. De La Cruz and R. Carelli, "Dynamic Modeling and Centralized Formation Control of Mobile Robots," *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, Paris, 2006, pp. 3880-3885.

[5] F. G. Rossomando, C. Soria, H. D. Patino, and R. Carelli, “Model reference adaptive control for mobile robots in trajectory tracking using radial basis function neural networks,” in *Latin American applied research*, April 2011, vol. 41, pp.177-182 · April 2011

[6] X. Li, Z. Wang, J. Zhu and Q. Chen, "Adaptive tracking control for wheeled mobile robots with unknown skidding," 2015 IEEE Conference on Control Applications (CCA), Sydney, NSW, 2015, pp. 1674-1679.

[7] X. Yun and Y. Yamamoto, “On feedback linearization of mobile robots,” 1992

[8] K. Shojaei, A. Mohammad Shahri, A. Tarakameh, and B. Tabibian, “Adaptive trajectory tracking control of a differential drive wheeled mobile robot,” in *Robotica*, 2011

[9] R. Barzamini, A. R. Yazdizadeh and A. H. Rahmani, "A New Adaptive Tracking Control For Wheeled Mobile Robot," *2006 IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok: 2006, pp. 1-6.

[10] C. Canudas deWit, H. Khennouf, C. Samson, and O. J. Sørdalen, “Nonlinear control design for mobile robots,” in *Recent Trends in Mobile Robots*, Y. F. Zheng, Ed. Singapore: World Scientific, 1993, vol. 11, pp. 121–156.