# Models of a wheeled robot named Robudem and design of a state feedback controller for its posture tracking

Jean-Claude Habumuremyi
Department Of Mechanical Engineering (MSTA), Royal Military Academy,
08 Hobbema Str., 1000, Brussels, Belgium
Jean-Claude.Habumuremyi@rma.ac.be

**ABSTRACT**

Wheeled robots are still frequently used for industrial applications and for research such as in remote surveillance and security, humanitarian demining, planetary exploration, rescue missions…The Robudem is a wheeled robot that will be tested for such tasks. In this paper, we have studied possible kinematics models of this robot based on the pure rolling and non-slipping constraints of the wheels on the ground. Then we have studied the way to control the posture of the robot when local trajectories (an important behavior for mobile robots) are specified. The Robudem robot is a nonholonomic system and the design of its controller is a major research topic in the theory on nonlinear control systems. This paper describes the use of one method named "the dynamic extension algorithm" to control the tracking posture of the Robudem robot.

## 1. Introduction

Wheeled robots are mainly used in industry and in research…principally due to their engineering simplicity (easy to build and to control than legged robots) and their low specific resistance (high energy efficiency). It is in the purpose of outdoor research in intelligent navigation that the Robotic Lab at the Royal Military Academy of Belgium has purchased the Robucar TT named Robudem (Figure 1) from the Robosoft Company. This paper deals with the kinematics models and the feedback control of this robot.

Figure 1: Robudem Robot

The models and the design of controllers of Wheeled Mobile Robot have been addressed in many papers.

In [1], P. F. Muir and C. P. Neuman established the methodology for modeling, analyzing, designing and controlling wheeled mobile robots. Their approach is parallel to the methodology applied on stationary manipulators. They extended it to take into account appropriate characteristics of wheeled mobile robots as the multiple closed-link chains, the contact between a wheel and a planar surface, and nonactuated and nonsensed wheel degrees-of-freedom. Their methodology allows first to model the kinematics of each wheel which constitutes the robot using the Sheth-Uicker convention (equivalent to the Denavit-Hartenberg convention used for the kinematics model of stationary manipulators) to define the transformation matrices. Then they amalgamate the information of individual wheel to obtain the kinematics model of the robot. From the established kinematics model, they designed the controllers and they also derived the conditions of pure rolling. If the pure rolling was not satisfied, they applied the least-squares fit to obtain the rolling.

J.C. Alexander and J.H. Maddocks [2] follow the same methodology as above to model Wheeled Mobile Robot. The major difference with the formalism developed by Muir and Neuman is that their analysis of cases in which pure rolling fails is based on physical models of friction. They describe the slippage which results from the wheels configuration using Coulomb's Law to model friction. The application of the 2 methods quoted above increases the complexity in the modeling due to the fact that in contrast to robotic manipulators, wheeled robots are nonholonomic systems. The pure rolling and the non-slipping constraints make the kinematics and dynamics analyses more complicated with these methods. Others methods much appropriate in the analyses of wheeled mobile robots have been developed.

In [3], G. Campion, G. Bastin and B. D'Andrea-Novel give a general and unifying method to model a wheeled robot by taking into account the restriction to robot mobility induced by the constraints. They introduce the concept of degree of mobility and of degree of steerability and they show that all wheeled mobile robots can be classified into 5 classes. In [4], they investigate how point and tracking problems for wheeled mobile robots can be solved by state feedback linearization. In this paper, we apply the method in [3] on bicycle models to derive known kinematics models in literature of the car-like vehicles and unknown models of our particular robot named Robudem. We have followed this procedure because the configuration of the Robudem robot is a very special case of car-like vehicles. We also consider, in this paper, the practical design of a feedback controller on such kind of robots using dynamic extension algorithm as described in general case in [5]. This controller will try to follow defined local trajectories from sensors in the purpose for example to avoid an obstacle, to reach a goal…

## 2. Kinematics model

### 2.1 Generality

To establish the kinematics model of wheeled robots, we accept the presence of nonholonomic constraints due to the rolling without slipping condition between the wheels and the ground. We analyze the case of fixed and steering wheels (Figure 2) which constitute the robots that we consider. The position of the wheel is characterized

by 3 constants: $\alpha$ , *l, r* and its motion with respect to the robot by 2 time-varying angles $\phi(t)$ and $\varphi(t)$ (or the speed $V(t) = r\varphi(t)$ ). $\phi$ is a constant in the case of the fixed wheels. We also defined the base frame with origin at 0 and the moving frame with origin P. The robot posture can be described in terms of the two coordinates *x, y* of the origin P of the moving frame and by the orientation angle $\theta$ of the moving frame, both with respect to the base frame. With these descriptions, the 2 following constraints can be deduced [3, 5]:

*Pure rolling constraint:* $\cos(\alpha+\phi+\theta)\dot{x} + \sin(\alpha+\phi+\theta)\dot{y} + l\sin\phi\dot{\theta} - V = 0$ (1)

*Non-slipping constraint:* $\sin(\alpha+\phi+\theta)\dot{x} - \cos(\alpha+\phi+\theta)\dot{y} - l\cos\phi\dot{\theta} = 0$ (2)

## 2.2 Car-Like Vehicle

About the car-like vehicle, we can distinguish 2 configurations:

- The rear wheels are driving (Figure 3): the engine motor by means of a differential system drives rear wheels. The front wheels are steering by mean of an Ackerman mechanism.
- And the front wheels are driving (Figure 4): the engine motor drives and steers the front wheels using a differential system and an Ackerman mechanism.

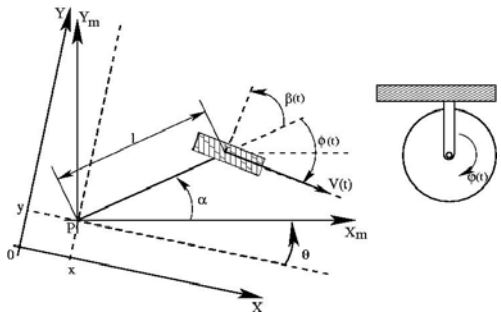


Figure 2: Fixed and steering wheels

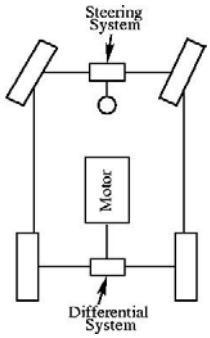


Figure 3: The rear wheels driving vehicle

The kinematics models of the car-like vehicles can be established from a bicycle model. We can indeed easily find the following equations from the imaginary wheels (Figure 5) placed at the middle of the front and rear axles:

$V_{2i} = V_2\dfrac{\cos\phi_2}{\cos\phi_{2i}}$ (3), $V_{2e} = V_2\dfrac{\cos\phi_2}{\cos\phi_{2e}}$ (4), $V_{1i} = V_1\left(1-\dfrac{C}{2R}\right)$ (5), $V_{1e} = V_1\left(1+\dfrac{C}{2R}\right)$ (6)

and $\cot g\phi_2 = \cot g\phi_{2e} - \dfrac{C}{2L} = \cot g\phi_{2i} + \dfrac{C}{2L}$ (7) (Ackerman condition).

### 2.2.1 Rear wheels driving model

We can establish the rear wheels driving model by considering the bicycle model shown on Figure 6. The two wheels are imaginary ones which are at the middle axles of the rear wheels driving vehicle (Figure 3). The relations of the speeds and the angles of the real wheels with imaginary wheels are obtained from equations (3), (4), (5), (6) and (7).

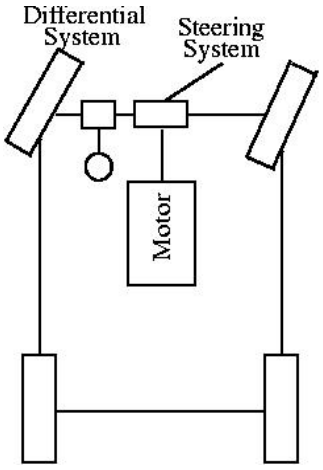


Figure 4: The front wheels driving vehicle

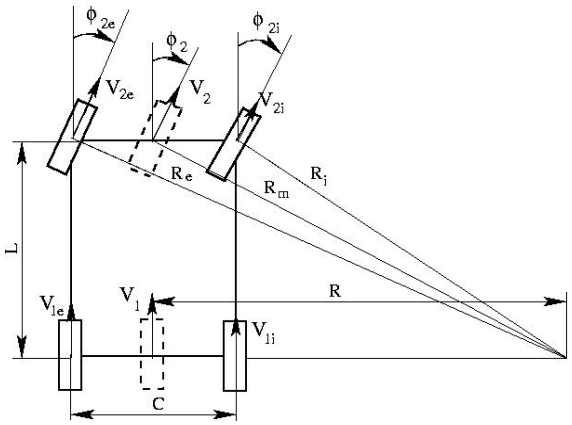


Figure 5: Car-Like and bicycle models

We can establish for the two wheels the following equations by mean of equations (1) and (2):

*Pure rolling constraint of the wheel 1:* $\cos\theta.\dot{x} + \sin\theta.\dot{y} = V_1$ (8)

*Non-slipping constraint of the wheel 1:* $-\sin\theta.\dot{x} + \cos\theta.\dot{y} = 0$ (9)

*Non-slipping constraint of the wheel 2:* $-\sin(\theta+\phi_2).\dot{x} + \cos(\theta+\phi_2).\dot{y} + L\cos\phi_2.\dot{\theta} = 0$ (10)

By solving equations (8), (9) and (10), we have:

$$\dot{x} = V_1.\cos(\theta) \quad (11)$$

$$\dot{y} = V_1.\sin(\theta) \quad (12)$$

$$\dot{\theta} = V_1.\frac{tg\phi_2}{L} \quad (13)$$

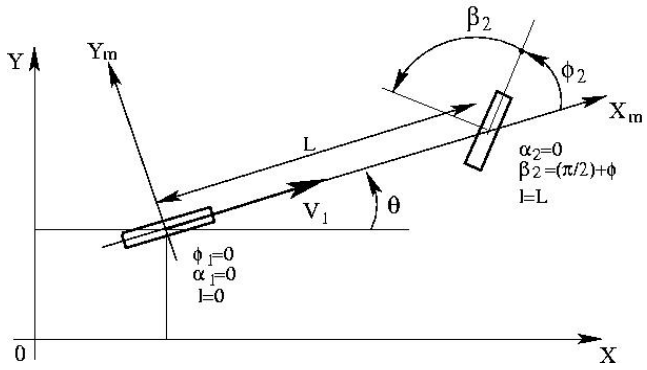


Figure 6: Bicycle model: rear wheels driving model

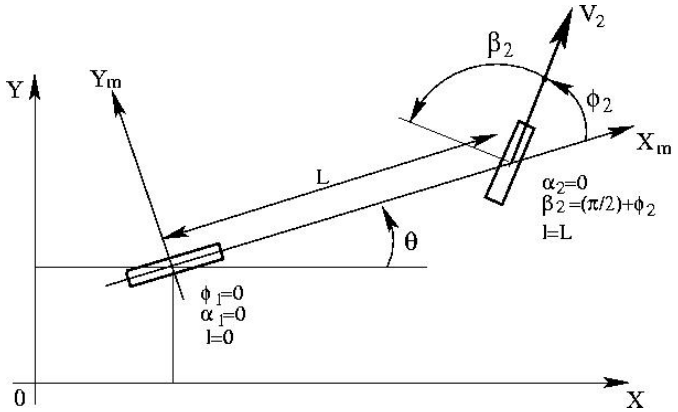


Figure 7: Bicycle model: front wheels driving model

**2.2.2 Front wheels driving model**

The bicycle model of the front wheels driving model is shown on Figure 7.

The following equation can be established:

*Pure rolling constraint of the wheel 2:* $\cos(\theta+\phi_2).\dot{x} + \sin(\theta+\phi_2).\dot{y} + L.\sin\phi_2.\dot{\theta} = V_2$ (14)

By solving system of equations (14), (9) and (10) we obtain:

$$\dot{x} = V_2 \cos\phi_2 \cos\theta \quad (15)$$

$$\dot{y} = V_2 \cos\phi_2 \sin\theta \quad (16)$$

$$\dot{\theta} = V_2 . \frac{\sin\phi_2}{L} \quad (17)$$

## 2.3 Robudem robot

The Robudem robot has this of particular: it is a combined front and rear wheels driving vehicle because an individual driving motor is attached at each wheel (Figure 8). Also the two axles can be steered which allow the robot to move in dual drive.

### 2.3.1 Single drive

As said above, in single drive the kinematics model of the Robudem is the combined of the rear (equations (11), (12) and (13)) and front (equations (15), (16) and (17)) wheels driving models established before. It is important that the two models produce the same speeds to avoid conflicts between motors (not good for the structure of the robot and its control) i.e. $V_1 = V_2 \cos\phi_2$ (18)

This relation (18) expresses that if we would like the wheels of the Robudem robot to move accordingly the 2 constraints quoted above (pure rolling and no-slipping), the speed of the imaginary rear wheel (i.e. rear wheels) must change in relation with the angle of the steering angle of the front imaginary wheel (i.e. steering front wheels). At the moment the low level control of the Robudem robot did not allow such operation. We can use in this case, one of the model as an approximation (for example, if steering angles of the wheels are not enough big).We can also establish an approximation, at the mathematical point of view, of the model of the Robudem robot in single drive by considering all the constraints ((9), (10), (11) and (15)) and by using the LSE (Least Square Estimator) methods. This method will allow us to have a model which minimizes the error to the expected model. The matrices which express the constraints of the Robudem robot are: $A\xi = y$ (18) where:

$$A = \begin{pmatrix} -\cos(\phi_2+\theta) & -\sin(\phi_2+\theta) & -L\sin\phi_2 \\ -\cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -\sin(\phi_2+\theta) & \cos(\phi_2+\theta) & L\cos\phi_2 \end{pmatrix},\ \xi = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T \text{ and } y = \begin{bmatrix} -V_2 & -V_1 & 0 & 0 \end{bmatrix}^T .$$

Equation (18) should be modified by incorporating an error vector *e* to account modeling error, as follows: $A\xi + e = y$ (19).

Now we want to search for a $\xi = \hat{\xi}$ which minimizes the sum of squared error defined by $E(\xi) = e^T e = (y - A\xi)^T (y - A\xi) = \xi^T A^T A\xi - 2y^T A\xi + y^T y$ .We obtain: $\hat{\xi} = (A^T A)^{-1} A^T y$ (20).

By solving the equation (20), we have:

$$\dot{x} = \frac{1}{2}(\cos\phi_2 V_2 + V_1)\cos\theta \quad (21)$$

$$\dot{y} = \frac{1}{2}(\cos\phi_2 V_2 + V_1)\sin\theta \quad (22)$$

$$\dot{\theta} = V_2 \frac{\sin \phi_2}{L} \quad (23)$$

As it is the same speed which is send to the low level control computer, this implies that $V_2 = V_1 = V$. In this particular case, equation (21), (22) and (23) becomes:

$$\dot{x} = V \cos^2 \frac{\phi_2}{2} \cos\theta \quad (24)$$

$$\dot{y} = V \cos^2 \frac{\phi_2}{2} \sin\theta \quad (25)$$

$$\dot{\theta} = V \frac{\sin \phi_2}{L} \quad (26)$$

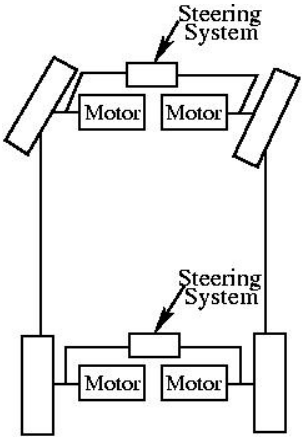

Figure 8: Robudem robot

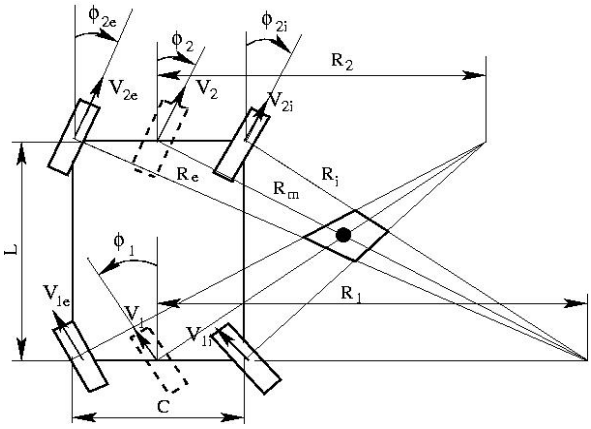

Figure 9: Dual drive

**2.3.2 Dual drive**

In dual drive, the slipping of the wheels is unavoidable because the front-wheels and the rear-wheels will not intersect at one ICR (Instantaneous Center of Rotation) due to the 2 Ackerman mechanisms (as shown on Figure 9). We can suppose that the robot will rotate at the center determined by the axis of imaginary wheels (Figure 9).

These equations can be established by considering the bicycle model (Figure 10) of the Robudem robot in the dual drive:

*Pure Rolling constraint of the wheels 1 and 2*

$$\cos(\phi_1 + \theta)\dot{x} + \sin(\phi_1 + \theta)\dot{y} - V_1 = 0 \quad (27)$$

$$\cos(\phi_2 + \theta)\dot{x} + \sin(\phi_2 + \theta)\dot{y} + L \sin\phi_2 \, \dot{\theta} - V_2 = 0 \quad (28)$$

*Non-slipping constraint of the wheels 1 and 2*

$$\sin(\phi_1 + \theta)\dot{x} - \cos(\phi_1 + \theta)\dot{y}_1 = 0 \quad (29)$$

$$\sin(\phi_2 + \theta)\dot{x} - \cos(\phi_2 + \theta)\dot{y} - L \cos\phi_2 \, \dot{\theta} = 0 \quad (30)$$

By solving the system of equations (27), (28) and (29), we obtain:

$$\dot{x} = V_1 \cos(\phi_1 + \theta) \quad (30)$$

$$\dot{y} = V_1 \sin(\phi_1 + \theta) \quad (31)$$

$$\dot{\theta} = \frac{V_2 - V_1 \cos(\phi_1 - \phi_2)}{L \sin\phi_2} \quad (32)$$

And solving the system of equations (27), (28) and (30), we obtain:

$$\dot{x} = V_1 \cos(\phi_1 + \theta)$$
$$\dot{y} = V_1 \sin(\phi_1 + \theta)$$
$$\dot{\theta} = \frac{V_1 \sin(\phi_1 - \phi_2)}{L \cos\phi_2} \quad (33)$$

Here also to avoid the conflicts between motors, we must change the speeds according to the steering angles. The constraint relation obtained by considering (32) and (33) is:

$V_2 \cos\phi_2 = V_1 \cos\phi_1$ (34)

We can also obtain the approximation of the model using the LSE (by solving equations (27), (28), (29) and (30)):

$$\dot{x} = -\frac{1}{4}\cos(\phi_1 - \theta)V_1 + \frac{3}{4}\cos(\phi_1 + \theta)V_1 + \frac{1}{4}\cos(\phi_1 - \theta)V_2 + \frac{1}{4}\cos(\phi_2 + \theta)V_2 \quad (42)$$
$$\dot{y} = \frac{1}{4}\sin(\phi_1 - \theta)V_1 + \frac{3}{4}\sin(\phi_1 + \theta)V_1 - \frac{1}{4}\cos(\phi_1 - \theta)V_2 + \frac{1}{4}\cos(\phi_2 + \theta)V_2 \quad (43)$$
$$\dot{\theta} = \frac{-\sin\phi_1 V_1 + \sin\phi_2 V_2}{L} \quad (44)$$

If $V_2 = V_1 = V$ (the particular case of the actual low level control of the Robudem Robot), we have:

$$\dot{x} = V \cos(\phi_1 + \theta)$$
$$\dot{y} = V \sin(\phi_1 + \theta)$$
$$\dot{\theta} = \frac{V(\sin\phi_2 - \sin\phi_1)}{L} \quad (45)$$

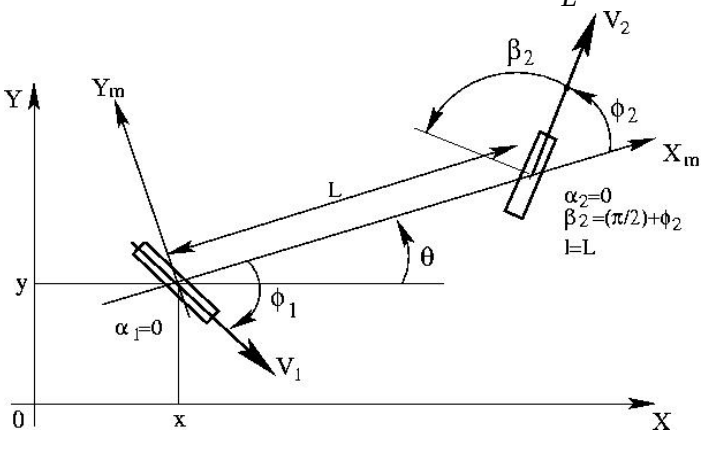


Figure 10: dual drive bicycle model

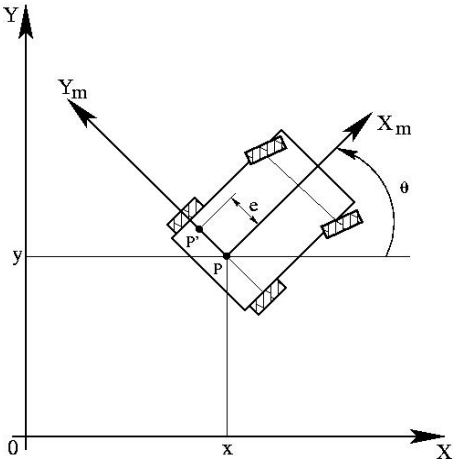


Figure 11: Point coordinate P'

## 3. Posture tracking of the Robudem robot

### 3.1 Dynamic extension algorithm

We will consider in this paper only the front wheels driving model. This model is one of the possible configurations of the Robudem Robot. The same principle can be followed for other configurations. The posture kinematics model of the front wheel drive robot is expressed by the following equations (equivalent to equations (16), (17) and (18) where we add a supplementary equation related to the control of the steering angles of the wheels):

$$\begin{pmatrix}\dot{\xi}\\ \dot{\phi}_2\end{pmatrix}=\begin{pmatrix}\Sigma(\phi_2,\theta) & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}V_2\\ \zeta\end{pmatrix}\ (46)\ \text{where}\ \dot{\xi}=\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix}\ \text{and}\ \Sigma(\phi_2,\theta)=\begin{pmatrix}\cos\phi_2\cos\theta\\ \cos\phi_2\sin\theta\\ \dfrac{\sin\phi_2}{L}\end{pmatrix}.$$

Such a robot has a restricted mobility and is not full state feedback linearizable. In this case, the kinematics model can not be solved by a continuous static time-invariant state feedback but by a dynamic (the robot is moving) linearizing state feedback. In order to apply the dynamic extension algorithm, we have to choose as output functions the coordinates of a point P' located on the axle of the fixed wheels (Figure 11), i.e.

$$h_1 = x - e\sin\theta \quad (47)$$

$$h_2 = y + e\cos\theta \quad (48)$$

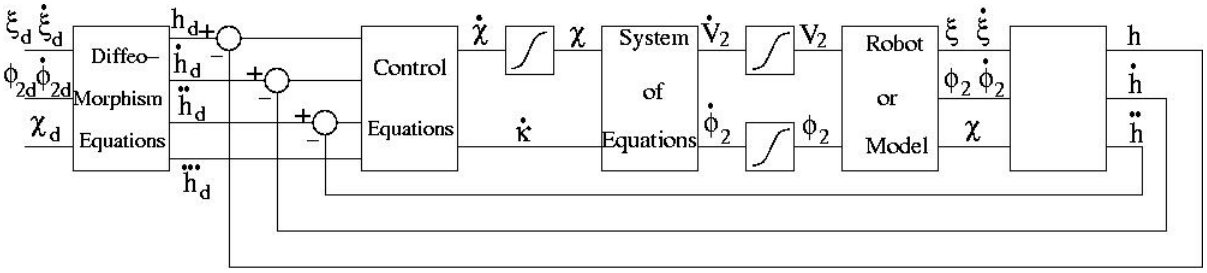


Figure 12: Block diagram of the controller

The first derivative of the output functions give:

$$\dot{h}_1 = (L\cos\phi_2 - e\sin\phi_2)\cos\theta\frac{V_2}{L} \quad (49)$$

$$\dot{h}_2 = (L\cos\phi_2 - e\sin\phi_2)\sin\theta\frac{V_2}{L} \quad (50)$$

If we define $\chi = \frac{\dot{V}_2}{L}(L\cos\phi_2 - e\sin\phi_2) - \frac{V_2}{L}(L\sin\phi_2 + e\cos\phi_2)\zeta$ (51) and

$\kappa = \frac{V_2^2}{L^2}(L\cos\phi_2 - e\sin\phi_2)\sin\phi_2$ (52), we obtain:

$$\ddot{h}_1 = \cos\theta\,\chi - \sin\theta\,\kappa \quad (53)$$

$$\ddot{h}_2 = \sin\theta\,\chi + \cos\theta\,\kappa \quad (54)$$

The third derivative of the output functions gives:

$\dddot{h} = f + g\,u$ (55) Where $f = \begin{pmatrix}-\sin\theta\,\chi\,\dot{\theta} - \cos\theta\,\kappa\,\dot{\theta}\\ \cos\theta\,\chi\,\dot{\theta} - \sin\theta\,\kappa\,\dot{\theta}\end{pmatrix}$, $g = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$, $h = \begin{pmatrix}h_1\\ h_2\end{pmatrix}$ and $u = \begin{pmatrix}\dot{\chi}\\ \dot{\kappa}\end{pmatrix}$.

We have also $\dot{\kappa} = \frac{2V_2\dot{V}_2}{L^2}(L\cos\phi_2 - e\sin\phi_2)\sin\phi_2 + \frac{V_2^2}{L^2}(L\cos(2\phi_2) - e\sin(2\phi_2))\zeta$ (56)

We define $\dddot{h}_d, \ddot{h}_d, \dot{h}_d$ and $h_d$ as the transformation obtained from the coordinate of the desired trajectory. If we apply the control law

$$u = g^{-1}\left(\dddot{h}_d + K_1\left(\ddot{h}_d - \ddot{h}\right) + K_2\left(\dot{h}_d - \dot{h}\right) + K_3\left(h_d - h\right)\right) - f \quad (57)$$

We obtain (introducing (57) in (55)) the closed-loop dynamics

$$\dddot{e} + K_1\ddot{e} + K_2\dot{e} + K_3 e = 0 \quad (58)$$

Where $e = (h_d - h) = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} h_{1d} - h_1 \\ h_{2d} - h_2 \end{pmatrix}$, $K_1 = \begin{pmatrix} K_{11} & 0 \\ 0 & K_{12} \end{pmatrix}$, $K_2 = \begin{pmatrix} K_{21} & 0 \\ 0 & K_{22} \end{pmatrix}$ and $K_3 = \begin{pmatrix} K_{31} & 0 \\ 0 & K_{32} \end{pmatrix}$.

By an appropriate choice of the elements of the matrices $K_1$, $K_2$ and $K_3$ such that all roots of the polynomial $s^3 + K_{11}s^2 + K_{21}s + K_{31} = 0$ and the polynomial $s^3 + K_{12}s^2 + K_{22}s + K_{32} = 0$ are in the open left-half plane, we can see that $\lim_{t\to\infty} e(t) = 0$ (i.e. the tracking of the posture).

## 3.2 Implementation and experimentation

Figure 12 shows the block diagram of the controller. The output of the controller (after the control law is applied) is $\dot{\chi}$ and $\dot{\kappa}$. Next we integrate $\dot{\chi}$ and we obtain $\chi$. We have to solve the system of equations (51) and (56), to have $\dot{V}_2$ and $\zeta$. The solvability of this system of equations is possible only when $V_2$ is different of 0 i.e. the robot must be in movement to apply the algorithm above.

The case study is shown on Figures 13, 14 and 15. It is a typical case of obstacle avoidance but with the control of the speed, the posture and the orientation of the robot. The simulate robot is initially at the Cartesian coordinate (0,0) (the middle of the rear axle is the reference point), has a speed of *1.2 m/s* and an orientation of *0°*. The desire trajectory, speed (*1 m/s*) and orientation (*0°*) are shown with dashed lines in respectively Figures 13, 14 and 15. We can see on these figures that the tracking of the desired postures of the simulated robot is acceptable.

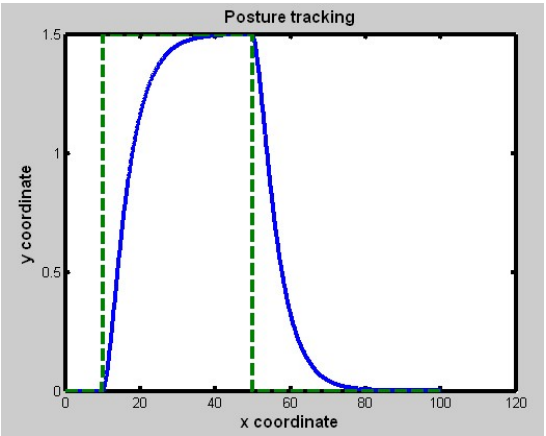


Figure 13: Desired and actual Trajectories

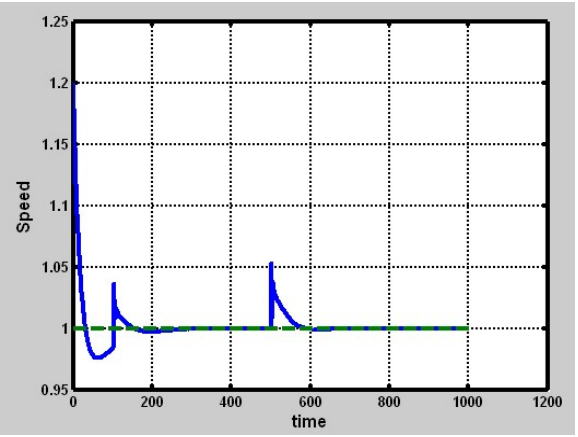


Figure 14: Desired and actual speed

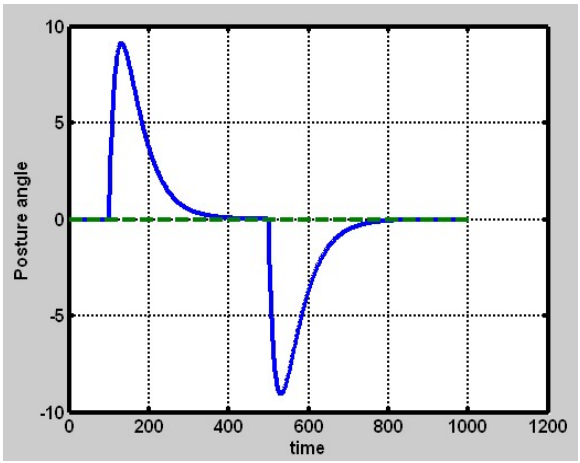


Figure 15: Desired and actual orientation

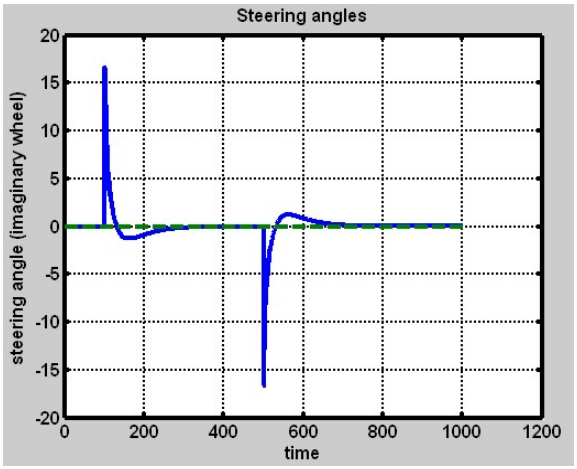


Figure 16: steering angles of the imaginary wheel

## 4. Conclusion

In this paper, we derived the models of the car-like vehicles by considering the pure rolling and the non-slipping constraints of the wheels and applying them on a bicycle model. We also established different models of the Robudem robot depending on its configuration (single or dual drive). Then we showed from one configuration, how to track a specified posture by means of dynamic linearizing state feedback using the dynamic extension algorithm. The case study showed the performance of the design controller on the simulated robot. On the real robot, we had to solve the problem of the transition between the static states of the robot to the dynamic states and reciprocally because the algorithm is only applicable when the robot is in movement. It is also important to determine the way of choosing the parameters of the controller because its behavior depends greatly on this choice. In spite of some failures in the tracking of the defined posture of the robot, due mainly to the 2 problems mentioned above, certain tests on the robot proved the efficiency of this algorithm. In future, we will investigate how to solve the two problems.

## References

[1] P. F. Muir and C. P. Neuman, Kinematic modeling of wheeled mobile robots, J. Robotic Systems 4(2): 281-333, 1987

[2] J. C. Alexander and J. H. Maddocks; on the kinematics of wheeled Mobile Robots, Int. J. of Robotics Research, Vol. 8, n° 5, pp 15-17, 1989

[3] G. Campion, G. Bastin and B. d'Andréa Novel; Structural Properties and classification of kinematic and dynamic models of wheeled mobile robots, IEEE Trans. On Robotics and Automation, vol. 12, pp. 47-62, 1996

[4] B. d'Andréa-Novel, G. Campion and G. Bastin; Control of nonholonomic wheeled mobile robots by state feedback linearization, Int. J. of Robotics Research, Vol 14, pp. 543-559, 1995

[5] F. Jensen, J. Jensen, L. Weiss, A.H. Jorgensen and R. G. Rebolledo; Mobile robot: Instrumentation, Model Verification and Control, Internal Report, Aalborg University, June 2004