Accuracy Improvement for a Redundant Vehicle

Julien Aragones, Geovany Araújo Borges, Alain Fournier

Accuracy improvement for a redundant vehicle

Julien Aragones, Geovany A. Borges and Alain Fournier
LIRMM, UMR no 55060, CNRS/Université Montpellier II
161, rue Ada - 34392 - Montpellier - Cedex 5 - France
{aragones,borges,fournier}@lirmm.fr

ABSTRACT

This paper presents solutions to reduce the effects of wheel slippage and kinematic model uncertainty on low-level control and odometry of over-actuated wheeled mobile robots. Odometry aims at numerical integration of the robot direct kinematic model, whose errors are in part dues to model uncertainties. A parameter identification can significantly improve the odometry accuracy. Wheel slippage introduces large positioning errors. Therefore, robot control must be as smooth as possible, which is achieved by an approach to build a $C2$ continuous motion. The last contribution is specific to omnidirectional robots with offset steered wheels. The redundancy property is used to reduce wheel slippage.

Keywords: $C2$ continuous motion, slippage avoidance, odometry improvement, wheeled mobile robots, redundant robots.

1 INTRODUCTION

One of the main characteristics of mobile robots is their theoretically infinite workspace. Concerning wheeled mobile robots (*WMRs*), this important property is given by the use of friction based joints. Even if they are suited for evolutions on smooth, flat and rigid floors, they are subject to wheels slippage [1].

The *WMRs* kinematic constraints are characterized by the kinematic model, they can only be written in differential form. These models have been widely explored and are expressed in a systematical form in [2]. They are involved in robot control [3] and in velocity estimation, which is very sensitive to model errors and wheel slippage. Omnidirectional robots have full plane mobility. Three actuators can generally provide the three robot degrees-of-freedom (*dofs*), but most robots are over-actuated. This actuation redundancy has been used to change the wheels arrangement of universal wheels [4][5]. It can also be used to complete a subtask [6].

This paper presents studies on three independent and complementary methods for improving accuracy of omnidirectional, over-actuated robots: (i) geometrical parameters estimation, which is very important to achieve a more reliable dead-reckoning positioning, (ii) smooth $C2$ continuous motion generation, and (iii) order reduction of actuation redundancy, both supposed to reduce the wheels slippage. Simulation results show that the slippage can be avoided during dynamic motions. Experiments illustrate the odometry accuracy gain obtained by parameter estimation.

The paper is organized as follows. Section 2 presents the kinematic modelling of a class of over-actuated wheeled mobile robot. Section 3 describes a non-linear identification procedure of geometrical parameters. Section 4 presents the continuous motion generator based on Bezier's formalism, as well as a policy for on-line changing the trajectory parameters. In section 5, order reduction of actuation redundancy using a particular torque distribution is discussed.

2 KINEMATIC MODELLING

For mobile robots, the kinematic model is classically computed by writing the rolling without slipping constraint at the wheel/ground contact point [2]. This model is separated in a posture kinematic model depending on the robot mobility and a configuration kinematic model depending on the robot geometry.

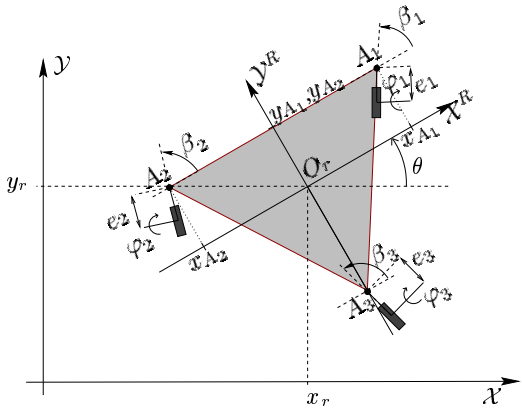
The model considered here is the configuration kinematic model of Omni robot (Figure 1(a)). This model links the posture derivative vector, which can be considered as the control vector, with respect to the configuration velocities, which are the wheel angular velocities. A relation between the robot velocity and the wheel angular velocities is found by applying the vector field relations. Figure 1(b) shows the geometrical parameters involved in the kinematic model. Considering the *i-th* wheel, the parameters are its radii r_i, its offset e_i, and the coordinates (x_{A_i}, y_{A_i}) of the wheel direction axis A_i in the two-dimensional robot frame. Its configuration is described by the direction angle β_i and the posture angle φ_i.

Considering the three axis mechanism, the model parameters are given by

$$\boldsymbol{\lambda} = \begin{pmatrix} \mathbf{e} \\ \mathbf{r} \\ \mathbf{x}_A \\ \mathbf{y}_A \end{pmatrix}, \text{with} \begin{cases} \mathbf{e} = (e_1, e_2, e_3)^T, \\ \mathbf{r} = (r_1, r_2, r_3)^T, \\ \mathbf{x}_A = (x_{A_1}, x_{A_2}, x_{A_3})^T, \\ \mathbf{y}_A = (y_{A_1}, y_{A_2}, y_{A_3})^T. \end{cases}$$

The configuration kinematic relations for the *i-th* wheel

(a)



(b)

Figure 1: (a) Omni and its (b) geometrical parameters.

are given by [2]

$$\begin{pmatrix} \dot{\beta}_i \\ \dot{\varphi}_i \end{pmatrix} = \begin{pmatrix} -\sin(\beta_i)/e_i & \cos(\beta_i)/e_i \\ \cos(\beta_i)/r_i & \sin(\beta_i)/r_i \\ \left(x_{A_i}\cos(\beta_i) + y_{A_i}\sin(\beta_i) - e_i\right)/e_i \\ \left(x_{A_i}\sin(\beta_i) - y_{A_1}\cos(\beta_1)\right)/r_i \end{pmatrix} \cdot \dot{\boldsymbol{\xi}}_r, \quad (1)$$

with $\boldsymbol{\xi}_r = (x_r, y_r, \theta)^T$. The posture kinematic relations are obtained by replacing $\dot{\boldsymbol{\xi}}_r$ as

$$\dot{\boldsymbol{\xi}}_r = \mathbf{R}(\theta) \cdot \dot{\boldsymbol{\xi}} \quad (2)$$

with $\boldsymbol{\xi} = (x, y, \theta)^T$ being the robot coordinates in the world frame and

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

Such relations depend on the wheel configuration as well as on the different geometrical parameters. In the sequel, the model sensitivity to $\boldsymbol{\lambda}$ is discussed. Then a parameter identification procedure is presented and its effect on posture estimation is discussed.

2.1 Sensitivity to geometrical errors

In order to compute the model sensitivity to geometrical parameters, the velocity of the direction axis A_i is considered. Equation 1 is decomposed into

$$\begin{pmatrix} \dot{\beta}_i + \dot{\theta} \\ \dot{\varphi}_i \end{pmatrix} = \begin{pmatrix} -\sin(\beta_i)/e_i & \cos(\beta_i)/e_i \\ \cos(\beta_i)/r_i & \sin(\beta_i)/r_i \end{pmatrix} \begin{pmatrix} v_{x_{A_i}} \\ v_{y_{A_i}} \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} v_{x_{A_i}} \\ v_{y_{A_i}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -y_{A_i} \\ 0 & 1 & x_{A_i} \end{pmatrix} \dot{\boldsymbol{\xi}}_r. \quad (5)$$

The robot configuration kinematic model is obtained by writing these equations for the three robot wheels:

$$\begin{pmatrix} \dot{\beta}_1 + \dot{\theta} \\ \dot{\beta}_2 + \dot{\theta} \\ \dot{\beta}_3 + \dot{\theta} \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{pmatrix} = \mathbf{J}_{p_1} \cdot \mathbf{J}_{p_2} \cdot \dot{\boldsymbol{\xi}}_r \quad (6)$$

In eq. (6), $\mathbf{J}_{p_1}$ is a 6×6 non-singular matrix depending on the wheel configuration and the wheels offset and radii. $\mathbf{J}_{p_2}$ is a 6×3 matrix depending on the direction axis position. Equation (6) is now inverted to express the direction axis velocity with respect to the wheels angular velocities:

$$\begin{pmatrix} v_{x_{A_i}} \\ v_{y_{A_i}} \end{pmatrix} = \begin{pmatrix} -e_i\sin(\beta_i) & r_i\cos(\beta_i) \\ e_i\cos(\beta_i) & r_i\sin(\beta_i) \end{pmatrix} \begin{pmatrix} \dot{\beta}_i + \dot{\theta} \\ \dot{\varphi}_i \end{pmatrix} \quad (7)$$

In the sequel, the symbol Δ is used to indicate the perturbation on a given variable. After the differentiation of (5) and (7) and the elimination of $(\Delta v_{x_{A_i}}, \Delta v_{y_{A_i}})^T$, the relation linking the velocity uncertainty with the geometrical parameters uncertainties and the measure uncertainties is found. For the *i-th* wheel, this relation is expressed as follows:

$$\begin{aligned} & - \begin{pmatrix} 1 & 0 & -y_{A_i} + e_i\sin(\beta_i) \\ 0 & 1 & x_{A_i} - e_i\cos(\beta_i) \end{pmatrix} \Delta\dot{\boldsymbol{\xi}}_r \\ & + \begin{pmatrix} (\dot{\beta}_i + \dot{\theta})\sin(\beta_i) \\ (\dot{\beta}_i + \dot{\theta})\cos(\beta_i) \end{pmatrix} \Delta e_i \\ & + \begin{pmatrix} \dot{\varphi}_i\cos(\beta_i) \\ \dot{\varphi}_i\sin(\beta_i) \end{pmatrix} \Delta r_i + \begin{pmatrix} 0 & \dot{\theta} \\ \dot{\theta} & 0 \end{pmatrix} \begin{pmatrix} \Delta x_{A_i} \\ \Delta y_{A_i} \end{pmatrix} \\ & + \begin{pmatrix} -e_i\sin(\beta_i) \\ e_i\cos(\beta_i) \end{pmatrix} \Delta\dot{\beta}_i + \begin{pmatrix} r_i\cos(\beta_i) \\ r_i\sin(\beta_i) \end{pmatrix} \Delta\dot{\varphi}_i \\ & + \begin{pmatrix} -e_i\cos(\beta_i)(\dot{\beta}_i + \dot{\theta}) - r_i\sin(\beta_i)\dot{\varphi}_i \\ -e_i\sin(\beta_i)(\dot{\beta}_i + \dot{\theta}) + r_i\cos(\beta_i)\dot{\varphi}_i \end{pmatrix} \Delta\beta_i. \\ = \; & 0 \end{aligned} \quad (8)$$

Eq. (8) is extended to the three robot wheels and used to find the Direct Kinematic Model (*DKM*) uncertainties due to the geometrical and/or the measure uncertainties. Considering only the uncertainties on geometrical parameters, one has for the whole vehicle

$$\mathbf{J}_p \cdot \Delta\dot{\boldsymbol{\xi}}_r = \mathbf{J}_g \cdot \Delta\boldsymbol{\lambda}. \quad (9)$$

The sensitivity matrix to geometrical uncertainties can be found from several ways. The system written in (9) is over-determined. Thus, in order to propagate the geometrical parameters perturbation $\Delta\lambda$ to the robot velocity perturbation $\Delta\dot{\boldsymbol{\xi}}_r$, we use

$$\Delta\dot{\boldsymbol{\xi}}_r = \left(\mathbf{J}_p^{\dagger} \cdot \mathbf{J}_g\right) \cdot \Delta\boldsymbol{\lambda}, \quad \text{or} \quad \Delta\dot{\boldsymbol{\xi}}_r = \mathbf{J}_{sp} \cdot \Delta\boldsymbol{\lambda} \quad (10)$$

with † being the Moore-Penrose pseudo-inverse operator. The sensitivity matrix $\mathbf{J}_{sp}$ is a 3×12 matrix. Supposing that the entries of $\Delta\boldsymbol{\lambda}$ are uncorrelated random normal variables with zero mean, the covariance matrix $\mathbf{C}_{\lambda}$ of $\Delta\boldsymbol{\lambda}$ is diagonal. Supposing a small perturbation model, $\Delta\dot{\boldsymbol{\xi}}_r$ is also assumed to be normally distributed with zero mean and covariance matrix given by

$$\mathbf{C}_{\dot{\boldsymbol{\xi}}_r} = \mathbf{J}_{sp} \cdot \mathbf{C}_{\boldsymbol{\lambda}} \cdot \mathbf{J}_{sp}^{T}. \quad (11)$$

Thus, $\mathbf{C}_{\dot{\boldsymbol{\xi}}_r}$ dependents on the wheels configuration β_i, their angular velocities $\dot{\beta}_i$ and $\dot{\varphi}_i$, and the robot angular velocity $\dot{\theta}$.

2.1.1 Sensitivity analysis

An analytical form of $\mathbf{J}_p^{\dagger}$ is very complicated to obtain. Thus, a study of the *DKM* sensitivity to geometrical errors is done using the numerical form of $\mathbf{C}_{\dot{\boldsymbol{\xi}}_r}$. Let's assume a displacement composed of two translatory motions, along X and Y axes (motion *A*), and of a motion involving the three degrees of freedom (motion *B*). These motions are executed with the robot at its maximum speed.

During motion *A*, the uncertainties on the translatory axe are not correlated with those on the others two axes. The resulting covariance matrix is given by

$$C_{\dot{\boldsymbol{\xi}}_r} = \begin{pmatrix} 0.0333 & 0 & 0 \\ 0 & 0.0003 & 0.0059 \\ 0 & 0.0059 & 0.12 \end{pmatrix}. \quad (12)$$

It is noticeable that the maximum uncertainty concerns the robot orientation axis. During motion *B*, a correlation between all the three axis is observed. In a given wheel configuration, the following covariance matrix has been computed:

$$C_{\dot{\boldsymbol{\xi}}_r} = \begin{pmatrix} 0.033 & 0.033 & 0.04 \\ 0.033 & 0.0397 & 0.057 \\ 0.04 & 0.057 & 0.274 \end{pmatrix}. \quad (13)$$

Once more, the maximum uncertainty is on the orientation axis.

The following conclusions and the possible solutions were obtained based on these and other extensive simulations:

1. As expected, the uncertainties on geometrical parameters can generate in some conditions large errors on velocity and posture estimation. A parameter estimation procedure can minimize such errors;
2. For this model, the largest errors are always on the orientation axis. The use of an auxiliary proprioceptive sensor to measure the robot orientation can significantly improve the velocity estimation.

3 PARAMETER IDENTIFICATION

3.1 Odometric model

Parameter identification is intended for improving the robot odometric model, which is obtained by discretizing the posture kinematic model. The inverse posture kinematic model is given by stacking eqs. (1)-(2) corresponding to the three wheels in a matrix form as

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}, \boldsymbol{\lambda}, \theta) \cdot \dot{\boldsymbol{\xi}}, \quad (14)$$

where $\mathbf{q} = (\beta_1, \beta_2, \beta_3, \varphi_1, \varphi_2, \varphi_3)^T$ is the robot configuration parameters and $\mathbf{J}(\mathbf{q}, \boldsymbol{\lambda}, \theta)$ is a 6×3 matrix. The posture kinematic model is

$$\dot{\boldsymbol{\xi}} = \mathbf{J}^{\dagger}(\mathbf{q}, \boldsymbol{\lambda}, \theta) \cdot \dot{\mathbf{q}}. \quad (15)$$

The simplest way of discretize (15) is to apply the first order linearization:

$$\hat{\boldsymbol{\xi}}_k = \hat{\boldsymbol{\xi}}_{k-1} + \mathbf{J}^{\dagger}(\mathbf{q}_{k-1}, \boldsymbol{\lambda}, \hat{\theta}_{k-1}) \cdot (\mathbf{q}_k - \mathbf{q}_{k-1}), \quad (16)$$

with k being the discrete time index. It should be observed that $\mathbf{J}^{\dagger}$ depends of $\mathbf{q}_{k-1}$, $\boldsymbol{\lambda}$ and $\hat{\theta}_{k-1}$, where $\mathbf{q}_{k-1}$ is usually the most accurate variable, even when using non expensive encoders. $\hat{\theta}_{k-1}$ is a component of $\hat{\boldsymbol{\xi}}_{k-1}$, which can be rapidly uncertain, in part due to the geometrical parameters uncertainty (*c.f.*, section 2.1.1). Other source of uncertainty is an approximated kinematic model, which can improved by adding more variables (*e.g.*, the axis inclination angles). Nevertheless, such a solution increases the model complexity and jeopardizes its use for real-time pose estimation using odometry. This is the well known modeling paradigm, where a trade-off between model complexity and model usability must be established. Further, from a certain number of parameters, model errors are less important than other non-systematic sources, such as navigating on uneven floors and wheel slippage. Therefore, an alternative solution for minimizing odometry errors is to estimate $\boldsymbol{\lambda}$.

3.2 Identification procedure

In order to identify the parameter vector $\boldsymbol{\lambda}$, a stochastic identification procedure has been applied. Such a procedure relies on the integration of measurements provided by a high precision laser gyrometer. Indeed, it is based on an extended Kalman filter (EKF) applied to state estimation of the following stochastic system:

$$\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{n-1}, \quad (17)$$

$$\Delta\theta_n = h_n(\boldsymbol{\lambda}_n) + w_n. \quad (18)$$

In the above, (17) is the process evolution model and (18) represents the measurements. w_n is the measurement noise, supposed to be Gaussian with zero mean

and variance $\mathbf{C}_{w_n}$. It should be observed that eq. (17) does not consider any parameter changing during the identification experiment (*i.e.*, without associated noise).

The measurement variable is the robot heading variation $\Delta\theta_n$ during each time step, provided by a high precision laser gyrometer. $h_n(\boldsymbol{\lambda}_n)$ is the heading changing computed by the odometric model:

$$h_n(\boldsymbol{\lambda}_n) = \mathbf{c}\cdot\mathbf{J}^{\dagger}(\mathbf{q}_{n-1},\boldsymbol{\lambda}_n,\theta_{n-1})\cdot(\mathbf{q}_k-\mathbf{q}_{k-1}). \quad (19)$$

Even if $\mathbf{J}^{\dagger}$ depends on θ_{n-1}, this variable has no influence on the measurements, given the row-vector $\mathbf{c} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$. Thus, θ_{n-1} may be provided by the non-accurate odometry, or by integrating the gyrometer measurements.

Supposing a sequence of measurements $\Delta\theta_n$, $n = 1,\ldots,N$, the Kalman filter updating equations for the estimated geometrical parameters $\hat{\boldsymbol{\lambda}}_{n-1}$ are

$$\begin{aligned}\hat{\boldsymbol{\lambda}}_n &= \hat{\boldsymbol{\lambda}}_{n-1} + \mathbf{k}_n\cdot(\Delta\theta_n - h_n(\hat{\boldsymbol{\lambda}}_{n-1})),\\ \mathbf{C}_{\hat{\boldsymbol{\lambda}}_n} &= \left(\mathbf{I}_{12} - \mathbf{k}_n\cdot\nabla^T_{\hat{\boldsymbol{\lambda}}_{n-1}}h_n\right)\cdot\mathbf{C}_{\hat{\boldsymbol{\lambda}}_{n-1}},\end{aligned} \quad (20)$$

with $\mathbf{C}_{\hat{\boldsymbol{\lambda}}_n}$ being the covariance matrix associated to $\hat{\boldsymbol{\lambda}}_n$. The Kalman gain is given by

$$\mathbf{k}_n = \frac{\mathbf{C}_{\hat{\boldsymbol{\lambda}}_{n-1}}\cdot\nabla^T_{\hat{\boldsymbol{\lambda}}_{n-1}}h_n}{\nabla_{\hat{\boldsymbol{\lambda}}_{n-1}}h_n\cdot\mathbf{C}_{\hat{\boldsymbol{\lambda}}_{n-1}}\cdot\nabla^T_{\hat{\boldsymbol{\lambda}}_{n-1}}h_n + C_{w_n}}. \quad (21)$$

In order to minimize the risk of filter divergence, it is a common practice to artificially increase the covariance matrix $\mathbf{C}_{w_n}$. Indeed, w_n should encompass not only the sensor measurement uncertainty, represented by the variance $\sigma^2_{\Delta\theta_n}$, but also the measurement model errors [7]. Hence, we follow

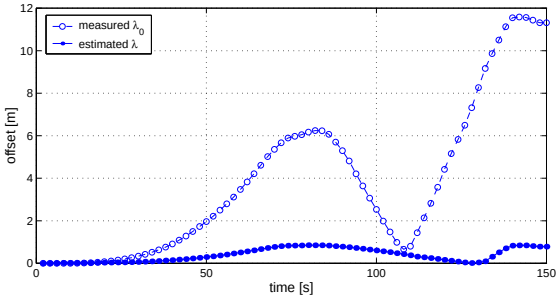
$$\mathbf{C}_{w_n} = \sigma^2_{\Delta\theta_n} + (\mu_n)^2 \quad (22)$$

where μ_n is a design parameter. In this way, this filter formulation is equivalent to that of the normalized gain recursive least-squares algorithm [8]. As such, μ_n have a strong influence on the estimator convergence speed. However, filter convergence should not be interpreted as parameter convergence to the true values, given that a non-linear approximated model is used for odometry.

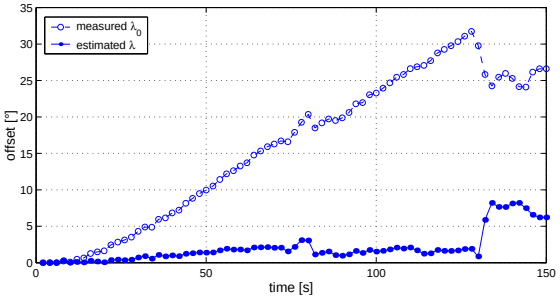
3.3 Identification experiment

For parameter excitation, the experimental datum was acquired with the Omni platform in different environments executing simple and complex trajectories. The amount of measurements for identification was $N = 142,880$, in a total of 714.40 seconds. The initial parameter values were obtained with measures made on the physical robot:

$$\hat{\boldsymbol{\lambda}}_0 : \begin{cases} \mathbf{r} = (\ 0,1\,m \quad 0,1\,m \quad 0,1\,m\ )^T, \\ \mathbf{e} = (\ 0,05\,m \quad 0,05\,m \quad 0,05\,m\ )^T, \\ \mathbf{x}_A = (\ 0,325\,m \quad -0,325\,m \quad 0\,m\ )^T, \\ \mathbf{y}_A = (\ 0,188\,m \quad 0,188\,m \quad -0,376\,m\ )^T. \end{cases}$$



(a) Position offsets



(b) Heading offsets

Figure 2: Offsets of pose estimation presented by the measured $\hat{\lambda}_0$ and estimated $\hat{\lambda}$ geometric parameters.

(23)

The initial covariance matrix is $\boldsymbol{\Lambda}_{\hat{\boldsymbol{\lambda}}_0} = (0,005\,m)^2\cdot\mathbf{I}_{12}$, with $\mathbf{I}_{12}$ being the identity matrix of dimension 12. This initialization for $\boldsymbol{\Lambda}_{\hat{\boldsymbol{\lambda}}_0}$ was that allowed better results. Using $\mu_n = 160\cdot\sigma_{\Delta\theta_n}$, the identified parameters were:

$$\hat{\boldsymbol{\lambda}} : \begin{cases} \mathbf{r} = (\ 0,0988\,m \quad 0,0989\,m \quad 0,1012\,m\ )^T, \\ \mathbf{e} = (\ 0,0443\,m \quad 0,0424\,m \quad 0,0446\,m\ )^T, \\ \mathbf{x}_A = (\ 0,3143\,m \quad -0,3302\,m \quad 0,0170\,m\ )^T, \\ \mathbf{y}_A = (\ 0,1916\,m \quad 0,1926\,m \quad -0,3816\,m\ )^T. \end{cases} \quad (24)$$

Validation of the identified parameters are presented in Figure 2. In this figure, one has the position and heading offsets in the time, presented by odometry using the measured parameters (23) and the estimated ones (24). The validation was carried out in a real environment, following a closed-loop trajectory, resulting in traveled distance of $56\,m$. The offsets are with respect to a more accurate dead-reckoning method which fuses encoder and laser gyrometer readings. Such an approach allows reliable comparison of the different geometrical parameters sets given the short duration of this experiment. It is worth noting the improvements on odometry-based pose estimation using the estimated parameters.

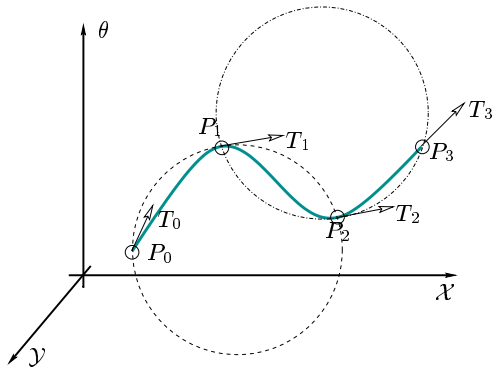


Figure 3: Trajectory building.

4 $C2$ CONTINUOUS MOTION GENERATION

Another objective of this study is to provide a motion generator ensuring an accurate motion, passing on the via points at the given velocity. The via points and the robot velocities are defined at an upper level, either by a path planner, or by a sensory system. This approach defines a geometrical trajectory which can be covered with different velocities.

4.1 Trajectory definition

The trajectory is defined in a piecewise manner. A segment is a piece of trajectory included between two consecutive via points. A segment geometry is computed while the robot is on the first segment via point. In order to simplify notations, the four consecutive points involved in the design of the current segment are called P_0, P_1, P_2, P_3. These points are considered to define the segment geometry in the 3D space $\mathcal{X} \times \mathcal{Y} \times \theta$.

Three consecutive via points are obviously on a circle. Using these circles, the tangents and the curvatures are found (see Figure 3). The tangents and curvatures are used to calculate the control points of the curve. Using the Bezier's formalism, the trajectory segment is described by a fifth degree polynomial

$$\boldsymbol{\xi}_n(u) = ((((\mathbf{p}_a \cdot u + \mathbf{p}_b) \cdot u + \mathbf{p}_c) \cdot u + \mathbf{p}_d) \cdot u + \mathbf{p}_e) \cdot u + \mathbf{p}_f,$$

where $\mathbf{p}_a$, $\mathbf{p}_b$, $\mathbf{p}_c$, $\mathbf{p}_d$, $\mathbf{p}_e$, $\mathbf{p}_f$ are three-dimensional vectors. During a segment, the Bezier's parameter u belongs to $[0, 1]$.

Using this definition, it is possible to modify on-line the trajectory, even during robot motion, without disturbing the $C2$ continuity. These changes should be done to correct the trajectory tracking (in our case, an integrated system using map-based positioning) or to avoid obstacles.

The correction principle is shown in Figure 4. Suppose an error ϵ is measured on point P_1, such that the actual point is P_1': $\epsilon = P_1' - P_1$. The tangents and curvatures of the segment P_1-P_2 are defined by the two circles passing through (P_0, P_1, P_2), and (P_1, P_2, P_3). In order to keep a $C2$ continuous trajectory, the next modifiable point is P_3.

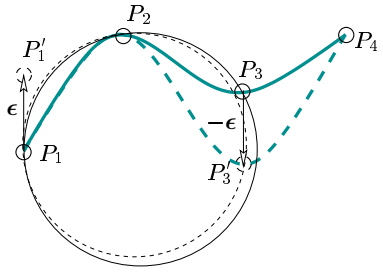


Figure 4: Trajectory correction principle.

Denote $P_3' = P_3 - \epsilon$ the corrected point. The new circle (P_1, P_2, P_3') appears in dashed drawing in figure (Fig. 5). It allows us to compute the new tangent and curvature on point P_2. The trajectory obtained after the correction on P_3 is plotted in dashed line. By changing the tangent and curvature on point P_2, the reference trajectory is modified from the point P_1 in order to reach the planned via points in a continuous manner.

4.2 Motion definition

In order to complete the motion, a velocity law is needed. The velocity law expresses the curvilinear abscissa with respect to the time. To keep the second order continuity for the movement, the velocity law must be at least $C2$ continue. A sinus plus slope law is chosen. Denoting $s\prime(t)$ the curvilinear velocity, the motion generation law is made of three phases:

1. *Acceleration phase*: for $0 \leq t < \Delta t_1$:

$$s'(t) = \frac{V_1 - V_m}{2} \cos(\frac{\pi t}{\Delta t_1}) + \frac{V_1 + V_m}{2}.$$

2. *Velocity step phase*: for $\Delta t_1 \leq t < \Delta t_1 + \Delta t_2$:

$$s'(t) = V_m.$$

3. *Deceleration phase*: for $\Delta t_1 + \Delta t_2 \leq t \leq \Delta t_1 + \Delta t_2 + \Delta t_3$:

$$s'(t) = V_m - \left\{ \frac{V_2 - V_m}{2} \cos(\frac{\pi(t - \Delta t_1 - \Delta t_3)}{\Delta t_2}) + \frac{V_m - V_2}{2} \right\}.$$

The duration of each phase depends on the via points velocities and on the robot acceleration capabilities:

$$\Delta t_1 = \frac{\pi}{2} \frac{|V_m - V_1|}{A_m}, \quad (25)$$

$$\Delta t_2 = \frac{1}{V_m} \left\{ L_t - \frac{V_m + V_1}{2} \Delta t_1 - \frac{V_m + V_2}{2} \Delta t_3 \right\}, \quad (26)$$

$$\Delta t_3 = \frac{\pi}{2} \frac{|V_m - V_2|}{A_m}, \quad (27)$$

where V_m and A_m are the maximum allowed velocity and acceleration on the segment.

The remaining step is the trajectory and the velocity law synchronization. Once the geometry is defined, the curvilinear abscissa is interpolated by a polynomial curve $P(u)$. The velocity law is then integrated to know each sample time the covered path $s(t_i)$. For a given

sample time t_i, the parameter value $u(t_i)$ is given by the root of $P(u) - s(t_i)$ belonging to $[0, 1]$.

The posture computation is summarized as:

$$t_i \longrightarrow s(t_i) \overset{P^{-1}(u_i)}{\longrightarrow} u(t_i) \overset{\xi_n(u_i)}{\longrightarrow} \xi_n(t_i). \tag{28}$$

Applying this method, a $C2$ continuous motion is computed. It is actually implemented in our robot for real-time motion definition. It executes at low level in a map-based posioning system, where absolute pose correction is performed at regular time intervals of two seconds. Thus, the trajectory must be recomputed every time the robot reaches a pass point.

5 ORDER REDUCTION OF ACTUATOR REDUNDANCY

Omnidirectional mobile robots have full plane mobility. According to conventional or universal wheels use, the actuator number is different. Beyond three, the robot is redundantly actuated. The additional actuators can be used to realize a subtask. This redundancy, often used for manipulator robots, is not usually studied in mobile robotics.

For omnidirectional mobile robots equipped with universal wheels, the remaining *dofs* are used to change the wheel arrangement. The robot is then equipped with a continuously variable transmission [4] [5]. Concerning omnidirectional robots equipped with conventional wheels, the redundancy degree is often higher. However, the load distribution has shortly been discussed [6].

In the sequel, the redundancy degree is first studied after expressing the Jacobian matrix in a velocity frame. Then the actuation redundancy is used to reduce wheel slippage.

5.1 Redundancy degree

The actuator torques and the operational forces and torque are related by [2]:

$$\mathbf{F}_{\boldsymbol{\xi}_r} = \mathbf{J}_p^T \cdot \boldsymbol{\Gamma}_A \tag{29}$$

with $\mathbf{F}_{\boldsymbol{\xi}_r} = (F_{x_r}, F_{y_r}, M_\theta)^T$ and $\Gamma_A = (\Gamma_{\varphi_1}, \Gamma_{\varphi_2}, \Gamma_{\varphi_3}, \Gamma_{\beta_1}, \Gamma_{\beta_2}, \Gamma_{\beta_3})^T$. The Jacobian matrix $\mathbf{J}_p^T$ is then expressed in the Frenet frame. It is dependent on $\alpha = \arctan(\dot{y}_r/\dot{x}_r)$:

$$\mathbf{J}_p \cdot \mathbf{R}(\alpha) = \begin{pmatrix} \mathbf{J}_\varphi \\ \mathbf{J}_\beta \end{pmatrix} =$$

$$\begin{pmatrix} \frac{\cos(\beta_1-\alpha)}{r_1} & \frac{\sin(\beta_1-\alpha)}{r_1} & \frac{x_{A_1}\sin(\beta_1)-y_{A_1}\cos(\beta_1)}{r_1} \\ \frac{\cos(\beta_2-\alpha)}{r_2} & \frac{\sin(\beta_2-\alpha)}{r_2} & \frac{x_{A_2}\sin(\beta_2)-y_{A_2}\cos(\beta_2)}{r_2} \\ \frac{\cos(\beta_3-\alpha)}{r_3} & \frac{\sin(\beta_3-\alpha)}{r_3} & \frac{x_{A_3}\sin(\beta_3)-y_{A_3}\cos(\beta_3)}{r_3} \\ \frac{-\sin(\beta_1-\alpha)}{e_1} & \frac{\cos(\beta_1-\alpha)}{e_1} & \frac{x_{A_1}\cos(\beta_1)+y_{A_1}\sin(\beta_1)-e_1}{e_1} \\ \frac{-\sin(\beta_2-\alpha)}{e_2} & \frac{\cos(\beta_2-\alpha)}{e_2} & \frac{x_{A_2}\cos(\beta_2)+y_{A_2}\sin(\beta_2)-e_2}{e_2} \\ \frac{-\sin(\beta_3-\alpha)}{e_3} & \frac{\cos(\beta_3-\alpha)}{e_3} & \frac{x_{A_3}\cos(\beta_3)+y_{A_3}\sin(\beta_3)-e_3}{e_3} \end{pmatrix}.$$

The subspaces generated by the traction and the direction actuators torques can be dissociated:

$$\begin{pmatrix} \mathbf{F}_\beta \\ \mathbf{F}_\varphi \end{pmatrix} = \begin{pmatrix} \mathbf{J}_\varphi^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_\beta^T \end{pmatrix} \cdot \boldsymbol{\Gamma}_A, \tag{30}$$

and $\mathbf{F}_{\boldsymbol{\xi}_r} = \mathbf{F}_\varphi + \mathbf{F}_\beta$, where $\mathbf{F}_\varphi$ and $\mathbf{F}_\beta$ gives the redundancy degree. In the general case, there are six actuators to generate three *dofs*. Thus, the redundancy degree is three. Whatever the motion, the redundancy degree is always non zero. Therefore the actuator redundancy can always be used.

5.2 Slipping minimization

Considering a fully actuated three castor wheels mobile robot, the redundancy degree is at least two. It is also possible to use actuation redundancy to complete a subtask during the robot motion.

5.2.1 Solving under-determined systems

The torques, needed to realize the motion, are found by solving the non-linear system (29). This under-determined can be solved (i) by adding constraint equations to reduce the under determination, (ii) by using the Moore-Penrose pseudo-inverse to find the least-squares solution, or (iii) by the general well known formulation $\boldsymbol{\Gamma}_A = (\mathbf{J}_p^T)^\dagger \cdot \mathbf{F}_{\boldsymbol{\xi}_r} + (\mathbf{I} - (\mathbf{J}_p^T)^\dagger \cdot (\mathbf{J}_p^T)) \cdot \varepsilon$, where $\varepsilon \in \mathbb{R}^6$ is used to complete a subtask.

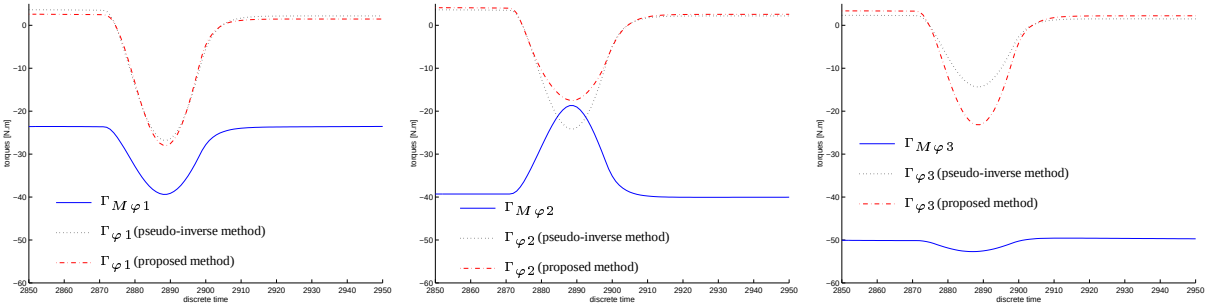
A lot of techniques have been developed to allow manipulator robots to avoid singular positions or to keep the joints far from their articular limits. In mobile robotics, these techniques have been discussed in [6] but they have never been applied to solve mobile robot troubles.

5.2.2 Slippage avoidance

The Moore-Penrose pseudo-inverse is often encountered in robot control but is not always the best solution. An off-centered wheels mobile robot has very large normal load variations on its actuated wheels due to their offset. The Jacobian pseudo-inverse distribute the torques on a geometrical purpose. This solution, satisfactory for fixed or centered wheels, can introduce wheel slippage in the case of caster wheels. The idea is to use the actuator redundancy to distribute the Cartesian forces while keeping the wheels as far as possible from their slippage conditions.

The dynamical normal load on each wheel is computed, depending on the their arrangement and on the robot inertial forces [9]. To avoid wheel slippage, the tangential force provided by the actuator torques must be lower than the Coulomb friction limit. Another important condition is to provide a smooth and continuous load distribution. If the cost function switches from one wheel to the others, the desired torque breaks can introduce slippage. Any torque combination belonging to the Jacobian kernel, can be added to the Moore-Penrose solution without changing the whole motion.

In a first step the contribution to motion of traction and



(a) Plots of Γ_{φ_1} and $\Gamma_{M\varphi_1}$

(b) Plots of Γ_{φ_2} and $\Gamma_{M\varphi_2}$

(c) Plots of Γ_{φ_3} and $\Gamma_{M\varphi_3}$

Figure 5: Simulation results.

direction actuators is checked by computing $\mathbf{F}_\varphi$ and $\mathbf{F}_\beta$. The torque vector ε is given by the linear combination $\varepsilon = |\mathbf{F}_\varphi| \cdot \varepsilon_\varphi + |\mathbf{F}_\beta| \cdot \varepsilon_\beta$, where ε_φ is used to act on traction torques and ε_β is used to act on orientation torques. This formulation avoids the switching from one wheel to the others. The distribution is smooth and continuous.

The orientation torques are computed by the Moore-Penrose solution. The maximum admissible traction torques $\mathbf{\Gamma}_{M\varphi} = (\Gamma_{M\varphi_1}, \Gamma_{M\varphi_2}, \Gamma_{M\varphi_3})^T$ are computed:

$$\Gamma_{M\varphi_i} = r_i \sqrt{(\mu_i \cdot F_{n_i})^2 - (\Gamma_{\beta_i}/e_i)^2}. \qquad (31)$$

ε_φ is obtained by differentiating the cost function

$$f(\Gamma_{\varphi_1}, \Gamma_{\varphi_2}, \Gamma_{\varphi_3}) = \sum_{i=1}^{3} \frac{\Gamma_{M\varphi_i}}{|\mathbf{\Gamma}_{M\varphi}|} (\Gamma_{M\varphi_i} - \Gamma_{\varphi_i})^2. \quad (32)$$

The same calculus is repeated for the steering axes β_1, β_2 and β_3 in order to compute ε_β. In a last step, ε is computed and introduced in the general solution formulation defined in section 5.2.1.

During dynamic motions, the torque distribution on the wheels is quite different from the Moore-Penrose solution. In the plots of Figure 5, one has the torque limits $\Gamma_{M\varphi_1}$, $\Gamma_{M\varphi_2}$ and $\Gamma_{M\varphi_3}$, as well as the torques Γ_{φ_1}, Γ_{φ_2} and Γ_{φ_3}, computed by the classic Moore-Penrose pseudo-inverse approach and by the proposed method. It should be pointed out that with the classic approach the torque limit $\Gamma_{M\varphi_2}$ is not respected (see 5(b), above). In order to assure such a constraint, transfers the exceeding torque of Γ_{φ_2} to Γ_{φ_1} and Γ_{φ_3}, thus minimizing wheel slippage due to the robot's dynamics.

6 CONCLUSION

This paper presented three independent studies for accuracy improvement of an over-actuated, omnidirectional wheeled mobile robot. The proposed contributions aim to be used at low-level control and dead-reckoning based positioning of this class of mobile robots.

The kinematic model depending on the geometrical parameters is necessary for both robot control and posture estimation. It has been presented an identification procedure for the robot geometrical parameters using measurements provided by a laser gyrometer. In a second time, a motion generator providing a $C2$ continuous motion has been presented. The generator guarantees continuous velocity or acceleration controls which reduce wheel slippage. These controls can be used with torque distribution function to avoid the slippage by keeping the wheels far from their slippage condition. A particular torque distribution is discussed, and a simulation illustrates how it reduces the slippage during the dynamical motions.

REFERENCES

[1] S. Shekhar, "Wheel rolling constraints and slip in mobile robots," in *IEEE International Conference on Robotics and Automation*, 2000.

[2] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transaction on Robotics and Automation*, vol. 12, pp. 47–62, February 1996.

[3] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Springer Verlag, 1996.

[4] K. S. Byun, S. J. Kim, and J. B. Song, "Design of a four-wheeled omnidirectional mobile robot with variable wheel arrangement mechanism," in *IEEE International Conference on Robotics and Automation*, 2002.

[5] M. Wada and H. Asada, "Design and control of a variable footprint mechanism for holomonic omnidirectional vehicles and its application to wheelchair," *IEEE Transaction on Robotics and Automation*, vol. 15, no. 6, pp. 978–989, 1999.

[6] B. Yi and W. Kim, "The kinematics for redundantly actuated omni-directional mobile robots," in *IEEE International Conference on Robotics and Automation*, 2000.

[7] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[8] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, second ed., 1999.

[9] J. Aragones and A. Fournier, "Commande dynamique d'un robot omni-directionnel redondant," in *IEEE Conférence Internationale Francophone d'Automatique (CIFA)*, (France), 2002.