Let $\mathbf{g_1}$, $\mathbf{g_2}$ be the columns of the 3×2 matrix on the right-hand side of (45), then

$$\mathrm{rank}[\ \mathbf{g_1} \quad \mathbf{g_2} \quad [\mathbf{g_1}, \mathbf{g_2}]\ ] = 3. \tag{46}$$

Hence the posture variables are controllable.

When $b \neq 0$, let $\mathbf{g_1}$ and $\mathbf{g_2}$ be the columns of the 4×2 matrix on the right-hand side of (43), then

$$\mathrm{rank}[\ \mathbf{g_1} \quad \mathbf{g_2} \quad [\mathbf{g_1}, \mathbf{g_2}] \quad [\mathbf{g_2}, [\mathbf{g_1}, \mathbf{g_2}]]\ ] = 4. \tag{47}$$

Thus, ${}^{\mathrm{R}}\mathbf{p_E}$ is controllable. □

The control laws in actuator failure are proposed and demonstrated in [6].

VI. Conclusion

In this paper, the mobility augmentation method was proposed to produce omnidirectional motion with conventional wheeled bases with the degree of mobility less than three. Combined with a revolute/prismatic joint, some conventional wheeled bases with the degree of mobility less than three could produce omnidirectional motion if some conditions were satisfied. The structural conditions for mobility augmentation were derived and several examples were provided. Such a nonconventional omnidirectional mechanism did not suffer from typical problems that conventional omnidirectional mechanisms have.

One of the illustrated structures was further improved to develop the OmniKity robot series. A novel gear train was devised so that, in addition to the omnidirectional property, the mechanism provides a fault-tolerance property in case of the failure of any one of the three motors.

References

[1] G. Campion, G. Bastin, and B. D'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 47–62, 1996.

[2] S. L. Dickerson, "Control of an omni-directional robotic vehicle with mecanum wheels," in *National Telesystems Conf.*, 1991, pp. 323–328.

[3] L. Ferriere and B. Raucent, "ROLLMOBS, a new universal wheel concept," in *IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 1877–1882.

[4] R. Holmberg and O. Khatib, "Development and control of a holonomic mobile robot for mobile manipulation tasks," *Int. J. Robot. Res.*, vol. 19, no. 11, pp. 1066–1074, 2000.

[5] M.-J. Jung, H.-S. Shim, H.-S. Kim, and J.-H. Kim, "The miniature omnidirectional mobile robot OmniKity-I (OK-I)," in *IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 2686–2691.

[6] M.-J. Jung and J.-H. Kim, "Fault tolerant control strategy of OmniKity-III," in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 3370–3375.

[7] S. M. Killough and F. G. Pin, "Design of an omnidirectional and holonomic wheeled platform prototype," in *IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 84–90.

[8] P. F. Muir and C. P. Neuman, "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot," in *IEEE Int. Conf. on Robotics and Automation*, 1987, pp. 1772–1778.

[9] M. Wada and S. Mori, "Holonomic and omnidirectional vehicle with conventional tires," in *IEEE Int. Conf. on Robotics and Automation*, 1996, pp. 3671–3676.

[10] M. West and H. H. Asada, "Design and control of ball wheel omnidirectional vehicles," in *IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 1931–1938.

[11] B. Yi and W. K. Kim, "The kinematics for redundantly actuated omni-directional mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 2485–2492.

Optimal Mobile Robot Pose Estimation Using Geometrical Maps

Geovany Araujo Borges and Marie-José Aldon

***Abstract*—We propose a weighted least-squares (WLS) algorithm for optimal pose estimation of mobile robots using geometrical maps as environment models. Pose estimation is achieved from feature correspondences in a nonlinear framework without linearization. The proposed WLS approach yields optimal estimates in the least-squares sense, is applicable to heterogeneous geometrical features decomposed in points and lines, and has an $O(N)$ computation time.**

***Index Terms*—Heterogeneous features, nonlinear optimization, optimal 2-D pose estimation, weighted least-squares.**

I. Introduction

In mobile robot absolute localization, depending on the employed approach, we encounter the problem of estimating the 2-D transformation parameters (rotation and translation) that relate two sets of geometrical features extracted from two different maps. In the context of absolute localization, we have a global map $\mathcal{M}$, which may be continuously updated, and a local map $\mathcal{M}'$ which is composed of geometrical features extracted from exteroceptive sensor readings such as a video camera [1], a laser rangefinder [2], ultrasonic ranging [3], [4], or a combination of them [5]. Usually, maps are composed of heterogeneous geometrical features such as points, which correspond to corners and vertical edges in the environment, and lines which represent walls and sides of polygonal obstacles. Map building using different types of features allows a greater reliability for the pose estimation.

Most of the proposed methods which use geometrical features for map-based pose estimation are divided into three phases: pose prediction, feature matching between maps $\mathcal{M}$ and $\mathcal{M}'$, and pose estimation from the matched features. Pose prediction provides an initial guess of the vehicle's pose and an associated uncertainty measure, which are used as prior information for feature matching between $\mathcal{M}$ and $\mathcal{M}'$. From the feature correspondences, the estimation phase provides a pose estimate and an associated uncertainty measure which optimize some criterion. The work presented in this paper only concerns the pose estimation phase.

II. Background

The most used approach for map-based mobile robot pose estimation using geometrical features is the extended Kalman filter (EKF) [6], [7]. The formulation of the mobile robot localization problem using the EKF is elegant and allows one to fuse multisensory data. The problem is generally written as that of estimating the state of a discrete stochastic nonlinear dynamic system which evolves using proprioceptive sensor readings u as

$$z(k) = g(z(k-1), u(k)) + v(k) \tag{1}$$

with v modeling a Gaussian noise with distribution $N(\mathbf{0}, \boldsymbol{P}_v(k))$ (multivariate normal distribution with zero mean and covariance $\boldsymbol{P}_v(k)$).

Manuscript received February 14, 2001; revised November 7, 2001. This paper was recommended for publication by Associate Editor M. Buehler and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by Capes, Brasília, Brazil, under Grant BEX2280/97-3

The authors are with the Robotics Department, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, UMR CNRS/UM2, C55060 Montpellier, France (e-mail: borges@lirmm.fr; aldon@lirmm.fr).

Publisher Item Identifier S 1042-296X(02)01775-5.

The system state represents the vehicle's pose $z = (x, y, \theta)^T$, and $z(k)$, $k = 0, 1, \ldots$ is assumed to be a Gaussian random sequence. Equation (1) is usually derived from the kinematics model of the robot, where u is provided by motion encoders coupled to the robot axes. At some moment, exteroceptive measurements $\boldsymbol{y}'$ are taken following

$$\boldsymbol{y}'(k) = \boldsymbol{h}(z(k), y(k)) + \boldsymbol{w}(k). \tag{2}$$

In the case of map-based positioning, y is a vector containing the feature parameters of the global map $\mathcal{M}$ and $\boldsymbol{y}'$ is its local observation in $\mathcal{M}'$. Obviously, such observation depends on $\boldsymbol{z}$ as explicitly shown in (2). The measurement is perturbed by a Gaussian noise $\boldsymbol{w}$, following $N(\boldsymbol{0}, \boldsymbol{P}_w(k))$. In the above models, $\boldsymbol{g}$ and $\boldsymbol{h}$ are intrinsical nonlinear functions of the robot pose $\boldsymbol{z}$. The problem here becomes obtaining estimates of the first and second moments of $\boldsymbol{z}(k)$, $\hat{z}(k)$ and $\boldsymbol{P}_{\hat{z}}(k)$, respectively, which are sufficient to compute the distribution of $z(k)$.

In the EKF framework, system state estimation encompasses two phases: prediction and estimation. During the former, a prediction of the robot pose and of its associated covariance matrix is obtained by linearizing (1) as

$$\hat{z}(k|k-1) = \boldsymbol{g}(\hat{z}(k-1), u(k)) \tag{3}$$

$$\boldsymbol{P}_{\hat{\boldsymbol{z}}}(k|k-1) = \nabla \boldsymbol{g} \cdot \boldsymbol{P}_{\hat{\boldsymbol{z}}}(k-1) \cdot \nabla \boldsymbol{g}^T + \boldsymbol{P}_v(k) \tag{4}$$

with $\nabla \boldsymbol{g}$ being the derivative of g with respect to z evaluated at $\hat{z}(k-1)$. When the measurements arrive, the EKF estimation phase is performed following the innovation form [8]

$$\nu = \boldsymbol{y}'(k) - \boldsymbol{h}(\hat{z}(k|k-1), \boldsymbol{y}(k)) \tag{5}$$

$$\boldsymbol{S}(k) = \nabla \boldsymbol{h} \cdot \boldsymbol{P}_{\hat{z}}(k|k-1) \cdot \nabla \boldsymbol{h}^T + \boldsymbol{P}_w(k) \tag{6}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}_{\hat{\boldsymbol{z}}}(k|k-1) \cdot \nabla \boldsymbol{h}^T \cdot \boldsymbol{S}^{-1}(k) \tag{7}$$

$$\hat{z}(k) = \hat{z}(k|k-1) + \boldsymbol{K}(k) \cdot \nu \tag{8}$$

$$\boldsymbol{P}_{\hat{z}}(k) = \boldsymbol{P}_{\hat{z}}(k|k-1) - \boldsymbol{K}(k) \cdot \boldsymbol{S}(k) \cdot \boldsymbol{K}^T(k). \tag{9}$$

In (5)–(9), ν is the innovation and $\boldsymbol{S}(k)$ is its estimated covariance, $\boldsymbol{K}(k)$ is the Kalman filter gain, and $\nabla \boldsymbol{h}$ is the derivative of $\boldsymbol{h}$ with respect to $\hat{z}$ evaluated at $\hat{z}(k|k-1)$. $\boldsymbol{P}_w$ incorporates the measurement model uncertainties, as well as the uncertainties of $\boldsymbol{y}$ and $\boldsymbol{y}'$. These equations are obtained by applying the original linear Kalman filter formulation [9] to the linearized measurement model around the predicted pose $\hat{z}(k|k-1)$.

With a fast and recurrent form, this filter has been successfully used for real-time mobile robot localization. However, the user should be aware of some limitations of the EKF and of the known proposed solutions in order to achieve a good design.

- Due to the linearization of the measurement model (2), this filter is very sensitive to nonlinear aspects of $\boldsymbol{h}$ and gives suboptimal estimates. With such an approximation, neither $P_{\hat{\boldsymbol{z}}}$ corresponds to the actual covariance, nor is the EKF a minimum variance estimator [6]. The growth of the $\boldsymbol{P}_v(k)$ and $\boldsymbol{P}_w(k)$ matrices are proposed solutions to minimize the effects of the linearization on the nonconvergence of $\boldsymbol{P}_{\hat{\boldsymbol{z}}}$, resulting in suboptimal but consistent estimates ($\boldsymbol{P}_{\hat{\boldsymbol{z}}}(k) > \boldsymbol{P}_{\boldsymbol{z}}(k)$). This solution requires an offline tuning using experimental data.
- If the prediction $\hat{z}(k|k-1)$ is far enough from the actual value of z, nonconvergence problems may arise due to truncation errors. The iterated EKF [7] is a proposed solution.
- The errors induced by linearization (2) on the pose estimate as well as in the associated covariance matrix are not known in advance. An alternative approach consists of estimating $\boldsymbol{P}_w(k)$ and noise bias in the same way that the adaptive EKF [10]. Julier *et al.* [11] have proposed a more accurate filter which is able to achieve consistent covariance propagation through the estimation process.
- Since the filter assumes a Gaussian distribution of the estimate $\hat{z}$, multiple mode distributions cannot be directly handled. Such cases can arise from multiple correspondences in real cluttered environments as well as from the nonlinear aspect of the problem. One solution is to apply the multiple hypothesis Kalman filter formulation as shown in [7]. Monte Carlo localization is another approach recently developed which approximates the pose distribution as a large set of filters. It allows us to represent non-Gaussian multimodal distributions. In [12], there is an application using grid maps.
- The computing time of the EKF rapidly grows in $O(N^3)$, with N being the number of corresponding features. This is the consequence of the inversion of matrix $\boldsymbol{S}(k)$ in (7). A proposed solution is the use of the EKF sequential formulation [7], where measurements are sequentially presented to the filter at time k. However, the filter may not converge if the presentation sequence is not well chosen. In [13], the sequential formulation is successfully used, on which a special treatment for multiple correspondences has been applied. The sequential filter has an $O(N)$ computation time.

All the above issues contribute to the difficulty of fine tuning the EKF, as cited by Julier *et al.* in [11]. Some proposed alternatives deal with specific issues and give suboptimal estimates by using the linearized measurement model. When all operating conditions assure good predictions $\hat{z}(k|k-1)$ and *almost perfect* feature extraction for local map building, satisfactory results can be obtained with the EKF. However, such operating conditions cannot be guaranteed when navigating in highly cluttered and noncontrolled environments. We mean that in such environments false feature correspondences may occur as well as wheel slippage. As a consequence, the Gaussian assumption is no longer valid, and in such a case the filter presents no robustness. Furthermore, it is very difficult to evaluate which part of the estimation errors is due to the linearization procedure.

In this paper, we propose a new algorithm for two-dimensional (2-D) pose estimation that provides an optimal pose estimate given two sets of corresponding heterogeneous features. Using a unified representation for these features, the problem is formulated as a nonlinear weighted least-squares (WLS) optimization. This results in a batch, nonapproximated optimal pose estimator in $O(N)$ computing time growing. Previous versions of the proposed algorithm were presented in [14] and [15].

The problem discussed in this paper is formulated in Section III. Section IV introduces a unified feature representation for 2-D lines and points. The pose and uncertainty estimation procedures are derived in Section V and Section VI, respectively. A complexity analysis of the proposed solution is described in Section VII. Section VIII reports a simulation on which we discuss the limitations of the proposed approach. Experimental results in a cluttered real environment are presented in Section IX.

III. Problem Statement

It is established that the robot pose $\boldsymbol{z} = (\boldsymbol{t}^T, \theta)^T$, with $\boldsymbol{t} = (t_x, t_y)^T$, represents a 2-D coordinate transformation such that $\boldsymbol{t} \in \mathbb{R}^2$ and $-\pi < \theta \leq \pi$. These parameters relate a global 2-D coordinate system $\mathcal{XY}$ to another local coordinate system $X'Y'$. In the context of mobile robot localization, the maps $\mathcal{M}$ and $\mathcal{M}'$ are respectively attached to XY and $X'Y'$. Also, vector $\boldsymbol{t}$ corresponds to the 2-D robot position $(x, y)^T$ relative to the frame of the global map $\mathcal{M}$, and the orientation of the mobile robot with respect to the X axis is given by the angle θ.

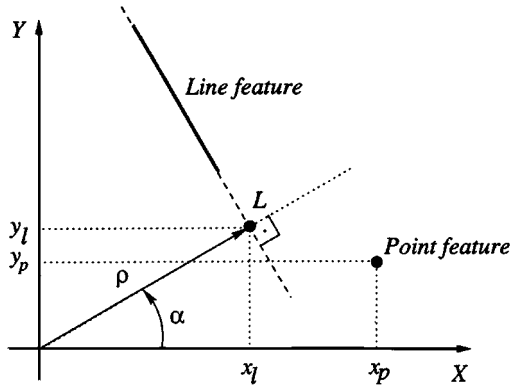

Fig. 1. Features description in the 2-D Euclidian space.

Thus, $(p_x, p_y)^T$ being the coordinates of a static point in XY and $(p'_x, p'_y)^T$ its coordinates in $X'Y'$, we have

$$\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \boldsymbol{R}(\theta) \cdot \left\{ \begin{pmatrix} p_x \\ p_y \end{pmatrix} - \boldsymbol{t} \right\} \tag{10}$$

with

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{11}$$

From $\mathcal{M}$ and $\mathcal{M}'$, we have two sets of N corresponding geometrical features named, respectively, $\mathcal{F} = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N\}$ and $\mathcal{F}' = \{\boldsymbol{f}'_1, \ldots, \boldsymbol{f}'_N\}$. These features may be lines and/or points. We suppose that the correspondence problem between features of the same type has been solved, i.e., features $\boldsymbol{f}_n$ and $\boldsymbol{f}'_n$ are corresponding. The correspondences are weighted by positive scalar factors μ_n. Each feature vector $\boldsymbol{f}_n$ ($\boldsymbol{f}'_n$) is associated with its uncertainty, represented by a 2×2 covariance matrix $\boldsymbol{P}_{\boldsymbol{f}_n}$ ($\boldsymbol{P}_{\boldsymbol{f}'_n}$).

IV. The Unified 2-D Feature Representation

We propose a new unified 2-D feature representation for points and lines in the Euclidean space (Fig. 1). It allows to embed lines and points in a unique WLS framework. Features are represented by a vector $\boldsymbol{f} = (x, y)^T$ in the global coordinate system XY ($\boldsymbol{f}'$ in the $X'Y'$ local coordinate frame), with:

- $(x, y)^T = (x_p, y_p)^T$ for point features;
- $(x, y)^T = (x_l, y_l)^T$ for line features [$(x_l, y_l)^T$ are the coordinates of the point L on the line, such that $x_l = \rho \cdot \cos(\alpha)$ and $y_l = \rho \cdot \sin(\alpha)$] (see Fig. 1).

For a more reliable pose estimation, we consider only the infinite line parameters. The end-points may be used as point features if they are sufficiently reliable.

For a given static feature in the environment, the relation between vectors $\boldsymbol{f}$ and $\boldsymbol{f}'$ in the two coordinate frames XY and $X'Y'$, respectively, is derived from (10). For points, these vectors are directly related by (10), resulting in

$$\boldsymbol{f}' = \boldsymbol{R}(\theta) \cdot \{\boldsymbol{f} - \boldsymbol{t}\}. \tag{12}$$

In order to develop the motion transformation between two lines, we use a parametric form which relates any point $\boldsymbol{p}$ belonging to a line with parameters $\boldsymbol{f}$ by means of a parameter k. In this way, it suffices to find the parameters $\boldsymbol{f}'$ of the transformed line knowing that the belonging points $\boldsymbol{p}'$ follow (12). Thus, we introduce the parametric line form

$$\boldsymbol{p}(k) = \boldsymbol{f} + k \cdot \boldsymbol{l}. \tag{13}$$

In (13), k is a scalar and $\boldsymbol{l} = (\sin(\alpha), -\cos(\alpha))^T$ is the unitary vector that defines the line direction. We can verify that $\boldsymbol{f}$ and $\boldsymbol{l}$ are orthogonal: $\boldsymbol{l}^T \cdot \boldsymbol{f} = 0$. The corresponding points $\boldsymbol{p}'(k)$ in $X'Y'$ are obtained by applying the coordinate frame transformation to $\boldsymbol{p}$ as

$$\boldsymbol{p}'(k) = \boldsymbol{R}(\theta) \cdot \{\boldsymbol{p}(k) - \boldsymbol{t}\} = \boldsymbol{R}(\theta) \cdot \{\boldsymbol{f} - \boldsymbol{t}\} + k \cdot \boldsymbol{R}(\theta) \cdot \boldsymbol{l}. \tag{14}$$

From (14), we can see that the line passes through the point $\boldsymbol{R}(\theta) \cdot \{\boldsymbol{f} - \boldsymbol{t}\}$ and has as unitary direction vector $\boldsymbol{l}' = \boldsymbol{R}(\theta) \cdot \boldsymbol{l}$ in the $X'Y'$ coordinate frame. Thus, $\alpha' = \alpha - \theta$ in $X'Y'$. From the definition of the line parameter vector, $\boldsymbol{f}'$ is given by the coordinates of the point on the transformed line that is closest to the origin of $X'Y'$. Thus, this point is parameterized by k' such that

$$\begin{aligned} k' &= \arg\min_k \left\| \boldsymbol{p}'(k) \right\|^2 \\ &= \arg\min_k \left\| \boldsymbol{R}(\theta) \cdot \{\boldsymbol{f} + k \cdot \boldsymbol{l} - \boldsymbol{t}\} \right\|^2 \\ &= \arg\min_k \left\| \boldsymbol{f} + k \cdot \boldsymbol{l} - \boldsymbol{t} \right\|^2. \end{aligned} \tag{15}$$

We can verify that $k' = \boldsymbol{l}^T \cdot (\boldsymbol{t} - \boldsymbol{f}) = \boldsymbol{l}^T \cdot \boldsymbol{t}$, since $\boldsymbol{f}$ and $\boldsymbol{l}$ are orthogonal. Thus, by applying k' to (14), the $\boldsymbol{f}' = \boldsymbol{p}'(k')$ parameter vector is

$$\boldsymbol{f}' = \boldsymbol{R}(\theta) \cdot \left\{ \boldsymbol{f} - (\boldsymbol{I}_2 - \boldsymbol{l} \cdot l^T) \cdot t \right\} \tag{16}$$

with $\boldsymbol{I}_2$ being the 2-D identity matrix. The explicit form of $\boldsymbol{I}_2 - \boldsymbol{l} \cdot l^T$ is

$$\boldsymbol{I}_2 - \boldsymbol{l} \cdot l^T = \begin{pmatrix} \cos^2(\alpha) & \sin(\alpha)\cos(\alpha) \\ \sin(\alpha)\cos(\alpha) & \sin^2(\alpha) \end{pmatrix}.$$

The similarity between (12) and (16) allows us to define a unified frame transformation as

$$\boldsymbol{f}' = \boldsymbol{R}(\theta) \cdot \left\{ \boldsymbol{f} - \boldsymbol{E} \cdot \hat{\boldsymbol{t}} \right\} \tag{17}$$

with

$$\boldsymbol{E} = \begin{pmatrix} a^2 & c \\ c & b^2 \end{pmatrix}.$$

The entries of the $\boldsymbol{E}$ matrix depend on the feature type represented by $\boldsymbol{f}$: $a = b = 1$ and $c = 0$ for a point feature, and $a = \cos(\alpha)$, $b = \sin(\alpha)$, and $c = a \cdot b$ for a line feature.

V. Optimal Pose Estimation

In this section, we present the mathematical development that results in optimal estimates $\hat{\boldsymbol{t}} = (\hat{t}_x, \hat{t}_y)^T$ and $\hat{\theta}$ given the feature correspondences $\mathcal{F}$ and $\mathcal{F}'$. Thus, we propose to find $\hat{\boldsymbol{t}}$ and $\hat{\theta}$, as the global minimum of the weighted cost function

$$J(\hat{z}) = \sum_{n=1}^{N} \mu_n \cdot d^2(\boldsymbol{f}_n, \boldsymbol{f}'_n, \hat{\boldsymbol{t}}, \hat{\theta}). \tag{18}$$

where the distance $d^2(\boldsymbol{f}_n, \boldsymbol{f}'_n, \hat{\boldsymbol{t}}, \hat{\theta})$ is given by

$$d^2(\boldsymbol{f}_n, \boldsymbol{f}'_n, \hat{\boldsymbol{t}}, \hat{\theta}) = \left\| \boldsymbol{r}_n \right\|_2^2. \tag{19}$$

In (19),

$$\boldsymbol{r}_n = \boldsymbol{f}'_n - \boldsymbol{R}(\hat{\theta}) \cdot \left\{ \boldsymbol{f}_n - \boldsymbol{E}_n \cdot \hat{\boldsymbol{t}} \right\} \tag{20}$$

is the nth residual, $\boldsymbol{r}_n$ being a nonconvex function of the transformation parameters, and $J(\hat{z})$ may have multiple minima. The procedure adopted here consists in finding all local minima $\hat{z}^*$ and using as pose estimate $\hat{z}$ the $\hat{z}^*$ satisfying a selection criterion. In doing so, as a first

condition, all local minima of $J(\hat{z})$ satisfy the following set of equations:

$$\frac{\partial J(\hat{z})}{\partial \hat{t}_x} = \frac{\partial J(\hat{z})}{\partial \hat{t}_y} = \frac{\partial J(\hat{z})}{\partial \hat{\theta}} = 0. \tag{21}$$

Further development of (21) results in

$$\boldsymbol{M}\hat{t} - \boldsymbol{N}\hat{\beta} = \boldsymbol{q} \tag{22}$$

$$\hat{\beta}^T \cdot \left(\boldsymbol{r} + \boldsymbol{S}\boldsymbol{N}^T\hat{\boldsymbol{t}}\right) = 0 \tag{23}$$

with $\hat{\beta}^T = (\cos(\hat{\theta}) \quad \sin(\hat{\theta}))$, and

$$\boldsymbol{M} = \sum_{n=1}^{N} \mu_n \begin{pmatrix} c_n^2 + a_n^4 & c_n(a_n^2 + b_n^2) \\ c_n(a_n^2 + b_n^2) & c_n^2 + b_n^4 \end{pmatrix} \tag{24}$$

$$\boldsymbol{N} = \sum_{n=1}^{N} \mu_n \begin{pmatrix} -a_n^2 x'_n - c_n y'_n & a_n^2 y'_n - c_n x'_n \\ -c_n x'_n - b_n^2 y'_n & c_n y'_n - b_n^2 x'_n \end{pmatrix} \tag{25}$$

$$q = \sum_{n=1}^{N} \mu_n \begin{pmatrix} c_n y_n + a_n^2 x_n \\ c_n x_n + b_n^2 y_n \end{pmatrix} \tag{26}$$

$$r = \sum_{n=1}^{N} \mu_n \begin{pmatrix} -x_n y'_n + y_n x'_n \\ -y_n y'_n - x_n x'_n \end{pmatrix} \tag{27}$$

$$\boldsymbol{S} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{28}$$

From (22) and (23), we obtain

$$\hat{\beta}^T \boldsymbol{\Phi} \hat{\beta} + \hat{\beta}^T \boldsymbol{\zeta} = 0 \tag{29}$$

with

$$\boldsymbol{\Phi} = \boldsymbol{S}\boldsymbol{N}^T\boldsymbol{M}^{-1}\boldsymbol{N} \tag{30}$$

$$\zeta = \boldsymbol{r} + \boldsymbol{S}\boldsymbol{N}^T\boldsymbol{M}^{-1}\boldsymbol{q}. \tag{31}$$

We can verify that $\boldsymbol{\Phi} = (\zeta - \boldsymbol{r})\boldsymbol{N}$, and that $\boldsymbol{N}^T\boldsymbol{M}^{-1}\boldsymbol{N}$ is a symmetric matrix since $\boldsymbol{M}$ is also symmetric. Let the entries of $\boldsymbol{\Phi}$ and $\boldsymbol{\zeta}$ be referred to as

$$\boldsymbol{\Phi} = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix}, \quad \zeta = \begin{pmatrix} \boldsymbol{\zeta}_1 \\ \boldsymbol{\zeta}_2 \end{pmatrix}.$$

Thus, $\boldsymbol{N}^T\boldsymbol{M}^{-1}\boldsymbol{N}$ being symmetric and given the form of matrix $\boldsymbol{S}$, we have $\Phi_{11} = -\Phi_{22}$. With this simple relation, further development of (29) results in

$$\Phi_{11}\cos(2\hat{\theta}) + \frac{(\Phi_{12} + \Phi_{21})}{2}\sin(2\hat{\theta}) + \zeta_1\cos(\hat{\theta}) + \zeta_2\sin(\hat{\theta}) = 0. \tag{32}$$

As presented in Appendix A, the trigonometric equation (32) has at most four solutions for $\hat{\theta}$, named $\hat{\theta}_i^*, i = 1, \ldots, 4$. Since these solutions satisfy (21), they are the rotation components of the local minima and maxima of J. The translation $\hat{\boldsymbol{t}}_i^*$ associated with each $\hat{\theta}_i^*$ is obtained from (22) and given by

$$\hat{\boldsymbol{t}}_i^* = \boldsymbol{M}^{-1}(\boldsymbol{N}\hat{\beta}_i^* + \boldsymbol{q}) \tag{33}$$

with $\hat{\beta}_i^* = (\cos(\hat{\theta}_i) \quad \sin(\hat{\theta}_i))^T$. These solutions respect the first condition given by (21), which is also satisfied by the local maxima. Thus, as a restrictive condition, we retain as candidate solutions to $\hat{\theta}$ only the minima which satisfy $\partial^2 J(\hat{z})/\partial\hat{\theta}_i^{*2} > 0$. Necessarily, they can be at most two with real values. If we have only one real minima, it is used as pose estimate. On the other hand, we have verified two local minima cases in symmetric map configurations. Thus, in such cases, the final decision may be based on the side of view of line features or on the minimum distance with respect to the predicted pose. The pose estimation can fail if the feature configuration allows a complete non-approximate solution (see the simulation in Section VIII). This can be verified in real time by a test on the conditioning of the $\boldsymbol{M}$ matrix.

VI. Pose Uncertainty Estimation

The uncertainty measure associated with $\hat{z}$ can be used as a validation gate for the pose estimate, or for global map updating. In order to estimate this uncertainty, we assume that all residuals $\boldsymbol{r}_n$ follow a multivariate normal distribution with zero mean. So, we consider that they are independently and identically distributed (i.i.d.) as $N(\boldsymbol{0}, \boldsymbol{P_r})$. We also take into account the features' uncertainty given by the covariance matrices $P_{\boldsymbol{f}_n}$ and $P_{\boldsymbol{f}'_n}$.

In order to estimate the covariance matrix $\boldsymbol{P}_{\hat{z}}$, we propagate the uncertainties associated with $\boldsymbol{r}_n$, $\boldsymbol{f}_n$, and $\boldsymbol{f}'_n$ to the pose estimate $\hat{z}$, according to the methodology described in [16]:

$$\boldsymbol{P}_{\hat{z}} = \left(\frac{\partial g}{\partial \hat{z}}\right)^{-1} \cdot \left(\boldsymbol{\Lambda_r} + \boldsymbol{\Lambda_f} + \boldsymbol{\Lambda_{f'}}\right) \cdot \left(\frac{\partial g}{\partial \hat{z}}\right)^{-1} \tag{34}$$

with $\boldsymbol{g} = \partial J/\partial \hat{z}$ and

$$\boldsymbol{\Lambda_r} = \sum_{n=1}^{N} \frac{\partial g}{\partial r_n} \cdot \boldsymbol{P_r} \cdot \frac{\partial g}{\partial r_n}^T \tag{35}$$

$$\boldsymbol{\Lambda_f} = \sum_{n=1}^{N} \frac{\partial g}{\partial \boldsymbol{f}_n} \cdot \boldsymbol{P}_{\boldsymbol{f}_n} \cdot \frac{\partial g}{\partial \boldsymbol{f}_n}^T \tag{36}$$

$$\boldsymbol{\Lambda_{f'}} = \sum_{n=1}^{N} \frac{\partial g}{\partial \boldsymbol{f}'_n} \cdot \boldsymbol{P}_{\boldsymbol{f}'_n} \cdot \frac{\partial g}{\partial \boldsymbol{f}'_n}^T. \tag{37}$$

$\boldsymbol{P}_{\boldsymbol{f}_n}$ and $\boldsymbol{P}_{\boldsymbol{f}'_n}$ are provided as inputs, and $\boldsymbol{P_r}$ is the estimated residual covariance matrix. We use the weighted sample covariance as an estimate of $\boldsymbol{P_r}$ given by

$$\boldsymbol{P_r} = \frac{\sum_{n=1}^{N} \mu_n (\boldsymbol{r}_n \cdot \boldsymbol{r}_n^T)}{\sum_{n=1}^{N} \mu_n}. \tag{38}$$

$\partial g/\partial \hat{z}$ being the Hessian of J with respect to $\hat{z}$, this matrix is nonsingular on the local minima $\hat{z}$ of the cost function. Thus, its reciprocal always exists and is used to compute $\boldsymbol{P}_{\hat{z}}$. Equation (34) has been obtained by assuming independence between $\boldsymbol{r}_n$, $\boldsymbol{f}_n$, and $\boldsymbol{f}'_n$. However, we can see clearly from (20) that residuals and features $\boldsymbol{f}_n$ and $\boldsymbol{f}'_n$ are correlated. In the above development, we have assumed independence only to easily compute $\boldsymbol{P}_{\hat{z}}$. Furthermore, $\boldsymbol{P}_{\hat{z}}$ may take into account only the feature uncertainties by doing $\boldsymbol{P_r} = \boldsymbol{0}$. Instead, we prefer to consider the residuals because they embody spurious effects such as bias on feature parameters and matching errors.

VII. Algorithm Complexity

The WLS algorithm presents an $O(N)$ polynomial complexity. With respect to the batch version of the EKF which is $O(N^3)$, the WLS is far less complex. On the other hand, an $O(N)$ polynomial complexity is also achieved by the sequential version of the EKF which uses the recursive form of the EKF to treat the measurements sequentially, by assuming that all measurements are independent.

In Fig. 2, the computing time of the C language implementations of these algorithms was precisely measured in a real-time kernel, the RTX produced by Venturcom Inc., running on a PIII 450-MHz CPU. The corresponding features were randomly generated. For medium size maps with 0 corresponding features, the WLS and the sequential EKF present almost the same computing time, but they are three orders of

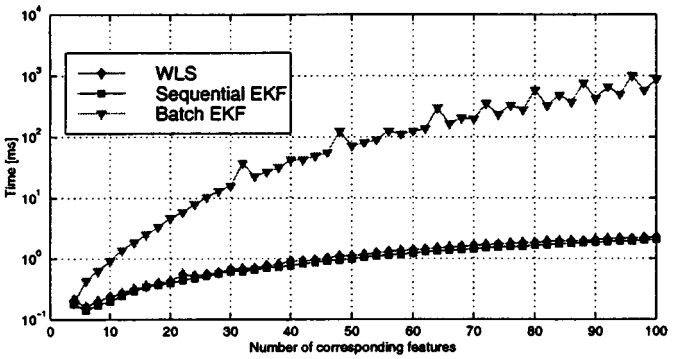

Fig. 2. WLS and EKF computing times in function of the number of corresponding features.

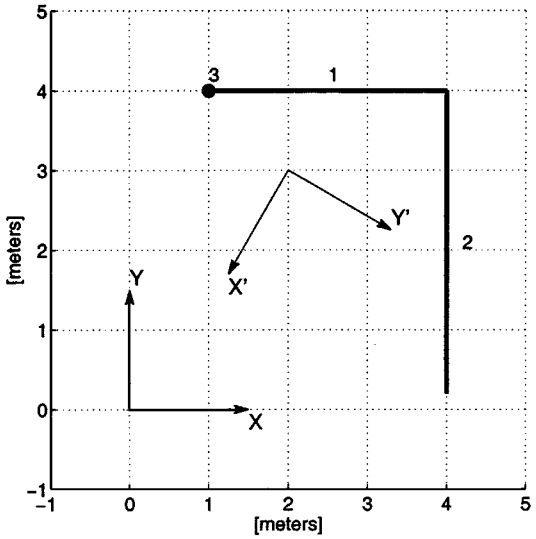

Fig. 3. The two lines and one point problem.

magnitude faster than the batch EKF. This results in a faster pose estimation, even with large environment maps. It must be pointed out that in a real-time localization system the localization task usually has a lower priority than control and data acquisition tasks. It means that the execution time for the EKF may be greatly increased when higher priority tasks get CPU control during its execution. Thus, the use of the batch EKF becomes prohibitive for a large number of feature correspondences in real-time mobile robot localization.

VIII. Simulation

In Fig. 3, we present the *two lines and one point problem*. The coordinate systems XY and $X'Y'$ are related by $\boldsymbol{t} = (2\text{ m}, 3\text{ m})$ and $\theta = 2\pi/3$ rad. This scenario can be found in real indoor mobile robot localization when a corner composed by the intersection of two walls (features 1 and 2) is detected, and only one of the wall extremities (feature 3) is reliably observed. This problem is not separable, i.e., it is not possible to estimate the frame transformation parameters from only one type of feature. An intuitive hybrid procedure may be firstly to reason only with the lines and find the two possible solutions which differ from π rad in rotation, and secondly to test each solution for the point and choose the one with the lowest residual. Nevertheless, it is not possible to extend this procedure to any greater number of features, and its estimates are suboptimal if the features do not match exactly.

Fig. 4. The Omni mobile robot.

The unified feature vectors for the problem in Fig. 3 are

$$\boldsymbol{f}_1 = \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \boldsymbol{f}_2 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \boldsymbol{f}_3 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$
$$\boldsymbol{f}'_1 = \begin{pmatrix} \frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{pmatrix}, \boldsymbol{f}'_2 = \begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix}, \boldsymbol{f}'_3 = \begin{pmatrix} \frac{\sqrt{3}+1}{2} \\ \frac{\sqrt{3}-1}{2} \end{pmatrix}.$$

The WLS method solves this problem yielding $\hat{\boldsymbol{t}} = (2\text{ m}, 3\text{ m})$ and $\hat{\theta} = 2\pi/3$ rad. In this simulation, we have employed equal correspondence weights $\mu_1 = \mu_2 = \mu_3 = 1$. This example illustrates the ability of our algorithm to solve the pose estimation problem with a minimum number of features. However, the problem must be well conditioned. This would not be the case if the point feature was not detected and if the only used features were the lines. This problem can be avoided by using the lines' intersection point as a feature. In this case, we have multiple minima of the cost function, with angles $-\pi/3$ rad and $2\pi/3$ rad. As discussed in Section V, the final decision would be taken based on the distance of the estimates with respect to a predicted pose, or by considering the lines' observation side. It is clear that a solution could not be given for two parallel lines.

IX. Experimental Evaluation

This experimental evaluation allowed us to compare the performance of the WLS and EKF algorithms in a real cluttered environment. It was carried out with our omnidirectional mobile robot Omni shown in Fig. 4. This platform is equipped with a laser rangefinder, a video camera, a high-precision gyrometer, and optical encoders associated with the motion axes. We have a PIII 450-MHz CPU-based computer installed on the robot, running Windows NT 4.0 with the real-time extension RTX. The global map used for this experimentation shown in Fig. 5 was hand built from a set of superposed range images. The initial robot pose was at the origin of the global frame. Ninety-five range images were acquired with the robot in motion during 95 s. The reference poses, shown as small squares (with some pose indexes), are provided by a combined gyrometer-encoders dead-reckoning technique. The high accuracy of this method allows us to use it as reference

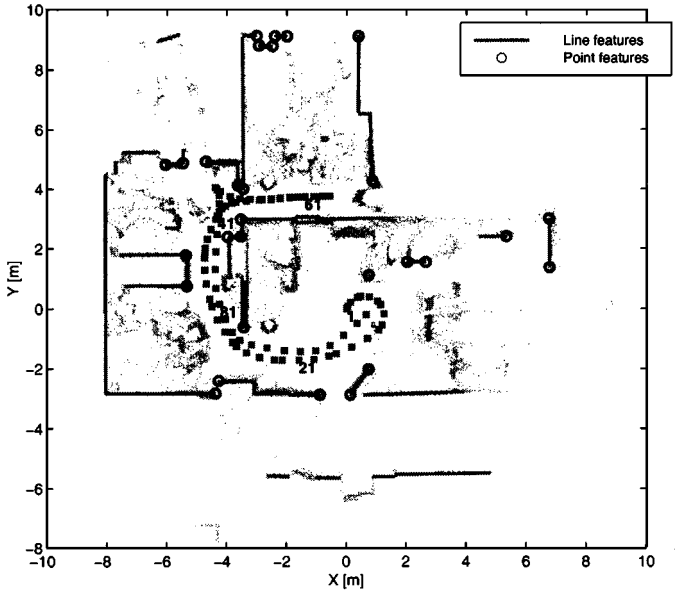

Fig. 5. The global map of the experimental environment. The superposed scans are shown to illustrate how cluttered the environment is.

for the limited time duration of this experiment. As the range images were acquired in motion, the deformation of these images was corrected in real time using the encoders data [14]. The line features of the local maps were extracted using a split-and-merge algorithm derived from the fuzzy C-means [17]. Point features are reliable breakpoints located at the extremities of the lines, and some ellipsoidal clusters corresponding to small obstacles. The unified feature representation was used for both algorithms. In these experiments, video images were not used. In the sequel, $\hat{z}_k(t|t-1)$ and $\hat{z}_k(t)$ are the predicted and estimated poses corresponding to the kth pose index.

In order to compare the EKF and the WLS under the same conditions, this evaluation was carried out off-line by using real data, i.e., laser range images, encoder readings, and reference poses at each pose index k, $k = 1, \ldots, 95$.

We have performed three experiments, denoted *A*, *B* and *C*, as follows.

- In experiment *A*: $\hat{z}_k(t|t-1)$ is computed from real encoder measurements, which method is far less precise than the reference pose. This experiment allows us to evaluate the algorithms performances when the prediction phase uses only odometry;
- In experiment *B*: $\hat{z}_k(t|t-1)$ is the reference pose contaminated by a simulated additive Gaussian noise with covariance matrix $\boldsymbol{P}_{\hat{z}_k}(t|t-1) = \mathrm{diag}\,(\,(0.25\ \mathrm{m})^2 \quad (0.25\ \mathrm{m})^2 \quad (3^\circ)^2\,)$. Here, the prediction uncertainty is larger than in experiment *A*.
- In experiment *C*: $\hat{z}_k(t|t-1)$ is the reference pose perturbed by two random noises: one Gaussian noise with covariance matrix $\boldsymbol{P}_w = \mathrm{diag}\,(\,(0.15\ \mathrm{m})^2 \quad (0.15\ \mathrm{m})^2 \quad (2^\circ)^2\,)$, and another noise with a uniform distribution in the range $[-0.3$ m, 0.3 m] for $\hat{x}_k(t|t-1)$ and $\hat{y}_k(t|t-1)$, and in the range $[-3^\circ, 3^\circ]$ for $\hat{\theta}_k(t|t-1)$. The Gaussian noise simulates the uncertainty propagated by odometry, and the uniform noise represents wheel slippage or other non-Gaussian error sources. The map matching and pose estimation procedures have knowledge of the Gaussian effect only. This experiment evaluates the robustness of the different approaches when errors on $\hat{z}_k(t|t-1)$ do not follow a Gaussian distribution.

For the kth pose index of the above experiments, the set of corresponding features is found by applying the classical Mahalanobis test to all possible feature correspondences between $\mathcal{M}$ and $\mathcal{M}'$ given the predicted pose $\hat{z}_k(t|t-1)$.

The conditions of experiments *B* and *C* require robustness of the pose estimation approaches. A fast robust iterated version of the WLS is obtained by using the following weighting function which is updated at each iteration:

$$\mu_n = \exp\left(-\frac{\|\boldsymbol{r}_n\|}{\eta}\right) \tag{39}$$

with $\eta = -2.6 \cdot median(\|r_1\|, \ldots, \|r_N\|)/\ln(0.1)$, $\boldsymbol{r}_n = \boldsymbol{f}'_n - \boldsymbol{h}(\hat{z}, \boldsymbol{f}_n)$ is the nth residual and η determines the acceptation window width.

For the first iteration, all weights are equal to 1. In this way, the robust WLS first computes a pose estimate which may be better then the $\hat{z}_k(t|t-1)$ provided by odometry. In our implementation, the maximum number of iterations is fixed at 5. During iterations, only the pose estimate is computed. Its associated covariance matrix is computed using the weights of the last iteration. This allows us to obtain a faster algorithm. If we use a robust function such as the Huber's weighting function [18], we obtain an iterated M-estimator. However, in order to converge, the M-estimator may require a large number of iterations. The proposed iterated estimator has presented satisfactory results in our experiments.

Table I presents the mean absolute errors (MAEs) of the algorithms for the three experiments (best results are in bold font). The MAE of ε_θ is almost the same for the two algorithms. However, the robust WLS gave better results for t_x and t_y estimation in all experiments. The results of the robust WLS are still better for experiments *B* and *C*. In these experiments, the theoretical foundations on which the EKF is based are not verified. In experiment *B*, $\hat{z}_k(t|t-1)$ is far from the true pose, and truncation errors caused by the linearization of the measurement model jeopardize the EKF performance. In experiment *C*, the uncertainties on $\hat{z}_k(t|t-1)$ are not Gaussian. Even if the differences do not seem very significant, it should be pointed out that they correspond to the mean values. Moreover, in a system based on feature tracking, an increase of the estimation errors at time step k is propagated to the next estimation cycle at $k+1$ by odometry and may result in the failure of the pose tracking.

In these experiments, we have used a handbuilt global map. Thus, it is likely that feature correspondences are not perfect and this may inject extra non-Gaussian noise to the pose estimation system. In order to obtain lower estimation errors with the EKF, extra artificial noise has been added in feature correspondences. This results in final estimates which are closer to the prediction, and observations have less influence on pose updating. Without adding such an artificial noise, our first evaluations led the filter to divergence.

X. Concluding Remarks

We have presented in this paper an absolute pose estimation algorithm for map-based mobile robot localization. The absolute pose estimate is optimal in the weighted least-squares sense. This method does not suffer from the nonconvergence problems associated with linearization as in the classical EKF approach. The computational time of the proposed solution is $O(N)$, N being the number of feature correspondences. The batch EKF has an $O(N^3)$ computational cost. Hence, the algorithm presented in this paper is an alternative to achieve more accurate mobile robot localization which is a necessary but not a sufficient condition for autonomous navigation. An experimental comparison of the WLS with the EKF algorithm has been reported, with a robust approach to compute the correspondence weights in the WLS. Since the algorithm only needs as input a set of corresponding features, its use for mobile robot relocalization is straightforward by using a global

TABLE I
EXPERIMENTAL EVALUATION: MAES FOR ROBUST WLS AND EKF ALGORITHMS.

	robust WLS			EKF		
	ε_{t_x}	ε_{t_y}	ε_θ	ε_{t_x}	ε_{t_y}	ε_θ
Experiment A	**6.88** cm	**5.77** cm	1.26 °	7.54 cm	8.21 cm	**1.06** °
Experiment B	**8.82** cm	**9.43** cm	1.47 °	12.15 cm	15.23 cm	**1.35** °
Experiment C	**8.61** cm	**7.58** cm	**1.13** °	11.63 cm	12.62 cm	1.22 °

matching approach to find such correspondences. Furthermore, an iterated reweighted M-estimator can be implemented by using robust weighting functions to compute the feature correspondence weights. Since such estimators are well known to be robust against outliers, which may result from false matches, we would expect more accurate pose estimation even in the occurrence of wheel slippage or nonsystematic odometric errors. In Section VIII, we have shown a simulation which illustrates the limitation of the proposed approach to compute an estimate in a noncomplete case. In such cases, failure can be detected and an alternative recursive and fast approach such as the sequential EKF can be used.

Finally, some additional considerations should be noted. This is a batch approach, not a recursive one. As a consequence, a pose estimate is only obtained after processing all current measurements. Thus, one could claim that the use of a batch approach instead of the sequential EKF filter is not very exciting when measurements are sequential (for example, with a laser rangefinder). In such a case, we agree that raw data are acquired sequentially. However, geometrical feature extraction is in most cases iterative and local map building using heterogeneous features is essentially a batch process. Therefore, even when raw data are acquired sequentially, the local map features are only available after the complete sensor data processing. A recent example is the work of Arras *et al.* [13]. In this paper, as mentioned in Section II, the EKF formulation was successfully used with a multisensory system composed of a video camera and a laser rangefinder. The processing of measurements is done sequentially, but only after feature extraction. So, in this case, the choice of the sequential filter is based only on the real-time requirements of the system, not on the manner on which data are acquired.

Recently, we have developed a multisensory localization system similar to that in [13], but our method uses explicitly local environment maps and our pose estimation is provided by the robust WLS version. Both implementations are more immune to robot motion when performing dynamic localization [15] (also referred as on-the-fly localization in [13]). In our case, multiple feature correspondences are accepted (they are very common), and it is the role of the pose estimator to give a larger weight to the most likely correspondences as the number of iterations increase. We have obtained promising results in long corridors at reasonable robot speeds, even in the presence of moving obstacles.

APPENDIX
SOLVING THE ROTATION TRIGONOMETRIC EQUATION

The roots $\hat{\theta}$ of (32) are the rotation components $\hat{\theta}_i^*$ of all local minima of the cost function J [see (18)]. By using Maple, a commercial symbolic mathematical software produced by Waterloo Maple, Inc., (32) has four roots $\hat{\theta}_i^*$, with $i = 1, \ldots, 4$, calculated as

$$\hat{\theta}_i^* = \arctan\left(-\frac{2\Phi_{11}x_i^2 - \mathbf{\Phi}_{11} + \zeta_1 x_i}{(\mathbf{\Phi}_{12} + \mathbf{\Phi}_{21})x_i + \zeta_2}, x_i\right) \tag{40}$$

where the x_i's are the roots of the following fourth-order polynomial:

$$\alpha_4 x^4 + \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0 = 0. \tag{41}$$

The polynomial (41) has real coefficients computed from the entries of $\mathbf{\Phi}$ and ζ as

$$\begin{aligned}
\alpha_4 &= 4\mathbf{\Phi}_{11}^2 + (\mathbf{\Phi}_{12} + \mathbf{\Phi}_{21})^2 \\
\alpha_3 &= 4\zeta_1\mathbf{\Phi}_{11} + 2\zeta_2(\mathbf{\Phi}_{12} + \mathbf{\Phi}_{21}) \\
\alpha_2 &= -4\mathbf{\Phi}_{11}^2 + \zeta_2^2 + \zeta_1^2 - (\mathbf{\Phi}_{12} + \mathbf{\Phi}_{21})^2 \\
\alpha_1 &= -2(\zeta_1\mathbf{\Phi}_{11} + \zeta_2(\mathbf{\Phi}_{12} + \mathbf{\Phi}_{21})), \\
\alpha_0 &= \mathbf{\Phi}_{11}^2 - \zeta_2^2.
\end{aligned}$$

In our implementation, the roots of (41) are computed by using an eigenvalue approach presented in [19].

REFERENCES

[1] A. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 969–978, Dec 1998.

[2] G. Borghi and V. Caglioti, "Minimum uncertainty explorations in the self-localization of mobile robots," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 902–911, Dec 1998.

[3] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 376–382, June 1991.

[4] J. L. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," in *IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 674–680.

[5] J. Neira, J. D. Tardós, J. Horn, and G. Schmidt, "Fusing range and intensity images for mobile robot localization," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 76–84, Feb 1999.

[6] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[7] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1987.

[8] M. A. Abidi and R. C. Gonzalez, Eds., *Data fusion in Robotics and Machine Intelligence*. New York: Academic, 1992.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Ser. D: J. Basic Eng*, vol. 82, pp. 35–45, 1960.

[10] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 219–229, Apr. 1999.

[11] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the non linear transformation of means and covariances in filters and estimators," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 477–482, Mar. 2000.

[12] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell. J.*, vol. 128, no. 1-2, pp. 99–141, May 2001.

[13] K. O. Arras, N. Tomatis, B. T. Jensen, and R. Siegwart, "Multisensor on-the-fly localization: Precision and reliability for applications," *Robot. Autonomous Syst.*, vol. 34, pp. 131–143, 2001.

[14] G. A. Borges, M.-J. Aldon, and T. Gil, "An optimal pose estimator for map-based mobile robot dynamic localization: Experimental comparison with the EKF," in *IEEE Int. Conf. Robot. Automation*, 2001, pp. 1585–1590.

[15] G. A. Borges and M.-J. Aldon, "Design of a robust real-time dynamic localization system for mobile robots," in *Proc. 9th Int. Symp. on Intelligent Robotic Systems*, 2001, pp. 425–434.

[16] R. M. Haralick, "Propagating covariances in computer vision," in *Int. Conf. Pattern Recognition*, 1994, pp. 493–498.

[17] G. A. Borges and M.-J. Aldon, "A split-and-merge segmentation algorithm for line extraction in 2-D range images," in *15th Int. Conf. on Pattern Recognition*, Sep 2000, pp. 441–444.

[18] P. J. Huber, *Robust Statistics*, 1st ed. New York: Wiley, 1981.
[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerial Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

Control of Cooperating Mobile Manipulators

Thomas G. Sugar and Vijay Kumar

***Abstract*—We describe a framework and control algorithms for coordinating multiple mobile robots with manipulators focusing on tasks that require grasping, manipulation and transporting large and possibly flexible objects without special purpose fixtures. Because each robot has an independent controller and is autonomous, the coordination and synergy are realized through sensing and communication. The robots can cooperatively transport objects and march in a tightly controlled formation, while also having the capability to navigate autonomously. We describe the key aspects of the overall hierarchy and the basic algorithms, with specific applications to our experimental testbed consisting of three robots. We describe results from many experiments that demonstrate the ability of the system to carry flexible boards and large boxes as well as the system's robustness to alignment and odometry errors.**

***Index Terms*—Compliance, locomotion and grasping, mobile robot cooperation.**

I. Introduction

We address the coordination of a small team of mobile manipulators that cooperatively perform such manipulation tasks as grasping a large, flexible object and transporting it in a two dimensional environment with obstacles. Such a system of robots is useful in material handling applications where there are no special purpose material handling devices (for example, conveyors, fixtures or pallets are not present). In particular, when the team is remotely controlled or supervised, the system can be used for the clean-up of hazardous waste material [1].

We focus on tasks that simply cannot be performed by a single mobile robot. As examples, consider the transportation of the large box or the flexible board shown in Fig. 1. The object, a box or a flexible board in our experiments, is large enough that one platform cannot carry it by itself without expensive, special purpose tooling that is specific to the object. In our approach, two or more robots are coordinated to accomplish the task of carrying an object. We demonstrate four key features of our system: (a) coordination and cooperation between tightly coupled mobile robots; (b) palm-like grasping of an object; (c) decentralized control of cooperating robots; and (d) robust grasping and transportation.

The physical interaction between the robots and the tight coupling required of autonomous controllers pose many challenging engineering problems including:

1) the manipulators must be capable of controlling the grasp forces in a robust fashion;
2) there needs to be an efficient way of communicating and sharing information in real time;
3) it should be possible to organize the robots differently for different tasks forcing the controllers to be independent and yet able to function in a tightly coupled architecture when carrying objects;
4) the robots must coordinate their trajectories in order to maintain a desired formation while maintaining the grasp; Unlike the task of pushing a box, the robots must maintain a formation while grasping and *carrying* the object;
5) the team must be robust with respect to errors that include robot positioning errors and modeling errors.

There is extensive literature on the subject of coordinating a group of robots. The behavior based control paradigm of [2]–[4] has been shown to be successful in controlling a large team of loosely coupled robots. It is possible to synthesize an impressive array of group behaviors [5] and coordinate robots for tasks like cooperative pushing [6]–[8], clustering in formations [9] and exploration [10] using variations of this basic approach. However, it is not clear that this approach can be used in its current form to maintain a tight formation and control grasp forces for holding and transporting objects.

When two or three robots are tightly coupled together in a specified formation for a specified manipulation task, the control problem is well-defined and the feedback control laws for coordinating a small team of robots are reasonably well understood [11]–[13]. The feedback laws for coordinated control of manipulation and locomotion are discussed by [14], [15]. Our work differs from such approaches in three respects.

In contrast to approaches in which the planning and control problems for the multirobot team are done centrally [16], our goal is to decentralize the control and planning to the extent possible. Our approach to decentralized control relies on the *decoupling* of the subproblems of controlling locomotion and manipulation [11], [17]. Second, our formulation will allow for changes in formation, as well as changes in the composition of the team. Third, our formulation allows for a small number of robots, although there are many unresolved questions about the optimal organization of large teams of robots.

Finally, it is important to note that our method for decoupling the control issues in manipulation and locomotion relies on a novel manipulator design. Ideally, we want each mobile manipulator to be equipped with an end effector that allows it to exert controlled forces and moments on the object and can accommodate small position and orientation errors resulting from the use of nonholonomic robots. However, the control of contact forces and moments is well known to be notoriously difficult [18]. We propose a parallel manipulator design in which the control of contact interactions are accomplished via a set of inexpensive position controlled motors. Further, as shown in the paper, we will only require one of the two or three team members to be equipped with an actively controlled end effector. While the details of the design of the manipulator are available elsewhere [19], we will review the basic principles in as much as they relate to the performance of the system.

An overall framework for the system is described. The experimental system and control architecture is given next. Finally, multiple experiments are presented that demonstrate the ability of our system to transport different objects.

II. The Team of Robots

A. Organization

In this section, we briefly discuss the organization of the team of mobile manipulators and outline the basic assumptions for this work. We

Manuscript received September 25, 2000; revised August 23, 2001. This paper was recommended for publication by Associate Editor L. Kavraki and Editor I. Walker upon evaluation of the reviewers' comments.

T. G. Sugar is with the Department of Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: thomas.sugar@asu.edu.).

V. Kumar is with the GRASP Laboratory, University of Pennsylvania. Philadelphia, PA 19104-6228 USA.

Publisher Item Identifier S 1042-296X(02)01375-7.