

Localization and trajectory following for multi-body wheeled mobile robots

Olivier Lefebvre Florent Lamiroux

LAAS-CNRS

7 avenue du Colonel Roche
31077 Toulouse cedex 4, France

olefebvr@laas.fr florent@laas.fr

Abstract— Autonomous navigation for wheeled mobile robots generally requires to plan a trajectory and to follow it while performing localization in the environment. In order to execute long-range motion, dead-reckoning localization is not precise enough and the robot must use landmark-based localization. Landmark-based localization can produce discontinuities in the robot position estimation, as it is well known with GPS systems, and perturb the trajectory following. If the robot is able to converge smoothly towards the trajectory, then these perturbations are easily managed. Otherwise, if the robot is more complex (for instance a multi-body mobile robot subject to several nonholonomic constraints) and navigates in a cluttered environment, we show that these perturbations can lead to collisions. The contribution of this paper is to state theoretically the problem and to propose a practical solution to trajectory following for multi-body wheeled mobile robots using landmark-based localization. This solution has been tested on a real robot towing a trailer.

I. INTRODUCTION

The issue of autonomous navigation for wheeled mobile robots has been extensively addressed in the last two decades. Research on nonholonomic path planning and nonholonomic systems control has reached a state of maturity, and there are numerous examples of mobile robots used in various contexts. In order to perform safe autonomous navigation on long-range motions, these robots also implement an obstacle avoidance method and landmark-based localization. The most successful examples are “museum guide” robots, and are generally round shape robots with unicycle-like kinematics [1], [2], [3].

Autonomous navigation for multi-body wheeled mobile robots navigating in cluttered environments has potential applications in automatic parking, driver assistance and trajectory verification for trucks for instance. But these systems and the applications envisaged induce new difficulties. Path planning for these systems is often time consuming and a two step approach is generally adopted. In a first step, a trajectory is planned in a model of the environment and in a second step, the trajectory is executed, while the robot is performing localization in a map of the environment and is reactively avoiding obstacles.

In this paper, we address the problem of trajectory following with landmark-based localization, while detecting collision. We claim that for multi-body wheeled mobile robots (a more precise characterization of the systems of interest is given later), landmark-based localization induce some perturbations in the trajectory following process, that can lead to collisions even with a reactive obstacle avoidance strategy.

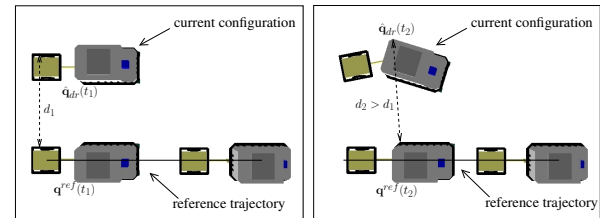


Fig. 1. For a multi-body wheeled mobile robot following a trajectory using a closed-loop feedback controller, the distance to the reference trajectory (a straight line on the figure) may increase before decreasing. The gap between the robot configuration and its reference trajectory might be due to a localization discontinuity.

Numerous previous works are aware of possible discontinuities in the landmark-based localization process and on their influence on the trajectory following process. But in these cases, the perturbation of the trajectory following is small because the robot has simple kinematics, and the problems can be ignored or solved by an adequate control law (see [4] for instance) or by growing the robot at the motion planning step.

To the best of our knowledge, the full consequences of localization discontinuities have never been directly addressed, especially in the challenging applications of multi-body wheeled mobile robots navigating in cluttered environments. Our contribution in this paper is to state theoretically the problem and to propose a practical solution.

The paper is organized as follows. In section II we give some general definitions used in the article and we highlight the specificity of trajectory following for multi-body wheeled vehicles. In section III we show that for a robot using a localization process providing a continuous position estimation (such as dead-reckoning), collision-free navigation can be guaranteed. Then in section IV we explain why the integration of landmark-based localization in the trajectory following process can lead to collisions for the systems of interest. Eventually, in section V, we propose an architecture for safe navigation of multi-body wheeled mobile robots and an algorithm to manage the perturbations of the landmark-based localization on the trajectory following process. We present some experimental results on robot Hilare2 towing a trailer, which performs long-range navigation in an environment with narrow passages.

II. TRAJECTORY FOLLOWING

A. Definitions

A robot is a kinematic chain, the configuration of which is denoted $\mathbf{q} \in \mathcal{C}$, where \mathcal{C} represents the configuration space of the robot. Without loss of generality we consider that the robot kinematic chain has a rigid body root whose pose in the plane is denoted $\mathbf{x} \in SE(2)$. The other internal configuration variables, for instance trailer orientation or configuration of an arm mounted on the platform, are denoted $\mathbf{q}_{int} \in \mathcal{C}_{int}$ and are expressed relatively to the root body:

$$\mathbf{q} = (\mathbf{x}, \mathbf{q}_{int}) \in \mathcal{C} = SE(2) \times \mathcal{C}_{int}$$

We define the distance between two configurations \mathbf{q}_1 and \mathbf{q}_2 as:

$$d_{\mathcal{C}}(\mathbf{q}_1, \mathbf{q}_2) = \max_{p \in \mathcal{R}} \|\mathbf{q}_1(p) - \mathbf{q}_2(p)\|, \quad (1)$$

where \mathcal{R} denotes the subset occupied by the robot. Then let \mathbf{x}_1 and \mathbf{x}_2 be two elements of $SE(2)$. We define the distance between \mathbf{x}_1 and \mathbf{x}_2 as:

$$d_{SE(2)}(\mathbf{x}_1, \mathbf{x}_2) = \max_{\mathbf{q}_{int} \in \mathcal{C}_{int}} d_{\mathcal{C}}((\mathbf{x}_1, \mathbf{q}_{int}), (\mathbf{x}_2, \mathbf{q}_{int})) \quad (2)$$

To any rigid body transformation $m \in SE(2)$ we associate a mapping from \mathcal{C} into \mathcal{C} that we denote by the same notation:

$$m : \begin{array}{l} SE(2) \times \mathcal{C}_{int} \rightarrow SE(2) \times \mathcal{C}_{int} \\ (\mathbf{x}, \mathbf{q}_{int}) \mapsto m \cdot \mathbf{q} = (m \cdot \mathbf{x}, \mathbf{q}_{int}) \end{array} \quad (3)$$

Thus m moves the root body of the robot while keeping internal configuration variables unchanged.

We define the distance between a configuration \mathbf{q} and an obstacle as:

$$d_{\mathcal{O}}(\mathbf{q}, \mathcal{O}) = \min_{\mathbf{p} \in \mathcal{R}, \mathbf{o} \in \mathcal{O}} \|\mathbf{q}(\mathbf{p}) - \mathbf{o}\|, \quad (4)$$

where \mathcal{O} denotes a subset of the workspace representing an obstacle.

1) *Dynamic Control System:* The robot kinematics is modeled as a driftless dynamic system:

$$\dot{\mathbf{q}} = \sum_{i=1}^k u_i \mathbf{X}_i(\mathbf{q}) \quad (5)$$

where $\mathbf{u} = (u_1, \dots, u_k)$ is the input vector and $\mathbf{X}_1, \dots, \mathbf{X}_k$ are the control vector fields.

Definition 1: An admissible trajectory over an interval $[0, T]$ is a couple (\mathbf{q}, \mathbf{u}) where \mathbf{q} is a piecewise C^2 mapping¹ from $[0, T]$ into \mathcal{C} and \mathbf{u} , called the input function of the trajectory, is a piecewise C^1 mapping from $[0, T]$ to \mathbb{R}^k satisfying:

$$\forall t \in [0, T] \quad \frac{d\mathbf{q}}{dt}(t) = \sum_{i=1}^k u_i(t) \mathbf{X}_i(\mathbf{q}(t))$$

These kinds of dynamic systems respect the following symmetry property:

Property 1: The control vector fields are independent of the choice of the reference frame:

$$\forall m \in SE(2) \text{ and } \forall \mathbf{q} \in \mathcal{C} \quad \mathbf{X}(m \cdot \mathbf{q}) = T_{\mathbf{q}} m \mathbf{X}(\mathbf{q})$$

¹To make notation lighter, we use the same notation for trajectory and configuration, and for input and input function.

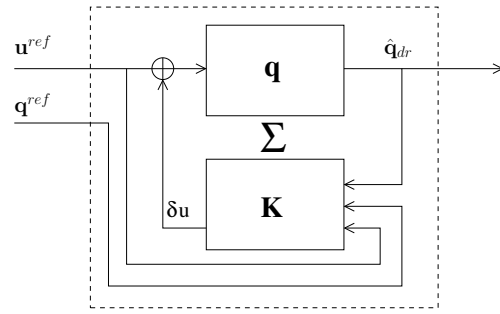


Fig. 2. The trajectory following closed-loop control system Σ

where $T_{\mathbf{q}}m$ is the tangent linear application of m at \mathbf{q} .

Thus we have the following corollary:

Property 2: $\forall t \in [0, T]$, if $\mathbf{q}(t)$ is the configuration reached by applying control \mathbf{u} on $[0, t]$ from initial state $\mathbf{q}(0)$, then applying the same inputs from initial state $m \cdot \mathbf{q}(0)$ will lead the system at time t to the configuration $m \cdot \mathbf{q}(t)$.

This property is also called group-symmetry in some references [5].

B. Closed-loop Trajectory Following

A closed-loop trajectory following process consists in computing an input to be applied to the robot, given its current configuration and a reference configuration.

1) *Closed-loop control law:* We denote by \mathbf{q}^{ref} and \mathbf{u}^{ref} respectively the reference configuration and reference input along an admissible reference trajectory.

Let us denote by Σ the closed-loop system built upon (5) and displayed on Figure 2. The evolution of the robot is represented by block \mathbf{q} and follows Equation (5) up to an input noise. The state of the system is the unknown configuration of the robot. The output $\hat{\mathbf{q}}_{dr}$ is a dead-reckoning measure of the state that follow Equation (5) up to a measure noise. In wheeled mobile robots, this measure is usually given by an odometry sensor possibly used with an Inertial Navigation System.

The control law denoted by K in the diagram computes the correction $\delta \mathbf{u}$ to be applied to the reference input \mathbf{u} . It takes as input a measure of the system configuration, a reference configuration and a reference input:

$$\delta \mathbf{u} = \mathbf{K}(\hat{\mathbf{q}}_{dr}, \mathbf{q}^{ref}, \mathbf{u}^{ref})$$

We do not give any detail about the implementation of the stabilizer K . A huge amount of bibliography exists in this domain, see [6] for an overview.

2) *Stability:* Several stability definitions have been proposed in control theory for trajectory following. In the forthcoming development, we will only assume the following property for multi-body wheeled mobile robots.

Property 3: There exists a positive real number D^{fol} such that if $(\mathbf{q}^{ref}, \mathbf{u}^{ref})$ is an admissible trajectory over $[0, T]$ fed forward to the closed-loop control system Σ , and if

$$\hat{\mathbf{q}}_{dr}(0) = \mathbf{q}^{ref}(0)$$

then for any $t \in [0, T]$,

$$d_C(\hat{\mathbf{q}}_{dr}(t), \mathbf{q}^{ref}(t)) < D^{fol} \quad (6)$$

Let us notice that with this stability property, the closed-loop control system keeps the estimation of the robot configuration $\hat{\mathbf{q}}_{dr}$ close to the reference configuration \mathbf{q}^{ref} . However, as the error $d_C(\hat{\mathbf{q}}_{dr}, \mathbf{q})$ is usually increasing with time, it is important to detect obstacles during trajectory following to avoid collisions. This problem is addressed in the following section.

III. COLLISION DETECTION, LOCALIZATION AND TRAJECTORY FOLLOWING

In order to avoid collisions, the robot must detect possible collisions of the planned trajectory with perceived obstacles, during the execution of the trajectory. Then it can use an obstacle avoidance strategy, but for the purpose of our study we focus on the obstacle detection task.

A. Obstacle

Let $\mathbf{q} = (\mathbf{x}, \mathbf{q}_{int})$ and $\hat{\mathbf{q}}_{dr} = (\hat{\mathbf{x}}_{dr}, \mathbf{q}_{int})$ be respectively the real (and unknown) and estimated robot configuration when an obstacle O is detected by on-board sensors. We denote by \mathcal{O} the unknown subset of the workspace occupied by obstacle O . Usually, only a part $\mathcal{O}^{visible}$ of the obstacle is visible. Up to sensing noise, the obstacle is seen in the robot frame as $\mathbf{x}^{-1}(\mathcal{O}^{visible})$. We denote by $\hat{\mathcal{O}}_{loc}^2$ the union of the visible part of the obstacle grown by the sensor maximal error D^{sensor} and of the space hidden by this visible part, that is we assume:

$$\mathcal{O} \subset \mathbf{x}(\hat{\mathcal{O}}_{loc}) \quad (7)$$

The subset $\hat{\mathcal{O}} = \hat{\mathbf{x}}_{dr}(\hat{\mathcal{O}}_{loc})$ is thus the estimated position in the global frame of the obstacle grown by D^{sensor} and of the corresponding hidden zone.

The desired clearance to obstacles is denoted $D^{clear} \in \mathbb{R}$. Thus a configuration \mathbf{q}^{ref} on the reference trajectory is considered as collision-free with obstacle O iff:

$$d_O(\mathbf{q}^{ref}, \hat{\mathcal{O}}) > D^{clear} \quad (8)$$

B. Collision detection task

The software architecture of the concurrent execution of the trajectory following **tFol** and collision checking **tCol** tasks is illustrated on Figure 3.

The collision detection process **tCol** is a periodic task that checks the trajectory for collision from the current abscissa s of the robot and over an interval of length Δs^{check} and updates an abscissa s^{stop} beyond which the trajectory following task is not allowed to go.

The trajectory following task **tFol** is periodic with the same frequency as this of the closed-loop control task. It manages the abscissa s along the trajectory, increasing the pseudo-velocity \dot{s} when possible and decreasing it when it has to stop at a given abscissa s^{stop} . The issue of computing a deceleration along the planned trajectory that respects the kinematic constraints of the system is out of the scope of this paper and we refer the reader to [7].

²The subscript *loc* means that the measure is done in the robot reference frame.

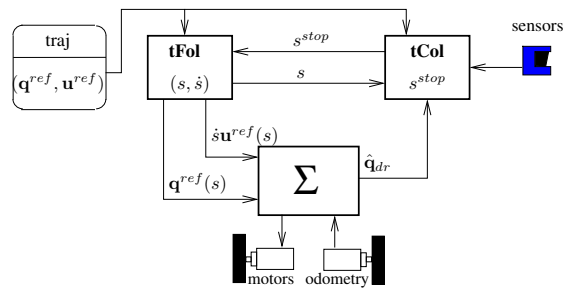


Fig. 3. Software architecture of concurrent trajectory following and collision detection tasks. **tFol** is a high frequency task, running at the same frequency as the mobile robot control module Σ . **tCol** is a lower frequency and lower priority task.

C. Condition for collision-free motion execution

Besides the necessity to check any part of the trajectory before following it, another condition is necessary for the motion execution process to be safe. This condition relates the measure error and the dead-reckoning error to the clearance distance.

1) *Dead-reckoning error*: Let $\hat{\mathbf{q}}_{dr}(s)$ and $\mathbf{q}(s)$ respectively denote the dead-reckoning estimation and the real (unknown) configuration at abscissa s along a planned trajectory $(\mathbf{q}^{ref}, \mathbf{u}^{ref})$. We define the dead-reckoning error between abscissas s_1 and s_2 as:

$$d_{SE(2)}(\hat{\mathbf{x}}_{dr}^{-1}(s_1) \cdot \hat{\mathbf{x}}_{dr}(s_2), \mathbf{x}(s_1)^{-1} \cdot \mathbf{x}(s_2)) \quad (9)$$

Property 4 (Collision-free motion): This property, the formal statement of which together with a proof will appear in forthcoming publication, states that if the admissible reference trajectory is tested as collision free with $D^{dr} + D^{fol} \leq D^{clear}$, then the robot is never in collision while following it, that is $\forall s \in [0, S], d_O(\mathbf{q}(s), \mathcal{O}) > 0$.

Therefore, Property 4 gives a formal framework to the trivial statement that a robot can follow without any collision a reference trajectory using only dead-reckoning localization. Collisions are avoided by growing the obstacles by the sum of the trajectory following error and the dead-reckoning error over intervals of collision detection.

IV. LANDMARK-BASED LOCALIZATION AND TRAJECTORY FOLLOWING

In the previous section, we have studied the problem of trajectory following and collision detection with dead-reckoning localization. For long-range motions in cluttered environments, we need to first plan an admissible collision-free trajectory using a map of the environment. In order to reach the goal configuration with bounded error, we need also to implement a landmark-based global localization method.

A. Landmark-based localization

Landmark-based localization estimates the position of the root body of the robot in the plane at a frequency usually far smaller than the trajectory following task frequency. Let $(l_n)_{n \in \{1, \dots, n^{loc}\}}$ be the sequence of abscissas at which landmark-based localization happens. For each of these abscissas, global localization is performed by fusing the

landmark-based and dead-reckoning localization with the popular Extended Kalman Filter for instance. The result of this fusion is denoted by $\hat{\mathbf{x}}^n(l_n)$. The rigid body transformation that moves $\hat{\mathbf{x}}_{dr}(l_n)$ to $\hat{\mathbf{x}}^n(l_n)$ is stored into:

$$m_n = \hat{\mathbf{x}}^n(l_n) \cdot \hat{\mathbf{x}}_{dr}^{-1}(l_n) \in SE(2) \quad (10)$$

Between two landmark-based localizations, the pose estimate of the robot is obtained by applying the latest m_n to the dead-reckoning localization: $\forall s \in [l_n, l_{n+1}]$.

$$\hat{\mathbf{x}}(s) = \hat{\mathbf{x}}^n(s) = m_n \cdot \hat{\mathbf{x}}_{dr}(s)w$$

B. Closed-loop Trajectory Following

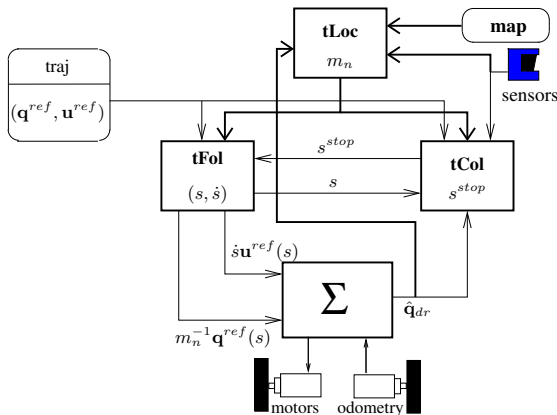


Fig. 4. Trajectory following control architecture with landmark-based localization. Task **tLoc** performs landmark-based localization and updates m_n after reading $\hat{\mathbf{x}}_{dr}(l_n)$. **tCol** uses m_n and $\hat{\mathbf{x}}_{dr}(s)$ to express in the global frame obstacles sensed in the robot frame. **tFol** feeds Σ forward with $m_n^{-1} \cdot \mathbf{q}^{ref}$.

To perform trajectory following based on global localization, it seems natural to replace the dead-reckoning estimate $\hat{\mathbf{q}}_{dr}(s)$ by the global localization estimate $\hat{\mathbf{q}}(s)$ as input to controller **K** in Figure 2. Unfortunately, for obvious security reasons the motion control module Σ is a black-box that is not allowed to rely on external data such as global localization. Using Property 1 we instead replace \mathbf{q}^{ref} by $m_n^{-1} \mathbf{q}^{ref}$ as input to Σ . We thus denote in this section

$$\mathbf{q}_{dr}^{ref}(s) \triangleq m_n^{-1} \mathbf{q}^{ref}(s) \quad (11)$$

The new control architecture is displayed in Figure 4. As just explained, in this new architecture, **tFol** computes $\mathbf{q}_{dr}^{ref}(s)$ before feeding it forward to Σ . Landmark-based localization is performed by a new task **tLoc**.

1) *Discontinuities in position estimation:* Introducing landmark-based localization induces discontinuities in the robot pose estimate. These discontinuities come from measure imprecisions and from errors in the landmarks map. In our control architecture, this discontinuity is changed into a discontinuity in the reference trajectory sent to the robot Σ (Equation 11). As a consequence, the trajectory fed-forward to the system is not admissible and the prerequisite of Property 4 is not satisfied anymore. Then the trajectory following process is not safe anymore as illustrated by the

example displayed in Figure 5, where a robot towing a trailer collide an obstacle after a localization discontinuity³. The reason of this collision is that the trajectory really executed by the robot has not been checked before being executed. It must be noticed that this figure is not a drawing, but it is the result of a realistic simulation.

For small size robots with simple kinematics, this problem can be overcome by increasing the size of the robot by an upper-bound of the localization errors. For multi-body wheeled mobile robots required to maneuver close to obstacles, increasing the clearance distance may forbid to perform high precision motions like docking to a platform or park along a sidewalk. The next section proposes a method to perform high precision motions with discontinuous localization. This method allows the distance between the reference trajectory and the obstacles to be smaller than the global localization imprecision.

V. COLLISION-FREE TRAJECTORY FOLLOWING FOR MULTI-BODY WHEELED MOBILE ROBOTS

In this section, we propose a trajectory following architecture robust to the perturbations of landmark-based localization. As noticed in the last section, localization discontinuities induce a catching up with the reference trajectory. And as shown on Figure 5, if this catching up is performed immediately, the robot follows a trajectory which has not been checked, and it can lead to collisions.

The principle of the method we propose is to simulate the catching up, and to perform it later on the trajectory, so that the trajectory followed by the robot is always checked beforehand.

A. Simulation module for catching up with the trajectory

Let $\bar{\Sigma}$ denote the simulation module of System Σ . $\bar{\Sigma}$ takes as input a reference trajectory $(\mathbf{q}^{ref}, \mathbf{u}^{ref})$ but the state and the output are the same and are denoted by $\bar{\mathbf{q}}$. The evolution of $\bar{\mathbf{q}}$ is computed from the model (5) and from the equation of the controller **K**. Moreover $\bar{\Sigma}$ outputs the reference input corrected by controller **K** and denoted as $\bar{\mathbf{u}}$. Figure 6 represents the simulation module $\bar{\Sigma}$. This simulation module satisfies the following property:

Property 5: $\forall \epsilon > 0, \forall \mu > 0, \exists T > 0$, such that

$$\begin{aligned} & \text{if } d_c(\bar{\mathbf{q}}(0), \mathbf{q}^{ref}(0)) < \mu^4 \\ & \text{then } \forall s \geq T d_c(\bar{\mathbf{q}}(s), \mathbf{q}^{ref}(s)) < \epsilon \end{aligned}$$

and the output trajectory $(\bar{\mathbf{q}}, \bar{\mathbf{u}})$ is admissible.

This property is used to compute the catching up with the reference trajectory when a localization discontinuity occurs: starting from an initial state $\bar{\mathbf{q}}(0)$ which is the new localization, system $\bar{\Sigma}$ outputs the catching up trajectory from $\bar{\mathbf{q}}(0)$ to the reference trajectory \mathbf{q}^{ref} . \bar{S} denotes the abscissa at which the catching up ends, that is $d_c(\bar{\mathbf{q}}(\bar{S}), \mathbf{q}^{ref}(\bar{S})) < \epsilon$.

B. Control architecture

In the control architecture relative to this section (see Figure 7), the task **tCol** manages as before the collision

³Voluntarily excessive for explanation purposes

⁴Obviously, parameter μ is related to the convergence interval of the stabilizer and the maximal amplitude of localization discontinuities.

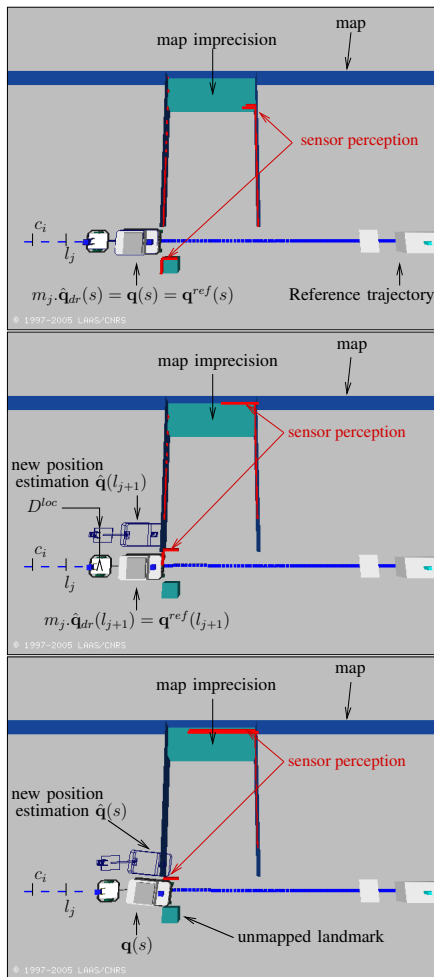


Fig. 5. Collision after a discontinuity of the localization, due to a map imprecision (an unmapped green rectangle added in the corridor). c_i is the abscissa at which the collision detection task started and l_j is the abscissa of the last landmark-based localization. The reference trajectory is checked as collision-free. On the first picture the robot is perfectly localized and it follows exactly its reference trajectory. Then on the second picture, at abscissa l_{j+1} , the landmark-based localization matches sensor perception with the map (in blue), which produces a discontinuity of D^{loc} in the localization (wire model). On the third picture the trajectory, which is now in collision. But the collision detection task has no time to detect this collision, and even if it could detect it instantaneously it would not have time to stop. To catch up with the reference trajectory, the closed-loop control tends to stabilize $\hat{\mathbf{q}}$ around \mathbf{q}^{ref} and the robot ($\mathbf{q}(s)$) turns right and hits the obstacle.

detection, but it also manages the effects of the localization discontinuities, as explained now. A rigid body transformation m is added in the definition of a trajectory, such that $\forall s \in [0, S]$:

$$\mathbf{q}_{dr}^{ref}(s) = m(s)^{-1} \mathbf{q}^{ref}(s)$$

\mathbf{q}_{dr}^{ref} being the trajectory fed forward to system Σ . At the beginning of the motion, m is uniformly initialized with the constant rigid-body transformation m_0 (Equation (10)) corresponding to the rigid-body motion that moves the dead-reckoning estimate to the global localization estimate at the beginning of the motion. We will see in the construction

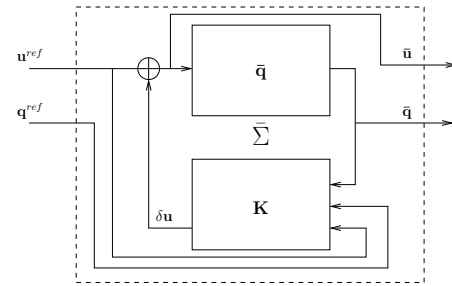


Fig. 6. Simulation module $\tilde{\Sigma}$ of System Σ . The input is a reference trajectory ($\mathbf{q}^{ref}, \mathbf{u}^{ref}$). The output is an admissible trajectory ($\tilde{\mathbf{q}}^{ref}, \tilde{\mathbf{u}}^{ref}$).

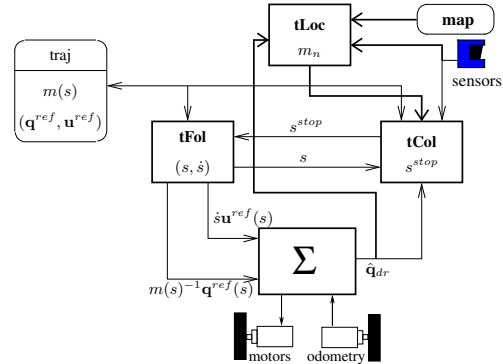


Fig. 7. Control architecture for collision-free trajectory following for multi-body wheeled mobile robots

of m that this mapping will be piecewise constant.

Algorithm 1 illustrated in Figure 8 describes the task **tCol**. The key-point of this algorithm is to express the reference trajectory in the so-called dead-reckoning frame using the relevant inverse rigid body transform $m(s)^{-1}$. We can easily check that the trajectory $m^{-1} \cdot \mathbf{q}^{ref}$ fed forward to system Σ is admissible, since it is continuous at s^{stop} and \tilde{S} after a catching up. Then the conditions of Property 4 are fulfilled, and we can guarantee the absence of collision with clearance distance $D^{fol} + D^{dr}$, D^{dr} being the dead-reckoning error between two obstacle perceptions.

Notice that in practice a catching up is performed only when the distance between the estimated robot position and the associated reference configuration is greater than a given threshold. Below this threshold, the closed-loop control can stabilize the system without collision.

C. Experimental results

Robot Hilare2 (a differential driven robot towing a trailer) was previously endowed with the control architecture displayed on Figure 4, and it implemented an obstacle avoidance strategy based on nonholonomic path deformation [8]. When navigating on long-range motion in cluttered environments using landmark-based localization⁵, we could observe residual collisions in narrow passages where the field of view of the sensor changes suddenly. We identified the discontinuities

⁵localization is performed by matching segments perceived by a laser range sensor with a map of segments.

Algorithm 1 tColInput: s , traj , \hat{O}_{loc} , m_n , \hat{x}_{dr} Output: traj , s^{stop} Initialization: $\forall s \in [0, S]$, $m(s) \leftarrow m_0$,**loop**

Collision Checking $\forall s^{check} \in [s, s + \Delta s^{check}]$
 $s^{stop} \leftarrow \text{CHECK} (m_n m(s^{check})^{-1} \cdot \mathbf{q}^{ref}(s^{check}),$
 $m_n \hat{x}_{dr}(\hat{O}_{loc}))$

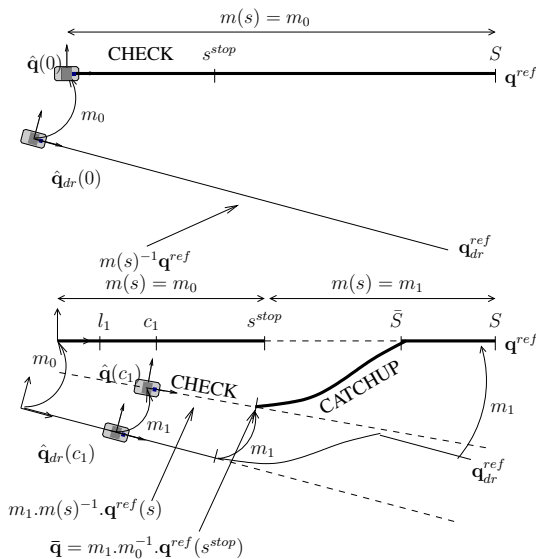
Trajectory Catching Up $\bar{\mathbf{q}}_0 \leftarrow m_n \cdot m^{-1}(s^{stop}) \cdot \mathbf{q}^{ref}(s^{stop})$ $(\bar{\mathbf{q}}, \bar{\mathbf{u}}, \bar{S}) \leftarrow \text{CATCHUP}$ trajectory from $\bar{\mathbf{q}}_0$ **if** $\bar{S} \leq S$ **then****Update of Trajectory** $\forall s \in [s^{stop}, \bar{S}]$, $(\mathbf{q}^{ref}(s), \mathbf{u}^{ref}(s)) \leftarrow (\bar{\mathbf{q}}(s), \bar{\mathbf{u}}(s))$ $\forall s \in [s^{stop}, S]$, $m(s) \leftarrow m_n$ **end if****end loop**

Fig. 8. **tCol** computations for collision-free trajectory following. The bold line is the reference trajectory \mathbf{q}^{ref} , and the line with standard width is the trajectory effectively fed forward to system Σ , that is $\mathbf{q}_{dr}^{ref}(s)$. On the picture at the top, the robot is at abscissa 0 and task **tCol** checks trajectory until abscissa s^{stop} . On the picture at the bottom, robot is at abscissa $c_1 > l_1$ and the landmark-based localization produces a discontinuity: $m_0 \neq m_1$. **tCol** checks the trajectory really executed by the robot (see Algorithm 1) and computes a trajectory from $\bar{\mathbf{q}}$ to catch up with the reference trajectory.

of the localization due to map imprecision and unmodeled objects in these locations as the reasons of the collisions.

Then we have implemented Algorithm 1, and for the first time and repeatedly the robot was able to navigate without collision on long-range trajectory, as displayed on Figure 9. The trajectory is executed at one stroke and is more than 50 meters long. The desired clearance is 5 centimeters, which has to be compared to the 80 cm width and more than 200cm length (with the trailer) of the robot.



Fig. 9. On the left, the trace of collision free long-range navigation with robot Hilare2 towing a trailer. Red ellipses display locations where localization discontinuity may occur, because the visible landmarks change suddenly. Configuration $\mathbf{q}_{picture}$ is visible on the right picture.

VI. CONCLUSION

In this paper we addressed an original issue: the effect of landmark-based localization discontinuities on trajectory following and obstacle avoidance for mobile robots. We have formalized the collision free motion execution when the localization process provides a continuous position estimation, such as dead-reckoning localization. With landmark-based localization however, the estimation of the robot position can be discontinuous. This perturbation does not only need to be smoothed before being integrated in the closed-loop control, but it also can lead to collisions. Therefore we have proposed an algorithm together with a navigation architecture that postpones the effects of these perturbations, and allows the robot to perform safe navigation. This approach has been successfully tested on a multi-body mobile robot on long-range navigation.

REFERENCES

- [1] W. Burgard, A. Cremers, D. Fox, D. Hhnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot," in *Proc. of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.
- [2] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.
- [3] R. Siegwart and all, "Robox at expo.02: A largescale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42, no. 3–4, pp. 203–222, 2003.
- [4] G. Kim, W. Chung, S. Han, K. Kim, M. Kim, and R. Shinn, "The autonomous tour-guide robot jenny," in *International Conference on Intelligent Robots and Systems*. Sendai, Japan: IEEE/RSJ, Sept. 2004.
- [5] P. Cheng, E. Frazzoli, and S. LaValle, "Exploiting group symmetries to improve precision in kinodynamic and nonholonomic planning," in *International Conference on Intelligent Robots and Systems*. Las Vegas, NE: IEEE/RSJ, Oct. 2003, pp. 631–636.
- [6] A. D. Luca, G. Oriolo, and C. Samson, *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences. NY: Springer, 1998, ch. Feedback Control of a Nonholonomic Car-like Robot, pp. 171–253.
- [7] D. Bonnafous and F. Lamiroux, "Sensor based trajectory following for nonholonomic systems in highly cluttered environment," in *International Conference on Intelligent Robots and Systems*. Las Vegas, NE: IEEE/RSJ, Oct. 2003, pp. 892–897.
- [8] F. Lamiroux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 967–977, Dec 2004.