Waypoints guidance of differential-drive mobile robots with kinematic and precision constraints

Patrice Boucher

Robotica / FirstView Article / July 2014, pp 1 - 24
DOI: 10.1017/S0263574714001921, Published online: 22 July 2014

Link to this article: http://journals.cambridge.org/abstract_S0263574714001921

How to cite this article:

Request Permissions : Click here
Waypoints guidance of differential-drive mobile robots with kinematic and precision constraints
Patrice Boucher*

Perception and Robotics Laboratory, Faculty of Electrical Engineering, Polytechnique Montreal, 2500, Chemin de Polytechnique, Canada

(Accepted June 19, 2014)

SUMMARY
This paper proposes a new kinematic controller for the waypoints guidance of robotic mobile platforms. A notable feature of the controller is its ability to process the raw sequence of waypoints to produce smooth reference velocities from control laws that are derived by taking into account a driving profile including the velocity limits, the acceleration limits, the motion modes through each waypoint (forward or backward) and the precision constraints that are required to ensure accurate waypoints traversal. A mathematical analysis demonstrates the convergence of the movements through the waypoints sequence. In addition, we present a simple way to adapt the driving profile in order that the platform reaches the last waypoint at a prescribed time. A feed-forward unit is finally described, that compensates for delays and first-order poles in the velocity response of the platform. Various simulations and experiments on real robotic platforms demonstrate the behavior and the effectiveness of the solution.

KEYWORDS: Waypoints guidance; path-following; kinematic constraints; smooth navigation; temporal planning; mobile robots.

List of Symbols

Space parameters
$w_i$: waypoint $i$.
$\{x, y\}$: 2D coordinates of $w_i$.
$\theta$: platform’s orientation.
$\chi$: platform’s configuration.
$r_i$: radius of the area around $w_i$.
$\gamma_i$: accuracy related to the waypoint $i$.
$d$: distance between the platform and the current target.
$\phi$: misalignment of the platform’s orientation relative to the target.
$s_{dec}$: threshold from which the deceleration start.

Motion parameters
$m_i$: Longitudinal motion direction (backward or forward) through $w_i$.
$v$: Longitudinal velocity.
$\omega$: Rotational velocity.
$\alpha$: Longitudinal acceleration.
$\alpha_r$: Rotational acceleration.
$\beta$: Decrease coefficient of the longitudinal velocity according to the orientation error magnitude.
$\zeta$: Decrease exponent of the longitudinal velocity.
$\{c_{left}, c_{right}\}$: Commands through the left motor and the right motor.
$\lambda$: Pole in the velocity response
$\lambda_o$: Longitudinal.
$\lambda_r$: Rotational.
$F$: Forward.
$B$: Backward.

* Corresponding author. E-mail: patrice.boucher@polymtl.ca
1. Introduction

Point-to-point navigation, parking, narrow passage traversal as well as human following\(^1\) are common tasks of a robotic mobile platform requiring that it moves in the environment. Each of these problems can be represented by a set of waypoints that the platform must successively reach in order to accomplish the requested task.

The first issue to be addressed in waypoints guidance is to ensure that the platform moves through each waypoint unimpeded by any perturbations arising from its dynamic properties, errors in its localization or skidding of the platform on the floor. This is accomplished by the establishment of appropriate control laws.

A second issue, particularly important in practical applications, is to minimize execution time while providing a well-suited navigation which fulfills a set of constraints of behavioral interest, such as velocities and acceleration limits. Basically, these constraints avoid actuator overload and jerky movements. In addition, they are chosen to generate intuitive, smooth, fast and comfortable movements, coined graceful movements by.\(^2,3\) Graceful movements are relevant in transport applications for safeguarding furniture as well as for providing for the comfort of passengers.

In previous works, waypoints guidance is typically provided by (1) defining a smooth motion path that links the waypoints together and by (2) controlling stable movements of the platform through this reference path in spite of errors in control. Each aspect has been extensively investigated, respectively, through path generators\(^4-6\) and path followers.\(^2,7-10\) Different issues related to this approach of waypoints guidance include the need: to update the reference path according to motion errors,\(^11\) to deal with singularities in control\(^9\) as well as to generate graceful movements.\(^2,3,10\) The complexity of these issues has encouraged the development of many solutions, each having their own strengths and weaknesses. A common weakness is the difficulty to obtain a predictable navigation behavior because of the sensitivity it exhibits towards the configuration of abstract control parameters. Another is that smoothness of navigation is strictly related to that of the reference path, which is problematic in obstructed areas in which the only feasible path is often sharp (e.g. at the corner of a hallway) and even jerky (e.g. through a narrow door frame). Our approach has the advantage of providing for smooth navigation even through sharp, jerky effective paths.

Some authors propose kinematic control laws that directly drive the platform’s movements through the waypoints. This alternative approach has proven to be challenging because of the nonlinear and constrained-driven nature of the problem that cannot be solved, for example, with standard linear controllers.

This paper contributes by proposing the first kinematic controller that ensures platform convergence through an arbitrary raw sequence of waypoints (without the need for pre-planning) while inherently accounting for velocity and acceleration limits. Moreover, it is the first proposed solution that allows to manage transitions of backward versus forward motion modes and to adjust the parameter of the control law according to the required accuracy of waypoint traversal. A driving profile is defined as the set of parameters that affect the navigation behavior, including velocity and acceleration limits, the precision of waypoints traversal, the motion modes and two coefficients of smoothness.

Our approach, first presented in ref. [12], was initially motivated for a smart wheelchair\(^1\) that carries out automatic tasks in a restrained, obstructed environment. This paper introduces several enhancements to,\(^12\) namely:

1. A thorough analysis of the mathematical basis of the command laws.
2. An analytic demonstration of convergence with well-defined criteria (only a numerical demonstration was presented in ref. [12]).
3. An approach for adapting the driving profile to a prescribed duration of the waypoints-following.
Waypoints guidance of differential-drive mobile robots

Table I. Parameters of kinematic controllers.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Raw sequence of waypoints</th>
<th>Inputs</th>
<th>Constraints</th>
<th>Setup</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Soft motion path</td>
<td>Soft motion trajectory</td>
<td>Longitudinal velocity</td>
<td>Surrounding environment</td>
</tr>
<tr>
<td>13</td>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>(v)</td>
<td>pr</td>
<td>pr</td>
</tr>
<tr>
<td>2</td>
<td>v</td>
<td></td>
<td></td>
<td>pr</td>
<td>pr</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>pr</td>
<td>pr</td>
</tr>
<tr>
<td>18</td>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>v</td>
<td></td>
<td></td>
<td>pr</td>
<td>pr</td>
</tr>
</tbody>
</table>

Proposed | v                        |                   |              | pr    | pr      | 1     | 2     |       |                  |                       |                   |                   |                       |                     |                   |

4. An approach that compensates for delays and first-order poles in the velocity response of the platform.

5. Extended simulations and experiments with different mobile platforms that demonstrate the motion behavior, the stability and convergence properties of our solution.

The rest of the paper is organized as follows. In Section 2 previous works are compared together and with our solution. The problem statement is presented in Section 3. The kinematic laws are detailed in Section 4, followed by a description of their use in the waypoints-following control in Section 5. The convergence analysis is demonstrated in Section 6. Section 7 explains how to achieve temporal constraints and Section 8 proposes a feed-forward compensator for delays and poles. Experimental results are finally presented in Section 9.

2. Comparison with Previous Works

Many approaches have been proposed for controlling the movements of a mobile platform. Table I presents a sample of relevant works aimed at this general objective. It allows to compare previous kinematic controllers according to their inputs, constraints, number of setup parameters and outputs.

Smoothness of the movements is often inferred via smoothness of the motion path through relevant path-planning solutions4–6 that typically incorporate curvature constraints on the path. However, smooth path traversal does not implicitly bring about smooth movements in practice, as a kinematic control can output jerk velocity transitions in response to configuration errors of the platform. In order to further improve smoothness of the movements, bounds can be imposed on the magnitudes of the velocity and acceleration. These constraints can be either incorporated a priori in the derivation of the control laws (cf. "pr" in Table I), or imposed a posteriori (cf. "po" in Table I) to the kinematic controller outputs. The first solution has the advantage that the controller outputs implicitly satisfy the kinematic constraints. In the latter solution, the constraints act as disturbances that can impact the stability, efficiency and convergence of the movements through the target.
Different control strategies allow the platform to follow either (I) a raw sequence of waypoints, (II) a soft motion path or (III) a soft motion trajectory (motion path with time constraints). Each of these previous objectives involves different definitions of the configuration error of the platform.

In the first objective (I), the configuration error is obtained by the distance and the heading misalignment of the platform with the current target. In regard to the problem of waypoints following, the first objective is the simplest in that it avoids the complexity of path-planning and trajectory planning. This complexity is related to the connection between the problem of path-planning and the problem of movement generation. Effective path-planners typically consider which movement can be executed by the platform according to its dynamic limitations and kinematic constraints, which are usually already taken into account at bottom levels in the dynamic controller and kinematic controller, respectively. Kinematic controllers that directly input the raw sequence of waypoints provide a unified solution to the problem of path generation and movement generation.

In the previous category, the kinematic control of considers a limit on the longitudinal velocity that is fixed from a fuzzy logic set of rules according to the path curvature and the distance of the target. Each rule corresponds to a state in which the curve is either smooth, appropriate or narrow and the target close, medium or far. The longitudinal velocity, as well as the angular velocity, are first proportional to the configuration error of the platform before being limited towards the kinematic constraints. With this approach, the behavior of movements is sensitive to the value of the proportional gains and to the values defining the qualitative state of the curve and the qualitative proximity of the target. In particular, there is no guarantee that the global movement is efficient (i.e. void of useless oscillations and loss of time). The approach computes a motion trajectory between two waypoints with boundary conditions from a variational optimization that minimizes a cost function which incorporates costs of discomfort associated with velocity, acceleration and travelling time. The optimal trajectory converges with the iterative procedure of Newton trust-region. Finally, the work reported in ref. [2] proposes kinematic control laws in which the angular velocity is set, taking into account an input longitudinal velocity in such a way that the robot follows a manifold through the target. This manifold is obtained from a convergence analysis based on Lyapunov theory.

In the second objective (II), the configuration error defined by is related to the misalignment between the platform orientation and a heading orientation computed in such a way that the platform motion converges to a reference motion path. In ref. [17], the longitudinal velocity is assumed constant, while an algorithm computes a steering angle and a consequent rotational velocity command aimed at allowing the car to follow the road. The work reported in ref. [2] computes the longitudinal velocity by solving an optimization problem that incorporates kinematic constraints, and then uses it as an input to the kinematic controller of.

In the third objective (III), the configuration error defined by is stated as the difference between the configuration of the real platform with the configuration of a virtual platform that follows a trajectory at prescribed times with respect to a velocity profile (cf. for an example of velocity planning according to kinematic constraints). proves the convergence by the Lyapunov theory. computes a path manifold according to the curvature constraints related to the mechanical limitations of the platform. This path manifold is then employed in a sliding mode kinematic controller that computes a reference angular velocity.

The work reported in ref. [22] regulates the heading of the mobile robot according to the surrounding obstacles in regard to the trajectory constraints imposed by the dynamic limitation of the platform.

Many kinematic controllers incorporate the rotational velocity in a closed-loop control that takes into account the dynamic properties of the mobile robot without consideration of kinematic constraints. While this approach can ensure a fast convergence of the platform with a fine-tuning of the control parameters, it may be incompatible with the objective of ensuring smooth movements. Indeed, this objective is heavily related to the presence of kinematic constraints, that act as disturbances when they are applied a posteriori to the control outputs.

The number of parameters in the control laws is an indicator of the complexity of their setup and to the spread of behaviors they can provide. In Table I, the term control parameters refers to parameters that mainly affect the efficiency and the stability of the movements, while the term softness parameters refers to those that mainly impact the shape of the actual motion path. Softness parameters are relevant for the user’s ability to modify the navigation according to subjective preferences.

An experimental comparison of the approaches of Table I is a delicate task. First, different objectives involve different performance indexes. Second, a constraint is connected to a rule that
restricts the navigation possibilities of the mobile platform. Different navigation rules involve different waypoints-following properties regardless of the kinematic control approach that is employed. One that wants to compare controllers derived from different sets of rules may be tempted to define a set of shared rules and to impose a posteriori to each controller the rules that are not yet included in its set. However, this approach may prove of little interest since a controller derived in relation to specific rules is obviously subject to perform better with them than another derived from different rules. Third, the sensitiveness of many approaches to their setup may lead to very specific results. Accordingly, relevant comparisons may call for multiple tests in different behavioral setups of the approaches. For these reasons, the scope of this section is restricted to highlight fundamental properties of the solutions.

As mentioned in the previous section and shown in Table I, our kinematic controller is the first that ensures convergence of the platform movements to the raw waypoints sequence while taking into account appropriate velocity and acceleration limits to ensure smoothness of movement. It also introduces the novel approach of managing backward and forward motion modes while allowing to adjust the parameter of the control law according to the required accuracy of waypoint traversal (which is accomplished in Section 6 by adjusting the magnitude of a parameter).

3. Waypoints Guidance Controller: An Overview

Our waypoints guidance controller intends to drive a mobile platform through successive zones of N waypoints $w_i, i \in \{1, \ldots, N\}$ of coordinates $\{x_i, y_i\}$, as illustrated in Fig. 1. Each zone $i$ has a circular area with a radius $r_i$, named the admissible radius of traversal. The accuracy $\gamma_i$ of a waypoint $i$ corresponds to the inverse of the admissible radius, from which $\gamma_i = 1/r_i$.

Our kinematic controller generates, in real-time, the required velocities $\{v, \omega\}_\text{REF}$ to bring the mobile platform from its current position to the subsequent waypoints. These velocities are used as the reference of a dynamic controller as illustrated in Fig. 2.

The dynamic controller includes a feed-forward compensation unit, described in Section 8, followed by a closed-loop velocity controller that generates the motor commands $\{c_{\text{left}}, c_{\text{right}}\}$. This architecture has the advantage of decoupling the position control provided by the kinematic controller to the platform’s dynamic properties (i.e. inertia, weight, surface friction). The configuration setup of the waypoints guidance controller can therefore be employed on many platforms insofar as the kinematic parameters are in keeping with platform limitations. Another advantage is that the waypoints guidance controller can be combined with interchangeable closed-loop velocity controllers.

The proposed approach is specially investigated for a differential-drive mobile robot. This type of robot has a nonholonomic constraint along the axis of two powered wheels. It is commanded by a longitudinal velocity $v$ and by a rotational velocity $\omega$.

Let the platform configuration be denoted by $\chi_k = \{x_k, y_k, \theta_k\}$. The configuration gap with a waypoint $w_i$, as illustrated in Fig. 3, is described by the following parameters:

- $d_k$: Euclidean distance between the platform’s position $\{x_k, y_k\}$ and the next waypoint $w_i$ at a step time $k$;
- $\phi_k$: misalignment of the platform’s orientation relative to the target defined as follows:

\[
\phi_k(x_k, y_k, \theta_k, x_i, y_i, m_i) = \begin{cases} 
\text{atan2}(y_i - y_k, x_i - x_k) - \theta_k & m_i = 1 \\
\text{atan2}(y_i - y_k, x_i - x_k) - \theta_k + \pi & m_i = -1 
\end{cases}
\]

(1)

The value of $\phi_k$ is included in $[-\pi, \pi]$ and $m_i$ sets the motion mode for reaching the waypoint $w_i$. The selected motion mode is forward with a value $m_i$ of 1 and backward with a value of $-1$. The motion mode is assumed to change only at the instant when the current target switches to the next waypoint of the sequence.

Navigation parameters, stored in a driving profile, refer to the set of motion characteristics that have important impacts on security, precision, overall pace of platform displacements. For security, mechanical restrictions, energy savings as well as for specific needs related to the application, the velocities and accelerations are normally bounded. The maximum rotational velocity is $\omega_{\text{max}}$. 


Waypoints guidance of differential-drive mobile robots

Fig. 1. (Colour online) An example of an admissible effective path (solid line) through a waypoints sequence.

\[ \vmax(m_i) = \begin{cases} v_F & m_i = 1 \\ v_B & m_i = -1 \end{cases} \]

The maximum longitudinal velocity, \( v_{\text{max}} \), depends on the motion mode \( m_i \).

The longitudinal acceleration limit and the rotational acceleration limit are represented by \( a_{\text{max}} \) and \( \alpha_{\text{max}} \), while the corresponding deceleration limits are represented by \( a_{\text{min}} \) and \( \alpha_{\text{min}} \).

For a direct convergence through the waypoints, as shown in Section 6, the longitudinal velocity is inhibited with the orientation error according to two smoothness parameters \( \beta \) and \( \xi \) that are given in Table II, named respectively the decrease coefficient and the decrease exponent of the longitudinal velocity. Section 6 shall present how to determine the minimum value of \( \beta \) that guarantees convergence. These parameters shape the impact of the orientation error on speed reduction.
Table II. Parameters of a driving profile.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_F$, $v_B$</td>
<td>m/s</td>
<td>Maximum longitudinal velocity in forward/backward motion mode</td>
</tr>
<tr>
<td>$\omega_{\text{max}}$</td>
<td>rad/s</td>
<td>Maximum rotational velocity</td>
</tr>
<tr>
<td>$a_{\text{max}}, a_{\text{min}}$</td>
<td>m/s$^2$</td>
<td>Maximum longitudinal acceleration/deceleration</td>
</tr>
<tr>
<td>$\alpha_{\text{max}}, \alpha_{\text{min}}$</td>
<td>rad/s$^2$</td>
<td>Maximum rotational acceleration/deceleration</td>
</tr>
<tr>
<td>$\xi$</td>
<td>-</td>
<td>Decrease exponent of the longitudinal velocity</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1/rad</td>
<td>Decrease coefficient of the longitudinal velocity according to the orientation error magnitude</td>
</tr>
<tr>
<td>$r_i$</td>
<td>m</td>
<td>Admissible radius of each waypoint traversal $i$</td>
</tr>
<tr>
<td>$m_i$</td>
<td>-</td>
<td>Motion mode $i$ (forward or backward)</td>
</tr>
</tbody>
</table>

This idea was first inspired from human driving behaviors. As noted by\textsuperscript{3,14}, human drivers generally tend to reduce the longitudinal velocity of their cars in proportion to the road’s sharpness. Moreover, a human also reduces the longitudinal velocity of the car in anticipation of a sharp road, even if the current road is straight. Experiments of Section 9 show that consideration of target velocities in the control laws of waypoint traversal (computed in Section 5) provides for this behavior of anticipating a sharp trajectory.

Navigation parameters are defined as positive according to Table II. All of them are inherently respected by the kinematic laws presented in the next section, except $a_{\text{max}}$ and $\alpha_{\text{max}}$ that are applied a posteriori.

4. Kinematic Laws

4.1. Rotational velocity computation

The objective is to find the rotational velocity law that provides the fastest reduction of an orientation error while observing the rotational velocity limit $\omega_{\text{max}}$ and deceleration limit $\alpha_{\text{min}}$. In order to prevent an overshoot, the rotational velocity must be null when the orientation error is null. In order to provide a closed-loop position control, the velocity law must vary according to the orientation error $\phi_k$ and, therefore, be time-independent.

For minimizing the compensation time, the rotational velocity $\omega_k^*$ is $\omega_{\text{max}}$ over an orientation error of $\phi_{s\text{dec}}$ with a gradual decrease according to $\alpha_{\text{min}}$ until $\phi_k = 0$. The sign of the rotational velocity is set in such a manner as to reduce the orientation error. The temporal evolution of the rotational velocity for $|\phi_k| < \phi_{s\text{dec}}$ can be stated by the following equation in which the rotation through the target is achieved for $t < 0$ up to $t = 0$ at time the orientation error is null:

$$\omega(t) = -\alpha_{\text{min}} \cdot t \cdot \text{sign}(\phi)$$

The orientation error is obtained by time integration of (2):

$$\phi(t) = -\int \omega(t) dt = \frac{\alpha_{\text{min}} \cdot t^2}{2} \text{sign}(\phi).$$

By isolating the variable $t$ in (3) and since $t \leq 0$, we obtain:

$$t = -\sqrt{\frac{2|\phi|}{\alpha_{\text{min}}}}.$$
By inserting (4) in (2), the rotational velocity law for $|\phi_k| < \phi_{s\text{dec}}$ is then:

$$\omega_k(\phi_k) = \sqrt{2|\phi_k| \alpha_{\text{min}} \cdot \text{sign}(\phi_k)}. \quad (5)$$

This equation is bounded by the maximum rotational velocity $\omega_{\text{max}}$. In Fig. 4, each curve is a command law that sets the rotational velocity according to the orientation error. Law A is that for which the rotational velocity is always maximal, taking into account the kinematic constraints $\omega_{\text{max}}$ and $\alpha_{\text{min}}$. This law involves a fixed rotational velocity of $\omega_{\text{max}}$ for $|\phi_k| \geq \phi_{s\text{dec}}$ with a gradual decrease according to $\alpha_{\text{min}}$ with the command law found in (5) for $|\phi_k| \leq \phi_{s\text{dec}}$:

$$\omega_{\ast k}(\phi_k) = \begin{cases} \sqrt{2|\phi_k| \alpha_{\text{min}} \cdot \text{sign}(\phi_k)} & |\phi_k| < \phi_{s\text{dec}} \\ \omega_{\text{max}} \cdot \text{sign}(\phi_k) & \text{else} \end{cases}. \quad (6)$$

The angle $\phi_{s\text{dec}}$ from which the deceleration starts is obtained at the intersection point of the two parts of Law A (6), namely when $\omega_k = \omega_{\text{max}}$. By replacing $\omega_k$ with $\omega_{\text{max}}$ and $\phi_k$ with $\phi_{s\text{dec}}$ in (5), and by isolating $\phi_{s\text{dec}}$, we obtain the following expression:

$$\phi_{s\text{dec}} = \frac{\omega_{\text{max}}^2}{2\alpha_{\text{min}}}. \quad (7)$$

Since Law A maximizes the rotational velocity, it provides an optimal solution for minimizing execution time. Indeed, all other laws that satisfy the kinematic constraints must fall below Law A in Fig. 4, such as Law B and Law C that involve equal or lower velocities and are associated with larger execution times.

Law A (6) is derived from an optimal time planning. However, since it inputs an orientation error rather than a time variable, it always remains suited to the current platform orientation despite any previous perturbations (e.g. floor sliding and errors in the dynamic control). This property ensures a closed-loop position control.

4.2. Longitudinal velocity computation

The objective is to find a longitudinal velocity law that efficiently reduces the distance between the platform and the next waypoint, given the maximum longitudinal velocity $v_{\text{max}}$, the maximum deceleration $a_{\text{min}}$ and a velocity $v_i$ on the next waypoint traversal. Note that $v_i, i \in \{1, \ldots, N\}$ represents the sequence of longitudinal velocities on each waypoint for command continuity. Their computation is addressed in Section 5. In order to provide a closed-loop position control, the longitudinal velocity law must also vary according to the configuration gap and be time-independent.

If there is no orientation error ($\phi_k = 0$), the optimal longitudinal velocity is $v_{\text{max}}$ over a critical distance of $d_{\text{idec}}$ with a decrease by $a_{\text{min}}$ until $v_i$ at the waypoint $w_i$ (where $d_k = 0$).
Waypoints guidance of differential-drive mobile robots

Assuming that the longitudinal velocity $v_k$ is equal to the velocity $v_i$ at time $t = 0$ on the target waypoint $i$, the temporal evolution of the velocity can be described, for $d_k < d_{sdec}$ and $t < 0$, by:

$$v(t) = \int_0^t a_{min} \, dt = -a_{min} \cdot t + v_i. \quad (8)$$

The temporal evolution of the distance $d$ with the target $w_i$ is consequently:

$$d(t) = \int_0^t v(t) \, dt = \frac{a_{min} \cdot t^2}{2} - v_i t \quad (9)$$

By isolating the time variable $t$ in (8), we have:

$$t(v) = \frac{v_i - v}{a_{min}} \quad (10)$$

which is next inserted in (9) to provide the longitudinal velocity $v$ as a function of the distance $d'$:

$$v(d) = \sqrt{2a_{min}d + (v_i)^2}. \quad (11)$$

The threshold distance $d_{sdec}$ from which begins the deceleration is obtained with $v = v_{max}$ in (11):

$$d_{sdec}(v_i) = \frac{v_{max}^2 - (v_i)^2}{2 \cdot a_{min}}. \quad (12)$$

Equation (12) is positive since $|v_i| \leq v_{max}$. By taking into account the motion mode\(^1\), the corresponding longitudinal velocity law is:

$$v^*_k(\phi_k, d_k, v_i, m_i) = \begin{cases} v_{max} \cdot m_i & d_k \geq d_{sdec} \\ m_i \cdot \sqrt{2d_k a_{min} + (v_i)^2} & d_k < d_{sdec} \end{cases} \quad (13)$$

and has the same close-loop behavior than the rotational velocity law.

Equation (13) must be adapted according to a non-null orientation error $\phi_k$. As aforementioned in Section 3, the velocity must decrease with the magnitude of the orientation error according to parameters $\beta$ and $\zeta$ (cf. Table II). In order to achieve such a behavior, (13) is modified as follows:

$$v^*_k(\phi_k, d_k, v_i, m_i) = \begin{cases} v_{max} \cdot m_i \frac{1 + |\beta \phi_k|^\zeta}{1 + |\beta \phi_k|^\zeta} & d_k \geq d_{sdec} \\ m_i \frac{2d_k a_{max} + (v_i)^2}{1 + |\beta \phi_k|^\zeta} & d_k < d_{sdec} \end{cases} \quad (14)$$

Section 6 demonstrates that this adaptation causes convergent movements through the waypoints sequence.

5. Waypoints-Following Control

The waypoints-following procedure is represented in Algorithm 1. The waypoints following starts with an initialization phase in which the longitudinal velocity on each waypoint traversal must ensure

\(^1\) Notice that $d_k$ is defined as positive and that the sign of the velocity $v^*_k$ always corresponds to the sign of the target velocity $v_i$. Otherwise, the platform would reach a null velocity (to change the direction mode) before reaching the waypoint, implying a deadlock situation.
velocity continuity, which involves a smooth progression between the longitudinal velocity for target $i - 1$ at time $k - 1$ to the target $i$ at time $k$.

In order to achieve this continuity of the velocity, a straightforward recursive computation is executed from the last waypoint of target velocity $v_{N\text{d}}$ up to the first waypoint. This calculation provides a set of traversal velocities $\{v_i, i = 1, \ldots, N\}$. At each recursion $i$, the velocity $v_i$ is obtained using (14) with: the separation distance $D_{i+1}$ between the current waypoint $i$ and the next one $i + 1$, the approximate platform’s orientation error $\phi_i$ at the time it reaches waypoint $i$ and the motion modes $m_i$ and $m_{i+1}$. Note that $v_i$ is null if the motion mode changes, implying $m_i \times m_{i+1} = -1$. A depiction of these parameters is presented in Fig. 5.

**Algorithm 1** Waypoint-following procedure

A. Choice of the motion modes $m_i, \forall i \in \{1, \ldots, N\}$ and of the target velocity on the last waypoint $v_{N\text{d}}$.

B. Recursive computation of the target longitudinal velocities on the waypoint traversals:

\[ v_N = v_{N\text{d}} \]

for $i = N$ to 2 do

if $m_{i-1} \neq m_i$ then

$v_{i-1} = 0$

else

- Update $d_{\text{dec}}(v_i)$ with (12);
- Compute the euclidean distance $D_i$ between $i - 1$ and $i$, and the rough orientation error $\phi_i$ at the instant the platform reaches $i$ from $i - 1$.
- Estimate $v_{i-1} (\phi_i, m_i, D_i, v_i)$ with (14);

end if

end for

C. Execute the waypoint-following at each step $k$:

Set $i = 1$.

while $i \leq N$ do

- Compute the configuration error: $d_k$, and $\phi_k$ with (1).
- Compute the target rotational velocity $\omega_k^*$ with (6) and the target longitudinal velocity $v_k^*$ with (13).
- Correct the values of $\omega_k^*$ and $v_k^*$ to account for the acceleration limits $a_{\text{max}}$ and $a_{\text{max}}$.
- Send the reference velocities to the dynamic controller.

if $d_k < r_i$ then

$i = i + 1 \{i$ is reached$}$

end if

Wait $\Delta T$

$k = k + 1$.

end while

At each execution time, the reference velocities are computed from (6) and (14) according to the current estimation of the platform’s pose, and are then limited in accordance with the acceleration limits $\{a_{\text{max}}, a_{\text{max}}\}$. A change of target waypoint occurs when the current one is reached, thus implying $d_k < r_i$.

Notice that the acceleration limits $a_{\text{min}}$ and $a_{\text{min}}$ are already considered in the kinematic laws and are thus inherently respected. Since parameters $a_{\text{max}}$ and $a_{\text{max}}$ are not considered in control laws, they can however affect position control: the lower their values are, the more they could disturb the effectiveness of movements (i.e. by generating sinusoidal movements).

6. Convergence Analysis

The kinematic laws of the waypoints guidance are assumed to provide convergence insofar as they cause the platform to reach consecutively, and in the first attempt, each waypoint within the desired accuracy. In order to prove convergence, we first present the conditions that may result in non-convergence traps. A condition for convergence is then deduced.
6.1. Non-convergence conditions
We can expect that non-convergence to a waypoint as a result of \( \{v^*_k, \omega^*_k\} \) can arise only if the following two conditions are simultaneously fulfilled:

1. the velocity command increases the separation distance from the waypoint;
2. if the velocity command persists, the separation distance may potentially never be reduced;

The variation of the Euclidian distance to waypoint \( i \) may be approximated by:

\[
\Delta d_k = - |v^*_k| \cdot \cos(\phi_k)
\]

The first condition for divergence is thus:

\[
|\phi_k| \geq \pi/2 \tag{15}
\]

The second condition is only satisfied if the curvature radius \( r_k \) of the effective path is larger than the current separation distance from the waypoint:

\[
|r_k| \geq d_k \tag{16}
\]

where:

\[
r_k = v^*_k / \omega^*_k. \tag{17}
\]

This condition is shown in Fig. 6 for an orientation error of \( \pi/2 \). All commands producing a path with a curvature radius greater than the current separation distance \( d_k \) will never generate a lower separation distance from the waypoint.

6.2. Convergence area of the configuration gaps
The combination of the rotational velocity law (6) with the longitudinal velocity law (14) provides a cardioid field for the actual paths. From Eqs. (6) and (13), the longitudinal velocity \( v^*_k \) and rotational
velocity $\omega^*_k$ vary only with respect to the configuration gap represented by $\phi_k$ and $d_k$, the traversal velocity $v_t$ and the motion mode $m_t$. From (17), the instantaneous curvature radius $r_k$ can then be expressed as a function of the configuration gap.

Below, we find the ranges of values of $d_k$ and $\phi_k$ for which all non-convergence conditions are fulfilled. These ranges are evaluated for each combination of the equation parts of the rotational velocity (related to the value of $\phi_k$ relative to $\phi_{sdec}$) and the longitudinal velocity (related to the value of $d_k$ relative to $d_{sdec}$).

**A-** ($d_k > d_{sdec}$) and $(|\phi_k| < \phi_{sdec})$ In this case, non convergence can only arise for $\phi_{sdec} \geq \pi/2$. The absolute value of the curvature radius is:

$$|r_k|^A = \frac{v_{max}}{\left(1 + |\beta \phi_k|^2\right)^{\frac{1}{2}}} \frac{1}{\sqrt{2 \phi_k} \alpha_{min}}$$ (18)

Let us assume that condition of non-convergence (15) is fulfilled, which means that $|\phi_k| \in [\pi/2, \min(\pi, \phi_{sdec})]$. From expression (18), the maximum value of the non-convergence radius $|r_k|^A$, below, is obtained with the minimal value of the orientation error $\phi_k = \pi/2$:

$$\left(r_{lim}^i\right)^A = \frac{v_{max}}{\left(1 + |\beta \pi/2|^2\right)^{\frac{1}{2}}} \frac{\sqrt{\pi} \alpha_{min}}{\omega_{max}}$$ (19)

Conditions of non-convergence (15) and (16) can therefore be simultaneously fulfilled for $d_k \in [d_{sdec}, \left(r_{lim}^i\right)^A]$ and $\phi_{sdec} > \pi/2$.

**B-** ($d_k > d_{sdec}$) and $(|\phi_k| > \phi_{sdec})$ The absolute value of the curvature radius is:

$$|r_k|^B = \frac{v_{max}}{\left(1 + |\beta \phi_k|^2\right)^{\frac{1}{2}}} \omega_{max}$$ (20)

The maximum value of the radius $|r_k|^B$ that fulfills the condition of non-convergence (15) is obtained with the minimal value of the orientation error $\phi$ over $\pi/2$ that satisfies $(|\phi_k| > \phi_{sdec})$, namely $\phi = \max[\pi/2, \phi_{sdec}]$:

$$\left(r_{lim}^i\right)^B = \frac{v_{max}}{\left(1 + |\beta \pi/2|^2\right)^{\frac{1}{2}}} \frac{\sqrt{\pi} \alpha_{min}}{\omega_{max}}$$ (21)

Conditions of non-convergence (15) and (16) can therefore be simultaneously fulfilled for $d_k \in [d_{sdec}, \left(r_{lim}^i\right)^B]$.

**C-** ($d_k < d_{sdec}$) and $(|\phi_k| < \phi_{sdec})$ In this case, non-convergence is possible for $\phi_{sdec} \geq \pi/2$. The maximum value of the curvature radius is obtained with $\phi_k = \pi/2$. Its absolute value is:

$$|r_k|^C = \frac{\sqrt{2d_k} \alpha_{min} + (v_t)^2}{\left(1 + |\beta \phi_k|^2\right)^{\frac{1}{2}}} \sqrt{\pi} \alpha_{min}$$ (22)

The maximum distance $\{d_k\}^{max}$ that satisfies condition (16) corresponds to the radius $|r_k|^C$:

$$|r_k|^C = \{d_k\}^{max} = \frac{\sqrt{2d_k} \alpha_{min} + (v_t)^2}{\left(1 + |\beta \pi/2|^2\right)^{\frac{1}{2}}} \sqrt{\pi} \alpha_{min}$$ (23)

The solution of the quadratic equation is

$$\left(r_{lim}^i\right)^C = \alpha_{min} + \frac{\sqrt{a_{min}^2 + (1 + |\beta \pi/2|^2)^2 (\pi \alpha_{min})(v_t)^2}}{\left(1 + |\beta \pi/2|^2\right)^{\frac{1}{2}}} \frac{\pi \alpha_{min}}{\sqrt{\pi}}$$ (24)
Waypoints guidance of differential-drive mobile robots

Fig. 7. (Colour online) Motion fields through a waypoint with different initial orientation errors.

since \((r_i^{lim})^C \geq 0\). Conditions of non-convergence (15) and (16) can therefore be simultaneously fulfilled for \(d_k \in [0, \min\{d_{sdec}, (r_i^{lim})^C\}]\).

D- \((d_k < d_{sdec})\) and \((|\phi_k| > \phi_{sdec})\) In analogy with previous demonstrations, the curvature radius is:

\[
(r_i^{lim})^D = \min + \sqrt{\min^2 + (1 + |\beta \cdot \max\{\pi/2, \phi_{sdec}\}|)^2 \omega_{max}^2 \omega_{max}^2 (v_i)^2},
\]

(25)

Conditions of non-convergence (15) and (16) can be simultaneously fulfilled for \(d_k \in [0, \min\{d_{sdec}, (r_i^{lim})^D\}]\).

In summary, the conditions for non-convergence (15) and (16) can only be satisfied inside a circle of radius \(r_{lim} = \max\{(r_i^{lim})^A, (r_i^{lim})^B, (r_i^{lim})^C, (r_i^{lim})^D\}\) centered on the current waypoint target, referred to as the radius of divergence. The platform’s movement diverges through the circle’s circumference if the platform is inside and converges through the circle’s circumference if the platform is outside. We can therefore expect that the platform’s movements always converge through the circle’s circumference, as confirmed by the simulation of Fig. 7, in which plain lines are used when the platform increases its distance from the target and dashed lines are used when the platform decreases its distance to the target.

Note that in each situation \(\{A, B, C, D\}\) given by (19), (21), (24) and (25), an increase of \(\beta\) causes a decrease of the radius of divergence. In fact, the more \(\beta\) increases, the more the effective path toward the waypoint flattens, as illustrated in Fig. 8 for an initial orientation error of \(3\pi/4\).

Let \(r(t, t_0, r_0)\) be the time evolution flow of the distance between the platform and a waypoint for a duration \(T\). The distance at time \(t_0\) is \(r_0\). According to Lyaponov’s theory, the radius of divergence \(r_{lim}\) is an equilibrium:

1. **Lyapunov uniformly stable**, since:

\[
\left| r(t, t_0, r_0) - r_{lim} \right| < \epsilon, \quad \forall \ t \geq 0, \ t_0 \in T \ and \ \forall \ r_0
\]

2. **Uniformly attractive**, since

\[
\lim_{t \to \infty} r(t, t_0, r_0) = r_{lim}, \quad \forall \ r_0 \ and \ t_0 \ in \ T
\]

3. **Uniformly asymptotically attractive**, as a consequence of being Lyapunov uniformly stable and Uniformly attractive.
Table III. Minimum admissible value of the decrease coefficient of the longitudinal velocity.

<table>
<thead>
<tr>
<th>Case</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\frac{1}{\pi} \sqrt{\frac{\alpha_{\text{max}}}{\alpha_{\text{min}}}} - 1$</td>
</tr>
<tr>
<td>B</td>
<td>$\max{\pi/2, \phi_{\text{dec}}} \sqrt{\frac{\alpha_{\text{max}}}{\omega_{\text{max}}}} - 1$</td>
</tr>
<tr>
<td>C</td>
<td>$\frac{2}{\pi} \sqrt{\frac{2\pi \alpha_{\text{max}} + (\pi/2)^2}{(\pi/2) \alpha_{\text{max}}}} - 1$</td>
</tr>
<tr>
<td>D</td>
<td>$\max{\pi/2, \phi_{\text{dec}}} \sqrt{\frac{2\pi \alpha_{\text{max}} + (\pi/2)^2}{(\pi/2) \alpha_{\text{max}}}} - 1$</td>
</tr>
</tbody>
</table>

6.3. Condition to guarantee convergence

In order to guarantee convergence, the decrease coefficient $\beta_{\text{min}}$ of the longitudinal velocity must be set in such a way that each radius of divergence $r_i$ is inferior to the corresponding admissible radius $r_{i,\text{lim}}$:

$$r_{i,\text{lim}} < r_i, \forall i = 1, \ldots, N.$$  \hspace{1cm} (26)

Table III presents, for each case $\{A, B, C, D\}$, the minimum value of the decrease coefficient $\beta_{\text{min}}$ that causes the radius of divergence $r_{i,\text{lim}}$ to be inferior to the admissible radius $r_i$. These equations are obtained by isolating the variable $\beta$ in Eqs. (19), (21), (24) and (25). Notice that imaginary values of $\beta$ resulting from these equations mean that convergence is ensured with $\beta = 0$.

By using the maximum velocity of waypoints traversal $v_i = \max\{v_F, v_P\}$ in equations of Table III with the minimal radii relevant to the application, the maximum value of $\beta_{\text{min}}$ among those of $\{A, B, C, D\}$ is effective for all waypoints sequences. In practice, we can choose a value of $\beta$ superior to this limit for robustness against perturbations.

7. Reaching a Final Time

An on-the-fly simulation of the waypoints guidance can predict the time $T_P$ for reaching the last waypoint of a sequence with a nominal driving profile having the kinematic parameters $\{v_0^0, v_{\text{max}}, a_{\text{max}}, a_{\text{min}}, \alpha_{\text{max}}, \alpha_{\text{min}}\}$. In order to reach the last target at a prescribed $T_d$, the required
Waypoints guidance of differential-drive mobile robots

A driving profile can be obtained as follows:

\[ v_F = v_0^p \cdot \eta \]  \hspace{1cm} (27)
\[ v_B = v_0^b \cdot \eta \]  \hspace{1cm} (28)
\[ \omega_{\text{max}} = \omega_0^{\text{max}} \cdot \eta \]  \hspace{1cm} (29)
\[ a_{\text{max}} = a_0^{\text{max}} \cdot \eta^2 \]  \hspace{1cm} (30)
\[ a_{\text{min}} = a_0^{\text{min}} \cdot \eta^2 \]  \hspace{1cm} (31)
\[ \alpha_{\text{max}} = \alpha_0^{\text{max}} \cdot \eta^2 \]  \hspace{1cm} (32)
\[ \alpha_{\text{min}} = \alpha_0^{\text{min}} \cdot \eta^2 \]  \hspace{1cm} (33)

with \( \eta = T_p / T_d \) a time-scale factor. The final time is achievable insofar as the second driving profile respects the system limitations.

It should be pointed out that since this strategy only affects the driving profile, it does not reduce the capability of the waypoints guidance controller to compensate (in terms of position) for perturbations in control (i.e. by generating reference velocities according to the current context). Since the driving profile does not change during control, any perturbation will however affect the effective realization time.

We will show in the experiments that an accurate kinematic control can provide a reasonable accuracy.

8. Compensation for Delays and Poles in the Dynamic Response

This section presents a feed-forward unit that compensates for a first-order pole and a delay in the velocity response of the platform.

The poles in the longitudinal and rotational velocity responses are denoted by \( \lambda^{lo} \) and \( \lambda^{ro} \), respectively. The values of the poles are understood to be strictly negative. We assume that both responses have an identical actuation delay \( \tau \). Since the following derivation is applicable to both velocity responses, we present that relating to longitudinal velocity and omit the distinction \( \text{lo} \) and \( \text{ro} \) for simplicity.

First, let the velocity response \( v_k \) to a velocity command \( \hat{v} \), be expressed in the following recurrence form:

\[ v_{k+1} = v_k e^{\lambda \Delta T} + \hat{v}_k (1 - e^{\lambda \Delta T}) \]  \hspace{1cm} (35)

where \( \Delta T \) is the step time.

8.1. Delay

The delay compensation is obtained by applying, at each step time \( k \), the command required at step \( k + \tau / \Delta T \) in which it is effective. The command law that compensates for the delay is designated as follows:

\[ v_k^\tau = \hat{v}_{k+\tau/\Delta T} \]  \hspace{1cm} (36)

where \( \hat{v}_{k+\tau/\Delta T} \) is the command related to the predicted platform state at time \( k + \tau / \Delta T \). The predicted platform state (position and velocity) is obtained by an on-the-fly simulation of the platform’s movement from knowledge of its current position and the sequence of reference velocities applied between \( k \) and \( k + \tau / \Delta T - 1 \).

The average prediction error caused by the localization noise and limited accuracy of the dynamic controller is dependent on the delay magnitude \( \tau \). For example, we assume that a delay of one
second — which we consider to be very wide for a closed-loop control of most mobile platforms — is still subject to cause, on average, small prediction errors even with state-of-the-art localization approaches and dynamic controllers.

8.2. Poles compensation
By isolating the velocity command in 35, we obtain the following equation:

\[ \hat{v}_k = \frac{v_{k+1} - v_k e^{\lambda \Delta T}}{1 - e^{\lambda \Delta T}}. \]  

(38)

Assuming that the velocity \( v^*_k \) corresponds to the actual velocity of the platform, the velocity command is thus given by:

\[ \hat{v}_k = \frac{v^*_{k+1} - v^*_k e^{\lambda \Delta T}}{1 - e^{\lambda \Delta T}} \]  

(39)

8.3. Consideration for the command saturation
If the acceleration from step \( k \) to step \( k + 1 \) is \( a_k \), we have:

\[ v^*_{k+1} = v^*_k + a_k \Delta T. \]  

(40)

Inserting (40) in (39), we obtain:

\[ \hat{v}_k = \frac{v^*_k + a_k \Delta T}{1 - e^{\lambda \Delta T}} \]  

(41)

If \( \hat{v}^{sat} \) is the saturation command, we need \( |\hat{v}_k| \leq \hat{v}^{sat} \), thus:

\[ \left| v^*_k + a_k \Delta T \frac{1}{1 - e^{\lambda \Delta T}} \right| \leq \hat{v}^{sat} \]  

(42)

The maximum value of the left part is obtained with \( v^*_k = v_{max} \) and \( a_k = a_{max} \). Thus, the values of \( v_{max} \), \( a_{max} \), and \( \Delta T \) must be chosen in accordance with the following condition:

\[ \left| v_{max} + a_{max} \frac{\Delta T}{1 - e^{\lambda \Delta T}} \right| \leq \hat{v}^{sat} \]  

(43)

9. Experimental Results
The waypoints guidance controller was implemented into a module as part of a control architecture build on Acropolis. The same implementation is used both for simulations and real experiments.

9.1. Simulations
9.1.1. Substantiation of the driving profile. In this simulation, we use a basic waypoints sequence to ascertain that all parameters of the driving profile are borne out in practice.

The driving profile is: \( \{ v_F = 1.5 \text{ m/s}, v_B = 0.8 \text{ m/s}, \omega_{max} = 0.5 \text{ rad/s}, a_{min} = 0.3 \text{ m/s}^2, a_{max} = 0.4 \text{ m/s}^2, \alpha_{min} = 0.4 \text{ m/s}^2, \alpha_{max} = 0.5 \text{ rad/s}^2, \xi = 3, \beta = 7, r = 0.13 \text{ m} \} \). From the equations of Table III, the minimum value of the decrease exponent of the longitudinal velocity \( \beta \) providing an admissible radius of 0.13 m is 4.43. By choosing a value of 7, the radius of divergence is

\[ 2 \text{ For example, if a mobile platform that moves at 0.5 m/s has suddenly to stop because of an obstacle ahead, a non-compensated delay of one second will cause the platform to stop 50 cm farther from this current position, which could in many cases engender collisions.} \]
Waypoints guidance of differential-drive mobile robots

Fig. 9. Simulation setup for the experiment of Figs. 10 and 11.

Fig. 10. (Colour online) A basic waypoints guidance with two waypoints \( \{w_1, w_2\} \).

\[ r_{lim} = 0.002 \, \text{m}, \text{which is handily inferior to the admissible radius of 0.13 m. Convergence is therefore well ensured.} \]

Figure 9 presents the simulation setup employed in Acropolis. A waypoints sequence, a driving profile and a platform’s configuration (position and orientation) input the Waypoints-Following Controller. A clock triggers, with a frequency of 60 Hz \( (\Delta T = 1/60 \, \text{s}) \), an update of localization, reference velocities by the Waypoints-Following Controller and motors’ commands. The localization is performed by dead-recognition. The simulation is realized in real-time, during the course of which the data and simulation interface are recorded. This \textit{a priori} setup presupposes no localization error and no control error. The dynamic response of the motor is thus assumed to be nearly ideal. The main perturbation stems from the delay introduced by the exchange of information from one module to another. In particular, the output of the Waypoints-Following Controller is ready for the Motors module one period later, and the output of the localization is ready one period after integration of this latter velocity. This delay of 2/60 s gives rise to small oscillations of the rotational velocity at experiment’s end, as shown in Fig. 11. If the approach presented in Section 8 is used to compensate for this delay, we obtain the soft rotational response depicted in Fig. 12.

Figure 11 reveals that the maximum longitudinal velocity, longitudinal acceleration, longitudinal deceleration, rotational velocity, rotational acceleration and rotational deceleration are accurately reproduced. The longitudinal and the rotational velocities show progressive variations. The accuracy of navigation is also satisfied.

In this experiment, smoothness of path does not reflect smoothness of navigation. Indeed, the curve in Fig. 10 is very steep from waypoint \( w_1 \) to waypoint \( w_2 \) whereas the velocity varies smoothly nonetheless.

This simulation underscores the importance of reducing the longitudinal velocity in conjunction with the orientation error. Indeed, if the longitudinal velocity does not decrease with the orientation error, the only feasible way to maintain the platform on a very steep curve is by generating a high rotational velocity that may imply an overshoot of the kinematic constraints with a jerk movement. It is worth mentioning that this type of steep curve is often necessary in restrained areas such as hallways and intersections.

9.1.2. \textit{Practical examples}. We present practical examples of waypoints guidance achieved with the following driving profile: \( \{v_F = 1.5 \, \text{m/s}, v_B = 0.8 \, \text{m/s}, \omega_{max} = 0.5 \, \text{rad/s}, a_{min} = 0.5 \, \text{m/s}^2, a_{max} = 0.6 \, \text{m/s}^2, a_{min} = 0.4 \, \text{m/s}^2, a_{max} = 2.4 \, \text{rad/s}^2, \zeta = 3, \beta = 7, r = 0.13 \, \text{m}\} \). Let us remark that the velocity and acceleration limits are chosen for a smooth and effective navigation.
Waypoints guidance of differential-drive mobile robots

Fig. 11. (Colour online) Verification of a simple waypoint following \( \{w_1, w_2\} \).

Fig. 12. (Colour online) Rotational velocity obtained with a delay compensation of 1/30 s.

Four waypoints guidance are presented, namely with:

1. Spaced waypoints with an irregular progression in Fig. 13.
2. Well-aligned waypoints with a decreasing separation distance in Fig. 14.
3. Waypoints having a spiral progression in Fig. 15.
4. Waypoints having a star progression in Fig. 16.

Examination of the relevant figures reveals that the accuracy of waypoints traversal is satisfied in all waypoints guidances. Extreme velocities and accelerations are respected; moreover the movements are smooth and efficient (without useless oscillation and deceleration). These examples confirm that the proficiency of the waypoints guidance is not contingent on the sequence of waypoints.

9.1.3. Compensation for the delay and the poles in the dynamic response. Figure 17 presents an example of the effect of the feed-forward compensator for a platform having a pole in the
longitudinal velocity response and the rotational velocity response at $-3 \, \text{s}^{-1}$ ($\lambda' = \lambda'' = -3 \, \text{s}^{-1}$) and an actuation delay $\tau$ of 0.15 s. The dotted line represents the waypoints guidance without feed-forward compensation (and without closed-loop dynamic control), while the plain line represents the waypoints guidance with the feed-forward compensation. The longitudinal and rotational velocity responses are plotted in Fig. 18.
Waypoints guidance of differential-drive mobile robots

9.2. Experiments with a real mobile platform, temporal constraint
The objective of these experiments is to evaluate the capacity of a robotic platform, exempt of external perturbations from the surroundings, to comply with temporal constraints. The platform is mounted on a commercial wheelchair FreeStyle F11 of Sunrise Medical. In order to circumvent localization errors and floor perturbations, the platform was mounted on a jack during the trials. Its velocities were obtained from odometer sensors and its movements were simulated from dead-recognition (time

Without compensation, the slow dynamic response causes a sinusoidal trajectory and increases the execution time (25 s rather than 20 s with compensation). The feed-forward compensation reduces the magnitude of oscillations and provides for a straightforward trajectory with effective velocity responses. It should be noted that the velocity variation can mistakenly appear abrupt due to the experiment time duration of \( \sim 25 \) s. We can verify that the speed acceleration limit is respected and that navigation is smooth.

Fig. 16. (Colour online) Guidance through waypoints with a star progression.

Fig. 17. (Colour online) Two waypoints guidance with and without feed-forward compensation.
integration of velocities). The only perturbations, internal to the platform, come from the actuator system including the dynamic controller, circuitry and motors. This setup allows for evaluating the temporal accuracy that can be obtained under good conditions.

The nominal driving profile is: \(\{ v_F = 1.0 \text{ m/s}, v_B = 0.5 \text{ m/s}, \omega_{\text{max}} = 0.4 \text{ rad/s}, a_{\text{min}} = 0.25 \text{ rad/s}^2, a_{\text{max}} = 0.5 \text{ m/s}^2, \alpha_{\text{min}} = 0.3 \text{ rad/s}^2, \alpha_{\text{max}} = 1.0 \text{ rad/s}^2, \zeta = 2.0, \beta = 4.50, r = 0.15 \text{ m} \}\). The velocity response of this platform has a delay of 0.3 s, a pole of \(-4 \text{ s}^{-1}\) for the longitudinal velocity and a pole of \(-9 \text{ s}^{-1}\) for the rotational velocity, which are compensated by the feed-forward compensator. This unit then inputs a proportional-integral-derivative (PID) dynamic controller with a forget factor on the integral of the error.

Table IV presents the statistics for temporal constraints set to 50, 70 and 100 s. Ten tests per temporal constraint are performed. Statistics include the average duration \(T\), average error of duration \(\epsilon_T\), standard deviation of duration \(\sigma_T\), average deviation between the commanded and measured velocities for the longitudinal velocity \((\epsilon_{\text{lin}})\) and rotational velocity \((\epsilon_{\text{ang}})\), average accuracy of duration and the repetitiveness. The two last measures are defined as followed:

\[
\text{duration accuracy} = \left(1 - \frac{\epsilon_T}{T_d}\right) \times 100 \tag{44}
\]

\[
\text{repetitiveness} = \left(1 - \frac{\sigma_T}{T_d}\right) \times 100 \tag{45}
\]

In Table IV, the temporal constraint of 50 s is reached with an average accuracy of duration of 96.3%, that of 70 s is reached with an average accuracy of duration of 97.3% while that of 100 s yields an average accuracy of duration of 99.2%. The repetitiveness values are respectively 98.6%, 99.1% and 99.2% for the 50, 70 and 100 s duration constraints. Since the driving profile is updated for each temporal constraint in open-loop, this duration accuracy is related to the effectiveness of the feed-forward compensation unit, as confirmed by the decreasing values of the velocity deviations \(\epsilon_{\text{lin}}\) and \(\epsilon_{\text{ang}}\) as a function of increasing trial duration constraint. These experiments demonstrate that our approach can accurately fulfill temporal constraints in a repeated, predictive and smooth fashion.

### 9.3. Self-localization and waypoints guidance

This example is executed with a mini-ATRV in an obstructed environment. A self-localization\(^{25}\) is performed by a Kalman filter that fuses a transition model based on odometry measurements to an
Table IV. Statistics of 10 waypoints guidance for each temporal constraint.

<table>
<thead>
<tr>
<th>$T_d$</th>
<th>$\epsilon^T$</th>
<th>$\sigma^T$</th>
<th>$\epsilon^\text{line}$</th>
<th>$\epsilon^\text{ang}$</th>
<th>d.accuracy</th>
<th>repet.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s</td>
<td>s</td>
<td>m/s</td>
<td>rad/s</td>
<td>(%)</td>
<td>(%)</td>
</tr>
<tr>
<td>50</td>
<td>1.84</td>
<td>51.84</td>
<td>0.71</td>
<td>0.030</td>
<td>96.3</td>
<td>98.6</td>
</tr>
<tr>
<td>70</td>
<td>1.86</td>
<td>68.14</td>
<td>0.63</td>
<td>0.020</td>
<td>97.3</td>
<td>99.1</td>
</tr>
<tr>
<td>100</td>
<td>0.84</td>
<td>99.40</td>
<td>0.82</td>
<td>0.014</td>
<td>99.2</td>
<td>99.2</td>
</tr>
</tbody>
</table>

Fig. 19. (Colour online) Simultaneous self-localization and waypoints guidance.

observation model based on a points-matching approach using a reference map and surrounding environment detected by a rangefinder. In this environment, the localization error is $\sim 0.1$ m.

The following parameters values are used: \( v_F = 0.32 \text{ m}, v_B = 0.25 \text{ m}, \omega_{\text{max}} = 0.35 \text{ rad/s}, \omega_{\text{min}} = \omega_{\text{max}} = 0.1 \text{ m/s}, \alpha_{\text{max}} = 0.2 \text{ rad/s}^2, \alpha_{\text{min}} = 0.15 \text{ rad/s}^2, \beta = 3, \lambda = 2, r_i = 0.18 \text{ m} \forall i \). The velocity values are small to ensure safe navigation in the range of obstacles. The backward motion mode is automatically selected from $W_{10}$ to $W_{20}$ in order to reduce time for turning.

The estimated velocity response of this platform has a delay of 0.6 s\(^3\), a pole of $-8 \text{ s}^{-1}$ for the longitudinal velocity and a pole of $-8 \text{ s}^{-1}$ for the rotational velocity. However, we observed that the delay varied in time from about 30% of its nominal value. The dynamic response is thus less effectively compensated with respect to the previous experiment.

The effective path, represented by the solid line in Fig. 19, respects the accuracy condition. It is gently sinusoidal because of the fluctuations in the localization error. This fluctuation slightly slows down the average longitudinal velocity with reorienting, which provides a safe and accurate navigation in the narrow entrance of the room at the up-right region. Figure 20 shows the reference velocity of the platform as well as the effective velocity measured from the odometers. The reference angular velocity has small overshoots of the maximum velocity constraint due to the error in the estimation of the delay in the control loop considered by the compensator unit. Many temporary and small reorienting movements compensate for perturbations in platform localization so as to ensure stable navigation with nearly maximum velocities.

\(^3\) The robot was controlled by a remote machine that enables us to record a movie of its internal state evolution at the basis of its movements. However, the communication time increased the delay of control.
10. Conclusions and Future Works
This paper introduced a kinematic controller for waypoints guidance of mobile platforms that computes in real-time, with respect to a driving profile, the reference velocities ensuring that the platform movements converge through a successive set of waypoints within a predefined accuracy of traversal. A convergence analysis, assessed by experimental results, demonstrates that the controller can provide a smooth, accurate, fast and stable navigation for any waypoints sequences. A proposed unit for compensating the delay and the pole in the dynamic response of the platform has shown to be highly relevant, even essential, in the command of real robots. Experiments on a smart wheelchair demonstrated the plausibility of the approach for achieving a temporal constraint by adapting the driving profile.

In future works, we could strive to adapt the kinematic laws of waypoints guidance for other platforms including car-like robots, tri-cycle robots and walking robots. The kinematic laws should also be considered in video games to provide natural movements of virtual objects.

Finally, it would be relevant to extend the comparison study of this controller with previous ones through extensive experiments incorporating the complexity of each approach.

Acknowledgments
This work has been supported by the Natural Science and Engineering Council of Canada (NSERC) through Grant No. CRD 349481-06. The author wishes to acknowledge the contribution of several members of the Perception and Robotics Laboratory for their logistic support and their works on robotic platforms, namely Paul Cohen, Hay Nguyen, Sousso Kelouwani, Alexandre Fortin, Vincent Zalzal and Raphael Gava.

References
Waypoints guidance of differential-drive mobile robots


