Complementary-route based ICR control for steerable wheeled mobile robots

Mohamed Sorour, Andrea Cherubini, Abdellah Khelloufi, Robin Passama, Philippe Fraisse

Complementary-Route Based ICR Control for Steerable Wheeled Mobile Robots

Mohamed Sorour[1], Andrea Cherubini[2], Abdellah Khelloufi[3], Robin Passama[2], Philippe Fraisse[2]

Abstract

Emerging industrial applications involving mobile manipulation in the presence of humans is driving attention towards steerable wheeled mobile robots (SWMR), since these can perform arbitrary $2D$ planar trajectories, providing a reasonable compromise between maneuverability (necessary for human avoiding algorithms) and effectiveness. Instantaneous center of rotation (ICR) based kinematic models and controllers are the most suited for such robots, as they assure the existence of a unique ICR point at all times. However, unsatisfactory behavior do exist in numerous applications requiring frequent changes in the sign of the angular velocity command. This is typically the case for robot heading control: moving the ICR point from one border of the $2D$ ICR space to the other makes it pass by the robot geometric center, where only pure rotations are feasible. This behavior is not desirable and should be avoided. In this paper, we propose a novel complementary route ICR controller, where the ICR can go from one extreme to the other by means of border switching in one sample period. Thanks to this approach, fast response to the velocity commands is achieved with little steering motion. The new algorithm has been tested successfully in simulations and experiments, and is more time efficient with far more satisfactory behavior than the state-of-art direct route based controllers. These results have been also confirmed quantitatively, using a newly developed metric, the command fulfillment index (CFI).

Keywords: Steerable mobile robot, pseudo-omni mobile robot, nonholonomic omnidirectional mobile robot.

1. Introduction

Steerable wheeled mobile robots employing traditional wheels are getting more popular in industrial applications, due to their lower cost for a given load carrying capacity as compared to fully omnidirectional (holonomic) mobile robots (FOMR) employing swedish or other non-conventional wheels [1]. Other promising holonomic alternatives employ fully powered castor wheels [2, 3] or differential drive system with offset turret [4, 5]. However, since these are not yet commercially available, they are not used in industrial automation.

Despite being non-holonomic, SWMR robots can perform complex $2D$ planar trajectories, assuming correct initial wheel orientation, followed by steer coordination (to maintain a unique ICR point throughout operation). However, their kinematic structure has several challenging research problems, the most critical of which are: 1. proper steering coordination, to avoid actuator fighting [6, 7, 8, 9], 2. avoidance of kinematic and representational singularities [7, 10, 11, 12, 13, 14, 15, 16], and 3. fulfillment of steer joint performance limits while solving the previous problems [13, 17, 18, 19, 20, 21, 22, 23]. Kinematic modeling and control of SWMR is usually done either in the $2D$ ICR space [13, 16, 20, 21] or in the $3D$ Cartesian space [7, 10, 24, 25]. In [11, 26, 27], both are combined, since the ICR space is best suited for steering coordination (to avoid actuator fighting and wheel slippage) as it ensures the existence of a unique ICR point, while Cartesian space is utilized for robot speed control. Here, we also use ICR-based steer coordination, and the focus of this work is precisely to enhance ICR based controllers, through the design of a complementary route strategy.

In applications requiring that some heading angle is maintained (e.g., vision-based tasks where features must stay in the field of view), or that the translation verse changes, the ICR point is required to move long

[1]M. Sorour was with LIRMM, Université de Montpellier, CNRS, Montpellier, France and with PSA Peugeot-Citroen, Velizy Villacoublay, France at the time of this work and is now with **ADD**

[2]A. Cherubini, R. Passama and P. Fraisse are with LIRMM, Université de Montpellier, CNRS, Montpellier, France. firstname.lastname@lirmm.fr

[3]A. Khelloufi is with the University of Sciences and Technology Houari Boumediene USTHB, BP32 EL-ALIA, 16111 Bab Ezzouar Algiers, Algeria and with the Center for Developement of Advanced Technologies CDTA, 20 Aout 1956 City, Baba Hassen Algiers, Algeria. akhelloufi@cdta.dz

distances from one extreme of the workspace to the other, usually passing by the robot geometric center, where the feasible robot velocity is limited. In such scenarios, the state-of-art ICR based controllers will lead to unsatisfactory behaviors.

To solve this problem, here we propose a comparison between the direct and complementary ICR routes, the former (state-of-art approach) being the shortest straight line connecting the current and desired ICR points, while the latter (proposed here) connecting the ICR extreme borders via a border ICR point that is chosen to minimize the total ICR distance moved across borders. The 4 borders here define the maximum values that the ICR point is allowed to take on the x and y axes of the geometrically centered robot frame. Instead of moving directly to the desired ICR, the complementary route will move first to the optimum border point lying on the nearest border line, switch borders, *for example: from the $+y$ to the $-y$ border line* in one sample period, and finally move to the desired ICR point. A graphical representation of such process is shown in Fig. 1.

In [20], the direct ICR route is stereographic projected onto a unit sphere, where moving between complementary borders can be done at the pole of the unit sphere. However, the authors do not provide any investigations on border switching nor simulations showing the joint-space performance. To the best of the author's knowledge, the work presented in this paper is the first connecting ICR borders to obtain more efficient SWMR control.

One of the contributions of the method adopted here is to provide a solution that is decoupled from the high level command/perception controller. Here, the trajectory planning does not need any prior knowledge of the robot structure: only the current robot velocity and pose will suffice to close the feedback loop. The error between the provided command and the actual capability of the robot is handled locally thanks to the proposed controller. In order to quantitatively assess the enhanced performance, we introduce the command fulfillment index (CFI) that is based on the robot velocity error vector, and we use it to compare the direct and complementary route controllers.

In this work, the authors extend the kinematic control framework of [26] so that it:

1. can handle sign variations in the rotational speed commands, that are frequent in orientation (heading) control applications,
2. has better responsiveness (assessed via the CFI),
3. demands less steering motion (in terms of both velocity and acceleration).

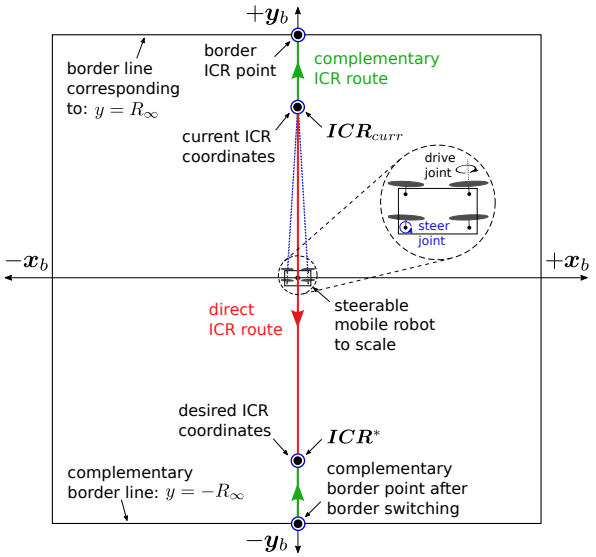

Figure 1: We bound the robot's $2D$ ICR space by 4 border lines corresponding to $x_b = \pm R_\infty$ and $y_b = \pm R_\infty$ defined in the robot frame $\mathcal{F}_b$ with origin at the robot center. An ICR point moving from the current to the desired value can follow a direct route (in red), or a complementary one (in green).

In the rest of the paper, Section II presents the necessary background on the kinematic model and controller. The complementary ICR route based controller is detailed in Section III. The complementary/direct route decision making algorithm is presented in Section IV. Experiments are depicted in Section V. Conclusions are finally given in Section VI.

2. Relevant Background

In this section, we briefly recall the SWMR kinematic model detailed in [25] and based on the pioneer works [24, 28, 29, 30, 31], together with the discontinuity robust ICR controller [26] that will be used in this paper.

2.1. Cartesian Space Kinematic Model

The schematic of a SWMR is shown in Fig. 2 for a 4 wheeled robot. However, the model is generic for any SWMR with $N \geq 3$ wheels. Let $\mathcal{F}_I = (o_I \,|\, \boldsymbol{x}_I, \boldsymbol{y}_I, \boldsymbol{z}_I)$ be the inertial frame, $\mathcal{F}_b = (o_b \,|\, \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b)$ the mobile base frame, with origin o_b located at the base geometric center, $\mathcal{F}_{hi} = (o_{hi} \,|\, \boldsymbol{x}_{hi}, \boldsymbol{y}_{hi}, \boldsymbol{z}_{hi})$ the i^{th} hip frame ($i = 1, \ldots, N$), attached to the fixed part of the steering joint, and related to the base frame by a fixed transformation matrix, and $\mathcal{F}_{si} = (o_{si} \,|\, \boldsymbol{x}_{si}, \boldsymbol{y}_{si}, \boldsymbol{z}_{si})$ the steering frame, attached to the movable part. The hip and steering frames share the same origin, with relative orientation β_i (the steering angle). Frame $\mathcal{F}_{wi} = (o_{wi} \,|\, \boldsymbol{x}_{wi}, \boldsymbol{y}_{wi}, \boldsymbol{z}_{wi})$

is attached to (but not rotating with) the i^{th} wheel, assigned such that $\boldsymbol{x}_{wi}$ points along the heading of the wheel, which rotates about $\boldsymbol{y}_{wi}$ by the driving angle ϕ_i. All the frames have the $\boldsymbol{z}$ axis pointing upwards. Let the mobile base pose w.r.t. the inertial frame define the 3×1 task space coordinates $\boldsymbol{\xi} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$. A left superscript is added to indicate the frame in which the pose is expressed, for instance ${}^I\boldsymbol{\xi}$ and ${}^b\boldsymbol{\xi}$ denote the robot pose, expressed in the inertial and base frames respectively. To lighten the notation in the sequel, unless otherwise specified, the left superscript is omitted for vectors expressed in the base frame.

Applying the *rolling with no slipping* and the *no lateral skidding* kinematic constraints to the i^{th} wheel velocity $\boldsymbol{v}_{ci} = \begin{bmatrix} v_{ti} & v_{ni} & 0 \end{bmatrix}^T$ at the ground contact point o_{ci} (expressed in the wheel frame $\mathcal{F}_{wi}$), with v_{ti} and v_{ni} respectively the i^{th} tangential and normal velocities, we obtain [26]:

$$\dot{\beta}_i = \frac{-\boldsymbol{g}(\beta_i)\ddot{\boldsymbol{\xi}}}{\frac{d\boldsymbol{g}(\beta_i)}{d\beta_i}\dot{\boldsymbol{\xi}}}, \tag{1}$$

$$\dot{\phi}_i = \frac{1}{r_w}\boldsymbol{f}(\beta_i)\dot{\boldsymbol{\xi}} + \frac{d}{r_w}\dot{\beta}_i, \tag{2}$$

$$\boldsymbol{f}(\beta_i) = \begin{bmatrix} c(\beta_i) & s(\beta_i) & d - h_{yi}c(\beta_i) + h_{xi}s(\beta_i) \end{bmatrix},$$

$$\boldsymbol{g}(\beta_i) = \begin{bmatrix} -s(\beta_i) & c(\beta_i) & h_{xi}c(\beta_i) + h_{yi}s(\beta_i) \end{bmatrix},$$

where $h_{xi} = \pm b$ and $h_{yi} = \pm a$ denote the position of the i^{th} hip frame origin o_{hi} in the base frame, $a, b \in \mathbb{R}^+$, $c(*)$ and $s(*)$ short hand $\cos(*)$ and $\sin(*)$, d and r_w are the wheel offset and radius respectively. The *no skidding* constraint imposes restrictions on the robot motion (wheels cannot move sideways), and forces the existence of a unique ICR point, around which the base and all wheels must rotate. From such constraint, we construct the kinematic constraint matrix $\boldsymbol{G}(\boldsymbol{\beta})$ as:

$$\boldsymbol{G}(\boldsymbol{\beta})_{N\times3} = \begin{bmatrix} \boldsymbol{g}(\beta_1)^T & \dots & \boldsymbol{g}(\beta_N)^T \end{bmatrix}^T. \tag{3}$$

Equations (1) and (2) represent the SWMR inverse actuation kinematic model (IAKM). To compute the task space velocity response from the joint space velocity measurements, we need the odometry model, i.e., the forward actuation kinematic model (FAKM):

$$\dot{\boldsymbol{\xi}} = \boldsymbol{F}^+_{(\rho)}(\hat{\boldsymbol{\beta}})(r_w\hat{\dot{\boldsymbol{\phi}}} - d\hat{\dot{\boldsymbol{\beta}}}), \tag{4}$$

$$\boldsymbol{F}(\hat{\boldsymbol{\beta}})_{N\times3} = \begin{bmatrix} \boldsymbol{f}(\hat{\beta}_1)^T & \dots & \boldsymbol{f}(\hat{\beta}_N)^T \end{bmatrix}^T.$$

In (4), the hat symbol $\hat{}$ indicates a measurement value and $\boldsymbol{F}^+_{(\rho)}(*)$ denotes the damped pseudo-inverse of $\boldsymbol{F}(*)$ [32, 33], evaluated using:

$$\boldsymbol{F}^+_{(\rho)}(*) = (\boldsymbol{F}^T(*)\boldsymbol{F}(*) + \delta^2\boldsymbol{I}_{3\times3})^{-1}\boldsymbol{F}^T(*),$$

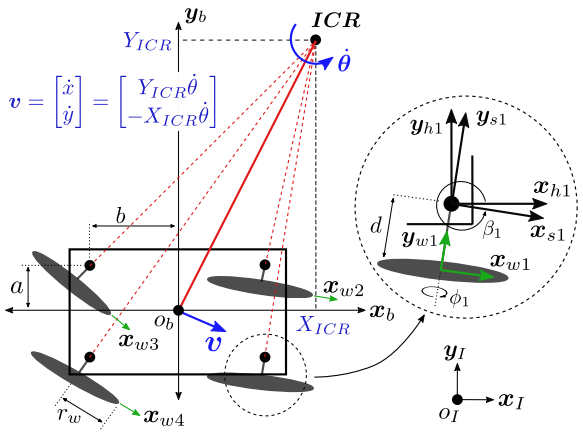

Figure 2: Schematic model of a four wheeled steerable robot.

with $\delta \in \mathbb{R}$, the damping factor, and $\boldsymbol{I}_{3\times3}$ the $3D$ identity matrix.

Remark 1. *In (2), there is a relationship with the steering velocity $\dot{\beta}_i$ since this wheel structure is off-centered. However, without loss of generality, we can assume that the driving velocity $\dot{\phi}_i$ is independent of the steering (the constant d/r_w can be automatically handled by low level controllers, since it will not result in any robot motion). In light of this, from a deeper look at (1), (2), and (4), we can see two separate systems, one for steering, the other for driving, and each depending on the required robot velocity/acceleration. Although the steering system can change the robot configuration, and as such the feasible velocity direction (the direction of the robot velocity vector), if $\dot{\boldsymbol{\phi}} = 0$, no motion will occur. Similarly, the driving system can change the magnitude of the robot velocity along the direction corresponding to the current steer configuration, but alone (i.e. when $\dot{\boldsymbol{\beta}} = 0$), it cannot change the robot velocity direction.*

2.2. Motion Discontinuity Robust ICR Controller

The enhanced motion-discontinuity robust controller framework originally introduced in [26] is depicted in Fig. 3, with the colored blocks representing the additions detailed in this paper. Such controller was developed to handle the velocity commands discontinuities that are frequent in human-robot collaborative/cooperative scenarios. The desired $3D$ Cartesian space robot motion ($\dot{\boldsymbol{\xi}}^*$, and $\ddot{\boldsymbol{\xi}}^*$) is generated by a high level controller (out of scope here). This is then mapped to the $2D$ ICR space, and the output desired ICR motion ($\boldsymbol{ICR}^*$, and $\dot{\boldsymbol{ICR}}^*$), along with the current ICR coordinates $\boldsymbol{ICR}_{curr}$ are fed to the ICR Point Velocity controller. This outputs a reference signal $\boldsymbol{ICR}_{ref}$ that is

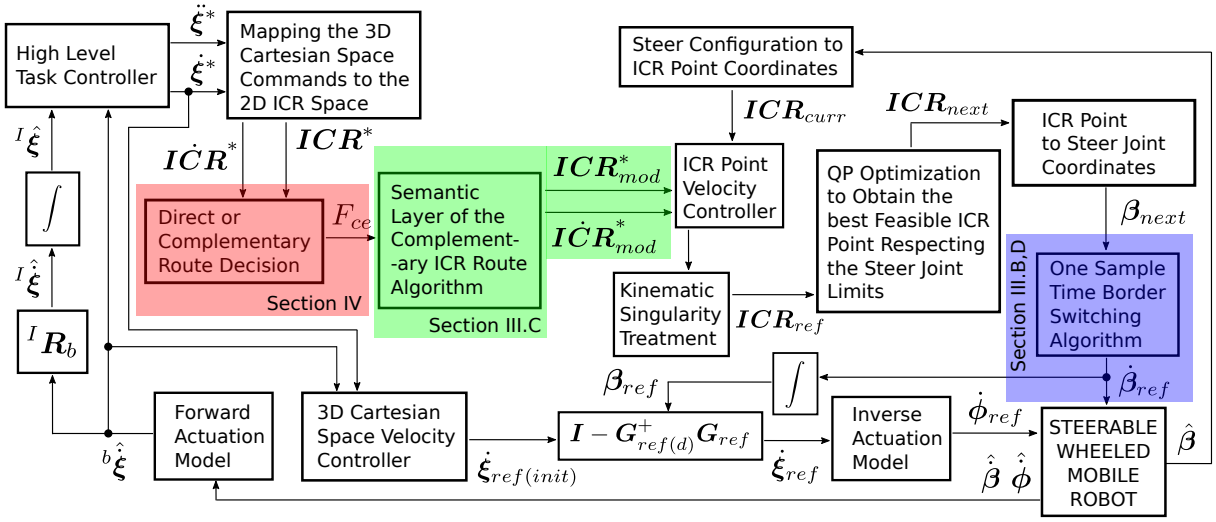

Figure 3: Motion-discontinuity robust controller [26] with the complementary controller additions shown in colored blocks.

then used by a quadratic programming (QP) optimization algorithm to decide the *"next sample time"* ICR coordinates $\boldsymbol{ICR}_{next}$ that will minimize the quadratic cost error: $\|\boldsymbol{ICR}_{ref} - \boldsymbol{ICR}_{next}\|_2^2$ while respecting the joint limits (formulated as linear constraints). We use $\boldsymbol{ICR}_{ref}$ rather than $\boldsymbol{ICR}^*$ in the cost function, to obtain a smooth behavior since the former is error-dependent. The corresponding steer joint reference signal $\boldsymbol{\beta}_{ref}$ is then evaluated (while fixing all numeric issues) and differentiated, to obtain the $\dot{\boldsymbol{\beta}}_{ref}$ that is sent to the robot low level controller. At the same time, a decoupled 3D Cartesian space robot velocity controller is implemented. Its initial output $\dot{\boldsymbol{\xi}}_{ref(init)}$ is projected onto the null space of the *"next sample time"* kinematic constraint matrix $\boldsymbol{G}(\beta_{ref})$ from (3), to obtain the feasible control signal $\dot{\boldsymbol{\xi}}_{ref}$ that is compatible with the *"next sample time"* robot configuration. The reference wheel rate $\dot{\boldsymbol{\phi}}_{ref}$ is then obtained using the IAKM from (2). For more details about each of the non colored blocks in Fig. 3, the reader is referred to our previous work [26].

The main goal of the ICR motion controller in [26] was to drive the $\boldsymbol{ICR}_{curr}$ to the desired one $\boldsymbol{ICR}^*$ via the controller output $\boldsymbol{ICR}_{ref}$. Instead, the work in hand aims at manipulating the desired signal $\boldsymbol{ICR}^*$, so that $\boldsymbol{ICR}_{ref}$ follows a complementary route instead of the direct one – whenever the former is found to be more time efficient. Such algorithm is summarized in 3 main blocks (colored) in Fig. 3, namely: *Direct or Complementary Route Decision* (detailed in Sect. 4), *Semantic layer of the complementary ICR route* (Sect. 3.3), and *One sample time border switching* (Sections 3.2 and 3.4).

3. Complementary ICR Route

In this section, the complementary ICR route algorithm is described/formulated. First, we show the difference with the direct route algoritmlm, which in some situations can have longer length. Then, we formulate a QP optimization problem, to find the shortest complementary route from current $\boldsymbol{ICR}_{curr}$ to desired ICR $\boldsymbol{ICR}^*$. We then describe in detail how to implement the algorithm in conjunction with the discontinuity-robust ICR controller recently developed in [26].

3.1. Complementary Versus Direct Routes

The direct and complementary ICR routes are depicted in Fig. 4 (upper) in red and green, respectively. The direct is the shortest direct straight line connection between the current point $\boldsymbol{ICR}_{curr}$, and the desired one $\boldsymbol{ICR}^*$. The complementary, on the other hand, is a concatenation of two successive straight lines *(detailed in next section)* via the border lines, defined below.

Definition 1. *$R_\infty \in]0, \infty[$ is a parameter representing the radius of curvature at infinity of the path followed by the robot, corresponding to pure translation motion. It serves as an upper limit to the values allowed for the x and y components of the* $\boldsymbol{ICR}$.

Remark 2. *Setting R_∞ to a value other than ∞ will result in non pure translation motion: there will always exist a rotation velocity component. Shall the desired command correspond to pure translation, the controller proposed in this work will provide the necessary corrective action.*

Definition 2. *A border line defines a bound (upper or lower) to the accessible workspace of the ICR in any of the $\mathcal{F}_b$ coordinates. As such, there exist 4 border lines: $x = \pm R_\infty$ and $y = \pm R_\infty$.*

Definition 3. *A border point is any point belonging to a border line.*

Definition 4. *The border is the perimeter of the square constructed from the 4 border lines.*

For the particular pair of $\boldsymbol{ICR}_{curr}$ and $\boldsymbol{ICR}^*$ shown in Fig. 4 (upper), the complementary route is the shortest and as such is expected to be time optimal. The ICR border point $\boldsymbol{ICR}_{bor} = \begin{bmatrix} X_{bor} & Y_{bor} \end{bmatrix}^T$ used to construct the complementary route belongs to the border line $y = R_\infty$, although any $\boldsymbol{ICR}_{bor}$ belonging to the 4 border lines can be used as well. Furthermore, there exist infinite points satisfying $y = R_\infty$, each resulting in a complementary route, as shown in Fig. 4 (middle), where the blue and magenta lines represent alternative solutions. This reflects the need to choose the optimal (shortest) complementary route, which in this example (and for this particular border line), is represented by the green line. In order to find the $\boldsymbol{ICR}_{bor}$ corresponding to the shortest complementary route (refer to Fig. 4 lower), we use quadratic programming.

The feasible domain is the perimeter of the square border (constructed from the 4 border lines) as depicted in Fig. 4 (lower). To compute the shortest complementary route, we divide the domain in four QP formulations subject to constraints, each representing one of the border lines, shown in Fig. 4 (lower) with different colors. The general formulation is given by:

$$\begin{aligned} \underset{\boldsymbol{ICR}_{i(bor)}}{\text{minimize}} \quad & f(\boldsymbol{ICR}_{i(bor)}) = \\ & \|\boldsymbol{ICR}_{curr} - \boldsymbol{ICR}_{i(bor)}\|_2^2 \\ & + \|\boldsymbol{ICR}^* - \boldsymbol{ICR}'_{i(bor)}\|_2^2, \\ \underset{i=[1,2,3,4]}{\text{subject to}} \quad & \boldsymbol{c}_i \boldsymbol{ICR}_{i(bor)} = (-1)^i R_\infty, \end{aligned} \tag{5}$$

where $\| * \|_2^2$ is the squared Euclidean norm, $\boldsymbol{ICR}_{i(bor)}$ is the ICR border point corresponding to the shortest (local minimum) complementary route using the i^{th} border line, $\boldsymbol{ICR}'_{i(bor)} = -\boldsymbol{ICR}_{i(bor)}$ and $\boldsymbol{c}_{1,2} = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $\boldsymbol{c}_{3,4} = \begin{bmatrix} 0 & 1 \end{bmatrix}$ indicate the constraint vectors. The set of analytic solutions to the optimization problem in (5) (recall that 1 local optimum solution will exist for each of the linear constraints) can be expressed as:

$$\begin{aligned} X_{bor} &\in \{X_{opt}, X_{opt}, R_\infty, -R_\infty\}, \\ Y_{bor} &\in \{R_\infty, -R_\infty, Y_{opt}, Y_{opt}\}, \end{aligned} \tag{6}$$

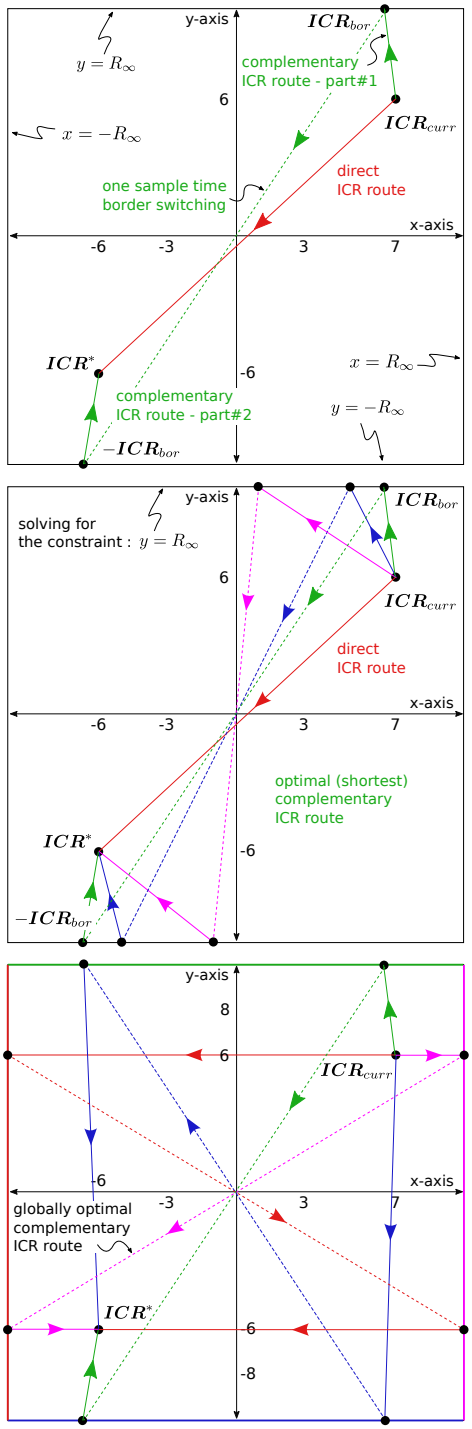

Figure 4: Illustrative scheme showing the complementary route versus the direct ICR routes (upper), the optimal complementary ICR route (green) among other possible routes (magenta and blue) satisfying one constraint $y = R_\infty$ (middle), and the global optimum complementary route (magenta) among the four local optima, each satisfying one straight line constraint (bottom).

with $X_{opt} = (X_{curr} - X^*)/2$, and $Y_{opt} = (Y_{curr} - Y^*)/2$. Finally, we compute the index j corresponding to lower cost f_{min}, to obtain the global minimum:

$$\begin{aligned} \left[f_{min}|j\right] &= \min_{i=[1,2,3,4]}(\{\left[f(\boldsymbol{ICR}_{i(bor)})|i\right]\}) \\ \boldsymbol{ICR}_{bor} &= \boldsymbol{ICR}_{j(bor)}. \end{aligned} \tag{7}$$

Using (7), we find the $\boldsymbol{ICR}_{bor}$ corresponding to the shortest complementary route (colored magenta in Fig. 4, lower) among the infinite points lying on the square shaped border.

3.2. One Sample Period ICR Border Switching

To maintain unique ICR existence (in accordance with the steer joint configuration) at all times, the ICR must switch border in one sample period. To do this, we must comply with the restrictions on the value of R_∞, dependent on the maximum change in steering angle that can be done in one sample time ($\delta\beta_{max}$ in Fig. 5) which is in turn related to the robot hardware (actuator) constraints. On the other hand, R_∞ is limited to certain range of upper values, since the optimization algorithm run by the motion discontinuity-robust controller will have limited freedom in the feasible region at ICR points far from the robot center moving at high ICR velocity $\dot{\boldsymbol{ICR}}_{ref}$ (refer to Fig. 6). In such cases, the feasible region for $\boldsymbol{ICR}_{next}$ will be confined to move towards the ICR space borders, a favorable behavior for the complementary route motion, but not for the normal (most usual) operating conditions. To avoid such inconvenience, we introduce the extended version $R_{\infty(e)}$ which is used only for complementary (not for direct) route execution.

Definition 5. *$R_{\infty(e)} = \{x \in \mathbb{R}^+ : R_\infty < x < \infty\}$, is an extended value of the upper limit allowed for the $\boldsymbol{ICR}$ components, so that border switching can be performed in one sample period.*

Definition 6. *Border switching is a joint space operation where the steer joints are reconfigured from $\mathcal{M}(\boldsymbol{ICR}_{bor})$ to $\mathcal{M}(-\boldsymbol{ICR}_{bor})$, where $\mathcal{M} : \boldsymbol{ICR} \mapsto \boldsymbol{\beta}$ is a coordinate mapping from the ICR space to the steer joint space.*

From the geometry of the green triangle in Fig. 5, $R_{\infty(e)}$ can be evaluated in the worse case scenario as:

$$\begin{aligned} R_{\infty(e)} &\geq b * \tan\left(\frac{\pi - \delta\beta_{max}}{2}\right) + o_{band} \\ \delta\beta_{max} &= \min(\{\dot{\beta}_{max} * t_s, \ddot{\beta}_{max} * t_s^2\}), \end{aligned} \tag{8}$$

where $\dot{\beta}_{max}$ and $\ddot{\beta}_{max}$ denote the maximum steering velocity and acceleration characteristics of the robot and

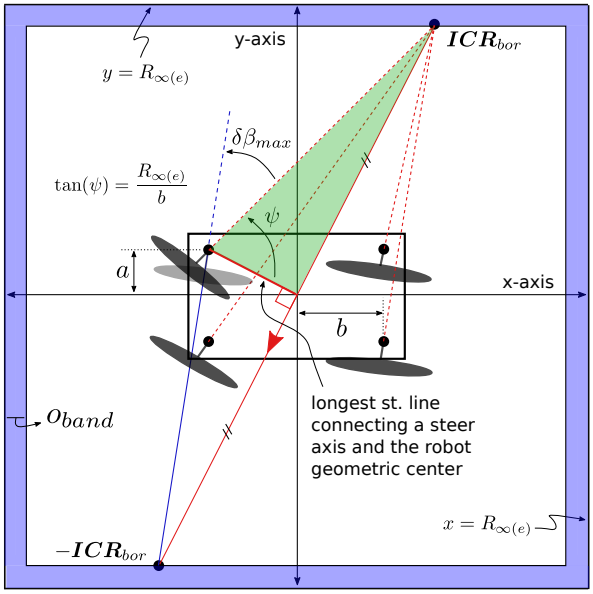

Figure 5: Evaluating the minimum value of $R_{\infty(e)}$ in the worse case scenario (biggest change in steer joint $\delta\beta$) to be able to perform border switching in one sample period. The mobile base schematic is magnified w.r.t. the ICR border lines for illustrative purposes.

o_{band} is a positive (hand tuned) scalar offset. For further clarification, R_∞ is usually selected in the range of $10 \sim 20$ m, whereas $R_{\infty(e)}$ as computed using (8) along with the robot and controller parameters provided in Table 1, evaluates to ≈ 31 m at sampling time $t_s = 0.025$ s (or ≈ 192 m at $t_s = 0.01$ s).

3.3. Route Planning

To implement the complementary ICR route, we first evaluate its time efficiency with respect to that of the direct route, using a quantitative metric (detailed in the next section). Once the route is verified as being more time efficient, the complementary enable flag F_{ce} is set to 1 allowing its execution. Such route requires the following set of 4 consecutive steps $S = \{S_1, \ldots, S_4\}$ with S_1: moving to the optimum border point $\boldsymbol{ICR}_{bor}$ at R_∞, S_2: extending the border lines to $R_{\infty(e)}$, S_3: switching border in one sample period, S_4: retracting the border lines back to R_∞. At this step, the complementary route algorithm is disabled ($F_{ce} = 0$) and the direct route is then utilized finally to the desired $\boldsymbol{ICR}^*$. The elements of the Boolean set S are initialized to 1 at the beginning of the complementary route execution. These steps are implemented by manipulating the modified ICR desired

point $\boldsymbol{ICR}^*_{mod}$, and the ICR controller gain λ as follows:

$$\begin{aligned}
&\forall S_1 \wedge F_{ce} \wedge (|\boldsymbol{ICR}_{err(c)}| > c_{th}) : \\
&\qquad \begin{cases} \boldsymbol{ICR}^*_{mod} = \boldsymbol{ICR}_{bor}, \\ \lambda = \lambda_n, \end{cases} \\
&Otherwise : S_1 = 0,
\end{aligned} \tag{9}$$

$$\begin{aligned}
&\forall S_1 \oplus S_2 \wedge F_{ce} \wedge (|\boldsymbol{ICR}_{err(e)}| > e_{th}) : \\
&\qquad \begin{cases} \boldsymbol{ICR}^*_{mod} = \boldsymbol{ICR}_{bor} * r_e, \\ \lambda = \lambda_e, \end{cases} \\
&Otherwise : S_2 = 0,
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\forall S_1 \oplus S_2 \oplus S_3 \wedge F_{ce} \wedge (\boldsymbol{ICR}_{curr} \neq \boldsymbol{ICR}'_{bor}) : \\
&\qquad \begin{cases} \boldsymbol{ICR}^*_{mod} = \boldsymbol{ICR}'_{bor} * r_e, \end{cases} \\
&Otherwise : S_3 = 0,
\end{aligned} \tag{11}$$

$$\begin{aligned}
&\forall S_1 \oplus S_2 \oplus S_3 \oplus S_4 \wedge F_{ce} \wedge (|\boldsymbol{ICR}'_{err(c)}| > c_{th}) : \\
&\qquad \begin{cases} \boldsymbol{ICR}^*_{mod} = \boldsymbol{ICR}'_{bor}, \\ \lambda = \lambda_n, \end{cases} \\
&Otherwise : \\
&\qquad \begin{cases} \boldsymbol{ICR}^*_{mod} = \boldsymbol{ICR}^*, \\ S_4 = 0, \\ F_{ce} = 0. \end{cases}
\end{aligned} \tag{12}$$

In the above, $r_e = R_{\infty(e)}/R_\infty$ is the extension ratio, $\boldsymbol{ICR}_{err(c)} = \boldsymbol{ICR}_{bor} - \boldsymbol{ICR}_{curr}$, and $\boldsymbol{ICR}_{err(e)} = \boldsymbol{ICR}_{bor} * r_e - \boldsymbol{ICR}_{curr}$ are the complementary and extended error vectors respectively, e_{th}, and c_{th} are the corresponding positive scalar thresholds. Two positive scalar gains λ_e, and λ_n are used for the extended and normal ICR motion controllers respectively. Finally, $\boldsymbol{ICR}'_{err(c)} = \boldsymbol{ICR}'_{bor} - \boldsymbol{ICR}_{curr}$. Equations (9 - 12) represent the semantic core (planning) of the complementary route algorithm, with the high level desired signal $\boldsymbol{ICR}^*$ modified to $\boldsymbol{ICR}^*_{mod}$ based on the choice of ICR route to follow. The modified desired signal will be used by the ICR controller [26] (refer to Fig. 3):

$$\begin{aligned}
\dot{\boldsymbol{ICR}}_{ref} &= \dot{\boldsymbol{ICR}}^* + \lambda \boldsymbol{ICR}_{err}, \\
\boldsymbol{ICR}_{err} &= \boldsymbol{ICR}^*_{mod} - \boldsymbol{ICR}_{curr}.
\end{aligned} \tag{13}$$

3.4. Automatic Joint Space Border Switching

In (11), we modify the desired ICR so that it is on the complementary border, but the actual border switching is done in the steering joint space. We switch by first checking if the $\boldsymbol{ICR}^*_{mod}$ and the $\boldsymbol{ICR}_{curr}$ satisfy:

$$\begin{aligned}
&(|X^*_{mod} \pm R_{\infty(e)}| \wedge |X_{curr} \mp R_{\infty(e)}|) \vee \\
&\qquad (|Y^*_{mod} \pm R_{\infty(e)}| \wedge |Y_{curr} \mp R_{\infty(e)}|) \leq o_{band}.
\end{aligned} \tag{14}$$

This simply indicates that at least one component (X or Y) of both the desired and current ICR points is in the opposite border band. Then, the complementary steer joint variables corresponding to $\boldsymbol{ICR}'_{bor} = -\boldsymbol{ICR}_{bor}$ are evaluated using the SWMR geometric model:

$$\beta'_{i(bor)} = \arctan 2\Big(- Y_{bor} - h_{yi}, -X_{bor} - h_{xi}\Big) - \frac{\pi}{2}. \tag{15}$$

However, we are interested in the minimum change in steering angle to achieve the border switch, which can be obtained by first computing the required steering change $\delta\beta_{i(bs)}$ using (15) and the current steer joint coordinates:

$$\delta\beta_{i(bs)} = PosB(\beta'_{i(bor)}) - PosB(\beta_{i(curr)}), \tag{16}$$

$$PosB(\delta\beta_{i(bs)}) = \begin{cases} \delta\beta_{i(bs)} + 2\pi, & \forall \delta\beta_{i(bs)} < 0 \\ \delta\beta_{i(bs)} - 2\pi, & \forall \delta\beta_{i(bs)} \geq 2\pi \end{cases},$$

where $PosB(*)$ is a function that guarantees that the output angles are positive and bounded in the range $[0, 2\pi[$. The minimum change is then obtained from (16) as follows:

$$\delta\beta_{i(bs)(min)} = \begin{cases} \delta\beta_{i(bs)} - 2\pi, & \text{if } \delta\beta_{i(bs)} > 3\pi/2 \\ \delta\beta_{i(bs)} - \pi, & \text{if } \delta\beta_{i(bs)} > \pi/2 \\ \delta\beta_{i(bs)} + 2\pi, & \text{if } \delta\beta_{i(bs)} < -3\pi/2 \\ \delta\beta_{i(bs)} + \pi, & \text{if } \delta\beta_{i(bs)} < -\pi/2 \end{cases}. \tag{17}$$

Finally, the $4D$ vector $\delta\boldsymbol{\beta}_{(bs)(min)}$ is added to the current steer position vector to conclude the border switching phase:

$$\boldsymbol{\beta}_{ref} = \boldsymbol{\beta}_{curr} + \delta\boldsymbol{\beta}_{(bs)(min)}. \tag{18}$$

4. Direct OR Complementary

In order to select the best route, here defined as the one that is more time efficient, we need an estimate of the total time required by each. To do this we use the Jacobian relating the steering velocity to that of the ICR point:

$$\dot{\boldsymbol{ICR}}_{max} = \boldsymbol{J}^+_{(\rho)}(\boldsymbol{ICR}_{curr}) * \dot{\boldsymbol{\beta}}_{max}, \tag{19}$$

$$\boldsymbol{J}(\boldsymbol{ICR}) = \begin{bmatrix} \boldsymbol{J}_1 & \dots & \boldsymbol{J}_N \end{bmatrix}^T, \tag{20}$$

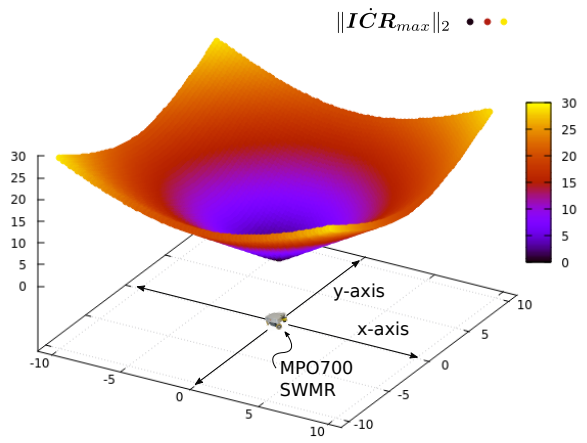

Figure 6: Variation of the maximum ICR velocity with robot's ICR position, featuring the Neobotix-MPO700 industrial mobile robot approximately to scale, with $R_\infty = 10m$.

$$\boldsymbol{J}_i = \begin{bmatrix} \dfrac{-\widetilde{Y}}{\widetilde{X}^2 + \widetilde{Y}^2 + \delta_3} & \dfrac{\widetilde{X}}{\widetilde{X}^2 + \widetilde{Y}^2 + \delta_3} \end{bmatrix},$$

where $\widetilde{Y} = (Y - h_{yi})$, $\widetilde{X} = (X - h_{xi})$, with $\delta_3 \in \mathbb{R}^+$ a damping factor. We obtain the Jacobian matrix in (19) by substituting by $\boldsymbol{ICR}_{curr}$ in the general relation (20). Fig. 6 shows the evolution of $\|\dot{\boldsymbol{ICR}}_{max}\|_2$ over the entire R_∞ bounded ICR space, computed using (19) with the parameters of the Neobotix MPO-700 mobile robot provided in Table 1. As depicted, among equal length direct and complementary routes, the latter will be more time efficient. Both routes are then divided into an equal, even number of samples $2 * n_p$, $n_p \in \mathbb{Z}^+$, and at each, $\dot{\boldsymbol{ICR}}_{max}$ is computed using (19) and used to estimate the time required by each route as follows:

$$\begin{aligned} T_d &= \sum_{k=1}^{2*n_p} \frac{S_d}{\|\boldsymbol{J}^+_{(\rho)}(\boldsymbol{ICR}_{k(d)}) * \dot{\boldsymbol{\beta}}_{max}\|}, \\ T_c &= \sum_{k=1}^{n_p} \frac{S_c}{\|\boldsymbol{J}^+_{(\rho)}(\boldsymbol{ICR}_{k(c)}) * \dot{\boldsymbol{\beta}}_{max}\|} \\ &\quad + \frac{S'_c}{\|\boldsymbol{J}^+_{(\rho)}(\boldsymbol{ICR}'_{k(c)}) * \dot{\boldsymbol{\beta}}_{max}\|}, \end{aligned} \tag{21}$$

$$\begin{aligned} \boldsymbol{ICR}_{k(d)} &= \boldsymbol{ICR}_{curr} + \frac{k}{2 * n_p} * \boldsymbol{ICR}_{err}, \\ \boldsymbol{ICR}_{k(c)} &= \boldsymbol{ICR}_{curr} + \frac{k}{n_p} * \boldsymbol{ICR}_{err(c)}, \\ \boldsymbol{ICR}'_{k(c)} &= \boldsymbol{ICR}'_{bor} + \frac{k}{n_p} * (\boldsymbol{ICR}^* - \boldsymbol{ICR}'_{bor}), \end{aligned}$$

with T_d, and T_c are the time duration estimates, $S_d = 0.5 * \|\boldsymbol{ICR}_{err}\|/n_p$, $S_c = \|\boldsymbol{ICR}_{err(c)}\|/n_p$, and $S'_c = \|\boldsymbol{ICR}^* - \boldsymbol{ICR}'_{bor}\|/n_p$ the distance increments for the direct, complementary part#1, and part#2 routes respectively (refer to Fig. 4, upper). It is worth noting that the estimates provided by (21) are not exact and are only useful for comparison purposes, since the error driven ICR controller will not provide maximum ICR velocity during the whole route.

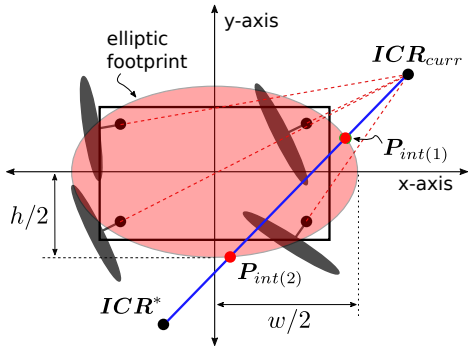

Figure 7: The intersection of the direct ICR route with the elliptic footprint of the mobile robot should favor the use of the complementary route.

Remark 3. *The complementary route described in this work is simply a straight line route passing by the 2D ICR space borders, parameterized by R_∞. We assume it is time optimal with respect to the straight line direct route even if it is longer in length, since it moves the ICR away from the robot geometric center. As such, much higher ICR velocities are achievable and longer ICR distances can be covered in shorter time with less steering, as depicted in Fig. 6. Although higher order routes (e.g., splines) could be used, here we focus on a simpler first order (straight line) route.*

In some cases, even if the complementary ICR route is found more time consuming than its direct counterpart, it is preferable to use it. In particular when the direct route will pass by/near singular configurations, triggering the singularity avoidance algorithms [26], that will consume time. Moreover, in the neighbourhood of singular configurations or at the robot geometric center, usually the feasible region for the ICR_{next} is very small, and it might happen (although rarely if controller parameters are tuned well) that no feasible solution is found. In practice, such situation occurs when the direct route passes by the robot footprint. To affect the decision in such cases, we construct an elliptic footprint, light red in Fig. 7, with geometric center coincident with the robot frame, and semiaxes lengths w, and h (values chosen arbitrarily), formulated as:

$$x^2/w^2 + y^2/h^2 = 1. \tag{22}$$

The intersection points $\boldsymbol{P}_{int(1,2)} = \begin{bmatrix} x_{int(1,2)} & y_{int(1,2)} \end{bmatrix}^T$ of such footprint with the direct route formulated as $y = m_d * x + b_d$, are:

$$x_{int(1,2)} = \frac{-w^2 m_d b_d \pm wh\sqrt{K_1 - b_d^2}}{K_1},$$

$$y_{int(1,2)} = \frac{h^2 b_d \pm whm_d\sqrt{K_1 - b_d^2}}{K_1},$$

where $K_1 = w^2 m_d^2 + h^2$. Then, we use the discriminant $D = K_1 - b_d^2$ to determine whether the direct route intersects the designed footprint or not ($D \geq 0$ if intersection occurs). Afterwards, we add a biasing factor T_b that is activated only if both the current and desired ICR points are outside the ellipse. Such biasing condition is checked by substituting by $\boldsymbol{ICR}_{curr}$, and $\boldsymbol{ICR}^*$ in (22) and formulated as:

$$C_b = \begin{cases} 1 & \text{if } \left(\frac{X_{curr}^2}{w^2} + \frac{Y_{curr}^2}{h^2} \geq 1\right) \wedge \left(\frac{X^{*2}}{w^2} + \frac{Y^{*2}}{h^2} \geq 1\right) \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the total time duration comparison is done using:

$$F_{ce} = \begin{cases} 1 & \text{if } (T_c \leq T_d + l(D)C_b T_b) \wedge \dot{\boldsymbol{ICR}}^* \neq 0 \\ 0 & \text{otherwise,} \end{cases} \quad (23)$$

where $l(*)$ is a logic function:

$$l(D) = \begin{cases} 1 & \forall D > 0 \\ 0 & \forall D \leq 0. \end{cases}$$

In (23), condition $\dot{\boldsymbol{ICR}}^* \neq 0$ ensures that we evaluate both routes and update the flag F_{ce} automatically only if a change in the desired ICR motion is detected. Flag F_{ce} is the output of the "direct or complementary" decision making algorithm (red colored block in Fig. 3).

5. Experiments

In this section, we describe/perform two experiments. The first (simulation) will show how the proposed complementary route controller can enhance the performance of the SWMR in response to diverse discontinuous velocity commands. To this end, we introduce the *command fulfillment index* as performance (quantitative) evaluation metric. The second experiment will feature a common application, where the mobile base must avoid obstacles while maintaining a particular heading angle so that a target is always in the vision cone. We show that the general performance is greatly enhanced with the proposed controller. Both the vision based controller and the obstacle avoidance algorithm are out of the scope of this work and consequently not detailed here, and the reader is referred to [34] for more information on these. The QP optimization has been implemented using the QuadProg++ library [35].

5.1. Command Fulfillment Index (CFI)

To compare the output velocity commands $\dot{\boldsymbol{\xi}}_{ref}$ of the two controllers (employing direct or complementary routes) we introduce the CFI (dimensionless quantity) metric. This is defined as:

$$CFI = 1 - \frac{\|\dot{\boldsymbol{\xi}}^* - \dot{\boldsymbol{\xi}}_{ref}\|_2}{2\|\dot{\boldsymbol{\xi}}_{max}\|_2}, \quad (24)$$

where $\dot{\boldsymbol{\xi}}_{max} = \begin{bmatrix} \dot{x}_{max} & \dot{y}_{max} & \dot{\theta}_{max} \end{bmatrix}^T$ is the maximum robot velocity (characteristic of the robot hardware). Note that the CFI does not depend on the estimated (from joint space measurements) robot velocity, since the low level (joint dynamics) control is out of the scope of this work. Instead, we aim at evaluating how near the "feasible" controller output velocity command $\dot{\boldsymbol{\xi}}_{ref}$ is to the desired one $\dot{\boldsymbol{\xi}}^*$. For SWMR, the value $0 < CFI < 1$ will be closer to 1 as the quality of the controller increases. Also note that the same values of both the desired $\dot{\boldsymbol{\xi}}^*$ and maximum $\dot{\boldsymbol{\xi}}_{max}$ velocities are used in comparing both controllers, so only $\dot{\boldsymbol{\xi}}_{ref}$ depends on the controller.

It is worth noting that the best CFI values are expected for a fully omnidirectional robot. However, it will not maintain $CFI = 1$ at all times being subject to discontinuous velocity commands: a sudden drop in the CFI value will happen once a new desired ICR point is demanded (see Fig. 10c, and Fig. 9c). It is generally expected that a FOMR will converge faster than a SWMR. Note that the CFI formulation in (24) depends on the response time of both the steering and driving joints. The FOMR structure has an advantage since it does not need steering. On the other hand, a SWMR structure has an advantage in the driving joint thanks to its conventional wheels that are generally expected to respond faster (to reach a discontinuously commanded drive speed) than omnidirectional (or Swedish) ones, due to their limited velocity/acceleration capabilities [36], and frequent slippage [37]. This argument implies that for applications subject to frequent discontinuous velocity commands, the SWMR structure may be equivalent or even out-perform the FOMR one.

5.2. Discontinuous Velocity Commands Simulation

To evaluate the proposed controller against discontinuous robot velocity commands in different situations, we should be able to set those commands in the ICR space. To this end, we use the kinematic relation between the $3D$ Cartesian velocity space and the $2D$ ICR space (refer to Fig. 2):

$$\boldsymbol{v}^* = \begin{bmatrix} \dot{x}^* \\ \dot{y}^* \end{bmatrix} = \begin{bmatrix} Y^*\dot{\theta}^* \\ -X^*\dot{\theta}^* \end{bmatrix}, \tag{25}$$

where X^* and Y^* are the components of each of the nine desired ICR points shown in Fig. 8, and a constant rotational velocity $\dot{\theta}^* = 0.05$ rad s^{-1} is applied throughout the simulation. Each of the 9 velocity commands is applied for a duration of 5 seconds. The $3D$ Cartesian velocity vector $\dot{\boldsymbol{\xi}}^* = \begin{bmatrix} \boldsymbol{v}^{*T} & \dot{\theta}^* \end{bmatrix}^T$ is then sent to the controller architecture shown in Fig. 3. The desired ICR points used (shown in Fig. 8), are selected arbitrarily to excite the controller in different situations, to show both the strengths and weaknesses. The controller parameters used throughout the simulations are shown in Table 1, with $R_\infty < R_{\infty(e)} < \infty$ values chosen as suggested in 3.2.

Simulation results for the complementary route controller and for the conventional one are shown in Figures 10 and 9 respectively. The first command $\boldsymbol{ICR}^* = \begin{bmatrix} 0 & R_\infty \end{bmatrix}^T$, applied during $t = [0, 5[s$ (t is the simulation time), shown in Fig. 8 requires no steering as it conforms with the initial steer joint configuration. Among numerous reasons, the motivation behind the first command is to show that even if no steering is required (the same case as in FOMR), the CFI is not equal to 1 (Fig. 10c, 9c) since it depends on how fast the required driving velocity can be reached, and in that spirit it is expected that the SWMR might out-perform the FOMR structure at some motion requirements, due to the limited performance capabilities [36, 37] of such non conventional wheel types.

Table 1: Robot and Controller parameters used in the experiments.

a	$0.19m$	b	$0.24m$
r_w	$0.09m$	δ	0.001
δ_1	1^{-9}	$\dot{V}_{max}$	$10m/s$
λ_n	7.7	R_{zone}	$0.015m$
$\ddot{\beta}_{max}$	$25rad./s^2$	t_s	$25ms$
w	$0.4m$	h	$0.3m$
$\dot{x}_{max}$	$0.5m/s$	$\dot{y}_{max}$	$0.5m/s$
$\ddot{x}_{max}$	$0.5m/s^2$	$\ddot{y}_{max}$	$0.5m/s^2$
$R_{\infty(e)}$	$33m$	e_{th}	$0.1R_\infty$
δ_3	0.01	n_p	50
d	$0.045m$	R_∞	$10m$
β_{th}	$0.005rad.$	$\dot{\beta}_{max}$	$2rad./s$
K_p	2	λ_e	30
$\dot{\theta}_{max}$	$0.5rad./s$	$\ddot{\theta}_{max}$	$0.5rad./s^2$
c_{th}	$0.05R_{\infty(e)}$	T_b	$3s$

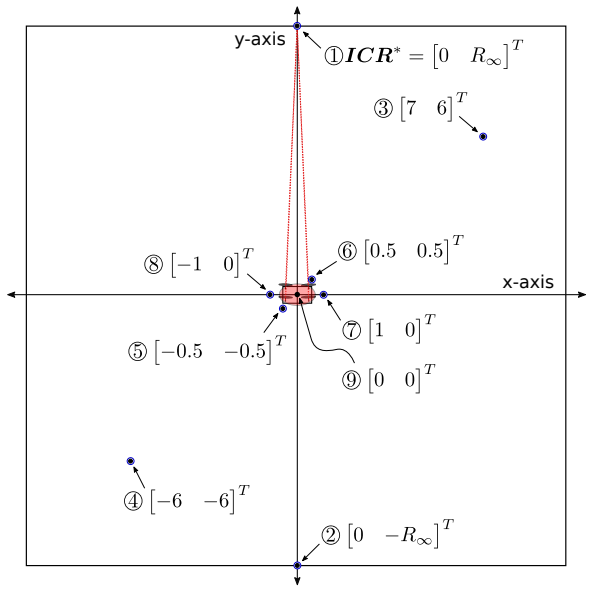

Figure 8: Desired (discontinuous) ICR points used in simulation. The relative dimensions of the mobile base schematic as well as the points are drawn at approximately $1:1$ scale for clarification. The initial steer joint configuration corresponds to the equivalent ICR point of the first velocity command vector.

The second command $\boldsymbol{ICR}^* = \begin{bmatrix} 0 & -R_\infty \end{bmatrix}^T$, applied during $t = [5, 10[$ s, requires ICR motion from one extreme of the ICR space to the opposite in the conventional (direct) controller. The ability to perform such discontinuous velocity command in short time is the core motivation behind this paper, since it is very common in heading control applications. In such applications, a small heading variation will result in a sign change for the rotational velocity, that in turn causes big jumps in the corresponding ICR. The X component of the desired and reference ICR points, for the complementary and direct routes is shown in Fig. 10a, 9a respectively. Similarly, the Y component is shown in Figures 10b and 9b. Recall that the reference ICR point is a feasible signal that can be performed by taking into account the robot maximum performance limits. By comparing in Fig. 10b, 9b the Y_{ref} output by both con-

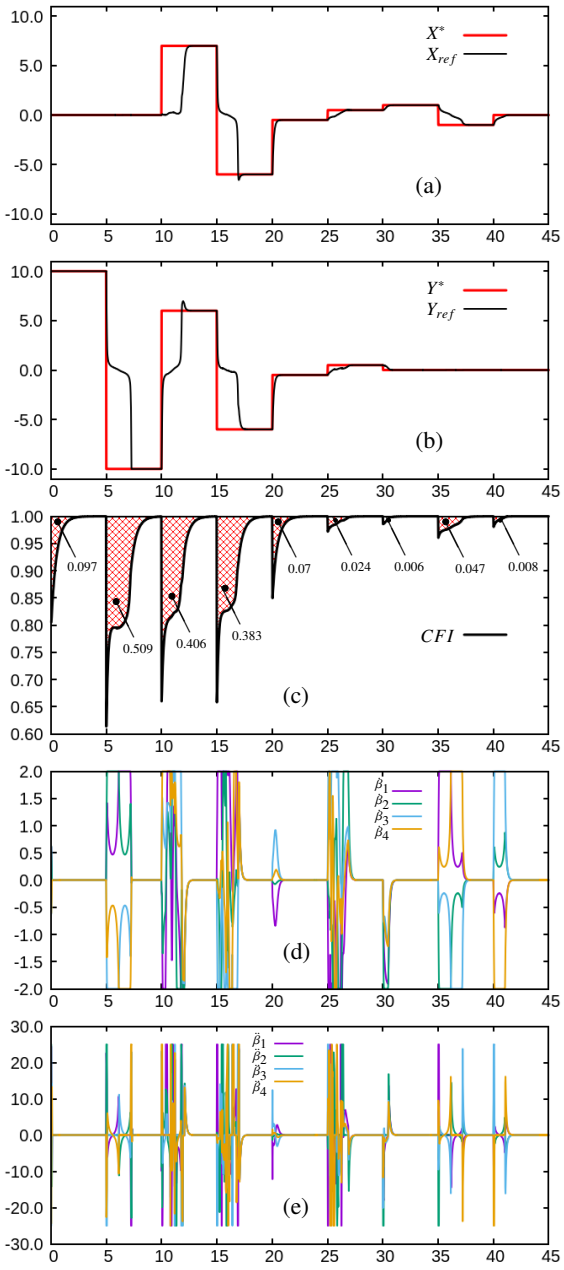

Figure 9: Simulation results employing the direct route (conventional) controller with (a), (b) the x, and y components respectively of the desired and reference (output of the controller) ICR points (in *meters*), (c) the command fulfillment index (*dimensionless*), (d), (e) the steering velocity (in *rad/s*) and acceleration (in *rad/s*^{2}) respectively. The values in (c) indicate the area above graph (proportional to the error in robot velocity) at each discontinuous velocity command. In all plots, the abscissa represents the simulation time instant (in *seconds*).

trollers, we clearly see the enhanced performance using the complementary route, where converging to the

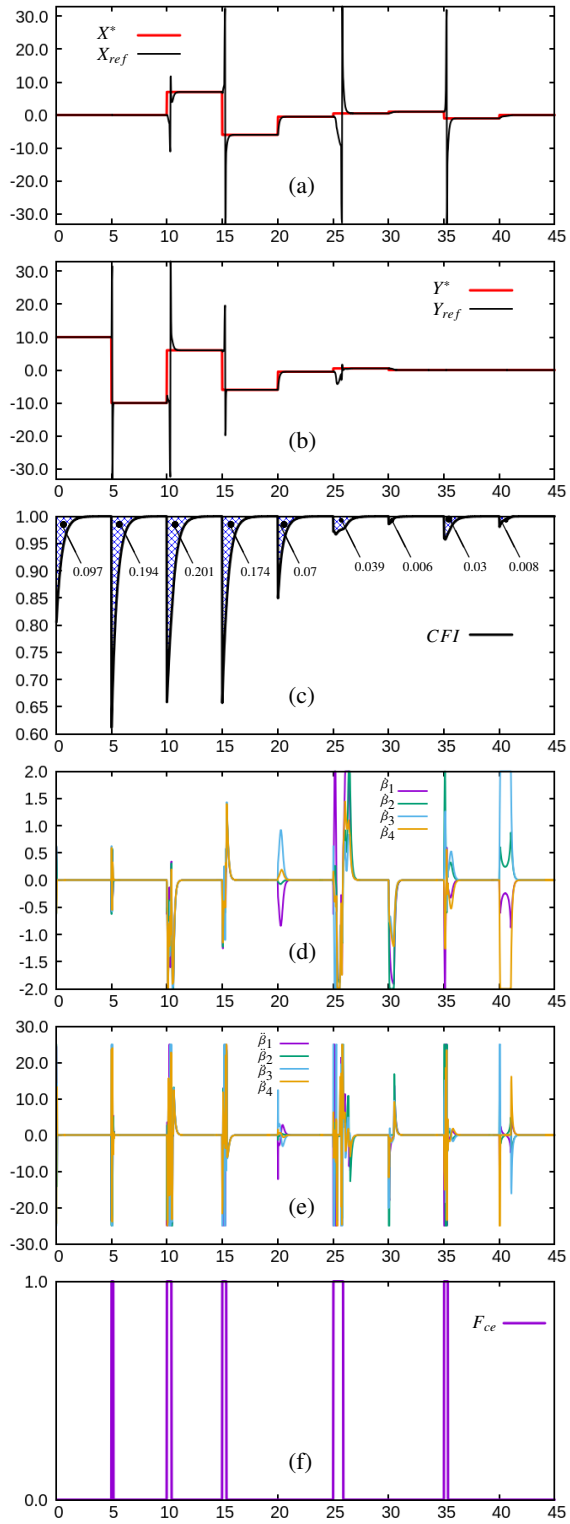

Figure 10: Simulation results of the complementary route controller with (a) to (e) as described in Figure 9 and (f) the time instants at which the complementary route algorithm is active.

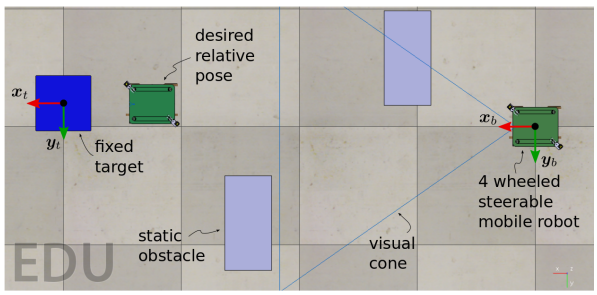

Figure 11: V-REP scene of the sensor-based navigation scenario, featuring the Neobotix-MPO700 mobile robot navigating to a target while performing obstacle avoidance. The simulated visual cone applies a constraint on the heading of the robot so that the target is always seen by the robot, as is common in vision-based navigation. Here we apply such constraint to show the utility of the proposed controller. $\mathcal{F}_t = (o_t, \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$ denotes the target object frame.

Y^* took only a fraction of a second ($0.2s$), whereas in Fig. 9b, the direct route took much longer to converge ($2.27s$).

The third $\boldsymbol{ICR}^* = \begin{bmatrix}7 & 6\end{bmatrix}^T$, and fourth $\boldsymbol{ICR}^* = \begin{bmatrix}-6 & -6\end{bmatrix}^T$ commands applied during $t = [10, 15[$ s, $t = [15, 20[$ s respectively, correspond to the example given earlier in Fig. 4 to enhance paper cohesion. These commands confirm the superior performance of the complementary route (refer to Figures 9 and 10) in terms of the convergence time of both the X and Y components of $\boldsymbol{ICR}_{ref}$. The 4 commands just described feature large discontinuous jumps in the desired ICR point, in which cases the performance of the complementary route clearly surpasses that of the direct route in terms of lowering both the time of convergence and the total Cartesian velocity error. The latter is evident from the smaller area above curves of the CFI plots in Fig. 10c as compared to Fig. 9c.

The fifth command $\boldsymbol{ICR}^* = \begin{bmatrix}-0.5 & -0.5\end{bmatrix}^T$, applied during $t = [20, 25[$ s, lies in the same quadrant as the fourth. Consequently, there is no advantage in using the complementary route; this is automatically detected using (21), and the direct route is used. The sixth command $\boldsymbol{ICR}^* = \begin{bmatrix}0.5 & 0.5\end{bmatrix}^T$, applied during $t = [25, 30[$ s, aims at testing the elliptic footprint avoiding algorithm. By comparing Figures 9, and 10 we see that the $\boldsymbol{ICR}_{ref}$ converging time for both is almost the same, although the CFI is lower in case of the direct route. This can be adjusted by lowering the value of the penalty factor T_b in (23). Instead, here, its value is intentionally left high enough ($3s$) to bias the decision in favor of the complementary route for two reasons. First: in practice, when the desired velocity is near the current one (especially near the robot geometric center), a small increase in the error does not tragically affect the application in hand. Second: avoiding ICR motion in the robot footprint region lowers the chances of triggering singularity avoidance algorithms, minimizing the steering as shown by comparing Figures 10d, 10e with Figures 9d, 9e, which is generally favorable.

Figure 12: Snapshots of the real sensor-based navigation experiment employing the complementary route ICR controller.

The seventh $\boldsymbol{ICR}^* = \begin{bmatrix}1 & 0\end{bmatrix}^T$, and eighth $\boldsymbol{ICR}^* = \begin{bmatrix}-1 & 0\end{bmatrix}^T$ commands applied during $t = [30, 35[$ s, $t = [35, 40[$ s respectively, show that the complementary route can be as efficient for small ICR jumps near the geometric center as it is for the far points in the ICR space. In moving from the seventh to the eighth, the conversion time duration is approximately $2.5s$ and $1s$ for the direct and complementary routes respectively. The corresponding areas above the CFI curve are $0.047s$, and $0.03s$ in the same order. Finally a pure rotation velocity command $\boldsymbol{ICR}^* = \begin{bmatrix}0 & 0\end{bmatrix}^T$ is applied during $t = [40, 45[$ s. In Fig. 10f, the periods in which the complementary route algorithm is active are shown. During the rest of the simulation period, the direct route is active, better explaining the proposed algorithm as a correcting module to enhance the overall performance. Aside better CFI values and shorter convergence time of $\boldsymbol{ICR}_{ref}$ to the $\boldsymbol{ICR}^*$, the proposed controller reduces the required steering velocity and acceleration, as clearly shown in Figures 9d, 9e, 10d and 10e,.

5.3. *Sensor-based navigation*

The performance evaluation detailed in previous subsection (Fig. 10, and Fig. 9) is a mere quantitative comparison. It shows that the complementary route controller outperforms the direct route one. Yet, in practice, the direct route controller may fail in several applications or provide unsatisfactory behavior at best. Here, we describe/perform one of these applications, where a mobile robot is required to move to a relative pose with respect to a visible target, while avoiding several static obstacles on its way. During this task, the robot

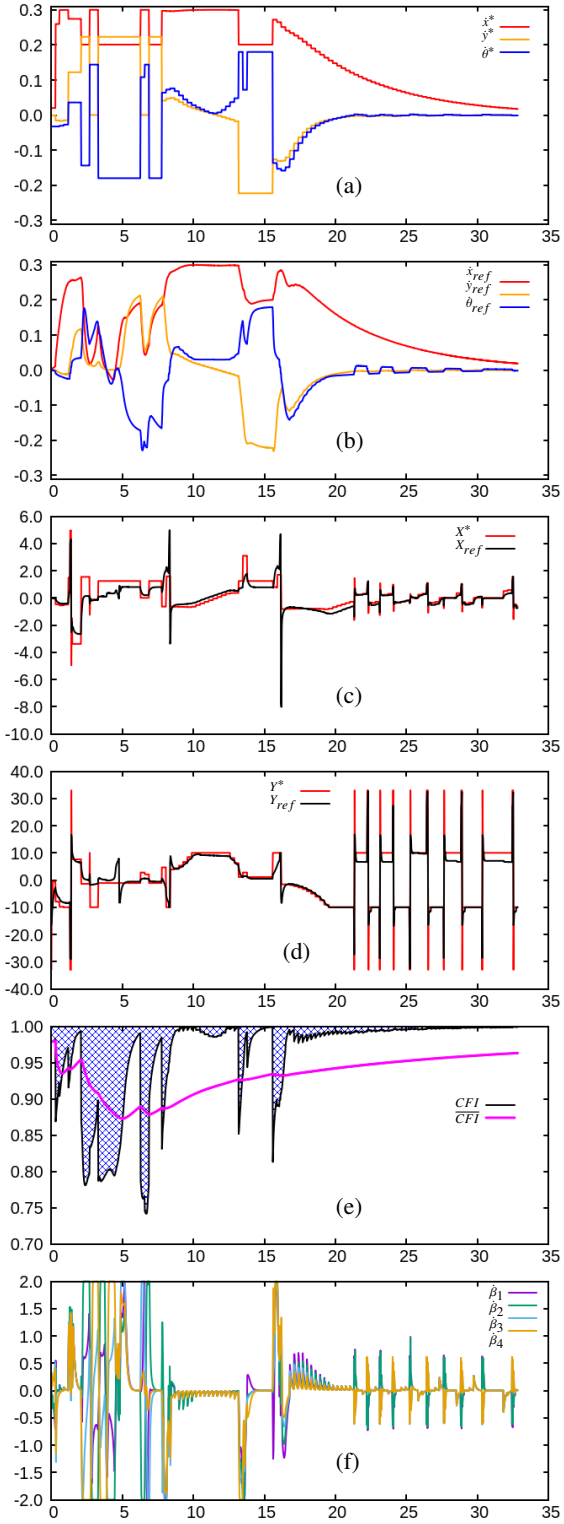

Figure 13: Simulation results of sensor-based navigation employing the complementary route controller.

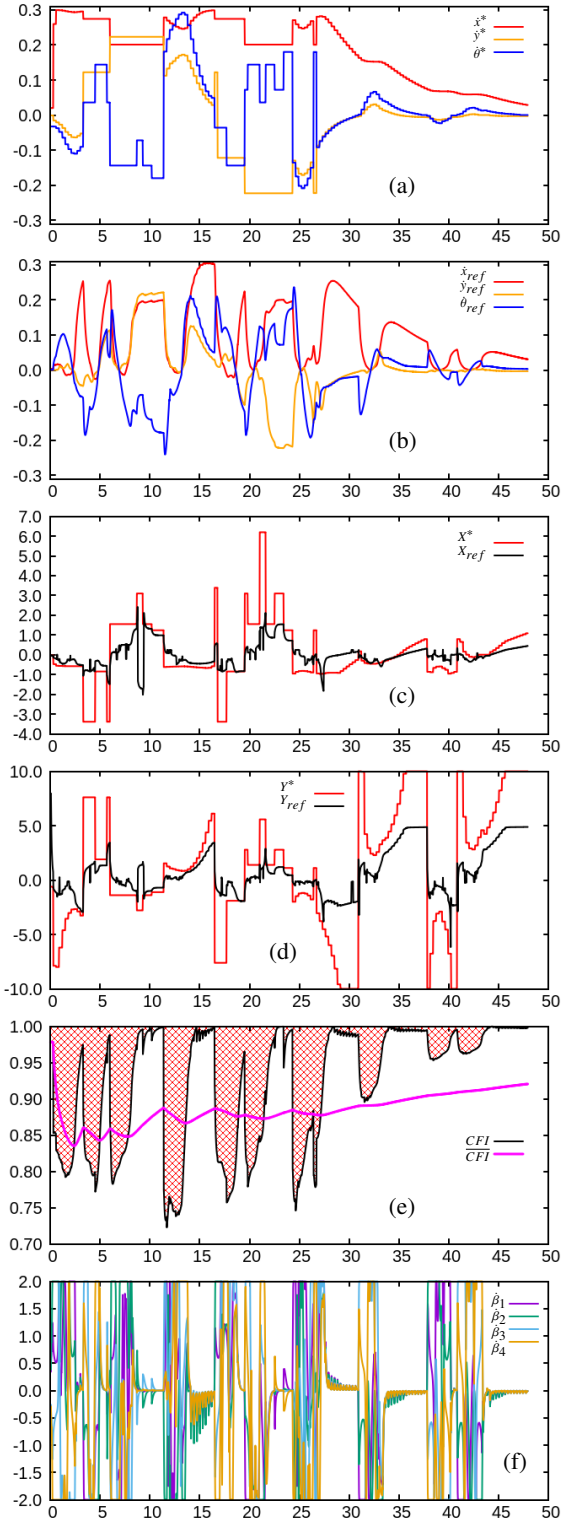

Figure 14: Simulation results of sensor-based navigation employing the direct route (conventional) controller.

has to maintain a certain heading angle with respect to the target to keep it within its field of view. A V-REP simulation of such application is shown in Fig. 11.

We have chosen a simple obstacle configuration for two reasons. First, the sensor-based navigation controller has been detailed and validated (in more complex scenarios) in [34], and is out of scope in this paper. Second, a simple scenario is preferable, to highlight the effect of the two controllers on the robot motion. In fact, the discontinuous $\dot{\boldsymbol{\xi}}^*$ output by the sensor-based navigation controller (the the "high level task controller" of Fig. 3) causes frequent sign changes of either component of the desired ICR. This phenomenon will make the direct route less satisfactory, independently from the obstacle configuration.

The discontinuous velocity commands $\dot{\boldsymbol{\xi}}^*$ are shown in Figures 13a and 14a for the complementary and direct route controllers, respectively. In Figures 13b and 14b, we show the output of the respective controllers. As seen, the complementary route approach is more responsive and converges more quickly to the desired signal. However, during the time period $t = [2, 5[$ s in Fig. 13b, the complementary route algorithm is less satisfactory in terms of following the desired robot velocity. Hence, during this period, the direct route is activated. The cause is visible in Figures 13c and 13d, that show the X and Y components of the $\boldsymbol{ICR}^*$ and $\boldsymbol{ICR}_{ref}$ signals, respectively. During this time period, $\boldsymbol{ICR}^*$ is near the robot geometric center (values of X^*, and Y^* closer to 0). Consequently, the direct route is more time efficient according to (21), and since the signal is highly discontinuous, the direct route controller output $\boldsymbol{ICR}_{ref}$, does not have enough time to converge to $\boldsymbol{ICR}^*$. It is worth noting that, whenever a new discontinuous velocity command is requested, all robot velocity components are affected until they converge to their corresponding desired values. This is because, in contrast with FOMR structured robots, the robot velocity components are coupled.

The X and Y components of $\boldsymbol{ICR}^*$ and $\boldsymbol{ICR}_{ref}$ signals for the direct route controller are shown in Figures 14c and 14d, respectively. By comparing these figures with their complementary route counterparts, the poor responsiveness of the first is evident, especially at pure heading control during the time period $t = [25, 48[$ s. The controller output $\boldsymbol{ICR}_{ref}$ takes longer to converge to $\boldsymbol{ICR}^*$. This is reflected on the longer simulation time required to reach the desired relative pose with respect to the target (48 instead of 33 seconds).

The command fulfillment index is shown in Fig. 13e, and 14e respectively in the usual order, along with the evolution of its moving average value $\overline{CFI}$ computed using:

$$\overline{CFI} = \sum_{0}^{N_s} \frac{CFI}{N_s},$$

where $N_s = t/t_s$ is the number of sampling instances (points) at time t. The CFI metric quantitatively shows the enhanced responsiveness discussed earlier for the complementary route controller. Recall that the area above the CFI curve (in hashed blue and red colors in the respective figures) is directly proportional to the robot velocity error.

In Figures 13f and 14f, we show the steering velocity for each controller. It is evident that the proposed algorithm is less demanding in terms of steering joint motion, apart from the issue encountered during the time period $t = [2, 5[$ s, where the direct route controller was activated.

Finally, this application scenario has been performed experimentally using the Neobotix-MPO700 industrial mobile robot equipped with laser scanners. The test using the direct route controller suffered severe vibrations due to the excessive steering required (refer to Fig. 14f) and is considered as a failure. On the other hand, the complementary route controller test was performed successfully, as shown in the video supplied with this paper[4], snapshots of which are provided in Fig. 12.

6. Conclusion

In this paper, a complementary route based ICR controller is introduced. Its performance against the conventional ICR controller is compared quantitatively using a novel evaluation metric, the *Command fulfillment index*. The two controllers are also compared in two case studies. The first features simple discontinuous velocity signals applied at low frequency for a detailed investigation of both controllers behavior. The second is a practical application (sensor-based navigation) in which command discontinuity occurs at much higher frequency. The complementary route ICR controller shows superior results in all the performed tests.

Acknowledgment

This work is supported by the French region Occitanie (project CoBot@LR) and by the PSA Robotics Department. It has also been supported by EPSRC under grant agreement EP/R02572X/1 (National Center for Nuclear Robotics).

[4] Also visible at: https://youtu.be/qsFAlTPs1yw

[1] L. Ferriere, B. Raucent, G. Campion, Design of omnimobile robot wheels, in: Proceedings of IEEE International Conference on Robotics and Automation, volume 4, 1996, pp. 3664–3670 vol.4.

[2] R. Holmberg, O. Khatib, Development and control of a holonomic mobile robot for mobile manipulation tasks, The International Journal of Robotics Research 19 (2000) 1066–1074.

[3] M. Wada, Virtual link model for redundantly actuated holonomic omnidirectional mobile robots, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., 2006, pp. 3201–3207.

[4] A. El-Shenawy, A. Wellenreuther, A. S. Baumgart, E. Badreddin, Comparing different holonomic mobile robots, in: 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 1584–1589.

[5] J. T. Costa, M. Yim, Designing for uniform mobility using holonomicity, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2448–2453. doi:10.1109/ICRA.2017.7989285.

[6] D. B. Reister, M. A. Unseren, Position and constraint force control of a vehicle with two or more steerable drive wheels, IEEE Transactions on Robotics and Automation 9 (1993) 723–731.

[7] B. Thuilot, B. Novel, A. Micaelli, Modeling and feedback control of mobile robots equipped with several steering wheels, IEEE Trans. on Robotics and Automation 12 (1996) 375–390.

[8] M. F. Selekwa, J. R. Nistler, Path tracking control of four wheel independently steered ground robotic vehicles, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 6355–6360. doi:10.1109/CDC.2011.6160677.

[9] C. Cariou, R. Lenain, B. Thuilot, M. Berducat, Automatic guidance of a four-wheel-steering mobile robot for accurate field operations, Journal of Field Robotics 26 (2009) 504–518.

[10] P. R. Giordano, M. Fuchs, A. Albu-Schaffer, G. Hirzinger, On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs, in: IEEE Int. Conf. on Robotics and Automation, 2009, pp. 4080–4087. doi:10.1109/ROBOT.2009.5152625.

[11] A. Dietrich, T. Wimbck, A. Albu-Schffer, G. Hirzinger, Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint, in: IEEE Int. Conf. on Robotics and Automation, 2011, pp. 6136–6142. doi:10.1109/ICRA.2011.5979549.

[12] C. P. Connette, C. Parlitz, M. Hagele, A. Verl, Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots, in: IEEE Int. Conf. on Robotics and Automation, 2009, pp. 4124–4130. doi:10.1109/ROBOT.2009.5152450.

[13] U. Schwesinger, C. Pradalier, R. Siegwart, A novel approach for steering wheel synchronization with velocity/acceleration limits and mechanical constraints, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2012, pp. 5360–5366. doi:10.1109/IROS.2012.6385644.

[14] C. Stoger, A. Muller, H. Gattringer, Kinematic analysis and singularity robust path control of a non-holonomic mobile platform with several steerable driving wheels, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015, pp. 4140–4145. doi:10.1109/IROS.2015.7353962.

[15] C. P. Connette, S. Hofmeister, A. Bubeck, M. Haegele, A. Verl, Model-predictive undercarriage control for a pseudo-omnidirectional, wheeled mobile robot, in: ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), 2010, pp. 1–6.

[16] C. P. Connette, A. Pott, M. Hagele, A. Verl, Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an icm representation in spherical coordinates, in: 2008 47th IEEE Conference on Decision and Control, 2008, pp. 4976–4983. doi:10.1109/CDC.2008.4738958.

[17] R. Oftadeh, R. Ghabcheloo, J. Mattila, A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013, pp. 4845–4851. doi:10.1109/IROS.2013.6697055.

[18] R. Oftadeh, M. M. Aref, R. Ghabcheloo, J. Mattila, Bounded-velocity motion control of four wheel steered mobile robots, in: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2013, pp. 255–260. doi:10.1109/AIM.2013.6584101.

[19] C. P. Connette, A. Pott, M. Hgele, A. Verl, Addressing input saturation and kinematic constraints of overactuated undercarriages by predictive potential fields, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 4775–4781. doi:10.1109/IROS.2010.5652685.

[20] C. Connette, M. Hgele, A. Verl, Singularity-free state-space representation for non-holonomic, omnidirectional undercarriages by means of coordinate switching, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2012, pp. 4959–4965. doi:10.1109/IROS.2012.6386131.

[21] S. Chamberland, E. Beaudry, L. Clavien, F. Kabanza, F. Michaud, M. Lauriay, Motion planning for an omnidirectional robot with steering constraints, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2010, pp. 4305–4310. doi:10.1109/IROS.2010.5648987.

[22] R. Oftadeh, R. Ghabcheloo, J. Mattila, Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels, in: IEEE Int. Conf. on Robotics and Automation (ICRA), 2014, pp. 2925–2931. doi:10.1109/ICRA.2014.6907280.

[23] R. Oftadeh, R. Ghabcheloo, J. Mattila, A time-optimal bounded velocity path-following controller for generic wheeled mobile robots, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 676–683. doi:10.1109/ICRA.2015.7139252.

[24] A. Betourne, G. Campion, Kinematic modelling of a class of omnidirectional mobile robots, in: IEEE Int. Conf. on Robotics and Automation, 1996, pp. 3631–3636. doi:10.1109/ROBOT.1996.509266.

[25] M. Sorour, A. Cherubini, R. Passama, P. Fraisse, Kinematic modeling and singularity treatment of steerable wheeled mobile robots with joint acceleration limits, in: IEEE Int. Conf. on Robotics and Automation, 2016, pp. 2110–2115. doi:10.1109/ICRA.2016.7487360.

[26] M. Sorour, A. Cherubini, P. Fraisse, R. Passama, Motion discontinuity-robust controller for steerable mobile robots, IEEE Robotics and Automation Letters 2 (2017) 452–459.

[27] C. Gruber, M. Hofbaur, Practically stabilizing motion control of mobile robots with steering wheels, in: 2014 IEEE Conference on Control Applications (CCA), 2014, pp. 1312–1317. doi:10.1109/CCA.2014.6981510.

[28] P. F. Muir, C. P. Neuman, Kinematic modeling of wheeled mobile robots, Journal of Robotic Systems 4 (1987) 281–340.

[29] G. Campion, G. Bastin, B. Novel, Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, IEEE Trans. on Robotics and Automation 12 (1996) 47–62.

[30] K. Low, Y. Leow, Kinematic modeling, mobility analysis and design of wheeled mobile robots, Ad. Robotics 19 (2005) 73–99.

[31] G. Campion, W. Chung, Springer Handbook of Robotics - Wheeled Robots, Springer Berlin Hei-

delberg, Berlin, Heidelberg, 2008, pp. 391–410. doi:10.1007/978-3-540-30301-5_18.

[32] Y. Nakamura, H. Hanafusa, Inverse kinematic solutions with singularity robustness for robot manipulator control, ASME Journal of Dyn. Sys., Measur., and Control 108 (1986) 163–171.

[33] C. W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods, IEEE Trans. on Sys., Man, and Cybernetics 16 (1986) 93–101.

[34] A. Khelloufi, A. Cherubini, N. Achour, R. Passama, Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints, in: 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2017.

[35] L. D. Gaspero, QuadProg++, 2018 (accessed April 22, 2017). Https://github.com/liuq/QuadProgpp.

[36] R. L. Williams, B. E. Carter, P. Gallina, G. Rosati, Dynamic model with slip for wheeled omnidirectional robots, IEEE Trans. on Robotics and Automation 18 (2002) 285–293.

[37] K. Nagatani, S. Tachibana, M. Sofne, Y. Tanaka, Improvement of odometry for omnidirectional vehicle using optical flow information, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2000, pp. 468–473. doi:10.1109/IROS.2000.894648.