2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

A Novel Time Optimal Path Following Controller with Bounded Velocities for Mobile Robots with Independently Steerable Wheels

Reza Oftadeh, Reza Ghabcheloo, Jouni Mattila

***Abstract*— Mobile robots with independently steerable wheels possess many high maneuverability features of omnidirectional robots while benefiting from better performance and capability of moving on rough terrains. However, motion control of such robots is a challenging task due to presence of singular configurations and unboundedly large steering velocities in the neighborhood of those singularities. Many proposed approaches rely on numerical solutions that keep the robot out of bulky regions around the singular points and hence lose some of the robot maneuverability. Based on a class of traditional path followers we design a new globally stable path following controller that exploits the high maneuverability of the platform. This design allows us to derive a set of closed-form analytical functions that describe the robot base velocity as a function of the wheels driving and steering velocities while abide to the robot non-holonomic constraints. Those functions are then utilized to find the maximum instantaneous velocity of the body that keeps the wheels velocities under the pre-specified bounds no matter how much the robot gets close or far from its singular configurations. The control algorithms developed in this paper have been evaluated on *iMoro*, a four wheel independently steered mobile manipulator designed and developed at IHA/TUT. Experimental data is also shown that show efficacy of the method.**

I. Introduction

The interest in mobile robots with active steering wheels has been increasing over time. This is due to their high maneuverability and flexibility while being able to operate on rough terrains and carry higher payloads with better efficiency compared with the other types of omnidirectional mobile robots. Such platforms are now being used in developing advanced mobile robots in many practical fields such as service robotics [1], [2], agricultural tasks [3] and space applications [4]. We as a part of the PURESAFE [1] project that aims to prevent human intervention in radioactive environments have designed and developed a four wheeled independently steerable mobile manipulator (iMoro) that is shown in Fig. 1. Our goal is to take advantage of such platforms maneuverability to perform manipulation and remote inspection in the confined spaces of CERN LHC tunnel.

The platform high maneuverability comes from its omnidirectional nature. The mobile platform is able to realize any arbitrary, independent set of linear and angular velocities but only after it has initially reoriented its wheels to a predetermined corresponding configuration. Hence, while those platforms have three Degrees of Freedom (DoF) [5], they are over-actuated [6] with all the actuators should abide the non-holonomic constraints of the robot. Hence, such robots are sometimes called non-holonomic omnidirectional robots[7] or pseudo-omnidirectional robots[8], [4].

The early researches on such robots have focused more on over-constrained nature [9] and correction of wheel odometry errors for deadreconing localization [10], [11]. Advances in sensors and actuators along with sensor fusion based algorithms for localization have solved many of those early issues. The interest is now shifted to analyze and develop more sophisticated control schemes for those robots [12], [13], [14].

However, presence of singularities both inherently [15] and in the presentation of the configuration space [16] makes design and realization of motion controllers a challenging task. One of the most popular ways to describe the platform configuration space and those singularities, is with the notion of Instantaneous Center of Rotation (ICR) [15]. The ICR of the robot body is defined on the horizontal Cartesian plane and with respect to the coordinate frame attached to the body. Each wheel is following an instantaneous circle with ICR at its center. As the ICR gets close to the wheel axis the radius of that circle becomes smaller. Hence, the driving velocity of that wheel decreases while its steering velocity unboundedly increases. When the ICR coincides with the wheel steering axis the driving velocity of the wheel becomes zero and its steering angle becomes undefinable. Hence, the 2D position vector of the ICR along with the angular velocity around it can serve as the state space for

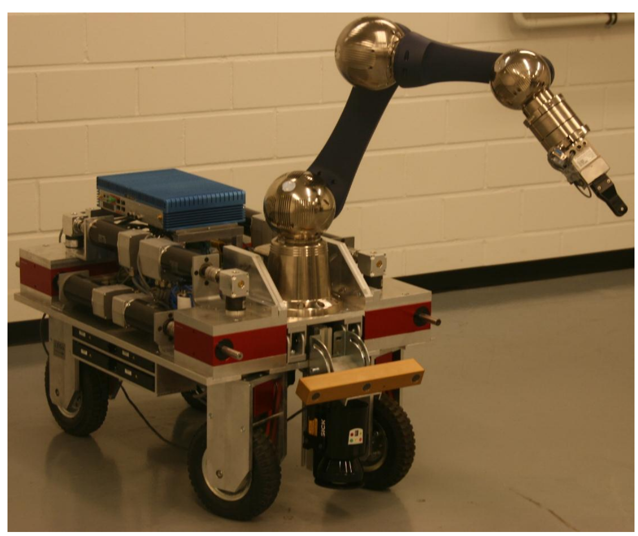

Fig. 1. The iMoro Mobile Manipulator

Authors are with Intelligent Hydraulics and Automation Department, Tampere University of Technology, 33101, Finland.
Corresponding author's email: oftadeh.reza@ieee.org.

[1]www.iha.tut.fi/puresafe/

the platform except when a wheel is singular and should be treated separately [15]. Moreover, when traversing along a straight line, ICR remains at infinity which can be regarded as a presentation singularity for that state space. Authors in [16] proposed an alternative ICR representation to avoid such singularities. Results in [16] are later extended in [17], where a singular-free switching state space has been proposed that addresses both inherent and presentation singularities. In another approach by [18], to avoid singularity, each wheel has been given an extra degree of freedom which makes the wheels footprint variable. However, the problem of operating in the close neighborhood of the singular configurations still remains unsolved.

Most of the solutions proposed so far try to plan ICR trajectories in singular free regions of platform velocity space [15]. Other solutions [8], [19] treat the singular configurations and their neighborhood as obstacles and solve a navigation problem based on potential field and/or model predictive control methods. However, in all of those methods considerable portions of the configuration space are avoided, thus reducing maneuverability of the platform. Even when realizing some simple maneuvers, at some points of the operation ICR will necessarily get relatively close to at least one wheel. One of the simple cases is when the platform is moving on a straight line while changing its heading by 180 degrees.

Furthermore, when the robot is required to follow a desired path and heading profile, ICR position has already been determined. Hence none of those approaches are suitable for path following problems. To clarify this matter, lets call the footprint of a wheel steering axis, the wheel path. Consider the platform body frame is moving on a regular curve called the body path with its natural parameter denoted $'s'$. Moreover, its heading, changes as a smooth function of s namely the heading function. Regardless of the velocities, the wheel path can be determined as a function of the body path and the heading function. Clearly, relative distance between the platform ICR and a wheel axis is the reciprocal of the wheel path curvature. Hence, a wheel is singular when the ICR coincides with its steering axis and equivalently when wheel path curvature is infinity.

In this paper, following ideas from classical path followers [20], we extend our earlier work[21] and develop a novel path following controller that guides the platform on a desired path and correct the heading accordingly. Consider the magnitude of the platform velocity is v, we show that stability of the origin of the error space is achieved regardless of v. Moreover, the control signals simplify the non-holonomic constraints and makes the curvature of the wheels paths independent of v and only as functions of desired body path, heading function and error signals. Hence, the steering and driving velocities of the wheels become linear proportions of v. Having v as an independent variable, this solution allows us to analytically find a maximum v at each sample time that keeps the actuators equal or less than their pre-specified bounds. The solution for a given path and heading function is time optimal since at each time step at least one actuator operates at its maximum velocity.

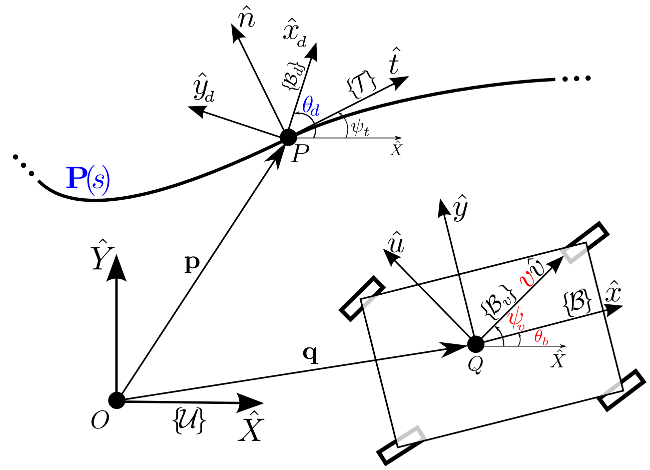

Fig. 2. Desired path and the required coordinate frames

The paper is organized as follows. In Section II, we describe the general architecture of the robot. Next, we define the problem at hand and the error space. In section IV, we describe the controller. Next, in Section V, by means of the proposed control signals we derive the analytical velocity constraints and based on that the optimal velocity of the platform is derived in VI. In the last section, we show the efficacy of the proposed solution through experiments done with the *iMoro* mobile platform.

II. General Descriptions

A. Robot's Architecture

iMoro, shown in Fig.1, consists of a rigid base and four wheels, each of which has two DOF. In fact, each wheel is equipped with two independent servo drives for steering and driving. The steering actuator rotates the whole wheel along its vertical axis, hence determining the heading of that wheel. The driving actuator drives the wheel and its contact point with the ground coincides with the wheel's vertical axis. The robot is equipped with off-the-shelf control drivers that provide velocity and position control options. Because of control strategy adopted in this paper, it is desirable to control the position of the steering and velocity of driving actuators. Hereafter, we will assume that control signals are four steering angles and four wheel speeds.

B. Notations and Symbols

We denote the vectors in bold with hat-sign($\hat{}$) for unit vectors. A left superscript to a vector denotes the frame in which it is expressed, except for the inertial frame, which is omitted for the sake of brevity. Moreover, for two arbitrary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a}.\boldsymbol{b}$ and $\boldsymbol{a} \times \boldsymbol{b}$ are the inner and cross products of the two vectors, respectively.

Fig. 2 depicts a schematic view of the platform together with a desired path. The coordinate frame $\boldsymbol{\mathcal{U}}\{\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}\}$ is the inertial frame with unit vectors $\hat{\boldsymbol{X}}$, and $\hat{\boldsymbol{Y}}$. Frame $\boldsymbol{\mathcal{B}}\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}\}$ is a fixed-body frame defining the heading of the robot, and $\boldsymbol{\mathcal{B}}_v\{\hat{\boldsymbol{v}}, \hat{\boldsymbol{u}}\}$ is the velocity frame, that is, unit vector $\hat{\boldsymbol{v}}$

determines the direction of the robot's base linear velocity vector and scalar v its magnitude. Both $\boldsymbol{\mathcal{B}}$ and $\boldsymbol{\mathcal{B}}_v$ are attached to the robot's base at point $\boldsymbol{Q}$ which can be chosen at will. Angles ψ_v and θ_b are the angle of $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{x}}$, respectively, in $\boldsymbol{\mathcal{U}}$. The tangent frame $\boldsymbol{\mathcal{T}}\{\hat{\boldsymbol{t}}, \hat{\boldsymbol{n}}\}$ is the Serret-Frenet frame at point $\boldsymbol{P}$ attached to the desired path $\boldsymbol{P}_d(s)$ and the angle between $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{X}}$ is denoted as ψ_t. To determine the desired heading, we define frame $\boldsymbol{\mathcal{B}}_d\{\hat{\boldsymbol{x}}_d, \hat{\boldsymbol{y}}_d\}$ attached to $\boldsymbol{P}$ such that the angle of $\hat{\boldsymbol{x}}_d$ in $\boldsymbol{\mathcal{U}}$ which is θ_d, equals the desired heading function $\theta_d(s)$. Note that both desired path $\boldsymbol{P}_d(s)$ and desired heading $\theta_d(s)$ are parametrized with the same parameter s. Moreover, vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ define points $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively, in $\boldsymbol{\mathcal{U}}$.

III. Problem Definition

The desired path $\boldsymbol{P}_d(s)$ is assumed to be a regular curve with bounded curvature on a horizontal plane. It is defined by a vector-valued function $\boldsymbol{P}_d : [0, L_P] \rightarrow \boldsymbol{R}^2$, where s and L_P are natural parametrization and length of the path, respectively. The curvature of $\boldsymbol{P}_d(s)$ which is a function of s is denoted as $C_c(s)$. Moreover, we assume that the desired heading function $\theta_d(s) : [0, L_P] \rightarrow \boldsymbol{R}$ is two times differentiable function of s.

As long as $\boldsymbol{P}_d(s)$ and $\theta_d(s)$ with mentioned conditions are defined, the scalar s determines the pos of the platform base. Moreover, the wheels configurations are uniquely determined since the wheels steering are tangent to the wheels paths. Hence, as long as the curvature of a wheel path is less than infinity the configuration of the whole platform is determinable by s. In case of errors, three values for errors in x, y directions and one for the heading in addition to s are needed to determine the platform configuration. Hence, along s we derive the following error signals to serve as state variables,

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = {}^{\mathcal{U}}\boldsymbol{R}_{\mathcal{T}}^{-1}(\boldsymbol{q} - \boldsymbol{p}) \tag{1a}$$

$$\theta_e = \theta_d - \theta_b \tag{1b}$$

$$\psi_e = \psi_t - \psi_v, \tag{1c}$$

where ${}^{\mathcal{U}}\boldsymbol{R}_{\mathcal{T}}$ is the rotation matrix $\boldsymbol{R}(\psi_t)$ that defines the rotation from frame $\boldsymbol{\mathcal{T}}$ to frame $\boldsymbol{\mathcal{U}}$. In this case the error signals x_e and y_e are position errors measured along $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$, respectively.

Time derivative of eqs. (1a) and (1b) results in,

$$\dot{x}_e = \dot{s}(C_c(s)y_e - 1) + v\cos(\psi_e) \tag{2a}$$

$$\dot{y}_e = -\dot{s}C_c(s)x_e - v\sin(\psi_e) \tag{2b}$$

$$\dot{\theta}_e = \frac{\partial \theta_d}{\partial s}\dot{s} - \omega_b \tag{2c}$$

in which, $\omega_b = \dot{\theta}_b$ is the angular velocity of $\boldsymbol{\mathcal{B}}$.

Problem 1: Given desired path $\boldsymbol{P}_d(s)$ and heading profile $\theta_d(s)$, derive *feedback control laws* for the speed and the steering angle of each wheel such that,

1) Path following: frame $\boldsymbol{\mathcal{B}}_v$ converges and follows frame $\boldsymbol{\mathcal{T}}$, that is, error signals x_e, y_e and ψ_e remain bounded and converge to zero. See eqs. (1a) and (1c).

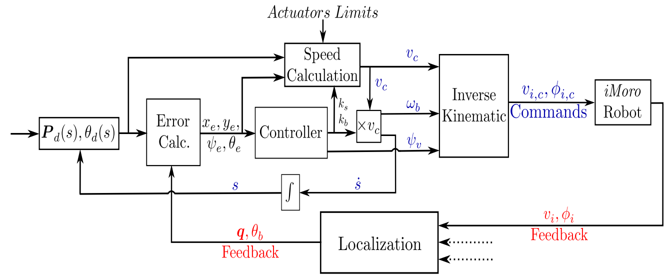

Fig. 3. Schematic block diagram of the whole system

2) Heading control: frame $\boldsymbol{\mathcal{B}}$ converges and follows frame $\boldsymbol{\mathcal{B}}_d$, that is, error signal θ_e remain bounded and converge to zero. See (1b).
3) Bounded control signal: rate of steering and speed of the wheels do not exceed predefined actuator bounds.

We consider the following assumptions to derive our solution. There is no mechanical constraint for wheels steering. The wheels are free-turn. The robot moves on a flat and horizontal plane. The base and the wheels are rigid and the wheels are non deformable. Furthermore, the condition of pure rolling without any side slippage is assumed for the wheels.

We will solve Problem 1 in two stages. First, assuming speed v a free variable, we use angular velocity ω_b and base linear velocity direction $\hat{\boldsymbol{v}}$ as control inputs and propose control laws to address subproblems 1 and 2 of Problem 1. This is presented next in Section IV. These control laws are then mapped to actuator signals (wheels steering and speed commands) using inverse kinematic relations. In the second stage, we use velocity magnitude v to address actuator bounds. This is addressed in Section VI. Fig. 3 schematically depicts the block diagram of control architecture. In this paper we do not discuss the localization block. We have described Inverse and Forward kinematic blocks in [22]. Many standard localization algorithms for mobile robots along with more specific approaches such as the one given in [23] are applicable. However, one of the key requirements of our approach is that the error signals should be continuous and differentiable which imposes the localization block to provide smooth signals for the platform pose.

IV. Derivation of Control Laws

Next we derive the control laws for ω_b and ψ_v (direction of $\hat{\boldsymbol{v}}$). Consider the error states eqs. (1a) to (1c). The control objective is to derive feedback control laws for ψ_v, ω_b, and $\dot{s}$ such that $x_e, y_e, \theta_e, \psi_e$ asymptotically converge to zero. Notice that $\dot{s}$ becomes an auxiliary control signal. First, define functions σ as,

$$\sigma(y_e) = -\operatorname{sgn}(v)\sin^{-1}\frac{k_2 y_e}{|y_e| + \epsilon} \tag{3}$$

where, $0 < k_2 \leq 1$ and $\epsilon > 0$. $\sigma(y_e)$ is a function that generates an appropriate approach angle from the robot to $\boldsymbol{P}_d(s)$. We assume that $v \geq 0$, so $\operatorname{sgn} v = 1$ and $\sigma(y_e)$

becomes independent of v. Since steering angle can take any value from $[-\pi, \pi]$, choice $v \geq 0$ puts no constraint on the configuration space.

Proposition 1: The feedback control laws given by,

$$\dot{s} = (k_1 x_e + \cos\sigma(y_e))v = k_s v \tag{4a}$$

$$\omega_b = (k_3 \theta_e + \frac{\partial \theta_d}{\partial s} k_s)v = k_b v \tag{4b}$$

$$\psi_v = \psi_t - \sigma(y_e) \tag{4c}$$

where $k_1, k_3 > 0$ solves sub-problems 1 and 2 of Problem 1. In particular, the origin of the error space is stable and is semi-globally exponentially stable if $v(t) \geq v_m > 0$.

Proof: Here we use similar Lyapunov functions as in [24] and [25], while new control laws are chosen to make curvatures independent of speed v, importance of which will be clear later. Consider the following Lyapunov function,

$$V_P = \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2 + \frac{1}{2}\theta_e^2 \tag{5}$$

which is positive definite and radially unbounded. The derivation of V_P along solution of eqs. (2a) to (2c) result in,

$$\dot{V}_P = -(k_1 x_e^2 + k_2 \frac{y_e^2}{|y_e| + \epsilon} + k_3 \theta_e^2)\, v(t) \tag{6}$$

which is negative, thus the origin is stable. For a given $d_1 > 0$, if $v(t) \geq v_m > 0$ and initially $|y_e(t_0)| < d_1$, it is easy to show that $\dot{V}_P < -\lambda V_P$. Thus, the origin is semi-globally exponentially stable [26]. Detail of the proof follows similar steps as in [20]. Now, note that substituting ψ_v from (4c) into (1c) results in $\psi_e = \sigma(y_e)$. Hence, ψ_e is bounded and when y_e converges to zero, error ψ_e will converge to zero. ■

It is worth noticing that gains k_s and k_b defined in (4a) and (4b), respectively, are independent of v. Thus rate of progress $\dot{x}_e$, $\dot{y}_e$, and $\dot{\theta}_e$ become proportional to scaler v. This can be easily shown by investigating $\dot{V}_P$ or substituting the control laws in (2a), (2b), and (2c) and they are given in Appendix. This means that as the platform goes faster it compensates the errors faster. As we show in the next section this choice simplifies the non-holonomic constraints to great extents. One way to derive k_1 and k_3 is to assume a maximum for v namely v_{max} and design constant error gains $k_{1,max}$ and $k_{3,max}$ based on v_{max} and then select k_1 and k_3 such as,

$$k_1 = \frac{k_{1,max}}{v_{max}} \ , \ k_3 = \frac{k_{3,max}}{v_{max}} \tag{7}$$

Last but not least, while the controller require the $\boldsymbol{P}_d(s)$ to be a regular curve, the actual desired path may consist of multiple regular curves that are non-smoothly connected together and a higher level state-machine can be designed to feed the regular segments individually to the controller. When the platform reaches the end of a segment the state machine stops the platform and rotates the steering wheels to comply with the start of the next segment and then feeds the new segment to the controller. Moreover, while a segment could be infinitesimally small in order to emulate a spot turn

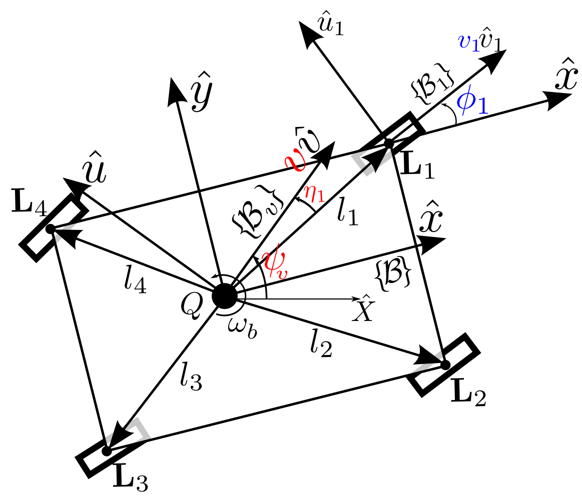

Fig. 4. Denoted kinematic parameters

for the platform, that higher level state machine can also be incorporated to provide the platform with exact spot turn at the end of any desired segment.

V. Derivation of Kinematics Constrains

In Fig. 4 the necessary variables are defined. Based on the figure, ${}^{\mathcal{B}}\boldsymbol{\ell}_i\,,\ i \in \{1,2,3,4\}$ are constant vectors related to the geometry of the robot with their magnitude being l_i. ϕ_i is the steering angle of the i^{th} wheel and the vector ${}^{\mathcal{B}}\hat{\boldsymbol{v}}_i$ can be written in the form of $[\cos(\phi_i) \ \ \sin(\phi_i) \ \ 0]^T$. $v_i \hat{\boldsymbol{v}}_i$ is the velocity vector of the attachment point $\boldsymbol{L}_i$ which also coincides with the wheel steering axis. η_i is the angle between $\hat{\boldsymbol{v}}$ and $\boldsymbol{\ell}_i$.

The following kinematic constraint maps the base velocity space to the wheels velocities.

$${}^{\mathcal{B}}\hat{\boldsymbol{v}} = \boldsymbol{R}(\psi_v - \theta_b)[1 \ 0 \ 0]^T \tag{8a}$$

$$v_i \ {}^{\mathcal{B}}\hat{\boldsymbol{v}}_i = v \ {}^{\mathcal{B}}\hat{\boldsymbol{v}} + \omega_b(\hat{\boldsymbol{z}} \times {}^{\mathcal{B}}\boldsymbol{\ell}_i) \tag{8b}$$

in which, $\hat{\boldsymbol{z}} = [0 \ 0 \ 1]^T$ and $\boldsymbol{R}(\psi_v - \theta_b)$ is the rotation matrix with angle $\psi_v - \theta_b$ around z-axis, that is, frame $\boldsymbol{\mathcal{B}}_v$ in $\boldsymbol{\mathcal{B}}$. The norm of (8b) and also its time differentiation can be simplified to,

$$v_i = \sqrt{v^2 + \omega_b^2 l_i^2 + 2v\omega_b l_i \sin\eta_i} \tag{9a}$$

$$v_i^2 \ \dot{\phi}_i = l_i \left(\dot{\omega}_b v - \dot{v}\omega_b\right)\cos\eta_i + v\left(\omega_v - \omega_b\right)\left(v + l_i\omega_b \sin\eta_i\right) \tag{9b}$$

in which, $\omega_v = \dot{\psi}_v$. Note that deriving (9b) requires some tedious but elementary algebraic manipulation. Notice the presence of acceleration terms $\dot{v}$ and $\dot{\omega}_b$ in the above equation and also its non-linearity with respect to v. In what follows we show that the choice of control signals in previous section cancel out those accelerations. It is virtually impossible to derive the velocity v based on the other variables. Moreover, the singular configuration is when $|\omega_b|$ becomes v/l_i, $\sin\eta_i$ becomes $-\operatorname{sgn}(\omega_b)$ and consequently $\cos\eta_i$ becomes zero. In this case v_i becomes zero and $\dot{\phi}_i$ is undefined. Notice that as the robot gets close to its singular configuration $\dot{\phi}_i$ becomes unboundedly large.

Here, we simplify above equations using the control signals presented in previous section. Based on (4b) and (4c), $\dot{\omega}_b$ and ω_v can be written as,

$$\dot{\omega}_b = k_b' \ v^2 + k_b \ \dot{v} \tag{10a}$$

$$\omega_v = k_v \ v \tag{10b}$$

in which, k_b' and k_v are independent of the platform speed v and are given in Appendix. Substitute $\dot{\omega}_b$, ω_b and ω_v from eqs. (4b), (10a) and (10b) into eqs. (8b), (9a) and (9b),

$$^{\mathcal{B}}\hat{\boldsymbol{v}}_i = \frac{^{\mathcal{B}}\hat{\boldsymbol{v}} + k_b(\hat{z} \times {}^{\mathcal{B}}\boldsymbol{\ell}_i)}{\sqrt{1 + k_b^2 l_i^2 + 2 l_i k_b \sin\eta_i}} \tag{11}$$

$$v = \frac{v_i}{\sqrt{1 + k_b^2 l_i^2 + 2 l_i k_b \sin\eta_i}} \tag{12a}$$

$$v = \dot{\phi}_i \frac{1 + k_b^2 l_i^2 + 2 l_i k_b \sin\eta_i}{k_b' l_i \cos\eta_i + (k_v - k_b)\,(1 + l_i k_b \sin\eta_i)} \tag{12b}$$

As described in Section II, the input values for steering and driving actuators of the wheel i are ϕ_i and v_i, respectively. Based on (11), the wheel steering angle is independent of v. Hence, even if the robot is stopped, the wheel path and so the steering angle can be determined. Moreover, k_b', k_b and k_v are functions of errors signals x_e, y_e, θ_e and desired variables $\boldsymbol{P}_d(s)$, $\theta_d(s)$ and their partial differentiations. Hence, for a given v_i, (12a) gives the velocity v that realizes that velocity. Correspondingly, given a $\dot{\phi}_i$, (12b) gives the velocity v that realizes that steering velocity. The curvature of the wheel path namely κ_i is $\dot{\phi}_i/v_i$ which is independent of v. Hence, as the platform moves toward its singular position and κ_i goes to infinity, (12b) reduces v with an appropriate rate to retain the given steering velocity.

Notice that right at the singular configuration, (12b) gives zero for v and the platform stops. Hence, if the robot were to be entrapped exactly in a singular configuration it would stop indefinitely. However, we state without proof that such configuration is an unstable equilibrium point of the system and practically it cannot become deadlock for the platform.

VI. Bounded Velocity Solution

In this section, we present our strategy to find the upper-bounds for v_c; the command signal for v, to limit the driving and steering velocities of wheels to prespecified bounds.

For a robot that has n independently steered wheels there are $2n$ velocity constraints to fulfill which are n deriving and n steering constraints. Consider the maximum driving velocity of a wheel is $V_{d,max}$ and the maximum steering velocity is $\dot{\phi}_{max}$. At each step of time, k_b', k_b and k_v are calculated so as the control signals ω_b and ψ_v. Next, substituting v_i by $V_{d,max}$ in (12a) and evaluate it for n wheels result in n candidates for the platform velocity. Equivalently, substituting $\dot{\phi}$ by $\dot{\phi}_{max}$ in (12b) and again evaluate it for all the wheels result in another set of n velocity candidates. Hence, there are $2n$ candidates for v namely $v_{c,j}$, we select v as,

$$v_c = \min(|v_{c,j}|), j \in \{1, 2, .., 2n\} \tag{13}$$

Note that functions (12a) and (12b) are strictly monotonic with respect to v_i and $\dot{\phi}_i$ respectively and so is their inverse with respect to v. Hence, the minimum of those eight candidates result in driving and steering velocities less than or equal to the given bounds. Moreover, based on this method at each point in time, at least one actuator is performing at its limit. When the wheels are far from the body ICR, the limit is determined by driving bond. When a wheel gets close to the body ICR, the limit is set by the steering bond. Hence, at each step the given velocity is the maximum feasible velocity to satisfy the actuators constraints and the solution is time optimal. In real robot, command speed v_c differs from v. It can be shown that the system is input-to-state stable with deviation $v - v_c$ as input. See [26] for more details.

VII. Experimental Results

In this section, experimental results of the proposed control system are presented. Experiments are done on *iMoro*, Fig. 1, with parameters shown in Table I. The limit for steering velocity mentioned in the table is considerably lower than feasible value to show the efficacy of the algorithm. Controller software is implemented in an embedded PC with real-time Linux [27].

Fig. 5 shows the actual path taken by the robot in the experiment based on wheel odometry. The desired path is a Bezier curve and the heading function is a simple polynomial of the path variable that changes from zero to 2π and hence require the robot to turn around itself while it follows the path. There is a relatively large intentional initial error between the start of the path and the initial position of the platform to test the performance of the controller. The figure shows that the robot properly follows the desired path and the heading profile.

Notice that even with some fluctuations due to the robot's unmodeled dynamics, the velocity boundaries are kept in comply with the velocity requirements. Fig.6 shows that the steering velocities remain below the bound, while Fig.7 shows that the driving velocities are saturated to the bound. Fig.8 show the maximum allowable speed command in the experiment. The figures show how the propose algorithm reduces the command speed (particularly sharply at critical areas) and thus retain the steering velocities below the prespecified limit.

TABLE I

SPECIFICATIONS OF *i*MORO ROBOT FOR THE EXPERIMENTAL STUDIES

Description	Quantity
Main Body Length (x direction)	655 mm
Main Body Width (y direction)	335 mm
Maximum Velocity of Steering Servo Motors	0.4 $^{rad}/_{sec}$
Maximum Velocity of Driving Servo Motors	50 $^{mm}/_{sec}$
Wheel Diameter	210 mm
Approximate Overall Mass	120 kg

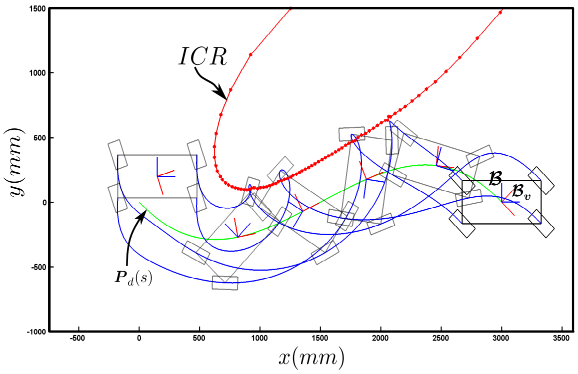

Fig. 5. Robot footprint in the experiment

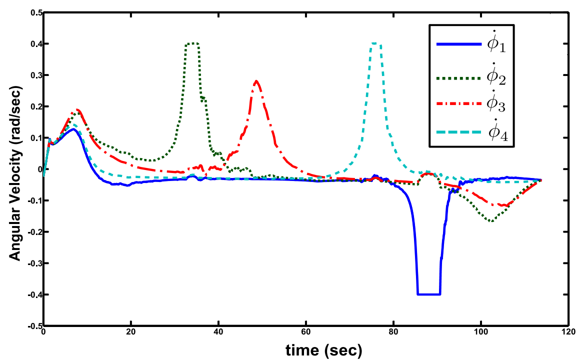

Fig. 6. Steering velocities $\dot{\phi}_i$

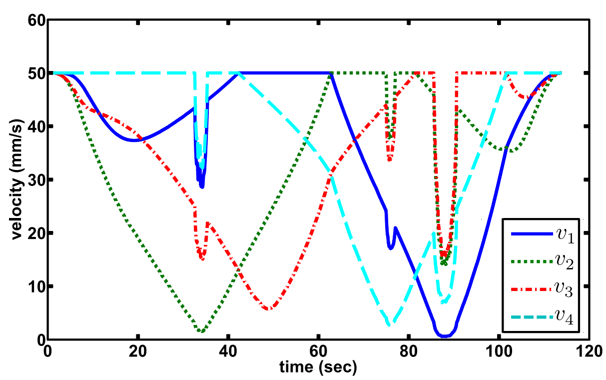

Fig. 7. Driving velocities v_i

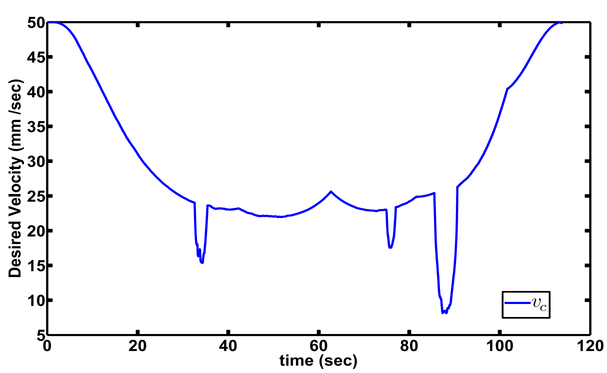

Fig. 8. Command signal for base speed v_c

VIII. Conclusion

This paper proposes a new solution to the path following problem of independently steered mobile robots. We show that utilizing this controller, the robot smoothly follow the desired path while it changes its heading based on a given desired function. Moreover, this design leaves the speed of the robot base as an arbitrary variable. We show that the proposed control signals significantly simplifies the non-holonomic kinematic constraints of the robot. Hence, the speed of the base can be determined analytically to keep the steering and driving velocities of the wheels under pre-specified values. Therefore, unlike many previous methods, our approach allows the robot to get close to its singular configurations and hence exploit its inherent maneuverability. The experiments show that even without considering any model-based approach or dynamic analysis, the designed path follower and related kinematics formulations are capable of bounding both steering and driving velocities in practice with some tolerances. Since this approach gives an analytical solution; it is capable of real-time implementation with low process costs. Our future work will target performance improvement of the controller for mobile manipulation purposes and its fault tolerance.

Appendix

Substituting the control signals (4b) and (4c) in the error states (2a), (2b), and (2c), they can be written as,

$$\dot{x}_e = \left(k_s \left(C_c y_e - 1\right) + \cos\left(\psi_e\right)\right) v = k_x v \tag{14a}$$

$$\dot{y}_e = -\left(k_s C_c x_e + \sin\left(\psi_e\right)\right) v = k_y v \tag{14b}$$

$$\dot{\theta}_e = \left(\frac{\partial \theta_d}{\partial s} k_s - k_b\right) v = k_\theta v \tag{14c}$$

Differentiating from (4c) and (4b) respectively yields to,

$$\omega_v = \left(\left(1 + \frac{\partial \sigma\left(y_e\right)}{\partial y_e} x_e\right) C_c k_s + \frac{\partial \sigma\left(y_e\right)}{\partial y_e} \sin(\psi_e)\right) v = k_v v \tag{15a}$$

$$\dot{\omega}_b = k_b' \, v^2 + k_b \dot{v} \tag{15b}$$

in which,

$$k_b' = k_2 k_\theta + \frac{\partial^2 \theta_d}{\partial s^2} k_s^2 + \frac{\partial \theta_d}{\partial s} k_1 k_x - \frac{\partial \theta_d}{\partial s} \frac{\partial \sigma\left(y_e\right)}{\partial y_e} k_y \sin \sigma(y_e) \tag{16}$$

Acknowledgments

This work, supported by the European Union's Seventh Framework Program under the Marie Curie Initial Training Network, was carried out within the framework of the PURESAFE, *Preventing hUman intervention for incREased SAfety in inFrastructures Emitting ionizing radiation*, under REA grant agreement number 264336.

REFERENCES

[1] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 20–33, 2012.

[2] B. Graf, U. Reiser, M. Hagele, K. Mauz, and P. Klein, "Robotic home assistant care-o-bot® 3-product vision and innovation platform," in *Advanced Robotics and its Social Impacts (ARSO), 2009 IEEE Workshop on*. IEEE, 2009, pp. 139–144.

[3] C. Cariou, R. Lenain, B. Thuilot, and M. Berducat, "Automatic guidance of a four-wheel-steering mobile robot for accurate field operations," *Journal of Field Robotics*, vol. 26, no. 6-7, pp. 504–518, 2009.

[4] U. Schwesinger, C. Pradalier, and R. Siegwart, "A novel approach for steering wheel synchronization with velocity/acceleration limits and mechanical constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5360–5366.

[5] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 1, pp. 47–62, 1996.

[6] P. F. Muir and C. P. Neuman, "Kinematic modeling of wheeled mobile robots," *Journal of robotic systems*, vol. 4, no. 2, pp. 281–340, 1987.

[7] J.-B. Song and K.-S. Byun, "Steering control algorithm for efficient drive of a mobile robot with steerable omni-directional wheels," *Journal of mechanical science and technology*, vol. 23, no. 10, pp. 2747–2756, 2009.

[8] C. Connette, C. Parlitz, M. Hägele, and A. Verl, "Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots," in *IEEE International Conference on Robotics and Automation, (ICRA)*. IEEE, 2009, pp. 4124–4130.

[9] F. G. Pin and S. M. Killough, "A new family of omnidirectional and holonomic wheeled platforms for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 4, pp. 480–489, 1994.

[10] D. B. Reister and M. Unseren, "Position and constraint force control of a vehicle with two or more steerable drive wheels," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 6, pp. 723–731, 1993.

[11] J. Borenstein, "Control and kinematic design of multi-degree-of freedom mobile robots with compliant linkage," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 1, pp. 21–35, 1995.

[12] Y. Mori, E. Nakano, and T. Takahashi, "Mechanism, control and design methodology of the nonholonomic quasi-omnidirectional vehicle odv9," *The International Journal of Robotics Research*, vol. 21, no. 5-6, pp. 511–525, 2002.

[13] M. Selekwa and J. Nistler, "Path tracking control of four wheel independently steered ground robotic vehicles," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. IEEE, 2011, pp. 6355–6360.

[14] A. K. Singh, D. Ghose, and K. M. Krishna, "Optimum steering input determination and path-tracking of all-wheel steer vehicles on uneven terrains based on constrained optimization," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3611–3616.

[15] B. Thuilot, B. d'Aandrea Novel, and A. Micaelli, "Modeling and feedback control of mobile robots equipped with several steering wheels," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 3, pp. 375–390, 1996.

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

[17] C. Connette, M. Hagele, and A. Verl, "Singularity-free state-space representation for non-holonomic, omnidirectional undercarriages by means of coordinate switching," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4959–4965.

[18] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 6136–6142.

[19] C. Connette, A. Pott, M. Hägele, and A. Verl, "Addressing input saturation and kinematic constraints of overactuated undercarriages by predictive potential fields," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 4775–4781.

[20] R. Ghabcheloo, "Coordinated Path Following," Ph.D. dissertation, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 2007.

[21] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila, "Bounded-velocity motion control of four wheel steered mobile robots," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Wollongong, Australia*, July 2013.

[22] ——, "Mechatronic design of a four wheel steering mobile robot with fault-tolerant odometry feedback," in *The 6th IFAC Symposium on Mechatronic Systems*, Hangzhou, China, 10-12 April 2013.

[23] L. Clavien, M. Lauria, and F. Michaud, "Instantaneous centre of rotation estimation of an omnidirectional mobile robot," in *IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA*. IEEE, 2010, pp. 5435–5440.

[24] D. Soetanto, L. Lapierre, and A. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," in *Proceedings of 42nd IEEE Conference on Decision and Control (CDC)*, vol. 2. IEEE, 2003, pp. 1765–1770.

[25] I. Kaminer, A. Pascoal, and O. Yakimenko, "Nonlinear path following control of fully actuated marine vehicles with parameter uncertainty," in *16th IFAC World Congress*, Prague, Czech Republic, 2005.

[26] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[27] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila, "Unified framework for rapid prototyping of linux based real-time controllers with matlab and simulink," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM),Kaohsiung, Taiwan*, July 2012, pp. 274 –279.