ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/308813002

A unified motion control and low level planning algorithm for a wheeled skid-steering robot

Conference Paper · September 2015

DOI: 10.1109/ETFA.2015.7301490

CITATION
1

READS
64

3 authors:

Dariusz Pazderski
Poznan University of Technology
54 PUBLICATIONS **455** CITATIONS
SEE PROFILE

Krzysztof R. Kozłowski
Poznan University of Technology
171 PUBLICATIONS **962** CITATIONS
SEE PROFILE

Tomasz Gawron
Poznan University of Technology
18 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Control system of of a manipulator for surgery View project

Control algorithmization for state- and input-constrained driftless nonholonomic systems in the context of complex motion tasks of mobile robots. View project

All content following this page was uploaded by Dariusz Pazderski on 03 January 2017.

The user has requested enhancement of the downloaded file.

A Unified Motion Control and Low Level Planning Algorithm for a Wheeled Skid-Steering Robot

Dariusz Pazderski and Krzysztof Kozłowski and Tomasz Gawron
Chair of Control and Systems Engineering
Poznań University of Technology
ul. Piotrowo 3a, 61-138 Poznań, POLAND
Email: dariusz.pazderski@put.poznan.pl, krzysztof.kozlowski@put.poznan.pl, tomasz.gawron@doctorate.put.poznan.pl

***Abstract*—In this paper a unified motion control and planning algorithm dedicated for the waypoint following task realized by a skid-steering is presented. In order to reduce excessive slip effects between robot wheels and ground it is assumed that the vehicle moves with bounded velocities and accelerations. The motion controller has been designed using formal methods to ensure asymptotically stable tracking of feasible reference trajectories. To account for practical motion tasks the trajectory tracking algorithm is complemented by a motion planner, which utilizes the differential flatness property of unicycle-like kinematics. During the motion planning stage an auxiliary trajectory connecting points in the configuration space and satisfying assumed phase constraints is generated. The resulting motion execution system has been implemented on a laboratory-scale skid-steering mobile robot, which served as platform for experimental validation of presented algorithms.**

I. Introduction

Motion control falls under fundamental issues considered in the field of mobile robotics. Generally, this problem can be challenging when one takes into account the fact that many robotic platforms are subject to nonintegrable differential constraints at kinematic and/or dynamic level. As a result, it is difficult or even impossible to realize an arbitrary trajectory even in an obstacle-free configuration space. An example of such a system is a differentially driven wheeled skid-steering vehicle which is governed by highly uncertain dynamics describing nonlinear wheels-ground interactions, as opposed to two-wheeled platforms frequently considered in the literature. In order to deal with this issue various models of robots subjected to slip effects have been presented to date. The detailed models assumes exact knowledge of interaction forces and can be effectively used for simulation and open-loop control design purposes – cf. [1]. As an alternative, a combination between kinematic and dynamic models was investigated in [2]. Other models based on kinematic approximation of nonitegrable velocity constraints can be found in [3], [4], [5], [6].

In this paper we postulate that modelling of a skid-steering vehicle should be practically oriented. We model robot motion with unicycle-like kinematics, which can be justified as a local approximation in the range of relatively low velocities and accelerations. In such conditions the upper bounds of disturbances introduced by slip effects are low and in consequence those disturbances can be attenuated by a closed-loop controller.

Investigated waypoint following motion task is often encountered in practical robotic applications, thanks to simplicity of its definition. The robot is driven through a prescribed sequence of waypoint configurations with consideration of feasible range of velocities (see e.g. [7]). No reference trajectory or path is given a priori.

In order to solve this problem we propose to combine a motion planner with a closed-loop tracking controller. Such an approach allows one to define a suitable motion plan, which takes into account desired velocity profiles, path curvature, etc. in order to limit skid/slip phenomena. Moreover, reduction of initial tracking error occurring during the planning stage results in significant reduction of transient response of the controller. Worth noting, that trajectory planning allows to introduce persistent excitation to the closed-loop control system which can improve convergence of waypoint configuration error.

The motion controller constructed in the sequel is designed using Lyapunov stability analysis to ensure global boundedness of configuration error for set-point control and asymptotic tracking for a specific class of reference trajectories. It is similar to control solution investigated in [8]. However, our control law can be viewed as a generalization of the previous algorithm derived utilizing Lie groups theory. Such an approach allows one to take advantage of intrinsic symmetry of the configuration space.

The motion planner described in this paper takes advantage of motion primitives in the form of polynomial segments to connect subsequent waypoint configurations. Recently, similar approach based on Bézier curves have been studied in [9]. In order to ensure continuous transition between path segments tangent and curvature conditions at waypoints have been considered as interpolation constraints. The algorithm is defined locally for each segment using symmetry in the configuration space. Then the trajectory is generated using a time scaling approach which takes into account the prescribed bounds of both linear and angular velocity. Moreover, continuous evolution of the velocity is considered.

This work was supported by The National Centre for Research and Development (NCBiR) grant "Autonomy in rescue and exploration robots", number PBS1/A3/8/2012.

978-1-4673-7929-8/15/$31.00 © 2015 European Union

Summarizing, the main purposes of this paper are:

- presentation of the unified motion controller and the basic planner used to complement properties of the closed-loop control algorithm
- illustration of the control system designed for motion control of skid-steering vehicle in a practical application
- experimental validation of controller robustness to bounded disturbances resulting from skid/slip effects and other unmodelled dynamics

The paper is organized as follows. In Section II an approximated model of a skid-steering vehicle is introduced. Next section is devoted to design of the tracking controller. Next, motion planner is presented taking into account path planning algorithm and design of velocity profile. In Section IV implementation of the control system is outlined and results of experiments are shown. Section V concludes the paper.

II. Model description

A. Vehicle kinematics and equivalent unicycle model

Consider a skid-steering robot moving on a plane as shown in Fig. 1. Robot body configuration is described by $q := \begin{bmatrix} x & y & \psi \end{bmatrix}^\top \in \mathbb{R}^2 \times \mathbb{S}^1$, with x, y being position coordinates and ψ denoting the vehicle orientation determined in the inertial frame. Assume that $v = [v_x \; v_y]^\top \in \mathbb{R}^2$ and $\omega \in \mathbb{R}$ are linear and angular velocities, respectively, expressed in the local frame fixed to the robot body.

Considering propagation of velocities on the plane one can derive the fundamental planar kinematics model

$$\dot{q} = \bar{R}(\psi) \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{1}$$

where $\bar{R}(\psi) = \begin{bmatrix} R(\psi) & 0 \\ 0 & 1 \end{bmatrix} \in \mathrm{SO}(3)$, while $R(\psi) \in \mathrm{SO}(2)$. Next, let us introduce slip angle $\delta \in \mathbb{S}^1$ and variable $\nu \in \mathbb{R}$ satisfying

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos\delta \\ \sin\delta \end{bmatrix} \nu. \tag{2}$$

Applying (2) to (1) gives the following equivalent unicycle-like kinematics

$$\dot{g} = \begin{bmatrix} X_1(g) & X_2 \end{bmatrix} u = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u, \tag{3}$$

where $g := \begin{bmatrix} x & y & \theta \end{bmatrix}^\top \in \mathbb{R}^2 \times \mathbb{S}^1$ is the transformed configuration with

$$\theta := \psi + \delta, \tag{4}$$

being an auxiliary orientation. Terms $X_1(g)$ and $X_2(g)$ correspond to control vector fields, while

$$u := \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \nu \\ \omega + \dot{\delta} \end{bmatrix} \tag{5}$$

is a new control input – cf. [6]. It should be noted that input transformation (5) is well defined assuming that time derivative of slip angle is bounded, which can be denoted as $\dot{\delta} \in \mathcal{L}_\infty$.

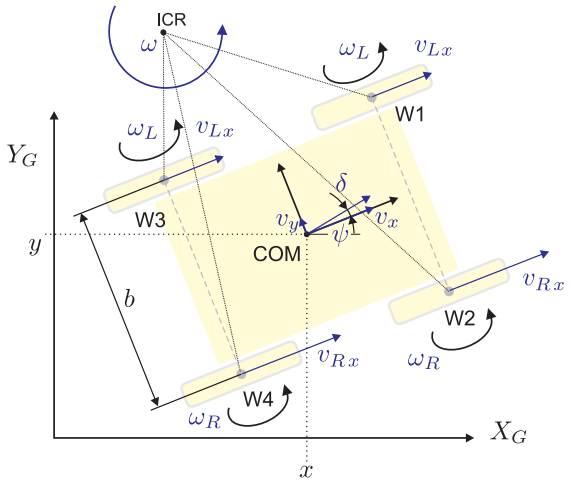

Fig. 1. Four wheeled skid-steering vehicle – planar kinematics.

Next, we investigate kinematics of the robot drive system. It is assumed that the vehicle is equipped with right and left pairs of mechanically coupled wheels rotating with angular velocities ω_R and ω_L, respectively. Thus, one can consider $w = \begin{bmatrix} \omega_R & \omega_L \end{bmatrix}^\top \in \mathbb{R}^2$ as a kinematic control input.

Comparing structure of this vehicle with the classic two-wheeled nonholonomic mechanism, it is clear that in the considered case straight lines collinear with wheel axes do not intersect. Hence, in order to guarantee sufficient mobility level of the robot body (cf. [10]), the conversion of rotary motion of wheels into linear motion is associated with the wheel slip phenomenon. To be more specific, linear velocities at contact points for each pair of wheels satisfy: $v_{Lx} = \sigma_L r \omega_L$ and $v_{Rx} = \sigma_R r \omega_R$, where r is the so-called effective radius of the wheels (for simplicity it is assumed that r is the same for each wheel) and σ_L, $\sigma_R \in \mathbb{R}$ are functions describing longitudinal slip. Recalling kinematics of a two-wheeled vehicle, one can find that velocities of the skid-steering vehicle are given by

$$v_x = \frac{r}{2}\left(\sigma_R \omega_R + \sigma_L \omega_L\right), \tag{6}$$

$$\omega = \frac{r}{b}\left(\sigma_R \omega_R - \sigma_L \omega_L\right). \tag{7}$$

Next, applying the following auxiliary slip functions

$$\forall \omega_R + \omega_L \neq 0, \; \sigma_v := \frac{\sigma_R \omega_R + \sigma_L \omega_L}{\omega_R + \omega_L},$$

$$\forall \omega_R - \omega_L \neq 0, \; \sigma_\omega := \frac{\sigma_R \omega_R - \sigma_L \omega_L}{\omega_R - \omega_L}.$$

to (6) and (7) allows one to introduce the following kinematics of the drive system

$$\begin{bmatrix} v_x & \omega \end{bmatrix}^\top = H(\sigma_v, \, \sigma_\omega) W w, \tag{8}$$

where $H(\sigma_v, \, \sigma_\omega) := \mathrm{diag}\{\sigma_v, \, \sigma_\omega\} \in \mathbb{R}^{2\times 2}$ describes slip effects, while $W := \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \in \mathbb{R}^{2\times 2}$ is dependent on geometric parameters of robot body.

Worth noting that lateral velocity v_y cannot be found explicitly based on input w. The similar statement concerns

also slip functions σ_v and σ_ω which are governed by the vehicle dynamics including a wheel-ground interaction model described in [11]. As shown in [12] and [4], this property implies that a skid-steering model cannot be globally determined at kinematic level.

Despite this issue, one can design a controller for system (1) taking advantage of unicycle model (3) as follows. Assume, that u_1 and u_2 are computed by the controller and v_y is known. Using (2) one has: $\delta = \arcsin(v_y/u_1)$ for $|u_1| \geq |v_y|$. Next, from (2) velocity v_x can be obtained. Taking time derivative of δ and using (5) one gets: $\omega = u_2 - \dot{\delta}$. Finally, given v_x and ω, one can compute wheels velocity w using inverse map of (8).

It should be emphasized that the considered control methodology has some limitations. Recalling that $|u_1| \geq |v_y|$ one can conclude that it can be applied when lateral slip is not significant, [3]. However, as a result of energy dissipation the lateral slip can be limited by decreasing of robot velocity, [13].

Following this idea, in this paper it is assumed that:

- slip functions σ_v and σ_ω are known and bounded,
- slip angle δ is known, v_y and $\dot{\delta}$ are bounded.

The correctness of those assumptions can be investigated based on dynamic model of the vehicle. This issue is out of scope of this paper.

B. Description on a Lie group

For the purposes of subsequent considerations, let us recall basic operators and terminology related to systems defined on a Lie group (cf. [14]). Lie groups serve as a tool for control design and generalized qualitative analysis of considered dynamical systems.

It is known that system (3) can be described on Lie group $G \sim \mathrm{SE}(2)$ with neutral element e and group operation given by

$$gh := g + \begin{bmatrix} R(\theta) & 0 \\ 0 & 1 \end{bmatrix} h,$$

where $R(\theta) \in \mathrm{SO}(2)$ and $g,\ h \in G$. Inverse of any element of $g \in G$ is represented by

$$g^{-1} = -\begin{bmatrix} R^\top(\theta) & 0 \\ 0 & 1 \end{bmatrix} g.$$

It can be shown that both $X_1(g)$ and X_2 are left invariant vector fields satisfying: $X_i(gh) = dl_g(h)\, X_i(h)$, where $i = 0, 1, 2$ and $dl_g(h)$ is a derivative of left translation defined by: $l_g(h) := gh$.

The tangent space of the group G can be determined by its Lie algebra $\mathfrak{g}$ with the basis constituted by X_1, X_2 and $X_3 = [X_2,\ X_1]$, where $[\cdot, \cdot]$ denotes the Lie bracket operator. Using vector-matrix notation the basis can be described by:

$$X(g) := [X_1\ X_2\ X_3](g) \in \mathbb{R}^{3\times 3}.$$

Taking advantage of X one can express any vector field $Z \in \mathfrak{g}$ evaluated at $g \in G$ in the Lie algebra basis as: $Z = X(g)\, z$, where $z \in \mathbb{R}^3$. Following this notation, kinematics (3) can be given by

$$\dot{g} = X(g)\, Cu,$$

where $C := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^\top \in \mathbb{R}^{3\times 2}$.

To facilitate algebraic notation we utilize a representation of Lie group G on its Lie algebra known as the adjoint representation and consider the following operator

$$Ad(g) := \frac{d}{d\tau}\left(g\tau g^{-1}\right)\Big|_{\tau=e} = \begin{bmatrix} R(\theta) & \left[\begin{smallmatrix} y \\ -x \end{smallmatrix}\right] \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}.$$

Assuming that this operator is expressed in the Lie algebra basis evaluated at neutral element e one can define: $Ad^X(g) := (X(e))^{-1} Ad(g) X(e)$.

III. Motion control and planning algorithm

A. Task description

In the sequel, we will develop motion control and planning algorithms dedicated for the waypoint following motion task. It is assumed that the sequence of prescribed waypoint configurations to be realized by the robot is generated by a high level planner or human operator. We are focused on a feedback control strategy driving the robot through subsequent waypoints.

We propose a two-stage algorithm composed of *a tracking closed-loop stabilizer* and *a trajectory planner* utilizing of polynomial splines. Hence, at the feedback control level, only the relatively simple trajectory tracking problem must be considered, because issues of acceleration/velocity constraints and skid/slip limiting are considered during the trajectory planning stage.

In general, an *admissible* reference trajectory $g_r(t) \in G$ can be defined by

$$\dot{g}_r = X(g_r)\, Cu_r, \tag{9}$$

where $u_r = [u_{r1}\ u_{r2}]^\top$ is reference input.

B. Closed-loop tracking controller

Assume that $g_r(t) \in G$ is a reference trajectory satisfying Eq. (9) with bounded reference input u_r and its time derivative $\dot{u}_r$.

In order to determine tracking error $\tilde{g}$ one can take advantage that g and g_r are elements of Lie group G. Then, the error can be defined on this Lie group as follows

$$\tilde{g} = g_r^{-1} g. \tag{10}$$

Taking time derivative of (10) and using differential operators (the details can be found in [14], [15]) one can represent open-loop dynamics by

$$\dot{\tilde{g}} = X(\tilde{g})\left(Cu - Ad^X\left(\tilde{g}^{-1}\right) Cu_r\right). \tag{11}$$

To clarify the considered method of stabilization, assume that reference input $u_r = 0$ and $g_r = \text{const}$.

Proposition 1 (Simple non-asymptotic stabilizer)**.** *Applying the following control law*

$$u := -C^\top X(\tilde{g})^\top K\tilde{g}, \tag{12}$$

where K is a symmetric positive definite gain matrix, to system (11) *with $u_r \equiv 0$, ensures Lyapunov stability of the closed-loop system such that*

$$\forall \tilde{g}(0) \in G, \forall t > 0 \quad \|\tilde{g}(t)\| \leq \|\tilde{g}(0)\|, \tag{13}$$

where $\|\cdot\|$ denotes Euclidean norm.

Proof. Consider a positive-definite quadratic form $V := \frac{1}{2}\tilde{g}^\top K\tilde{g}$. Substituting (12) to (11) with $u_r \equiv 0$ yields in

$$\dot{\tilde{g}} = -X(\tilde{g})CC^\top X(\tilde{g})^\top K\tilde{g}. \tag{14}$$

Next, calculating $\dot{V}$ and recalling that $K = K^\top$ one has: $\dot{V} = \tilde{g}^\top K\dot{\tilde{g}}$. Then using the closed-loop dynamics (14) in $\dot{V}$ implies that $\dot{V} = -\tilde{g}^\top KHK\tilde{g}$, where

$$H := X(\tilde{g})CC^\top X(\tilde{g})^\top. \tag{15}$$

Noticing that $CC^\top \succeq 0$ is a positive semidefinite matrix and recalling that the matrix congruence in (15) preserves positive semidefiniteness, one has $H \succeq 0$. As a result $\dot{V} \leq 0$ is negative semidefinite which indicates that V is a non-increasing function. Hence, stability result (13) has been proved. □

It is worth emphasizing that the presented control law is general in the sense that it is not restricted to a particular system on a Lie group. Now, we extend our consideration to the trajectory tracking problem. It can be concluded from (11) that term Cu should compensate drift $Ad^X\left(\tilde{g}^{-1}\right)Cu_r$. In particular, when $\tilde{g} \to e$ one has $Ad^X\left(\tilde{g}^{-1}\right) \to I$, where I denotes identity matrix. Thus, for small tracking error control input $u \to u_r$ which is obvious result for a feasible reference trajectory.

In order to define globally stabilizing tracking controller we are focused on a specific case, when the Lie group structure satisfies properties defined in Section II-B for the unicycle-like vehicle.

Proposition 2 (Asymptotic tracking controller)**.** *Control law defined as follows*

$$u := C^\top\left(-X(\tilde{g})^\top K\tilde{g} + Ad^X(\tilde{g}^{-1})Cu_r\right) + u^*, \tag{16}$$

where

$$u^* := \left(\tilde{x}\sin\tilde{\theta} - \tilde{y}\cos\tilde{\theta}\right)\begin{bmatrix} 0 & -1 \\ \alpha\mathrm{sinc}\,\tilde{\theta} & 0 \end{bmatrix} u_r \tag{17}$$

with $K := \mathrm{diag}\{k_1, k_2, k_3\} \in \mathbb{R}^{3\times 3}$ being diagonal positive definite gain matrix, $\alpha > 0$ denoting a positive coefficient (or a bounded scalar function), and $\mathrm{sinc}\,\phi := \frac{\sin\phi}{\phi}$ standing for an analytic function for $\phi \in \mathbb{R}$, applied to unicycle-like kinematics (3) *ensures global asymptotic tracking for a persistently exciting reference trajectory g_r (i.e. u_{r1} in* (9) *should not vanish).*

Proof. Using (16) in (11) yields

$$\begin{aligned}\dot{\tilde{g}} =& -X(\tilde{g})CC^\top X(\tilde{g})^\top K\tilde{g} \\ &+ X(\tilde{g})\left(\left(CC^\top - I\right)Ad^X(\tilde{g}^{-1})\right)Cu_r + Cu^*\big).\end{aligned} \tag{18}$$

Define a Lyapunov function candidate $V := \frac{1}{2}\tilde{g}^\top P\tilde{g}$ with $P := \mathrm{diag}\{\alpha, \alpha, 1\} \in \mathbb{R}^{3\times 3}$. Time derivative of V is $\dot{V} = \tilde{g}^\top P\dot{\tilde{g}}$. Taking into account closed loop dynamics (18) one has

$$\begin{aligned}\dot{V} =& -\tilde{g}^\top PHK\tilde{g} + \tilde{g}^\top PX(\tilde{g})\left(\left(CC^\top - I\right)Ad^X(\tilde{g}^{-1})\right) \\ &\times Cu_r + Cu^*\big),\end{aligned} \tag{19}$$

where matrix H is defined by (15). In the considered case one has

$$H = \begin{bmatrix} \cos^2\tilde{\theta} & \sin\tilde{\theta}\cos\tilde{\theta} & 0 \\ \sin\tilde{\theta}\cos\tilde{\theta} & \sin^2\tilde{\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

which implies that for positive-definite matrices P and K matrix product $PHK \succeq 0$.

Next, after taking into account (17) and computing the following term

$$\begin{aligned}h :=& X(\tilde{g})\left((CC^\top - I)Ad^X(\tilde{g}^{-1})Cu_r + Cu^*\right) \\ =& \begin{bmatrix} -\sin^2\tilde{\theta} & \tilde{y} \\ \sin\tilde{\theta}\cos\tilde{\theta} & -\tilde{x} \\ \alpha(\tilde{x}\sin\tilde{\theta} + \tilde{y}\cos\tilde{\theta})\mathrm{sinc}\,(\tilde{\theta}) & 0 \end{bmatrix} u_r.\end{aligned} \tag{21}$$

one can prove that the second term in (19) becomes $\tilde{g}^\top Ph = \alpha\left(-\tilde{x}\sin^2\tilde{\theta}\sin\tilde{\theta} + \tilde{y}\sin\tilde{\theta}\cos\tilde{\theta}\right)u_{r1} - \alpha\tilde{\theta}\mathrm{sinc}\,\tilde{\theta}\left(-\tilde{x}\sin\tilde{\theta} + \tilde{y}\cos\tilde{\theta}\right)u_{r1} = 0$. Recalling proof of Proposition 1 it can be concluded that $\dot{V}$ is negative semidefinite. It implies that $V \in \mathcal{L}_\infty$ is bounded. Hence, tracking error $\tilde{g}$ is also bounded. Consequently, u_1 and $u_2 \in \mathcal{L}_\infty$ are bounded for bounded reference signals u_{r1} and u_{r2}. Then, it can be shown that $\ddot{V}$ is bounded when $\dot{u}_{r1}$ and $\dot{u}_{r2} \in \mathcal{L}_\infty$. As a result $\dot{V}$ is uniformly continuous. Next, after considering Barballat's lemma one can conclude that $\dot{V}$ converges to zero.

Referring to (20) it can be shown that $\tilde{g}^\top PH\tilde{g} \to 0$ implies that $\lim_{t\to\infty}\tilde{\theta}(t) = 0$ and $\lim_{t\to\infty}\tilde{x}(t) = 0$. Using this result in (21) and recalling closed-loop system (18) it can be stated that trajectory $\tilde{g}$ converges to invariant set such that for $\tilde{x} = 0$ and $\tilde{\theta} = 0$ and $u_{r1} \neq 0$ one has $\lim_{t\to\infty}\tilde{y} = 0$. □

From the stability proof it follows that the asymptotic tracking of admissible reference trajectory with $u_{r1} \neq 0$ is guaranteed when $\alpha > 0$ and K is positive definite. Unfortunately, the presented stability analysis does not give details concerning the convergence rate. However, the gains can be selected based on the linearized model. Consequently, exposing only linear terms in (18) and taking advantage of results (21) and (20) one can obtain:

$$\dot{\tilde{g}} = \begin{bmatrix} -k_1 & u_{r2} & 0 \\ -u_{r2} & 0 & u_{r1} \\ 0 & -\alpha u_{r1} & -k_3. \end{bmatrix}\tilde{g} + O(\tilde{g}^2), \tag{22}$$

where $O(\tilde{g}^2)$ denotes higher order terms. Investigating the characteristic polynomial of matrix in (22) it can be found that it is similar to the controller considered in [8]. Based on this linearized model one can apply gain scheduling to achieve the desired performance of the controller in vicinity of the reference trajectory.

C. Motion planning algorithm

Similarly, *parking problem* understood as convergence of the robot posture to the desired constant position and orientation, can be solved implicitly taking advantage of the proposed methodology.

The controller presented in subsection III-B ensures asymptotic convergence assuming a persistent reference motion. Therefore it cannot be used directly to solve the waypoint following task, which implies piecewise constant reference configurations w.r.t. time. To cope with this limitation resulting from Brockett's obstruction [16] we use a local planner connecting initial robot configuration and subsequent waypoints by a C^2 (at least two times differentiable) admissible path. As a result, the waypoint following motion task comprising a sequence of set-point feedback control tasks is replaced by a task of tracking a persistently exciting trajectory. While in this case asymptotic stability of the closed-loop system cannot be guaranteed, it can be concluded that configuration error bounds depend on maximal disturbances acting on the system and initial tracking error minimized during motion planning stage. Thus, given relatively small disturbances, a configuration error low enough to be acceptable in practical conditions will be achieved.

To generate a persistently exciting admissible trajectory while minimizing the initial tracking error a simple local planner, which utilizes polynomial splines and takes into account prescribed velocity and acceleration bounds is proposed. The planning process consists of executing the same procedure for every subsequent pair of waypoints, beginning from the initial robot configuration, which is treated as the zeroth waypoint. This procedure is now described.

Assume that subsequent waypoint configurations are denoted by $g_{d(k-1)}$ and $g_{d(k)} \in G$, while T_k is the planned time of reaching k^{th} waypoint. Trajectory $g_r(t) \in G$ must be planned to satisfy the following conditions:

- A1 connectivity: $g_r(0) = q_{d(k-1)}$, $g_r(T_k) = q_{d(k)}$, where $T_k < \infty$ is a bounded time period
- A2 admissibility: $\forall t \in [0, T_k]$, $g_r(t)$ is a solution of Eq. (9)
- A3 boundedness of reference input u_r: it is assumed that $\forall t \geq 0$, $|v_{rx}| < U_1$ and $|\omega_r| < U_2$, where U_1 and U_2 are positive bounds
- A4 continuity of reference input: $\forall t \in [0, T_k] \quad u_{r1},\ u_{r2} \in C^1$

In order to satisfy those conditions trajectory $g_r(t)$ is designed in two stages. Firstly, a suitable path $q_d(s)$ taking into account assumptions A1 and A2 is found. Secondly, a time scaling procedure is used to determine path parametrization $s(t) \in [0,\ 1]$.

1) Path design: Let $\bar{g}_d$ be auxiliary configuration on the Lie group G considered between two points:

$$\bar{g}_d = g_{d(k)}^{-1} g_{d(k-1)}. \tag{23}$$

Next, define a path $\tilde{g}_d(s)$ parametrized by $s \in [0,\ 1]$ such that

$$\tilde{g}_d(0) = \bar{q}_d \text{ and } \tilde{q}_d(1) = e. \tag{24}$$

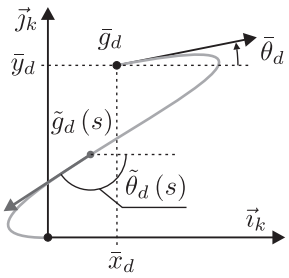

Fig. 2. A path segment connecting two waypoints defined in the local coordinate frame.

Then

$$g_d(s) = g_{d(k)} \tilde{g}_d(s). \tag{25}$$

connects subsequent points $g_{d(k-1)}$ and $g_{d(k)}$.

To simplify notation we define

$$\tilde{g}_d(s) := \left[\tilde{p}_d^{\top}(s)\ \ \tilde{\theta}_d(s)\right]^{\top}, \tag{26}$$

where $\tilde{p}_d(s) := \left[\tilde{x}_d(s) \quad \tilde{y}_d(s)\right]^{\top}$ denotes position variables.

Path connecting subsequent waypoints is constructed base on a 5^{th}-order polynomial $\gamma(s,a) = \sum_{i=0}^{5} a_i s^i$, where $a = \begin{bmatrix} a_0 & a_1 & \dots & a_5 \end{bmatrix}^{\top} \in \mathbb{R}^6$ denotes coefficients determining its shape. The first and second order derivatives of $\gamma(s,a)$ are denoted by $\gamma'(s,a)$ and $\gamma''(s,a)$, respectively. In the considered planning method it is assumed that the position path satisfies: $\tilde{p}_d(s) := [\gamma(s,a_x)\ \gamma(s,a_y)]^{\top}$, where a_x and $a_y \in \mathbb{R}^6$ are parameters, which can be computed for a given set of boundary conditions. As a result of (24) and assumption A1 we have:

$$\tilde{p}_d(0) = \begin{bmatrix} \bar{x}_d & \bar{y}_d \end{bmatrix}^{\top} \text{ and } \tilde{p}_d(1) = 0. \tag{27}$$

Considering assumption A2 and Fig. 2 one can conclude that a vector tangent to the path $\tilde{p}_d(s)$ at $s = 0$ and $s = 1$ should be properly oriented. The tangent conditions become

$$\tilde{p}_d'(0) = \begin{bmatrix} v_0 \cos\bar{\theta}_d & v_0 \sin\bar{\theta}_d \end{bmatrix}^{\top} \text{ and } \tilde{p}_d'(1) = \begin{bmatrix} v_f & 0 \end{bmatrix}^{\top}, \tag{28}$$

with v_0 and v_f being positive tuning parameters.

In order to guarantee continuity of the reference angular velocity, the path curvature should be properly shaped. In order to facilitate construction of a complex path consisting with more than one segment it is assumed that the curvature of a local path segment tends to zero at waypoint configurations. This assumption can be represented as follows:

$$\tilde{p}_d''(0) = \tilde{p}_d''(1) = 0. \tag{29}$$

Taking into account (27)-(29) one can compute polynomial parameters as follows: $\begin{bmatrix} a_x^{\top} & a_y^{\top} \end{bmatrix}^{\top} = P^{-1} Y$, where $P \in \mathbb{R}^{12\times 12}$ is the constant invertible regression matrix, while $Y := \begin{bmatrix} \tilde{p}_d^{\top}(0) & \tilde{p}_d^{\top}(1) & \tilde{p}_d^{\top}(0) & \tilde{p}_d^{\top}(1) & 0 \end{bmatrix}^{\top} \in \mathbb{R}^{12}$. Having polynomial parameters one can obtain $\tilde{p}_d(s)$ for $s \in [0,1]$ and compute other path-related variables.

Recalling admissibility condition for the path, the orientation variable and auxiliary inputs values along the path

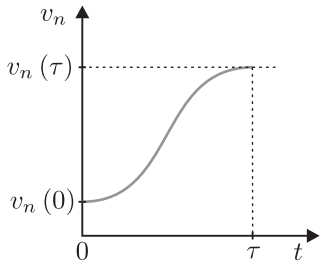

Fig. 3. Velocity profile proposed for trajectory planning.

can be computed using differential flatness. Consequently, the orientation satisfies

$$\tilde{\theta}_d(s) = \operatorname{atan2}\left(\mu\tilde{y}_d'(s), \mu\tilde{x}_d'(s)\right), \tag{30}$$

where $\operatorname{atan2}(\cdot,\cdot)$ stands for the four-quadrant inverse tangent function and $\mu \in \{-1, 1\}$ is a parameter defining motion strategy (forward $\mu := 1$ or backward $\mu := -1$). Tangent velocity along the path is given as

$$v_d(s) = \left\|\tilde{p}_d'(s)\right\|, \tag{31}$$

while nominal angular velocity satisfies

$$\omega_d(s) = \left(\tilde{y}_d''(s)\,\tilde{x}_d'(s) - \tilde{x}_d''(s)\,\tilde{y}_d'(s)\right)/v_n^2(s), \tag{32}$$

where $v_n \neq 0$.

The path planning process ends by computing path coordinates in the given inertial frame referring to (25).

2) Time parametrization: The velocity profile is designed taking into account tangent velocity on the path $\tilde{p}_d(s)$. It is assumed that nominal velocity v_n changes monotonically in the assumed time horizon $\tau > 0$ and satisfies

$$t \in [0,\ \tau]\ v_n(t) := \frac{v_n(\tau)-v_n(0)}{2}\left(1-\cos\left(\frac{\pi}{\tau}t\right)\right) + v_n(0), \tag{33}$$

where $v_n(0) \leq U_1$ and $v_n(\tau) \leq U_1$ are desired positive values – cf. Fig. 3. Integrating Eq. (33) one can calculate the following distance

$$l_\tau = \int_0^\tau v_n(t)\,dt = \tau\left(\frac{v_n(\tau)+v_n(0)}{2}\right). \tag{34}$$

Equation (34) can be used to find time τ for the assumed path length. It gives possibility to distinguish path fragments where the reference vehicle should start or stop gradually. From (34) one has

$$\tau = \frac{2l_\tau}{v_n(\tau)+v_n(0)}.$$

Let $t_s \geq 0$ be an increasing function such that $\frac{dt_s}{dt} > 0$. In order to calculate $s(t_s)$ we compare nominal velocity (33) with tangent velocity (31) and introduce the following scaling function $\xi_1(t_s) := \frac{v_n(t_s)}{v_d(s(t_s))}$. Next, for the given $v_n(t_s)$ we scale value of angular velocity given by (32) and check if the following inequality holds (cf. assumption A4): $\xi_1\left|\omega_d(s(t_s))\right| \leq U_2$. Define

$$\xi_2(t_s) := \max\left\{\xi_1\frac{\left|\omega_d(s(t_s))\right|}{U_2}, 1\right\}$$

and calculate $t_s(t) = \int_0^t \xi_2^{-1}(\zeta)\,d\zeta$, and $s(t_s) = \int_0^{t_s}\frac{\xi_1(\zeta)}{\xi_2(\zeta)}d\zeta$. Finally, the reference configuration is given by $g_r(t) := g_d(s(t_s(t)))$ while the reference velocities become

$$v_r(t) = \mu\frac{v_n(t_s(t))}{\xi_2(t_s(t))}, \tag{35}$$

and

$$\omega_r(t) = \frac{\xi_1(t_s(t))\,\omega_d(s(t_s(t)))}{\xi_2(t_s(t))}.$$

Recall that parameter μ in (35) is used to determine the sign of linear velocity.

IV. Experimental studies

The presented control system has been implemented in C++ language using ROS Hydro. The skid-steering platform RoKSIS used in experiments is presented in Fig. 4. The robot is driven by two Maxon motors governed by EPOS2 drivers connected to the on-board PC via CAN bus. The vehicle is localized using on-board camera uEye UI-1221LE-C-HQ 0.36 MPix equipped with 50 deg diagonal lens which records artificial passive markers with QR code fixed to the ceiling in the laboratory room – see Fig. 5. The vision system is implemented based on ArUco library, [17]. In order to improve properties of localization system a sensory fusion between data provided by camera and an internal predictor is realized. The whole system was integrated using ROS complemented by a specialized KSIS software framework providing: automatically verified system composition, building blocks for motion control/planning algorithm implementations and push-based low-latency communication between critical controller components (see [18] for details).

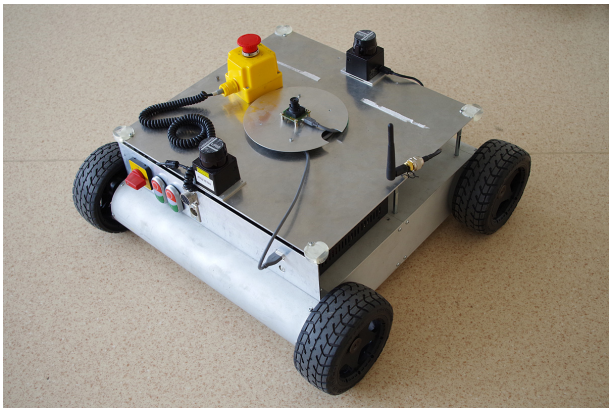

Fig. 4. Experimental four-wheeled SSMR robot RoKSIS with wheelbase width of 0.405 m and axes within 0.52 m.

Kinematic model (8) has been identified experimentally. Vehicle motion has been analysed in stationary conditions (wheel velocities defined by slow time-varying functions) along trajectories with various curvatures. Based on this analysis it was noticed that the slip functions are almost constant parameters given by: $\sigma_v = 1.1$ and $\sigma_\omega = 0.66$. Moreover, for considered paths lateral velocity v_y approaches zero, which

Fig. 5. Code markers used for absolute localization arranged into a square grid spaced by 1.5m

TABLE I
SELECTED PARAMETERS OF TRAJECTORY SEGMENTS.

i	j	v_0	v_f	$v_n\|_{P_i}$	$v_n\|_{P_j}$	direction
0	1	3	3	0	0	backward
1	2	1	2	0	U_1	forward
2	3	1	2	U_1	U_1	forward
4	0	1	2	U_1	0	forward

indicates that $\delta \approx 0$. As a consequence, it has been assumed that $\psi \approx \theta$, $v_x \approx u_1$, and $\omega \approx u_2$.

The proposed algorithm was verified experimentally for a selected scenario with waypoints chosen as follows (values are given in SI units): $g_{d(0)} = [0 \ -0.2 \ 0]^\top$, $g_{d(1)} = [1 \ -0.2 \ \pi/2]^\top$, $g_{d(2)} = [1 \ 1.2 \ \pi]^\top$, $g_{d(3)} = [0 \ 1.2 \ -\pi/2]^\top$ and $g_{d(4)} = g_{d(0)}$. Bounds of linear and angular velocities are given by coefficients U_1 and U_2, respectively. It is assumed that in the middle of each path segment nominal velocity $v_n = U_1$ while at the beginning and the end of the path segment nominal velocity is given by $v_n|_{P_i}$ and $v_n|_{P_j}$, respectively. The detailed parameters for each segment are collected in Table 1.

Gains of the controller have been chosen as follows: $K = \mathrm{diag}\{4, 8, 4\}$ and $\alpha = 10$, while frequency of the control loop has been set to $f_s = 30$ Hz. Various bounds of reference velocities have been considered (U_1 [m/s], U_2 [rad/s]): E1 – $U_1 = 0.2$ and $U_2 = 0.5$, E2 – $U_1 = 0.5$ and $U_2 = 0.8$, E3 – $U_1 = 1$ and $U_2 = 1$.

Both reference and obtained paths are presented in Fig. 6. Curvature of the reference path is bounded which should limit sharp robot turns in order to attenuate disturbances from skid/slip phenomena. Obtained results show that robot's path resembles the reference path closely especially when the path curvature is not significant. Accordingly, it can be noticed that tracking accuracy is deteriorated when the path curvature increases. This issue is a composition of two predominant effects. Firstly, for higher velocities slip phenomenon is more apparent and accuracy of the assumed kinematic model of the skid-steering robot can be significantly affected. It results in a disturbance introduced into the control-loop and decreased precision of estimator used for localization. Secondly, when the robot moves with higher velocities, the quality of vision based localization is also deteriorated.

The time plots of tracking errors are presented in Fig. 7. It can be seen they remain bounded during the given time horizon. It is evident that they increase with velocity. In particular, for higher speed of the vehicle the orientation error becomes significant. Comparing position errors one can state that magnitude of lateral component $\tilde{y}$ is typically higher. This is a result of phase constraints imposed on the vehicle and properties of the tracking controller. Additionally, it can be seen that at the final stage of control when the reference motion is stopped $\tilde{x}$, $\tilde{\theta}$ tend to values near zero, while $\tilde{y}$ remains in some vicinity of zero. This is a manifestation of Lyapunov non-asymptotic stability guaranteed by the controller when the reference trajectory is degenerated to a constant point.

The reference and desired velocities of the robot in experiment E3 are illustrated in Fig. 8. The continuous transition from stop phase to motion with maximum velocity can be easily seen. Moreover, one can observe how the reference velocities are profiled in order to satisfy the assumed constraints. From Fig. 8 follows that velocities calculated by the controller correspond to reference inputs u_{r1} and u_{r2}. In spite of that, higher disturbances can be observed with respect to angular velocity – it can be also confirmed by analysing orientation error presented in Fig. 7. In particular, the most difficult conditions are met when product $|u_{r1}u_{r2}|$ increases – then slip angle δ becomes more significant.

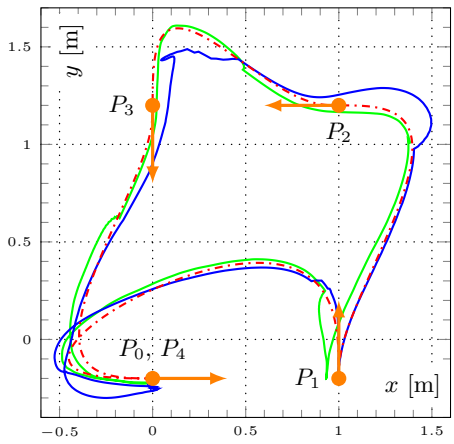

Fig. 6. Selected waypoints and paths in the Cartesian space: reference path (■), robot path in E1 (■) and E3 (■).

V. CONCLUSIONS

In this paper the unified controller for the skid-steering vehicle was presented. Developed algorithm has been applied in practice to control of laboratory mobile robot RoKSIS. The obtained results indicate that the presented control solution can be effectively used in robotic applications. Additionally, it can be concluded that a simple model of the skid-steering vehicle based on unicycle kinematics is sufficient for motion control in a limited range of velocities.

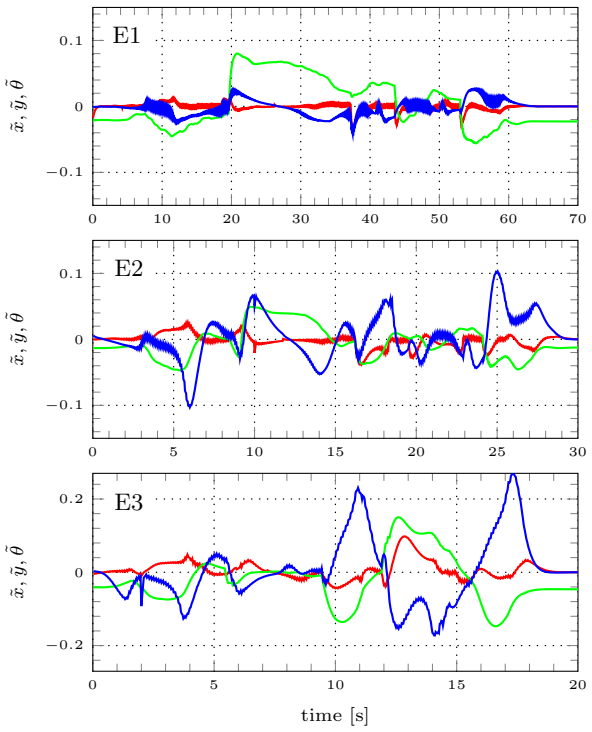

Fig. 7. Experiment E1-E3 – tracking errors for various limits of reference velocities: $\tilde{x}$ in [m] (■), $\tilde{y}$ in [m] (■), $\tilde{\theta}$ in [rad] (■).

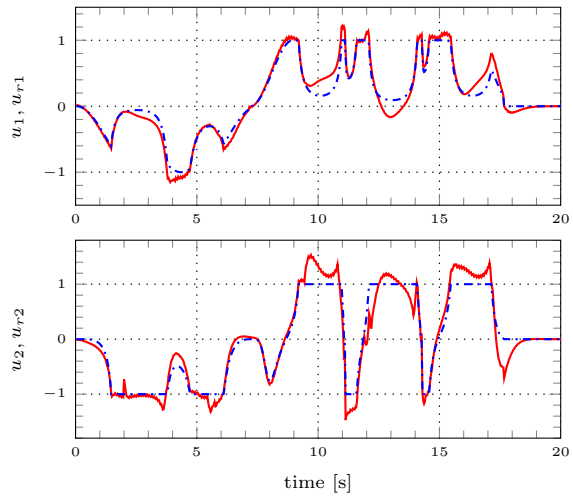

Fig. 8. Experiment E3 – computed (■) and reference (■) velocities : u_1, u_{r1} in [m/s], u_2, u_{r2} in [rad/s].

Presented control solution is relatively robust to external disturbances. It was shown that in spite that a closed-loop controller does not guarantee asymptotic stability at a constant reference configuration, the convergence to this configuration can be achieved with satisfactory accuracy by tracking a properly designed auxiliary trajectory. In such a case one can obtain some kind of practical stability.

Future work will be devoted to extension of the proposed approach. In particular, the planner will be enhanced by incorporating velocity fields to realize selected behavioural motion patterns. The problem of obstacle collision will be also taken into account.

We believe that performance of the presented control solution can be further improved for example by introducing more advanced estimation techniques to partially compensate negative influence of skid/slip phenomena.

REFERENCES

[1] K. Zadarnowska and A. Ratajczak, "Task-priority motion planning of wheeled mobile robots subject to slipping," in *Robot Motion and Control 2011*, ser. Lecture Notes in Control and Inform. Sci., K. Kozłowski, Ed. Springer London, 2012, vol. 422, pp. 75–85.

[2] M. Cholewiński, K. Arent, and A. Mazur, "Towards practical implementation of an artificial force method for control of the mobile platform Rex," in *Recent Advances in Automation, Robotics and Measuring Techniques*, ser. Advances in Intelligent Systems and Computing, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Springer International Publishing, 2014, vol. 267, pp. 353–363.

[3] L. Caracciolo, A. De Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, May 1999, pp. 2632–2638.

[4] D. Pazderski and K. Kozłowski, "Trajectory tracking of underactuated skid-steering robot," in *Proc. Amer. Contr. Conf.*, 2008, pp. 3506–3511.

[5] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *Proc. IEEE/RSJ Int. Conf. of Intelligent Robots and Systems (IROS)*, Oct 2007, pp. 1222–1227.

[6] D. Wang and C. B. Low, "Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 676–687, June 2008.

[7] M. Michałek and K. Kozłowski, "Motion planning and feedback control for a unicycle in a way point following task: The VFO approach," *Int. J. Appl. Math. Comput. Sci.*, vol. 19, no. 4, pp. 533–545, 2009.

[8] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sørdalen, "Nonlinear control design for mobile robots," in *Recent Trends in Mobile Robots*, Y. F. Zheng, Ed. Singapore: World Scientific, 1993, vol. 11, pp. 121–156.

[9] K. Simba, N. Uchiyama, and S. Sano, "Real-time trajectory generation for mobile robots in a corridor-like space using Bézier curves," in *Proc. IEEE/SICE Int. Symp. System Integration*, Dec 2013, pp. 37–41.

[10] G. Campion, G. Bastin, and B. d'Andréa Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots." *IEEE Trans. Robot. Autom*, vol. 12, no. 1, pp. 47–62, 1996.

[11] H. B. Pacejka, *Tire and Vehicle Dynamics*. SAE International, 2006.

[12] A. D. Lewis, "When is a mechanical control system kinematic?" in *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona, USA, December 1999, pp. 1162–1167.

[13] K. Kozlowski and D. Pazderski, "Practical stabilization of a skid-steering mobile robot-a kinematic-based approach," in *Proc. 3rd IEEE Int. Conf. on Mechatronics*, 2006, pp. 519–524.

[14] P. Morin and C. Samson, "Control of nonholonomic mobile robots based on the transverse function approach," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 1058 –1073, oct. 2009.

[15] D. Pazderski, K. Kozłowski, and D. Waśkowicz, "Control of a unicycle-like robot with trailers using transverse function approach," *Bull. Pol. Ac.: Tech.*, vol. 60, no. 3, pp. 537–546, 2012.

[16] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, R. W. Brockett, R. S. Millman, and H. J. Sussmann, Eds. Boston: Birkhäuser, 1983, pp. 181–191.

[17] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[18] T. Gawron. (2015) Ksis framework – control system middleware. [Online]. Available: http://ksisframework.tomaszgawron.pl

View publication stats