ViCoMoR 2014

3rd Workshop on Visual Control of Mobile Robots (ViCoMoR)

Half Day Workshop

September 18, 2014, Chicago, Illinois, USA
in conjunction with the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)

http://vicomor.unizar.es

Organizers

Youcef Mezouar
IFMA, Clermont Université - Institut Pascal, France
Gonzalo López-Nicolás
I3A - Universidad de Zaragoza, Spain

Contents

Web: http://vicomor.unizar.es
3rd Workshop on Visual Control of Mobile Robots (ViCoMoR)
September 18, 2014, Chicago, Illinois, USA, in conjunction with the
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)

The organization of this workshop was supported by Ministerio de Economía y Competitividad and European Union (projects DPI2012-32100 and DPI2012-31781), DGA-FSE (T04), and LabEx IMobS3 (ANR-10-LABX-16-01).

Aims and Scope

Among the variety of sensors available today, vision systems stand out because they provide very rich information at low cost. One of the main reasons for integrating vision in the control loop was the interest for increased flexibility of robotic systems. However, versatility of vision systems comes at the cost of higher data processing complexity. Visual control or visual servoing has been one of the major research issues in robotics for more than four decades. Although control theory and computer vision are both mature areas of research, important advances that bring new challenges are happening nowadays such as the advent of RGB-D cameras, the use of omnidirectional vision, or the development of robust control techniques.

The interest in this subject lies in the many potential robotic applications in industrial as well as in domestic settings that involve visual control of mobile robots. In this context, autonomous navigation and integration of vision in the control loop is still an open and ambitious research area. Thus, the purpose of this workshop is to bring together researchers with common interests in computer vision and the control of robots, and discuss topics related to the challenging problems of visual control of mobile robots.

Topics

Topics of interest include:

– Autonomous navigation and visual servoing techniques for mobile robots.

– Visual perception for visual control, visual sensors and integration of image information in the control loop.

– Visual control with constraints: nonholonomic constraints, motion in formation, distributed visual control, obstacle avoidance, etc.

– New trends in visual control, innovative solutions or proposals in the framework of computer vision and control theory.

Program committee

Nicolas Andreff	- Université de Franche-Comté, France
Helder Araujo	- ISR, University of Coimbra, Portugal
Antonis Argyros	- FORTH, Heraklion, Greece
Hector M. Becerra	- CIMAT, Guanajuato, Mexico
Enric Cervera	- Universitat Jaume-I, Spain
François Chaumette	- INRIA Rennes - IRISA, France
Andrea Cherubini	- LIRMM, University of Montpellier, France
Peter Corke	- Queensland University of Technology, Australia
Francisco Escolano	- Universidad de Alicante, Spain
Nicholas R. Gans	- University of Texas at Dallas, USA
Andrea Gasparri	- Università degli Studi Roma Tre, Roma, Italy
Jose J. Guerrero	- Universidad de Zaragoza, Spain
Koichi Hashimoto	- Tohoku University, Sendai, Japan
Seth Hutchinson	- University of Illinois at Urbana-Champaign, USA
Nicolas Mansard	- LAAS/CNRS, France
Philippe Martinet	- Ecole Centrale de Nantes, France
Roberto Naldi	- Università di Bologna, Italy
Cedric Pradalier	- GeorgiaTech Lorraine, France
Patrick Rives	- INRIA Sophia Antipolis, France
Carlos Sagues	- Universidad de Zaragoza, Spain
Davide Scaramuzza	- University of Zurich, Switzerland
Omar Tahri	- LE2I, University of Burgundy, France
Dimitris P. Tsakiris	- FORTH, Heraklion, Greece
Andrew Vardy	- Memorial University of Newfoundland, Canada
Xenophon Zabulis	- FORTH, Heraklion, Greece

Organizers

Youcef Mezouar
Clermont Université, IFMA, Institut Pascal,
BP 10448, F-63000 Clermont-Ferrand, France
CNRS, UMR 6602, IP, F-63171 Aubière, France
Email: youcef.mezouar@ifma.fr
Web: http://maccs.univ-bpclermont.fr/index.php/Profiles/MezouarY/

Gonzalo López-Nicolás
Instituto de Investigación en Ingeniería de Aragón - Universidad de Zaragoza
María de Luna 1, E-50018 Zaragoza. Spain
Email: gonlopez@unizar.es
Web: http://webdiis.unizar.es/~glopez

Instituto Universitario de Investigación
en Ingeniería de Aragón
Universidad Zaragoza

Invited speakers

Peter Corke, Aaron McFadyen, Inkyu Sa

Queensland University of Technology, Australia

Title: Image-based visual servoing for air vehicles

Abstract: This talk will discuss two applications of image-based visual servoing for air vehicles. First is an all-onboard system for close-quarters flying with respect to a pole-like structure, motivated by problems in infrastructure inspection for non-skilled pilots. The second is the case of collision avoidance for unmanned fixed wing UAVs, where IBVS control leads to safe avoidant motion along a conical spiral.

Helder Araujo

ISR, University of Coimbra, Portugal

Title: Pose from Images of Generalized Cameras. Applications in Navigation

Abstract: The use of generic camera models allows a unified treatment of all types of cameras. Non-central cameras are useful not only as a result of the fields of view provided, but also because of geometrical relationships between the images and the 3D objects. In particular navigation of robots can benefit from more stable and robust pose estimation methods. In this talk we present pose estimation methods for generic camera models that can be applied advantageously to some mobile robotic applications.

Elias Müggler, Davide Scaramuzza

University of Zurich, Switzerland

Title: Appearance-based, Active Vision applied to Autonomous Mapping from Micro Aerial Vehicles

Abstract: The problem of computing the optimal views to reconstruct an object or a scene has been studied for almost 30 years and is known in the robotics and computer vision communities as View Path Planning or Next Best View. The limitation of state-of-the-art approaches is that they only retain geometric information of the scene while discarding the scene appearance (i.e., texture). They work by maximizing the expected information gain on the basis of the robot pose and the current map. Instead, we propose to maximize the information gain on the basis of the robot pose, the map, as well as the "appearance" of the scene. We successfully demonstrate our approach in the case of realtime, monocular reconstruction from a micro aerial vehicle.

Ryan W. Wolcott, Ryan M. Eustice

University of Michigan, USA

Title: Visual localization for automated urban driving

Abstract: This talk reports on the problem of map-based visual localization in urban environments for autonomous vehicles. Self-driving cars have become a reality on roadways and are going to be a consumer product in the near future. One of the most significant road-blocks to autonomous vehicles is the prohibitive cost of the sensor suites necessary for localization. We propose to localize a single monocular camera within a 3D prior groundmap, generated by a survey vehicle equipped with 3D LIDAR scanners.

Program ViCoMoR 2014

September 18, 2014 - Morning session (8:30-12:00)

8:30

Pose from images of generalized cameras. Applications in navigation
Helder Araujo (Invited talk)

9:00

Appearance-based, active vision applied to autonomous mapping from micro aerial vehicles
Elias Muggler, Davide Scaramuzza (Invited talk)

9:30

Backstepping and sliding-mode techniques applied to an underactuated camera onboard a rotorcraft MAV
Jesus G. Villagomez, Manuel Vargas, Francisco R. Rubio

9:45

Path following: from mobile robotics to laser surgery
Jean-Antoine Seon, Brahim Tamadazte, Nicolas Andreff

10:00 - Coffee break

10:30

Image-based visual servoing for air vehicles
Peter Corke (Invited talk)

11:00

Visual localization for automated urban driving
Ryan W. Wolcott and Ryan M. Eustice (Invited talk)

11:30

Embedded vision-based localization and model predictive control for autonomous exploration
Helene Roggeman, Julien Marzat, Martial Sanfourche, Aurelien Plyer

11:45

Planar motion estimation from three-dimensional scenes
Xuebo Zhang, Congyuan Wang, Yongchun Fang, Kexin Xing

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2014)
September 18, 2014, Chicago, Illinois, USA

Backstepping and Sliding-mode Techniques Applied to an Underactuated Camera Onboard a Rotorcraft MAV*

Jesús G. Villagómez[1], Manuel Vargas[2] and Francisco R. Rubio[3]

***Abstract*— This paper presents the modeling and control of a multi-body air vehicle composed of a miniature aerial vehicle (MAV) and mechanical device, a camera positioner, augmented aerial platform which is considered as the logical next framework within the MAV civil applications. The goal is to improve the current operational profile of visual sensors onboard the MAV, by broading current aerial configurations for visual sensors tasks with novel capabilities to disengage the dynamic coupling due to typical setups. The Euler-Lagrange formalism is applied aiming at obtaining coupling terms between the aerial and the camera positioner systems. Results of two nonlinear control techniques applied to the MAV are presented. A backstepping and a sliding-mode techniques are first designed and then numerically simulated. Finally, we discuss the results of each approach.**

I. INTRODUCTION

The development and application of Miniature Air Vehicles (MAV) has been growing during last years, especially in military scopes, where MAVs have been used intensively for security purposes and reconnaissance flights during a conflict. Its low cost profile with respect to traditional manned aircraft and a wide range of applicability shows that these platforms are very useful tools to provide an extension of the environment perception to the operator, by means of exteroceptive sensors or cameras, for data collecting and later analysis or online area surveillance. This kind of aerial platforms provides an excellent alternative due to its operational functionalities as vertical take-off and landing (VTOL), maneuverability and hovering [1]. But due to its inherent limitations, especially if the payload is quite constrained, reduce dramatically the range of electromechanical devices to be attached to the frame to enhance its current functional profile. Works where the modeling of a system compounded by an industrial manipulator and an aerial platform have been recently presented [2][3]. Whereas this concept is quite promising, and lets the system to interact with the environment for cargo transportation or aerial grasping, represents new challenges in control theory, e.g. couplings between both mechanical system dynamics arises or the system payload may dynamically vary. Nevertheless, it is possible to consider other applications where it would be necessary to use an augmented platform, e.g. the use of an actuated optical sensor onboard the MAV. Computer vision has been used widely in MAVs, but most of previous works considered only the visual information provided by a camera rigidly attached to the airframe [4], for control purposes [5]-[6] or civil applications [7]. What is presented in this work is the addition of a 2 degrees of freedom (DOF) positioner camera jointly with a camera to an underactuated MAV, to achieve the 6 DOF control in the orientation subsystem of the camera frame. Mechanically there are two ways in which it can be done: the former consists on a $tilt - roll$ camera attached to the front of the MAV, and the later consists on adding a $pan - tilt$ camera to the bottom of the airframe, which is considered in this proposal. In most of the works concerning the use of an actuated camera exists at least one camera operator to control the visual sensor attitude [8]. In such cases the camera positioning problem is divided into two coupled control problems. Once the aircraft reaches its initial position, hovering or translating avoiding obstacles, guidance should be achieved in such a way that the camera could get its initial reference frame and see the object scene. Concurrently, the camera operator must compensate for the movements of the aircraft and reject these perturbations while keeps trying to track the interest object. By combining the 3 translational DOF plus the yaw angle provided by the motion of a quad-rotor with the 2 rotational DOF supplied by a camera positioner, or camera *gimbal*, it is possible to position the six degrees of freedom of the onboard camera. On this operational mode, also known as "fly-the-camera", references will be given from the visual system, which tracks and keeps the interest object's path with commands generated in the camera field-of-view (FOV) to move both devices. Previous works already considered this alternative, but only to improve the camera operator experience [9]. In contrast, what we propose is the full automation of the compounded system where references will be given from the camera frame's perspective and no camera operator must be considered. The outline of this work is organized as follows: section II describes the modeling of a camera positioner onboard a MAV through the motion in a plane of the quad-rotor and the camera positioner. In section III the problem statement is introduced and some assumptions for the control task are presented. Sections IV and V the nonlinear control techniques Backstepping and Sliding-mode are under consideration, respectively, and some numerical simulations are consequently introduced. Final remarks and perspectives are given in section VI.

*This work was partially supported by the spanish Ministry of Education (MECD) under national research projects $DPI2012-37580-C02-02$ and $DPI2013-44135-R$.

[1] Jesús G. Villagómez is with Department of Systems Engineering and Automation. University of Seville, Spain villagomez@us.es

[2] Manuel Vargas is with Department of Systems Engineering and Automation. University of Seville, Spain vargas@us.es

[3] Francisco R. Rubio is with Department of Systems Engineering and Automation. University of Seville, Spain rubio@us.es

II. SYSTEM MODELING

Our approximation for modeling the mechanics of the system is focused on the Planar Vertical Take Off and Landing (PVTOL) aircraft with the camera positioner in configuration $pan{-}tilt$, in the vertical plane. This model will be called tPVTOL ($tilt$ PVTOL), since the pan angle is fixed by definition. Consider a 1 DOF camera positioner system evolving within the plane, thanks to the flight of a miniature rotorcraft, which it is rigidly mounted to. This camera actuator is intended to position the camera in a desired angular position. Such positioning tasks are performed during near-hovering or in-motion maneuvers. The camera positioner's angular position and rates will be measured by an Inertial Measurement Unit (IMU) attached to the camera body. This configuration is considered as a multi-body mechanical system where the simultaneous operation generates dynamic couplings. Accordingly, based on weight of both the camera and the camera actuator and involved links lengths, the in-flight camera positioning operation could shift the rotorcraft center of gravity (CoG), generating reaction forces and consequently disturbing the inertial camera orientation.

A. Reference Frames and Notation

The following dynamic model of the aircraft with the camera positioner is presented for the case where the camera is pointing forwards (*tilt* $\backsim 0^o$), although the model is also valid for other setups, such as camera pointing downwards (*tilt* $\backsim 90^o$). The kinematics of the flying system comprises three right-handed reference coordinate systems [10].

- Let: $(\hat{x}_{\mathcal{W}}, \hat{y}_{\mathcal{W}}, \hat{z}_{\mathcal{W}})$, which defines the fixed inertial frame $\mathcal{W}$, whose origin $\mathcal{O}_{\mathcal{W}}$ is located at the Earth surface. For the planar case, the vector basis becomes $(\hat{x}_{\mathcal{W}}, 0, \hat{z}_{\mathcal{W}})$.
- Let: $(\hat{x}_{\mathcal{B}}, \hat{y}_{\mathcal{B}}, \hat{z}_{\mathcal{B}})$ be the body-fixed frame $\mathcal{B}$, whose origin $\mathcal{O}_{\mathcal{B}}$ corresponds to the geometric center of the quadrotor. Similarly, for the longitudinal case, the vector basis becomes $(\hat{x}_{\mathcal{B}}, 0, \hat{z}_{\mathcal{B}})$.
- Let $(\hat{x}_c, \hat{y}_c, \hat{z}_c)$ be the body-fixed frame $\mathcal{C}$, whose origin $\mathcal{O}_c$ corresponds to the center of the camera frame. The vector basis becomes $(\hat{x}_c, 0, \hat{z}_c)$ for the longitudinal case.
- The orientation of the rigid body is given by a rotation $\mathbf{R}: \mathcal{B} \rightarrow \mathcal{W}$, where $\mathbf{R} \in SO(2)$ is an orthogonal rotation matrix, with:

$$\mathbf{R}(\theta) = \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} \tag{1}$$

$$\mathbf{R}(\gamma) = \begin{pmatrix} cos(\gamma) & -sin(\gamma) \\ sin(\gamma) & cos(\gamma) \end{pmatrix} \tag{2}$$

where θ is the quad-rotor's pitch angle, and γ is the angle of the camera positioner respect to (w.r.t.) $\hat{y}_c$ (Fig. 1(a)).

B. Euler-Lagrange Model

In order to characterize the motion of the compounded system, we use the Euler-Lagrange formulation, whose main advantage consists in providing dynamics couplings in multi-body mechanical systems.

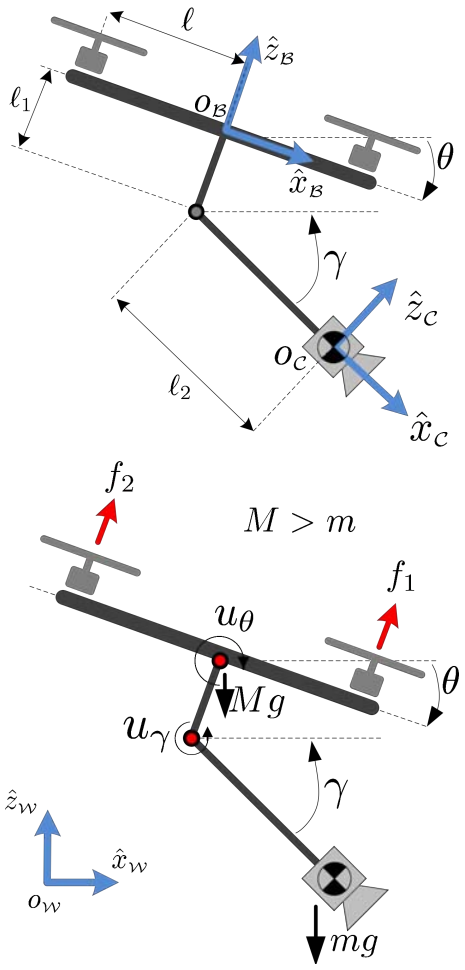


Fig. 1. Freebody diagram: (a) tPVTOL Frames of reference and (b) depiction (in red) of applied forces and moments.

Simple mechanical systems can be described by their Lagrangian, which is generally the difference between the kinetic energy and the potential energy.

1) Kinetic Energy
The total kinetic energy function $\mathcal{K}(\dot{\theta}, \dot{\gamma})$ of the compounded mechanical system resulting from the translational and rotation motion can be partitioned by the sum of the quad-rotor's kinetic energy, given by

$$\mathcal{K}_M(\dot{\theta}) = \tfrac{1}{2} I_Y \dot{\theta}^2 + \tfrac{1}{2} M \dot{x}^2 + \tfrac{1}{2} M \dot{z}^2 \tag{3}$$

and the camera actuator's kinetic energy,

$$\mathcal{K}_m(\dot{\theta}, \dot{\gamma}) = \tfrac{1}{2} I_y (\dot{\theta} - \dot{\gamma})^2 + \tfrac{1}{2} m \dot{x}_c^2 + \tfrac{1}{2} m \dot{z}_c^2 \tag{4}$$

where the position of the camera frame center is

$$\begin{aligned} x_c &= x - \ell_1 sin(\theta) + \ell_2 cos(\gamma) \\ z_c &= z - \ell_1 cos(\theta) - \ell_2 sin(\gamma) \end{aligned}$$

A description of the system's parameters is as follows: ℓ is the length between the position of each propeller and the geometric center of the airframe (considered symmetric). Length ℓ_1 is the distance from the quad-rotor's CoG to the camera actuator's joint, and ℓ_2 the distance between the center of this actuator and the center of the camera frame. I_Y and I_y are respectively the inertia mass-moment of the quad-rotor airframe and the system compounded by the camera and its actuator, while M and m represents, respectively, the mass of the quad-rotor (excluding *gimbal* and camera) and the sum of the masses of the assembly formed by the camera and the positioner (henceforth Camera Compounded System - CCS). In this case, it is considered that the entire mass of the CCS is located at the end of the positioner.

2) Potential Energy
Considering z the height of the quad-rotor's CoG, the potential energy term is given by the sum of both aircraft's and CCS's potential energies ([11])

$$P = Mgz + mg[z - (\ell_1 cos(\theta) + \ell_2 sin(\gamma))] \quad (5)$$

Once the kinetic and potential energy of the compounded system have been computed using (3), (4), and (5), the Lagrangian can be written as follows:

$$\begin{aligned} \mathcal{L} = & \tfrac{1}{2}(M+m)\dot{x}^2 + \tfrac{1}{2}(M+m)\dot{z}^2 + \tfrac{1}{2}(I_Y + m\ell_1^2)\dot{\theta}^2 \\ & + \tfrac{1}{2}m\ell_2^2\dot{\gamma}^2 + \tfrac{1}{2}I_y(\dot{\theta}-\dot{\gamma})^2 \\ & + m\ell_1\dot{\theta}(-\dot{x}cos(\theta) + \dot{z}sin(\theta)) \\ & - m\ell_2\dot{\gamma}(\dot{x}sin(\gamma) + \dot{z}cos(\gamma)) - m\ell_1\ell_2\dot{\theta}\dot{\gamma}sin(\theta-\gamma) \\ & - Mgz - mg[z - (\ell_1 cos(\theta) + \ell_2 sin(\gamma))] \end{aligned} \quad (6)$$

C. Equations of motion

Under certain assumptions the Lagrangian can be divided into two interconnected dynamics, the translational and rotational subsystems ([12]). These simplified equations of motion compose a decentralized dynamic model. Through the Euler-Lagrange formulation, equations modeling the overall motion of the tPVTOL are derived, i.e. the translational subsystem:

$$\begin{cases} u_x = (M+m)\ddot{x} + m\ell_1 sin(\theta)\dot{\theta}^2 - m\ell_1 cos(\theta)\ddot{\theta} \\ \qquad - m\ell_2 cos(\gamma)\dot{\gamma}^2 - m\ell_2 sin(\gamma)\ddot{\gamma} \\ u_z = (M+m)\ddot{z} + m\ell_1 cos(\theta)\dot{\theta}^2 + m\ell_1 sin(\theta)\ddot{\theta} \\ \qquad + m\ell_2 sin(\gamma)\dot{\gamma}^2 - m\ell_2 cos(\gamma)\ddot{\gamma} - (M+m)g \end{cases} \quad (7)$$

The corresponding model that describes the rotational motion is given by:

$$\begin{cases} u_\theta = (I_Y + I_y + m\ell_1^2)\ddot{\theta} - (I_y + m\ell_1\ell_2 sin(\theta-\gamma))\ddot{\gamma} \\ \qquad - m\ell_1 cos(\theta)\ddot{x} + m\ell_1 sin(\theta)\ddot{z} \\ \qquad + m\ell_1\ell_2 cos(\theta-\gamma)\dot{\gamma}^2 - mg\ell_1 sin(\theta) \\ u_\gamma = (I_y + m\ell_1\ell_2 sin(\gamma-\theta))\ddot{\theta} + (m\ell_2^2 - I_y)\ddot{\gamma} \\ \qquad - m\ell_2 sin(\gamma)\ddot{x} - m\ell_2 cos(\gamma)\ddot{z} \\ \qquad - m\ell_1\ell_2 cos(\gamma-\theta)\dot{\theta}^2 + mg\ell_2 cos(\gamma) \end{cases} \quad (8)$$

The system motion equations obtained from the Lagrange formulation, translational (7) and rotational (8), can be clustered in compact vectorial form.

1) Translational motion

$$(M+m)\ddot{\xi} = \mathcal{U}_\xi + \mathcal{W}_{(M+m)} + \mathcal{F}_c + \mathcal{F}_t \quad (9)$$

where $\xi = (x, z)^\top$ represents the 2D position of the quad-rotor. The control input is given by $\mathcal{U}_\xi = [u_x \ u_z]^\top = \mathbf{R}(\theta-\gamma)\cdot[0 \ f]^\top$ as the thrust vector, where $f = f_1 + f_2$ is the total upward thrust provided by frontal and rear motors. $\mathcal{W}_{(M+m)} = (0, (M+m)g)^\top$ denotes the total weight vector. The sum of the terms $\mathcal{F}_c$ and $\mathcal{F}_t$ consists on the coupling force vector resulting from the CCS, which, according to Newton's second Law, collects both the centrifugal forces provided by the camera positioner subsystem ($\mathcal{F}_c$) and tangential forces ($\mathcal{F}_t$) exerted in the aircraft, which are written as

$$\mathcal{F}_c = \begin{pmatrix} -m\ell_1 sin(\theta)\dot{\theta}^2 + m\ell_2 cos(\gamma)\dot{\gamma}^2 \\ -m\ell_1 cos(\theta)\dot{\theta}^2 - m\ell_2 sin(\gamma)\dot{\gamma}^2 \end{pmatrix} \quad (10)$$

$$\mathcal{F}_t = \begin{pmatrix} m\ell_1 cos(\theta)\ddot{\theta} + m\ell_2 sin(\gamma)\ddot{\gamma} \\ -m\ell_1 sin(\theta)\ddot{\theta} + m\ell_2 cos(\gamma)\ddot{\gamma} \end{pmatrix} \quad (11)$$

2) Rotational motion The rotational subsystem motion equation in compact vectorial form can be depicted as

$$\mathcal{J}\ddot{\eta} = \mathcal{U}_\eta + \mathcal{G}_\eta + \Gamma_c + \Psi_c \quad (12)$$

where $\eta = (\theta, \gamma)^\top$ represents the generalized coordinates vector regarding orientation, the control input as $\mathcal{U}_\eta = [u_\theta \ u_\gamma]^\top$, where u_γ is applied directly in the camera positioner joint and u_θ is obtained through $u_\theta = \ell\cdot(f_2 - f_1)$. Effects of the gravity on the system are collected in vector $\mathcal{G}_\eta$, as follows:

$$\mathcal{G}_\eta = \begin{pmatrix} mg\ell_1 sin(\theta) \\ -mg\ell_2 cos(\gamma) \end{pmatrix} \quad (13)$$

The coupling terms among the quad-rotor and the CCS are depicted in Γ_c, written as:

$$\Gamma_c = \begin{pmatrix} m\ell_1 cos(\theta)\ddot{x} - m\ell_1 sin(\theta)\ddot{z} \\ m\ell_2 sin(\gamma)\ddot{x} + m\ell_2 cos(\gamma)\ddot{z} \end{pmatrix} \quad (14)$$

Centrifugal forces provided by the camera positioner are collected in Ψ_c, as follows:

$$\Psi_c = \begin{pmatrix} -m\ell_1\ell_2 cos(\theta-\gamma)\dot{\gamma}^2 \\ m\ell_1\ell_2 cos(\gamma-\theta)\dot{\theta}^2 \end{pmatrix} \quad (15)$$

The matrix inertia term (invertible), defined as $\mathcal{J}$, has the following form:

$$\mathcal{J} = \begin{pmatrix} I_Y + I_y + m\ell_1^2 & -I_y - m\ell_1\ell_2 sin(\theta-\gamma) \\ I_y + m\ell_1\ell_2 sin(\gamma-\theta) & m\ell_2^2 - I_y \end{pmatrix} \tag{16}$$

III. PROBLEM STATEMENT

The goal is getting the tPVTOL to follow a predefined trajectory, given in the quad-rotor frame. Specifically, we are interested in control the camera frame position (x_c, z_c) and the system orientation, given by angles θ and γ, w.r.t. the inertial frame, through the motion of the quad-rotor plus the CCS. Nevertheless, the equations of motion are coupled, since the inertia matrix (16) has terms which depends on the attitude between the quad-rotor and the camera positioner. Considering the system close to its equilibrium point, some assumptions must be arranged in order to decouple both dynamics, as a first approach to control law design.

A. Model assumptions

For control purposes and complexity simplification, coupling terms in the rotational subsystem, i.e, sideward accelerations among the quad-rotor and the CCS (Γ_c), will not be considered as part of the dynamic model for control design, since in near-hovering flights, the effects on the rotational sub-system provided by lateral accelerations are practically close to zero. The couplings between the quad-rotor translational motion and rotational sub-system can then be bounded and decoupled. Similarly, effects of centripetal forces ($\mathcal{F}_c$) on the translational sub-system of the quad-rotor can be neglected due to the consideration of its small effect on the system. These assumptions yield a new form for the inertia matrix as:

$$\mathcal{J}_d = \begin{pmatrix} I_Y + I_y + m\ell_1^2 & 0 \\ 0 & m\ell_2^2 - I_y \end{pmatrix} \tag{17}$$

whereby rotational and translational subsystems are decoupled. Additional assumptions are also considered in this work:

1) The proposed control strategies will be tested by simulation, considering system parameter values collected in table (Tab. 1).

Parameter	Value	Unit
g	9.81	$\frac{m}{s^2}$
m	0.4	kg
M	0.955	kg
I_Y	0.43	$kg \cdot m^2$
I_y	0.25	$kg \cdot m^2$
ℓ_1	0.10	m
ℓ_2	0.10	m

TABLE I

SYSTEM MODEL PARAMETER VALUES

2) For experimental validation, it is considered propellers forces would be limited, with a value not exceeding $f_{max} = (M+m)\mathrm{g} = 13.29$ N. Consequently, the applied thrust in the quad-rotor frame will also be bounded, similarly to the applied torques in the quad-rotor frame, defined as u_θ.
3) The maximum torque provided by the camera positioner rotational joint should also be bounded in experimental tests, with a value $|u_\gamma| \leq 4mg\ell_2 = 1.56$ N·m.

B. Equations of motion

Considering (17) and previous assumptions, the system considered to be controlled is as follows. For the rotational subsystem:

$$\mathcal{J}_d \cdot \ddot{\eta} = \mathcal{U}_\eta + \mathcal{G}_\eta + \Psi_c \tag{18}$$

and the translational:

$$(M+m)\ddot{\xi} = \mathcal{U}_\xi + \mathcal{W}_{(M+m)} + \mathcal{F}_t \tag{19}$$

The model above can be rewritten in a state-space form $\dot{X} = f(X,U)$ by introducing $X = [x_1 ... x_8]^\top \in \mathbb{R}^8$ as state vector of the system.

$$\begin{array}{lllll|lllll} x_1 & = & \theta & & & x_5 & = & x & & \\ x_2 & = & \dot{x}_1 & = & \dot{\theta} & x_6 & = & \dot{x}_5 & = & \dot{x} \\ x_3 & = & \gamma & & & x_7 & = & z & & \\ x_4 & = & \dot{x}_3 & = & \dot{\gamma} & x_8 & = & \dot{x}_7 & = & \dot{z} \end{array} \tag{20}$$

From (18-20), the state-space system representation is obtained:

$$\dot{X} = \begin{pmatrix} x_2 \\ \frac{1}{b_1}(-a_1 cos(x_1 - x_3)x_4^2 + a_2 sin(x_1) + U_3) \\ x_4 \\ \frac{1}{b_2}(a_1 cos(x_1 - x_3)x_2^2 - a_3 cos(x_3) + U_4) \\ x_6 \\ \frac{1}{b_3}(-a_4 sin(x_1)x_2^2 + a_5 cos(x_3)x_4^2 + U_1) \\ x_8 \\ \frac{1}{b_3}(-a_4 cos(x_1)x_2^2 - a_5 sin(x_3)x_4^2 + a_6 + U_2) \end{pmatrix} \tag{21}$$

With:

$$\begin{array}{lll|lll} a_1 & = & m\ell_1\ell_2 & b_1 & = & m\ell_1^2 + I_Y + I_y \\ a_2 & = & gm\ell_1 & b_2 & = & m\ell_2^2 - I_y \\ a_3 & = & gm\ell_2 & b_3 & = & M+m \\ a_4 & = & m\ell_1 & & & \\ a_5 & = & m\ell_2 & & & \\ a_6 & = & g(M+m) & & & \end{array} \tag{22}$$

and the control input vector as:

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} u_x \\ u_z \\ u_\theta \\ u_\gamma \end{bmatrix} = \begin{bmatrix} f \cdot sin(\theta-\gamma) \\ f \cdot cos(\theta-\gamma) \\ \ell(f_2 - f_1) \\ u_\gamma \end{bmatrix} \tag{23}$$

IV. BACKSTEPPING CONTROL OF THE TPVTOL

The control strategy's goal is to drive the compounded system according to the commanded reference. It is worthwhile to note since the model given in (21) is constituted of translational (position) and rotational (attitude) decoupled subsystems, it is possible to propose a control scheme where the translational controller outputs desired attitude angle, i.e, pitch (θ) and camera angular position (γ), which are the angles to be tracked by the orientation controllers.

Using the Backstepping approach, based on [13], it is possible to synthesize the virtual control law forcing the system to follow the desired trajectory.

A. Attitude Control

For the first step, the tracking error is considered as:

$$z_1 = x_{1d} - x_1 \tag{24}$$

Considering the Lyapunov theorem of stability, we define a Lyapunov function z_1 positive definite

$$V(z_1) = \tfrac{1}{2} z_1^2 \tag{25}$$

and its time derivative negative semi-definite as:

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2) \tag{26}$$

The stabilization of z_1 is obtained introducing a virtual control input x_2 as

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \tag{27}$$

with $\alpha_1 > 0$. The equation (26) is rewritten as:

$$\dot{V}(z_1) = -\alpha_1 z_1^2 \tag{28}$$

By changing variables:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{29}$$

For the second step, the following augmented Lyapunov function is considered:

$$V(z_1, z_2) = \tfrac{1}{2}(z_1^2 + z_2^2) \tag{30}$$

The corresponding time derivative of (30) is:

$$\begin{aligned} \dot{V}(z_1, z_2) &= z_2\left(\tfrac{1}{b_1}(-a_1 \mathrm{c}(x_1 - x_3)x_4^2 + a_2 \mathrm{s}(x_1) + U_3)\right) \\ &- z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) \\ &- z_1(z_2 - \alpha z_1) \end{aligned} \tag{31}$$

where $\mathrm{c}(\cdot)$ and $\mathrm{s}(\cdot)$ stands for $cos(\cdot)$ and $sin(\cdot)$, respectively. The control output U_3 is then extracted, satisfying $\dot{V}(z_1, z_2) < 0$:

$$\begin{aligned} U_3 &= (z_1 - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2)\, b_1 \\ &+ a_1 \mathrm{c}(x_1 - x_3)x_4^2 \\ &- a_2 \mathrm{s}(x_1) \end{aligned} \tag{32}$$

Term $\alpha_2 z_2$ is added in order to stabilize z_1. A similar procedure is performed to compute the input control U_4:

$$\begin{aligned} U_4 &= (z_3 - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4)\, b_2 + a_3 \mathrm{c}(x_3) \\ &- a_1 \mathrm{c}(x_1 - x_3)x_2^2 \end{aligned} \tag{33}$$

with:

$$\begin{cases} z_3 = x_{3d} - x_3 \\ z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \end{cases} \tag{34}$$

B. Position Control

The procedure followed to compute control signals U_1 and U_2 is similar to the previous section.

$$\begin{aligned} U_1 &= (z_5 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6)\, b_3 + a_4 \mathrm{s}(x1)x_2^2 \\ &- a_5 \mathrm{c}(x_3)x_4^2 \end{aligned} \tag{35}$$

$$\begin{aligned} U_2 &= (z_7 - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8)\, b_3 + a_4 \mathrm{c}(x_1)x_2^2 \\ &+ a_5 \mathrm{s}(x_3)x_4^2 - a_6 \end{aligned} \tag{36}$$

with:

$$\begin{cases} z_5 = x_{5d} - x_5 \\ z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5 \\ z_7 = x_{7d} - x_7 \\ z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7 \end{cases} \tag{37}$$

Similarly, terms $\alpha_6 z_6$ and $\alpha_8 z_8$ with $(\alpha_6, \alpha_8) > 0$ are added to stabilize virtual control laws z_5 and z_7, respectively.

C. Backstepping Controller Numerical Simulation

Several simulations were performed on Matlab Simulink using the proposed dynamic model in (21) with the 8 parameters $(\alpha_1..\alpha_8)$ controller, which were adjusted simultaneously by observing the behaviour of the system. The initial condition was 0.2 (m) for the x-component in the translational and 0.1 (m) for the z-component. Similar initial conditions were applied for the rotational subsystem, with $[\theta_0 \ \gamma_0]^\top = [0.2 \ 0.1]^\top (rad)$. The goal for the control task was to achieve the desired position $[x_{5d} \ x_{7d}]^\top = [3 \ 5]^\top (m)$ fully stabilized in orientation, i.e., $[x_{1d} \ x_{3d}]^\top = [\theta_d \ \gamma_d]^\top = [0 \ 0]^\top (rad)$. Results were satisfactory, as depicted in figures (Fig. 2) and (Fig. 3).

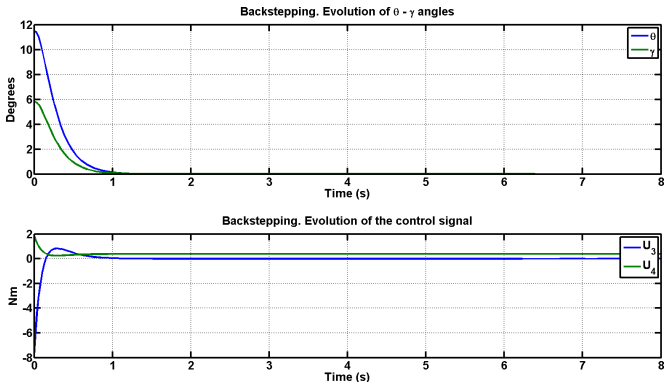


Fig. 2. Simulation: the backstepping controller has to stabilize the rotational subsystem and maintain the θ and γ angles close to zero.

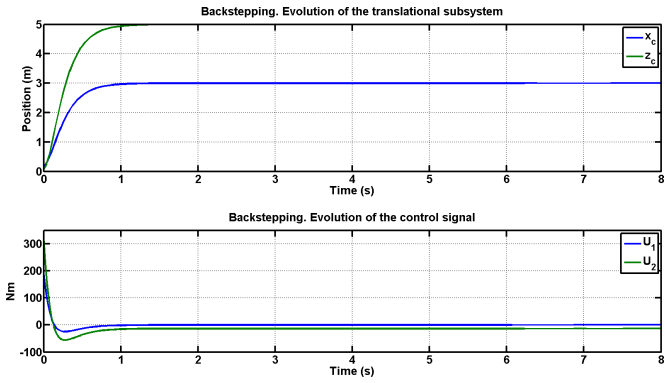


Fig. 3. Simulation: the backstepping controller has to stabilize the translational subsystem to the desired point. It works well where initial conditions are in a relatively small range.

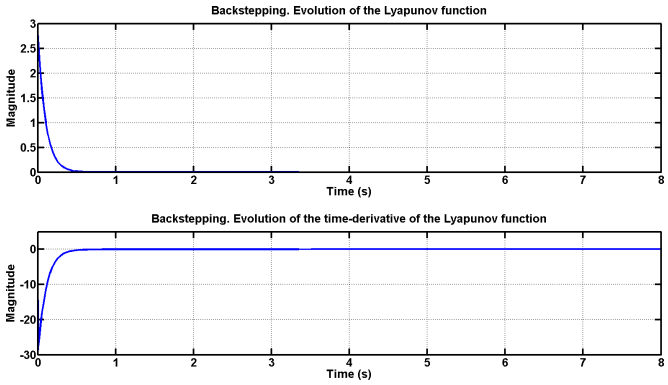


Fig. 4. Backstepping simulation: Lyapunov function behaviour is consistent with the stability theorem definition.

V. SLIDING-MODE CONTROL OF THE TPVTOL

The first step concerning the design of a Sliding-mode based control law is similar to the one used for the backstepping approach, using the state-space system defined in (Eq. 21).

A. Attitude Control

For the rotational subsystem, instead of using a second virtual control variable as z_2, the surface s_2 is used to be consistent with the sliding mode law definition:

$$s_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{38}$$

For the second step, we consider the augmented Lyapunov function:

$$V(z_1, s_2) = \tfrac{1}{2}(z_1^2 + s_2^2) \tag{39}$$

The chosen law for the attractive surface is the time derivative of (38) satisfying ($s\dot{s} < 0$):

$$\begin{aligned} \dot{s}_2 &= -k_1 sign(s_2) - k_2 s_2 \\ &= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 \\ &= \frac{1}{b_1}(-a_1 cos(x_1 - x_3)x_4^2 + a_2 sin(x_1) + U_3) \\ &- \ddot{x}_{1d} + \alpha_1(s_2 + \alpha_1 z_1) \end{aligned} \tag{40}$$

The control signal U_3 is then extracted:

$$\begin{aligned} U_3 &= (-\alpha_1^2 z_1 - k_1 sign(s_2) - k_2 s_2)\, b_1 \\ &+ a_1 cos(x_1 - x_3)x_4^2 - a_2 sin(x_1) \end{aligned} \tag{41}$$

And the same step is followed in order to get U_4:

$$\begin{aligned} U_4 &= (-\alpha_2^2 z_3 - k_3 sign(s_3) - k_4 s_3)\, b_2 \\ &+ a_3 cos(x_3) - a_1 cos(x_1 - x_3)x_2^2 \end{aligned} \tag{42}$$

with:

$$\begin{cases} z_3 = x_{3d} - x_3 \\ s_3 = x_4 - \dot{x}_{3d} - \alpha_2 z_3 \end{cases} \tag{43}$$

B. Position Control

Following a similar procedure, control signals U_1 and U_2 are then extracted:

$$\begin{aligned} U_1 &= (-\alpha_3^2 z_5 - k_5 sign(s_4) - k_6 s_4)\, b_3 \\ &+ a_4 sin(x_1)x_2^2 - a_5 cos(x_3)x_4^2 \end{aligned} \tag{44}$$

$$\begin{aligned} U_2 &= (-\alpha_4^2 z_7 - k_7 sign(s_5) - k_8 s_5)\, b_3 \\ &+ a_4 cos(x_1)x_2^2 + a_5 sin(x_3)x_4^2 - a_6 \end{aligned} \tag{45}$$

with:

$$\begin{cases} z_5 = x_{5d} - x_5 \\ s_4 = x_6 - \dot{x}_{5d} - \alpha_3 z_5 \\ z_7 = x_{7d} - x_7 \\ s_5 = x_8 - \dot{x}_{7d} - \alpha_4 z_7 \end{cases} \tag{46}$$

C. Sliding-mode Controller Numerical Simulation

Since the $sign(\cdot)$ function used in the proposed control law is numerically discontinuous, a new function definition $\mu(\cdot)$ was used to improve the system behaviour, avoiding the characteristic chattering effect of the sliding-mode techniques, obtaining a behavior similar to a ramp function, where:

$$\mu(x) = \begin{cases} 1 & x > 1 \\ -1 & x < 1 \\ x/\epsilon & oth. \end{cases} \tag{47}$$

with $\epsilon \simeq 10^{-1}$.

In order to evaluate the Sliding-mode based control law some numerical simulations were performed. Similarly to the previous case, controller parameters were simultaneously adjusted in order to get good performance in controller and system behaviour. For the translational subsystem, initial

conditions are 0.2 (m) for the x-component and 0.1 (m) for the z-component. Similar conditions were applied for the rotational subsystem case, with $[\theta_0\ \gamma_0]^\top = [0.2\ 0.1]^\top (rad)$. In this case, the goal for the control task was to stabilize the system in hovering conditions, i.e., to achieve the desired position $[x_{5d}\ x_{7d}]^\top = [0\ 0]^\top (m)$ and an orientation reference, given as $[x_{1d}\ x_{3d}]^\top = [\theta_d\ \gamma_d]^\top = [0\ 0]^\top (rad)$.

Numerical results obtained show that the proposed controller is able to stabilize both rotational and translational subsystems, avoiding the chattering effect in the control signal through the use of the (47) function. Satisfactory results are illustrated in figures (Fig. 5) and (Fig. 6). Numerically, the behaviour of the system is slightly worse than the approach using the Backstepping technique.

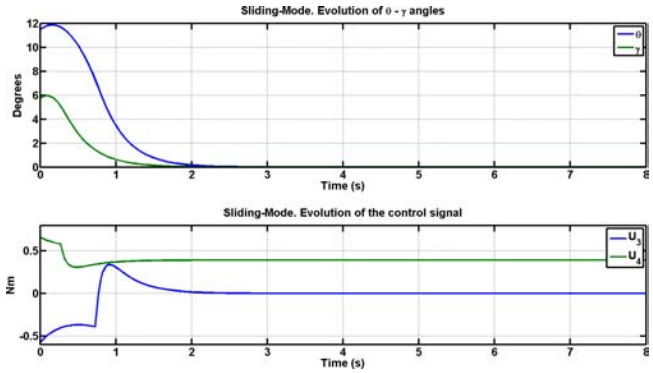


Fig. 5. Simulation: the sliding-mode controller has to stabilize the rotational subsystem and maintain the θ and γ angles close to zero. The behaviour of the system is slightly worse than the backstepping controller.

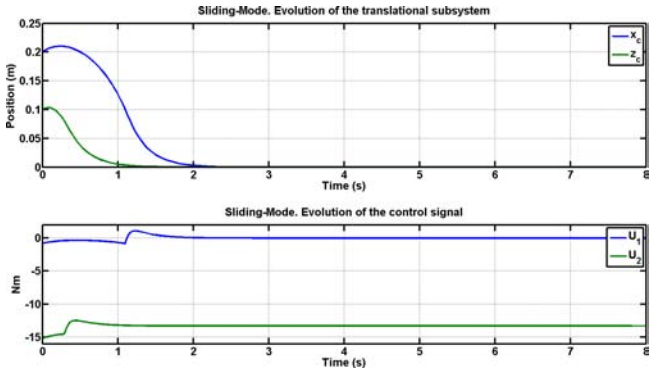


Fig. 6. Simulation: the sliding-mode controller has to stabilize the translational subsystem to the desired point. Results show that the performance of this controller is not as suitable as the backstepping.

VI. FINAL REMARKS AND PERSPECTIVES

This paper addressed a first approach for the modeling and control of a multi-body platform consisting of a quad-rotor and camera positioner. The aim is to use the two additional degrees of freedom provided by a camera positioner in combination with an underactuated quad-rotor aircraft to get an actuated system, allowing free arbitrary movement in the rotational subsystem of the camera in the space. We

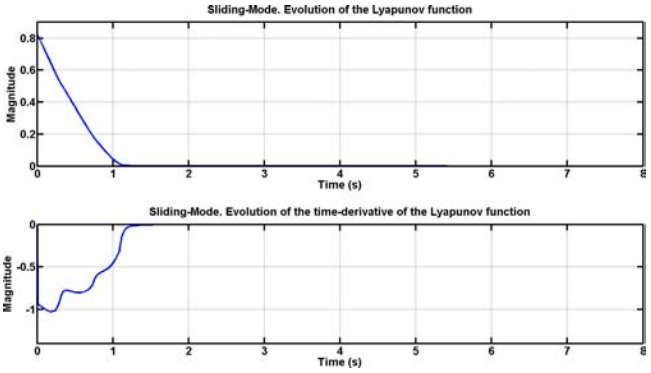


Fig. 7. Sliding-mode simulation: Lyapunov function behaviour is consistent with the stability theorem definition.

presented numerical simulation of two different nonlinear control techniques "Backstepping" and "Sliding-mode" applied to the tPVTOL. As it can be seen in the numerical results, the controller introduced using the backstepping approach provides average results. First tuning of the sliding-mode controller shows worse results stabilizing rotational and translational subsystems. Future work will comprise a better assignment to the controllers parameters in order to get better results.

REFERENCES

[1] G.V. Raffo, Robust Control Strategies for a quadrotor helicopter. An underactuated mechanical system. Ph.D.Thesis, Engineering School. University of Seville. 2011.

[2] J.Escareno, M.Rakotondrabe, G.Flores and R.Lozano, Rotorcraft MAV having an onboard manipulator: longitudinal modeling and robust control, in European Control Conference (ECC). Zürich, Switzerland, 2013, p.267-269.

[3] V. Lippiello, and F. Ruggiero, Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, p.3768-3773.

[4] P. Corke, Robotics, Vision and Control. Fundamental Algorithms in MATLAB, Springer, 2011.

[5] A. Chriette, T. Hamel and R. Mahony, Visual servoing for a scale model autonomous helicopter, in Proceedings ICRA. IEEE International Conference on Robotics and Automation, 2001, p. 1701-1706 vol.2.

[6] E. Altug, J.P. Ostrowski, and R. Mahony, Control of a quadrotor helicopter using visual feedback, in Proceedings ICRA. IEEE International Conference on Robotics and Automation, 2002, p.72-77 vol.1.

[7] F. Bonnin-Pascual, E. Garcia-Fidalgo, and A. Ortiz, Semi-autonomous visual inspection of vessels assisted by an unmanned Micro Aerial Vehicle, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, p.3955-3961.

[8] D. Lee, V. Chitrakaran, T. Burg, D. Dawson and B. Xian, Control of a remotely operated quadrotor aerial vehicle and camera unit using a fly-the-camera perspective, in 46th IEEE Conference on Decision and Control, 2007, p.6412-6417.

[9] A.E. Neff, D. Lee, V.K. Chitrakaran, D.M. Dawson, and T.C. Burg, Velocity control for a quad-rotor uav fly-by-camera interface, in SoutheastCon. Proceedings IEEE, 2007, p. 273-278.

[10] B. Etkin and L.D. Reid, Dynamics of Flight, New York: Wiley, 1959.

[11] I. Fantoni and R. Lozano, Nonlinear control for underactuated mechanical systems. In Communications and control engineering series, Springer, 2002.

[12] P. Castillo, R. Lozano and A.E. Dzul, Modelling and Control of Mini-Flying Machines, Springer, 2005.

[13] S. Bouabdallah, and R. Siegwart, Backstepping and sliding-mode techniques applied to an indoor micro quadrotor, Proceedings ICRA. IEEE International Conference on Robotics and Automation. 2005.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2014)
September 18, 2014, Chicago, Illinois, USA

Path following: from mobile robotics to laser surgery

Jean-Antoine Seon, Brahim Tamadazte, Nicolas Andreff

***Abstract*— Laser surgery requires accurate following of a path defined by the surgeon, while the velocity on this path is dependent on the laser-tissue interaction. Therefore, path following and velocity profile control must be decoupled. In this paper, non-holonomous control of the unicycle model is used to implement velocity-independent visual path following for laser microphonosurgery. Experiments at almost 600 Hz show an accuracy (mean) of less than 0.34 pixels ($\approx 15\ \mu$m) with a standard deviation of 0.21 pixels ($\approx 10\mu$m) in path following, and a relative velocity distortion of 7.5×10^{-8} %.**

I. INTRODUCTION

Laser surgery consists of the use of a laser source (instead of a scalpel) to cut tissue in which the laser beam vaporizes soft tissue. A large variety of surgical areas practice laser surgery: ophthalmology, gynaecology, otolaryngology, neurosurgery, and paediatric surgery [1]. This is especially true when it comes to microsurgery which requires an extreme precision [2]. In the past few decades, the microphonosurgery received a growing interest by both surgeons and researchers in the objective to develop more suitable and intuitive laser microsurgery system [3], especially in the laser-assisted procedure field. Vocal folds surgery is one of the most demanding procedures in terms of accuracy, particularly because of the specific tissue to be resected (thin, viscous, fragile, difficult healing, and lesions less than 1 mm) [4].

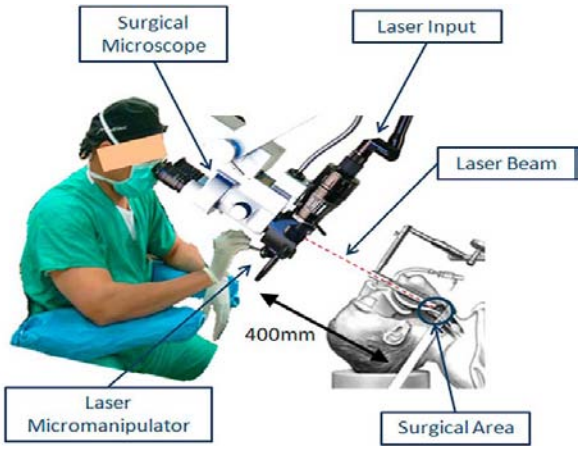


Fig. 1: Schematic view of the AcuBlade system.

The most popular laser-assisted microphonosurgery is undoubtedly the AcuBlade system [5]. The later consists of a laser micromanipulator placed outside the patient which is manually actuated by the surgeon. The use of this system requires extensive expertise due to the position of the laser source (positioned from a distance of 400 mm) (Fig. 1). Consequently, the laser surgery performances are highly dependent on the individual surgeon dexterity and skills [6], [7]. To overcome this drawbacks, the μRALP project develops a more compact (less than 20 mm of diameter) and intuitive flexible endoscopic system [8]. It can be inserted directly into the larynx and equipped with cameras, lighting source, laser source and two degrees of freedom (dof) steered mirror [9] to control the laser spot displacement on the target (Fig. 2). This new system allows the monitoring of ablation and resection of the vocal folds tumors in automatic mode. It means that the laser displacement on the vocal folds is steered by controlling the embedded 2 dof mirror using a vision feedback control scheme.

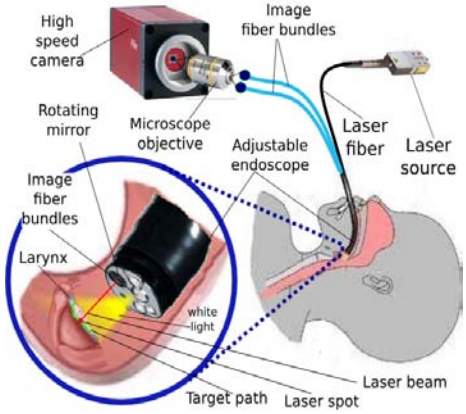


Fig. 2: Schematic view of the μRALP system.

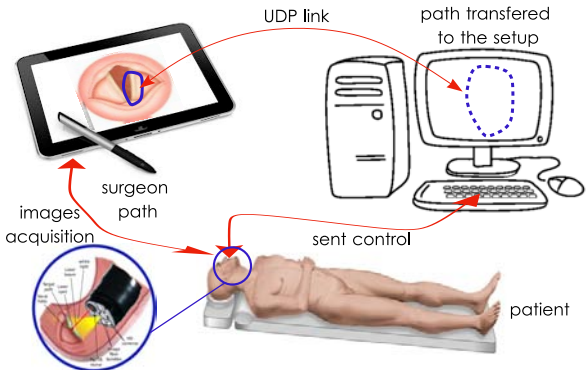


Fig. 3: Schematic view of the laser steering system.

In practice, the control consists of two tasks. The first

This work was supported by μRALP, the EC FP7 ICT Collaborative Project no. 288663 (http://www.microralp.eu), and by ACTION, the French ANR Labex no. ANR-11-LABX-01-01 (http://www.labex-action.fr).

The authors are with FEMTO-ST Institute, AS2M department, Université de Franche-Comté/CNRS/ENSMM/UTBM, 24 rue Savary, 25000 Besanon, France. nicolas.andreff@femto-st.fr

one is to ensure the laser velocity compatibility with laser-tissue interaction, in order to avoid carbonization of the tissue while achieving efficient incision or ablation. The second one is to ensure that the laser follows the desired geometric path drawn by the surgeon on the input device (a tablet in [8]) (Fig. 3). Those two tasks eventually define the trajectory (i.e., geometric path + velocity profile along the path) of the laser. However, it is not advisable to use standard trajectory tracking because the two tasks should be intuitively modifiable by the surgeon. In this paper, we focus on the second task: laser path following using the visual feedback independently from the velocity profile.

The first contribution of the paper is to consider laser path following as a non-holonomic problem, similar to the unicycle path following (Fig. 4(a)). The second contribution of the paper is to implement path following at high frequency to satisfy the constraints of laser-tissue interaction.

In the remainder of this paper, Section II gives a background on the existing path following methods in mobile robotics. In Section III, the implementation of the path following method is detailed. Section IV presents the experimental validation results.

II. PATH FOLLOWING IN MOBILE ROBOTICS

Path following in the mobile robotics has been widely studied in the literature for many years and there are several applications in the industrial field (e.g., autonomous vehicle control). Path following differs from trajectory following essentially by the fact that the notion of *time* is removed in the first one. Indeed, in trajectory following the robot is controlled to be at a location at a given time which is not the case in path following. However, as shown by Brockett [10], mobile robots are not stable with continuous steady state feedback laws. The first approach is to use discontinuous control laws [12] or piecewise continuous control law [13]. But these methods do not explicitly address the issue of robustness to the occurrence of uncertainties. Thereafter, [14] proposed to use a sliding mode based controller which has shown better behaviors. The second approach is to use continuous non-steady state feedback laws such as Samson in the early 90 [15]. Lyapunov methods are often used to design such type of control law [17], [19]. Indeed, it gives good results in terms of robustness and accuracy. It is also possible to use a chained system to design the controller with exact linearization as shown by Morin and Samson [20], [21]. In these studies, the path following speed profile is not known so, only the path and the distance between the robot and the curve are considered. Therefore, the tangential vectors of the curve are used to set the velocity and the orientation of the robot. In our work, both motion and behavior of a laser spot are considered as analogous to those of a unicycle robot. This means that the displacement of the laser spot on a target is governed by the control of an actuated small mirror on which the laser beam is reflected. The laser spot trajectory will be defined by the surgeon using a tablet or smartphone, on which are projected the vocal folds images in real-time. This trajectory is naturally unknown shape and a non-parameterized curve.

A. Kinematic Equations

Let us consider that the laser spot is represented by a 2D point $\mathbf{p} = (x,y)^\top$ in a fixed (reference) frame $\mathcal{R}_0$ (the image frame). Therefore, the kinematics of the laser are governed in the *Frenet* frame $\mathcal{R}_s$ (Fig. 4(b))frame by:

$$\dot{s} = \frac{v}{1-dC(s)}\cos\theta_e \quad (1)$$

$$\dot{d} = v\sin\theta_e \quad (2)$$

$$\dot{\theta}_e = \omega - \dot{s}C(s) \quad (3)$$

where s and $C(s)$ are respectively the curvilinear abscissa and the curvature, θ_e is the difference between the laser orientation $\underline{\mathbf{v}}$ and the tangential vector of $\mathcal{R}_s$, d is the distance of $\mathbf{p}$ to Γ, v represents the translational velocity amplitude of the laser spot in $\mathcal{R}_0$, and ω its rotational velocity carried by the axis ${}^0\underline{z}$ (Fig. 4(b)).

B. Control law

To establish the controller, [20] introduces a coordinates/variables transformation in the following manner: $\{s,d,\theta_e,v,\omega\} \iff \{z_1,z_2,z_3,u_1,u_2\}$ defined in $\mathbb{R}^2 \times \left(-\frac{\pi}{2},+\frac{\pi}{2}\right) \times \mathbb{R}^2$. This allows transforming locally (1), (2) and (3) in:

$$\dot{z}_1 = u_1 \quad (4)$$

$$\dot{z}_2 = u_1 z_3 \quad (5)$$

$$\dot{z}_3 = u_2 \quad (6)$$

where u_1 and u_2 are intermediate control inputs.

In our case, the coordinates/variables transformation is defined as:

$$z_1 = s \quad (7)$$

$$z_2 = d \quad (8)$$

$$z_3 = \Big(1-dC(s)\Big)\tan\theta_e \quad (9)$$

and,

$$u_1 = \dot{s} \quad (10)$$

$$u_2 = \Big(-\dot{d}C(s) - d\frac{\partial C(s)}{\partial s}\dot{s}\Big)\tan\theta_e + \Big(1-dC(s)\Big)\Big(1+\tan^2\theta_e\Big)\dot{\theta}_e \quad (11)$$

Moreover, to ensure that the distance d and the orientation error θ_e are servoed to 0, the stable proportional state feedback solution [20] is:

$$u_2 = -u_1\gamma_1 z_2 - |u_1|\gamma_2 z_3 \quad (12)$$

where, γ_1 and γ_2 are positive gains.

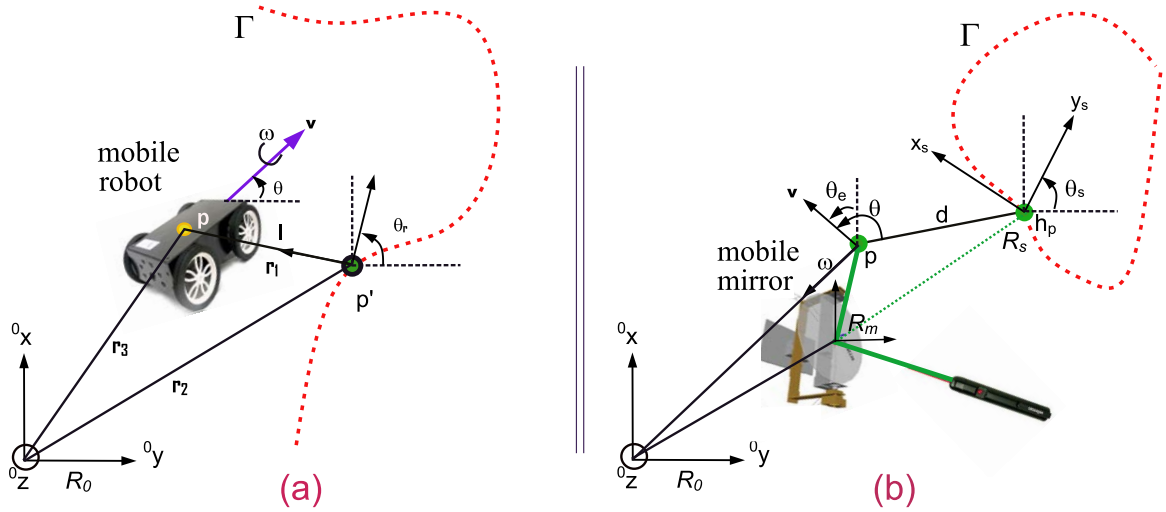


Fig. 4: Mobile robot modeling *versus* laser surgery modeling.

III. IMPLEMENTATION

Considering that the position of the laser spot $\mathbf{p}$ and the curvilinear coordinates s and $C(s)$ are well known at each iterations k of the process, the rotational speed ω can be computed by inverting (11):

$$\omega = \frac{u_2 + \left(\dot{d}C(s) + d\frac{\partial C(s)}{\partial s}\dot{s}\right)\tan\theta_e}{\left(1 - dC(s)\right)\left(1 + \tan^2\theta_e\right)} + \dot{s}C(s) \qquad (13)$$

Note that the singularity $dC(s) = 1$ is avoid as long as d is small. Then, using (12) and (13) allow defining the expression of the velocity direction at each step k of the path following process:

$$\underline{\mathbf{v}}_{k+1} = \frac{\underline{\mathbf{v}}_k + \omega^0\underline{\mathbf{z}} \times \underline{\mathbf{v}}_k}{\|\underline{\mathbf{v}}_k + \omega^0\underline{\mathbf{z}} \times \underline{\mathbf{v}}_k\|} \qquad (14)$$

where, $\times$ defines the cross-product and $\underline{\mathbf{v}}_k$ represents the current velocity direction of the laser spot.

Therefore, it is necessary to convert these velocities to the joint velocities $\dot{\mathbf{q}}_i$ ($i \in [1,2]$) through the inverse differential kinematic matrix $\mathbf{D}^{-1}$ of the mirror mechanism as $\dot{\mathbf{q}}_i = \mathbf{D}^{-1}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}$.

IV. VALIDATION

A. Experimental Set-up

The proposed path following technique was tested on a home-made experimental set-up (Fig. 5). It consists of a high speed camera EoSens®CXP (from Mikrotron®) characterized with a frame rate which can reach 10 000 frames per second with a resolution of 800 × 600 pixels, a laser pointer, a fixed mirror, and an actuated mirror (S-334) from Physical Instruments Inc. The latter contains two single axis units (α and β) working in series with one common pivot point characterized by a bandwidth of 1 kHz, a resolution of 0.2 μrad and a motion range of ± 25 *mrad*. With that motion range and appropriate alignment of the mirror with the image, the inverse kinematics can be approximated by: $\mathbf{D}^{-1} \approx \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$.

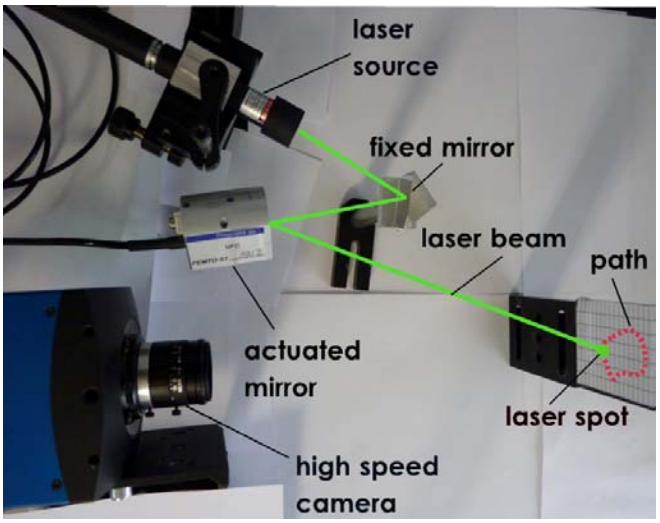


Fig. 5: Photography of the experimental set-up.

Concerning the time-varying position p of the laser spot in the image, it is tracked using ViSP library [23].

B. Experimental Validation

The proposed method have been implemented on the experimental set-up presented above to verify the relevance of the path following technique. To illustrate the results, we opted for a non-parametric and arbitrary shape curve as shown in Fig. 6. The initial position of the laser spot is placed at a distance d = 1 pixel and an orientation θ_e = 0 to the closed point of the curve Γ. The initial parameters (gains) are fixed as v = 100, γ_1 = 0.2, and γ_2 = 0.8.

Fig. 6 shows an image sequence describing the experimental validation of the proposed path following method. As shown by this figure, the reference path (yellow color) is perfectly followed by the laser spot (red color) even in the

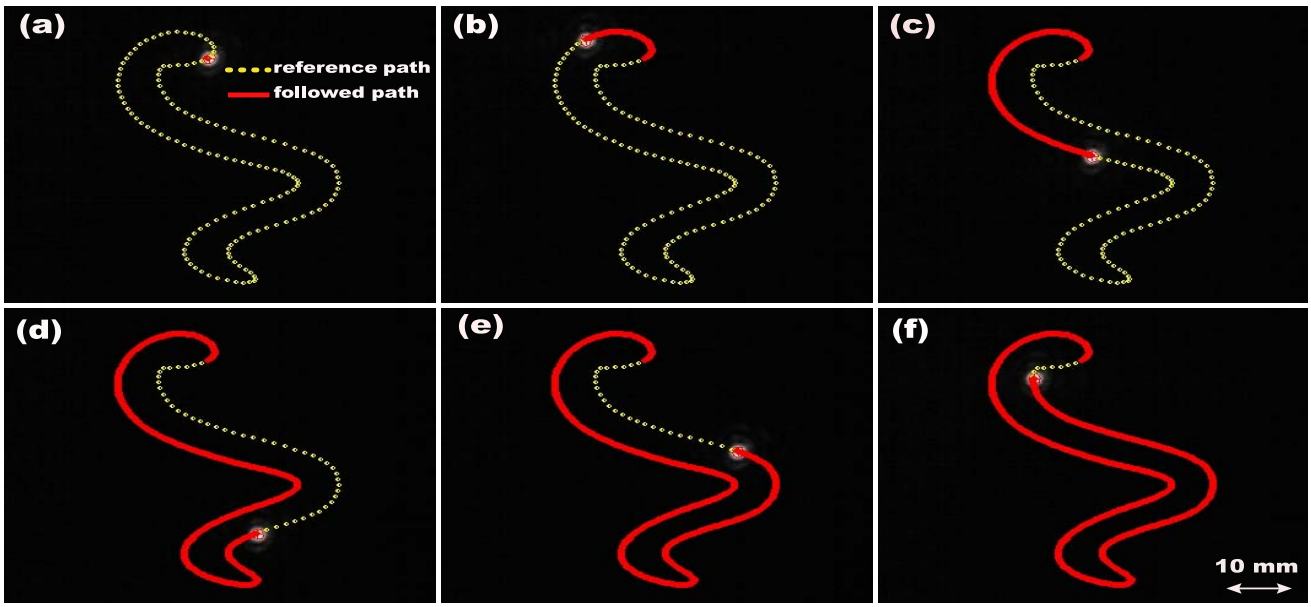


Fig. 6: Image sequence captured during the experimental validation of the path following: (a) to (f) represent the laser spot displacements (in red color) following the reference path (in yellow color).

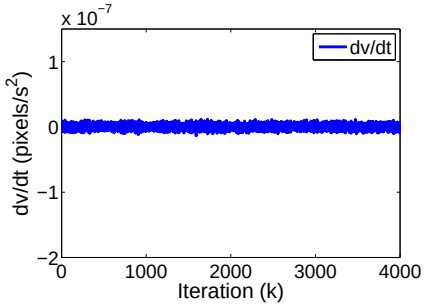
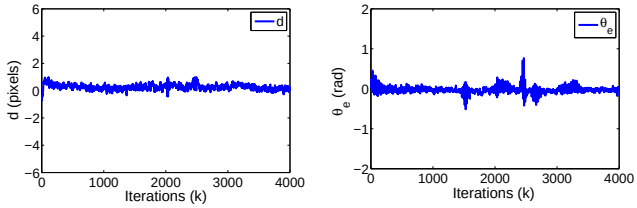


Fig. 7: Velocity profile during the path tracking process studied in Fig. 6.

TABLE I: Time-consuming of the main tasks of the process.

task	time (*ms*)
grabbing image	0.064
laser tracking	0.471
controller	0.002
sending control (USB 2.0)	0.989
others	0.176
entire process	**1.702 (f = 587 Hz)**

case where the curvature is significant. Similarly, Fig. 8(a) illustrates the evolution of the errors (distance d as well as orientation θ_e) *versus* the iterations number k (**each iteration takes 1.702 ms**). Note that peaks in Fig. 8(b) are due to the high curvature area. For several repetitive tests, the RMS (Root Mean Square) error as well as the standard deviation (STD) are computed (e.g., RMS(d) = 0.34 pixels $\equiv$ 15 μm and STD(d) = 0.21 pixels $\equiv$ 10 μm). Otherwise, the velocity variations are computed and shown in Fig. 7 in which it can be highlighted that the laser spot velocity profile during the entirety process remains almost constant (with a relative velocity distortion of 7.5×10^{-8} %).



(a) Distance error d. (b) Orientation error θ_e.

Fig. 8: Experimental results for the validation test shown in Fig. 6.

Note that in laser microphonosurgery, the laser scanning should works in high frequency i.e., at least of 200 Hz (sampling time) to avoid tissue carbonization during tumors ablation or resection. As shown in Table I, it can be highlighted that our approach takes only 1.702 milliseconds (ms) which is equivalent to a frequency of 587 Hz (sampling time) despite the use of a slow USB communication.

We also studied the effect of the velocity $\mathbf{v}$ on the quality (RMS and standard deviation of the error) for the path following using another shape curve. Thus, for different ranges of velocity $\mathbf{v}$ from 100 pixels/seconds to 500 pixels/seconds with a step of 100, the accuracy remains almost the same (RMS(d) $\in$ [0.072 to 0.083] pixels, and RMS(θ_e) $\in$ [0.072 to 0.083] rad and standard deviation within similar ranges.

V. Conclusion

In this paper, a vision-based control for laser steering using a path following approach was presented. To achieve this, we were inspired explicitly by the methods used in mobile (unicycle kinematic model) in order to develop an efficient controller scheme by taking into account the fact that the path following and the velocity profile must be decoupled.

Thus, the proposed path following method was based on the use of the curvilinear representation (i.e., curvilinear abscissa and curvature) which allows a better consideration of the laser steering problem, especially in case of laser surgery. This method was validated experimentally using a real-world validation set-up which was equipped with a high frequency (1 kHz) actuated mirror and high speed camera ($\approx$ 10 000 fps). The developed controller has shown more than satisfactory results in terms of accuracy (about 15 μm), robustness, repeatability and rapidity (about 587 Hz).

The next stages of this work will involve adapting the described materials for real laser surgery applications i.e., on a human cadaver using the entire endoscopy system developed in μRALP project. In addition, the velocity will have to be servoed with regard to the heat profile in the tissue and to the depth variations of the tissues with respect to the image.

REFERENCES

[1] W. Schwesinger and J. Hunter, "Laser in general surgery, in "*Surgical Clinics of North America*, 1992.

[2] V. Moiseev, "Laser technologies in microsurgery of extramedullary tumors, in "*Int. Conf. on Actual Problems of Electronic Instrument Engineering*, 2004, pp. 275–275.

[3] M. Remacle, "Laser-assisted microphonosurgery," in *Surgery of Larynx and Trachea*, M. Remacle and H. E. Eckel, Eds. Springer Berlin Heidelberg, 2010, pp. 51–56.

[4] H. Eckel, S. Berendes, M. Damm, and J. Klusmann, "Suspension laryngoscopy for endotracheal stenting, in "*Laryngoscope*, vol. 113, pp. 11–15, 2003.

[5] L. I. Israel. (2014, Feb.) Digital acublade system. israel@online. [Online]. Available: http://www.surgical.lumenis.com

[6] J. Ortiz, L. Mattos, and D. Caldwell, "Smart devices in robot-assisted laser microsurgery: Towards ubiquitous tele-cooperation," in *IEEE Int. Conf. on Rob. and Biomimetics*, 2012, pp. 1721–1726.

[7] N. Andreff, S. Dembélé, B. Tamadazte, Z. E. Hussnain, "Epipolar geometry for vision-guided laser surgery" in *Int. Conf. on Info. in Cont., Auto. and Robo.*, 2013, pp. 471–476.

[8] L. Mattos, D. Nikhil, G. Barresi, L. Gustani, G. Pirreti, "A novel computerized surgeonmachine interface for robot-assisted laser phonomicrosurgery"; in "Laryngoscope", DOI: 10.1002/lary.24566, 2014.

[9] S. Lescano, D. Zlatanov, M. Rakotondrabe, and N. Andreff, "Kinematic analysis of a meso scale parallel robot for laser phonomicrosurgery," in *IAK 2013 - 2nd Conf. on Interdisciplinary Applications in Kinematics*, A. Kecskemthy, Ed. Springer Berlin, 2013.

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

[11] A. Bloch and N. McClamroch, "Control of mechanical systems with classical nonholonomic constraints," in *IEEE Conf. on Decision and Control*, 1989, pp. 201–205.

[12] A. Bloch, M. Reyhanoglu, and N. McClamroch, "Control and stabilization of nonholonomic dynamic systems," in *IEEE Trans. on Automatic Control*, vol. 37, no. 11, pp. 1746–1757, 1992.

[13] J. P. Hespanha, D. Liberzon, and A. Morse, "Logic-based switching control of a nonholonomic system with parametric modeling uncertainty, in"*Systems and Control Letters*, vol. 38(3), pp. 167–177, 1999.

[14] T. Floquet, J.-P. Barbot, and W. Perruquetti, "Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems, in "*Automatica*, vol. 39, no. 6, pp. 1077–1083, 2003.

[15] C. Samson, "Velocity and torque feedback control of a nonholonomic cart," in *Advanced Robot Control*, Lecture Notes in Control and Information Sciences, C. Canudas de Wit, Ed.Springer Berlin Heidelberg, 1991, vol. 162, pp. 125–151.

[16] J.-B. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift, in "*Systems and Control Letters*, vol. 18, no. 2, pp. 147–158, 1992.

[17] J. Godhavn and O. Egeland, "A lyapunov approach to exponential stabilization of nonholonomic systems in power form," in *IEEE Trans. on Auto. Control*, vol. 42, no. 7, pp. 1028–1032, 1997.

[18] W. E. Dixon, Z. P. Jiang, and D. M. Dawson, "Global exponential setpoint control of wheeled mobile robots: A lyapunov approach," in *Automatica*, vol. 36, pp. 1741–1746, 2000.

[19] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via lyapunov techniques," in *IEEE Rob. Auto. Mag.*, vol. 2, no. 1, pp. 27–35, 1995.

[20] C. Samson, "Mobile robot control. part 2: Control of chained systems and application to path following and time-varying point stabilization of wheeled vehicles," INRIA, Tech. Rep. 1994.

[21] P. Morin and C. Samson, "Motion control of wheeled mobile robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 799–826.

[22] F. Frenet, "Sur les courbes à double courbure," in *J. des mathématiques pures et appliquées*. Joseph Liouville, 1852, pp. 437–448.

[23] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills, in"*IEEE Rob. and Auto. Mag.*, vol. 12, no. 4, pp. 40–52, 2005.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2014)
September 18, 2014, Chicago, Illinois, USA

Embedded vision-based localization and model predictive control for autonomous exploration

Hélène Roggeman, Julien Marzat, Martial Sanfourche, Aurélien Plyer

***Abstract*— This paper presents a complete mobile robot architecture for autonomous exploration in a GPS-denied unknown environment. The platform considered is equipped with wheel encoders, stereo-vision and depth sensors, the measurements of which are fused within an extended Kalman filter for robust localization. An occupancy grid of the environment is built on-line for environment reconstruction and obstacle detection. Based on this map, a model predictive control scheme autonomously defines safe exploration trajectories, while taking into account interaction with the imaging sensors. Experimental results demonstrate the embedded computational capability of this vision-based control loop.**

***Index Terms*— autonomous robot, exploration, model predictive control, robust estimation, visual odometry**

I. Introduction

Autonomous robots are interesting contenders to carry out surveillance and exploration missions in unknown or dangerous environments. The algorithms embedded on these platforms should be able to simultaneously achieve mapping, localization, trajectory definition and control. Much research has been conducted on mobile robots equipped with either global localization sensors (GPS, Vicon systems) or relative ones such as a laser range finder, which is expensive and consumes a lot of energy. An alternative choice is to rely mainly on visual sensors, since recent progress in embedded computational capabilities now allows fast image processing and three-dimensional environment modeling. Such a vision-based architecture has been shown to be practical on a Micro Air Vehicle for an exploration mission in [1].

Vision-based ego-localization has reached a high level of maturity in the last decade with many applications in aerial robotics or mobile robotics. From a methodological point of view, two approaches are often opposed despite recent convergent trends: Visual Odometry (VO) ans V-SLAM (Visual Simultaneous Localization and Mapping). Basically, VO estimates the relative motion of the camera between two successive images [2], [3] (monocular, stereo or depth image) while V-SLAM resolves the global localization problem by building a globally-consistent maps of the environment [4], [5]. However, in a recent movement of convergence, VO algorithms tend to be like V-SLAM algorithms which the constraint of globally consistent would be released. For a more detailed review, we advise the reading of the recent two-parts tutorial of D. Scaramuzza and F. Fraundorfer [6], [7].

H. Roggeman, J. Marzat, M. Sanfourche, A. Plyer are with ONERA – The French Aerospace Lab, F-91123 Palaiseau, France, firstname.lastname@onera.fr

Model Predictive Control (MPC) is an appealing control strategy to build and follow trajectories in unknown environments. It uses a dynamical model of a system to predict its future state on a time horizon. Using this prediction, a possibly multi-modal performance criterion is optimized at each timestep for computing control inputs that achieve the required goals [8]. Unlike most path planning methods, this scheme is able to take into account an accurate system model as well as environment changes, since new control inputs are computed on the basis of measurements acquired in real time. This is why it has been used in a few recent works dealing with exploration by mobile robots [9]–[12].

This paper details an embedded vision-based MPC coupled loop to address autonomous exploration of an unknown area by a mobile robot (see model in Section II), with obstacle avoidance. The algorithmic architecture is presented in Figure 1. The visual odometry algorithm (eVO) is described in Section III-A, an EKF filter for fusion of wheel and visual odometries in Section III-B and the environment mapping strategy in Section IV. The MPC autonomous guidance scheme is explained in Section V and experimental results are reported in Section VI.

II. Robot architecture and model

The experimental platform considered is a four-wheel Robotnik Summit XL (Figure 2) equipped with a Kinect sensor and a stereo rig composed of two electronically synchronized USB cameras separated by a 18cm long baseline equipped with 5.5mm S-mount lens. The cameras are configured to capture VGA frames at 20Hz. The stereo image flow is processed to estimate the robot trajectory and to provide depth maps in outdoor environments while Kinect gives depth maps in indoor environment. These sensors are linked to an embedded PC with an Intel quad-core i7 processor and a Nvidia GPU (GT640) in charge of data processing.

The state vector for this system is $\mathbf{x} = [x, y, \theta]^T$, where (x, y) is the robot position and θ its heading angle. The control input vector is $\mathbf{u} = [v, \omega]^T$, where v is the robot linear speed and ω its angular speed. These are related to the controlled rotation speeds of the wheels by

$$\begin{cases} v = \frac{r}{2}(\omega^l + \omega^r) \\ \omega = \frac{r}{2L}(\omega^r - \omega^l) \end{cases} \quad (1)$$

where L is the half-axis length and r the wheel radius. The discrete-time dynamical model $\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ can be

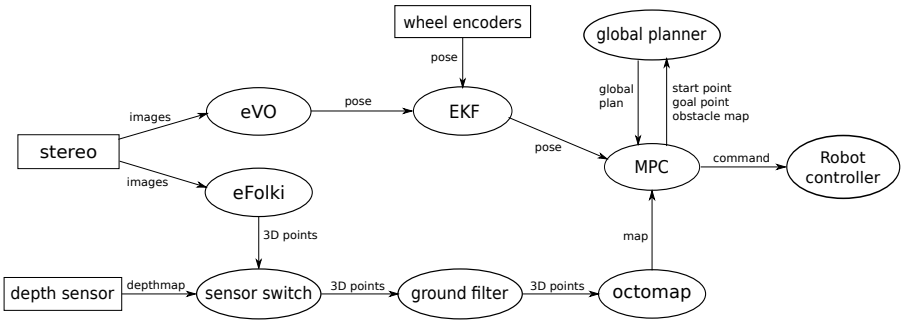


Fig. 1. Algorithmic architecture

written as

$$\begin{cases} x_k = x_{k-1} + t_e v_{k-1} \cos\theta_{k-1} \\ y_k = y_{k-1} + t_e v_{k-1} \sin\theta_{k-1} \\ \theta_k = \theta_{k-1} + t_e \omega_{k-1} \end{cases} \quad (2)$$

where t_e is the sampling period.

Fig. 2. Our mobile robotic platform equipped with two visual sensors (see text).

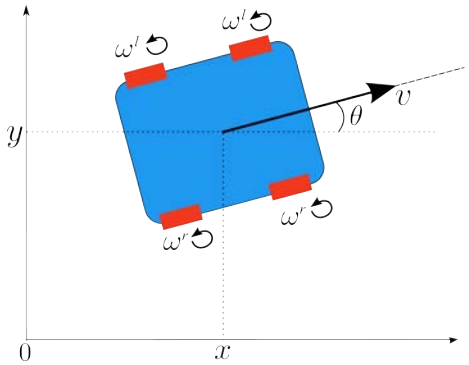


Fig. 3. Robot motion model.

III. Vision-based localization

A. Stereo Visual SLAM

Estimating the robot trajectory from its starting point relies on a stereo visual SLAM algorithm, called eVO [13]. Visual SLAM addresses the problem of ego-localization through the construction of a consistent map of the environment without prior information. Here we consider a map built from a limited number of 3D points located in a common frame defined as the robot frame at its starting point. Thanks to the calibrated stereo setup, the scale factor is known and the depth of landmarks in the sensor frame is measurable in a limited range (around 10 meters).

As in [5], the implemented algorithm is based on two tasks working in parallel: mapping and localization.

The mapping task consists in localizing new landmarks and discarding those which have gone out of sight. Except for the first stereo pair used to initialize the map, this operation is executed on-demand when the ratio of visible landmarks over the number of landmarks stored in the map falls below a threshold. The stereo pairs used to update the map are called *keyframes*. In practice, for each keyframe, a few hundred of Harris [14] or FAST [15] corners are extracted in the left image then matched by a coarse-to-fine exhaustive search along the epipolar lines. The relative-to-sensor localization of the novel landmarks is easily deduced from the disparity value. In order to be stored in the map, the landmarks are localized in the reference frame by applying the transformation corresponding to the current robot pose. In contrast to [1], [5], [16], the multiview refinement assuring a more precise landmark localization, inherent to SLAM techniques, were bypassed for computational reasons linked to the limited computational power of the original target robot. The map is then a collection of 3D points localized from a limited number of points of view. As shown on practical examples in [13], this scheme limits the estimation drift compared to standard dead-reckoning visual odometry techniques and delivers a pose at higher frequency. This operation takes approximately 20 ms on the embedded PC.

The localization task exploits unambiguous matchings between landmarks stored in the map and image features. These 2D-3D matchings are initialized during the mapping operation, then they are propagated along the visible landmark by tracking image features with KLT [14]. The full 6-degree-of-freedom pose (position and attitude) is deduced

from a two-steps procedure: the initial pose is estimated with the P3P algorithm using a RANSAC procedure [17], [18], which is then refined by minimizing the reprojection error of inlier matches selected by RANSAC. This task is carried out by considering only the left image. This operation is very fast (less than 10ms on the embedded PC).

This algorithm was benchmarked one year ago on the KITTI datasets [19], where it ranked first at the time of the submission. The measured drift on various sequences (KITTI or MAV) varies between 1 % and 2% of the trajectory length. In some intricate situation - the observed scene lacks of texture or when the robot navigates in highly dynamic environments-, the estimated trajectory can suddenly differ largely from the true trajectory. The proposed countermeasure consists in fusing by an EKF the pose parameters corresponding to a planar motion, denoted by $\mathbf{y}^{\text{evo}} = [x^{\text{evo}}, y^{\text{evo}}, \theta^{\text{evo}}]^{\text{T}}$ in what follows, with measurements coming from the wheel encoders.

B. EKF sensor fusion with outlier detection

As in [20], an extended Kalman filter (EKF) is used to cope with the nonlinear dynamical model (2). It provides an estimated state $\widehat{\mathbf{x}}$ and its associated covariance matrix $\mathbf{P}$ through the fusion of wheel encoders and stereovision measurements. It also allows outlier detection if the vision-based system faces momentarily an unstructured scene.

The estimated vehicle input vector, denoted by $\widehat{\mathbf{u}} = [\widehat{v}, \widehat{\omega}]^{\text{T}}$, can be computed from the wheel encoder measurements (averaged between the two wheels of each side) using (1). The prediction step of the filter based on this input information is thus equivalent to classical wheel odometry

$$\widehat{\mathbf{x}}_{k|k-1} = f\left(\widehat{\mathbf{x}}_{k-1|k-1}, \widehat{\mathbf{u}}_{k-1}\right) \tag{3}$$

The EKF framework also makes it possible to take into account the uncertainty related to wheel odometry (mostly due to wheel slip) in the input covariance noise matrix

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_v^2(k) & 0 \\ 0 & \sigma_\omega^2(k) \end{bmatrix} \tag{4}$$

The noise variances σ_v^2 and σ_ω^2 can be chosen either constant or proportional to the squared angular speeds of the wheels, since a larger error is to be expected with higher rates. The propagation of the state covariance is achieved by

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{\text{T}} + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^{\text{T}} \tag{5}$$

where

$$\mathbf{F}_{k-1} = \left.\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right|_{\widehat{\mathbf{x}}_{k-1|k-1}, \widehat{\mathbf{u}}_{k-1}} \tag{6}$$

$$\mathbf{G}_{k-1} = \left.\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\widehat{\mathbf{x}}_{k-1|k-1}, \widehat{\mathbf{u}}_{k-1}} \tag{7}$$

In the case of model (2), these matrices are equal to

$$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & 0 & -t_e \widehat{v}_{k-1} \sin\widehat{\theta}_{k-1} \\ 0 & 1 & t_e \widehat{v}_{k-1} \cos\widehat{\theta}_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\mathbf{G}_{k-1} = \begin{bmatrix} t_e \cos\widehat{\theta}_{k-1} & 0 \\ t_e \sin\widehat{\theta}_{k-1} & 0 \\ 0 & t_e \end{bmatrix} \tag{9}$$

The correction step of the EKF uses the visual odometry measurements $\mathbf{y}_k^{\text{evo}} = [x_k^{\text{evo}}, y_k^{\text{evo}}, \theta_k^{\text{evo}}]^{\text{T}}$, such that the innovation is equal to

$$\mathbf{r}_k = \mathbf{y}_k^{\text{evo}} - \widehat{\mathbf{x}}_{k|k-1} \tag{10}$$

and its covariance is

$$\mathbf{S}_k = \mathbf{P}_{k|k-1} + \mathbf{R}_k \tag{11}$$

where $\mathbf{R}_k = \text{diag}\left(\sigma_x^2, \sigma_y^2, \sigma_\theta^2\right)$, with noise variances related to the accuracy of eVO mentioned in Section III-A.

Finally, the updated state and covariance are

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{S}_k^{-1}\mathbf{r}_k \tag{12}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I}_3 - \mathbf{P}_{k|k-1}\mathbf{S}_k^{-1}\right)\mathbf{P}_{k|k-1} \tag{13}$$

The vision-based system may generate large localization errors if the scene is not structured enough (e.g., a plain wall). If undetected, this may cause the output of the EKF to be erroneous and as a consequence endanger the robot mission and safety. Following sensor fault diagnosis techniques, a robust outlier detection scheme is considered by monitoring the innovation (10). If it exceeds some threshold $\boldsymbol{\epsilon}_{\mathbf{r}}$, which can be selected proportional to the innovation covariance (11), then only the prediction step is keeped as the state estimate (i.e., classical wheel odometry). The visual odometry system is then re-initialized at each timestep with the new predicted state, until the innovation falls below the threshold. The filter then proceeds back to the update step.

Algorithm 1 Robust EKF

At each timestep k

1) Compute $\widehat{\mathbf{u}}_{k-1}$ from wheel encoders
2) Predict $\widehat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ using (3) and (4)
3) Using stereovision measurements $\mathbf{y}_k^{\text{evo}}$, Compute the innovation using (10) and (11)
4) **if** $\mathbf{r}_k > \boldsymbol{\epsilon}_{\mathbf{r}}$
   Use $\widehat{\mathbf{x}}_{k|k-1}$ as the estimated state and reset eVO
   else
   Compute the update $\widehat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ from (12) and (13)
5) Go back to 1 with $k \leftarrow k+1$

The behavior of the EKF is illustrated in Figure 4 on real data. It can be seen that the filter successfully rejects visual odometry outliers due to the unstructured scene (see detection signal in Figure 5) and also corrects the drift of wheel odometry when visual information is available.

IV. Online environment mapping

Online environment mapping consists in aggregating 3D measurements, in particular depthmaps, in a global model thanks to the poses estimated by the EKF previously described. Two kinds of depthmap are considered here : those captured by the depth active camera end those calculated from stereo pair by the dense optical flow algorithm called eFolki [21].

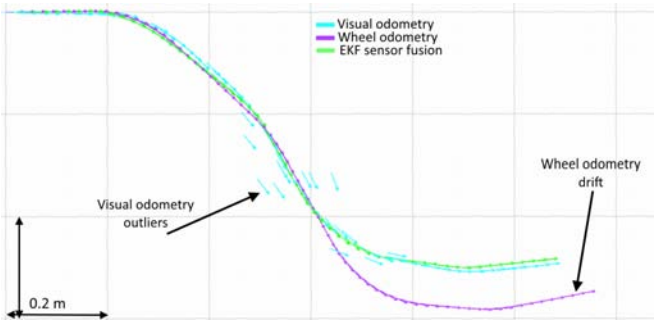


Fig. 4. Robust EKF sensor fusion (grid step size 1m)

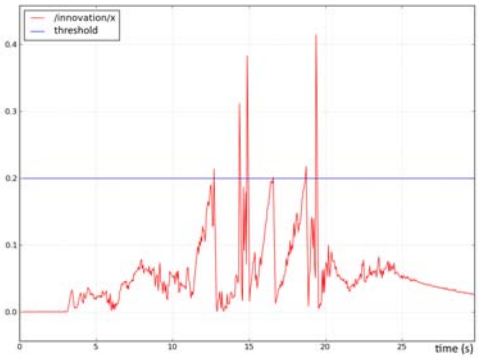


Fig. 5. Outlier detection via the innovation signal

A. Dense stereo-matching

eFolki is an evolution of the standard Lucas-Kanade [22] optical flow algorithm. Like for the dense LK algorithm, the basic problem is to register local windows centred around each image pixel $\mathbf{x}$ by minimizing a SSD (Sum of Squared Difference) criterion over a 2D motion vector $\mathbf{u}(\mathbf{x})$:

$$\sum_{\mathbf{x}'} w(\mathbf{x}'-\mathbf{x})\left(I_1(\mathbf{x}')-I_2(\mathbf{x}'+\mathbf{u}(\mathbf{x}))\right)^2, \qquad (14)$$

where w is a separable weighting function, uniform or Gaussian, of limited support $\mathcal{W}$, typically a square window parametrized by its radius r. The minimization of the criterion (14) is done by iterative Gauss-Newton coarse-to-fine pyramidal strategy as in a classical implementation of LK. However, using the first order expansion described in [23], an iteration can be completed with only one image interpolation per pixel, while the well-known PyramiLK algorithm [24] requires several image interpolations by pixel. The resulting code is remarkably fast on massively parallel architecture such as GPU.

The motion estimation greatly depends on the local image texture and fails in case of illumination changes. So, to achieve a high level of robustness in real-world environment, eFolki uses a Rank Transform [25] applied to the images before SSD minimization. In practice, the motion estimation in low texture areas is more noisy but the motion estimation stays convergent.

B. Multisensor occupancy grid

The default device is the depth active camera. However, in many situations especially outdoor, the depthmaps delivered by this sensor are incomplete or even empty. So, a mechanism based on the density of depthmaps acquired by active camera permits to switch automatically from active camera to passive stereorig.

The 3D point clouds output by this "sensor switch" (see figure 1) are then individually filtered for removing the ground plane thanks to a RANSAC-based search algorithm [26]. A prior is used to find the plane whose the perpendicular angle is near vertical.The result is then inserted in an Octomap model [27].

Octomap is a well-known implementation of a volumetric occupancy grid using an octree data structure. Each element of this data structure contains two probabilities, the occupancy one and the free one. These quantities are updated by a ray-tracing technique emulating a depth sensor: the ray between the sensor and a 3D point uprise the free probability of intersected voxels while the end-point uprise the occupancy probabilty of its corresponding voxel. Using octree offers a very efficient memory structure but the update strategy is very computationally intensive. Only low framerate (near 1Hz) are available due to this limitation.

After the Octomap update, we project the map in a three state 2D map : {unexplored, free, occupied}, and use it for autonomous guidance.

V. Guidance for autonomous vision-based exploration

Model Predictive Control (MPC) is a usual method for the guidance of autonomous vehicles in complex environments, taking into account differential constraints [28]. Convergence results for this receding horizon strategy can be found in [29].

A. MPC principles

Considering the current robot position $\mathbf{x}_k$, a sequence $\mathbf{U}_k$ of H_c control inputs is defined as well as the resulting sequence $\mathbf{X}_k$ of H_p predicted states using model (2).

$$\mathbf{U}_k = \{\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+H_c-1}\} \qquad (15)$$

$$\mathbf{X}_k = \{\mathbf{x}_{k+1}, \mathbf{x}_{k+2}, \ldots, \mathbf{x}_{k+H_p}\} \qquad (16)$$

Finite control horizon H_c and prediction horizon H_p are considered for tractability. If $H_c < H_p$, control inputs at timesteps larger than H_c should be considered either constant (for linear speed) or null (for angular speed). Each control input vector $\mathbf{u}_k$ is bounded within the compact set $\mathcal{U}$ as

$$\begin{aligned} -v_{\max} < v_k < v_{\max} \\ -\omega_{\max} < \omega_k < \omega_{\max} \end{aligned} \qquad (17)$$

and thus $\mathbf{U}_k \in \mathcal{U}^{H_c}$. A cost function $J(\mathbf{U}_k, \mathbf{X}_k)$ should be defined to quantify the mission requirements and constraints. The following optimization problem is then solved at each timestep k to find the optimal control sequence.

$$\begin{aligned} \mathbf{U}_k^* = \arg\min_{\mathbf{U}_k \in \mathcal{U}^{H_c}} J(\mathbf{U}_k, \mathbf{X}_k) \\ \text{with } \mathbf{x}_i \text{ satisfying (2)}, \\ \forall i \in [k+1; k+H_p] \end{aligned} \qquad (18)$$

The first component $\mathbf{u}_k^*$ of this sequence is then applied on the robot and the procedure is repeated at the next timestep using the new information gathered in the motion.

B. MPC costs

The main cost function is defined as

$$J = w_{\text{obs}} J_{\text{obs}} + w_u J_u + w_{\text{expl}} \left(b_e J_{\text{expl}} + \overline{b_e} J_{\text{nav}} \right) \quad (19)$$

where

- J_{obs} is the obstacle avoidance cost,
- J_u regulates the linear and angular speeds,
- J_{expl} is the exploration cost,
- J_{nav} the waypoint navigation cost.

J_u and J_{obs} are always active, while J_{expl} and J_{nav} correspond to separate mission phases and will never be active simultaneously: b_e is a boolean which is equal to 1 when exploration is active and to 0 when the robot should switch to waypoint navigation.

As detailed in what follows, the sub-costs $J_\bullet$ are all of unit norm. The weights $w_\bullet$ should be chosen to reflect their relative importance (see Section VI).

1) Obstacle avoidance: The obstacle avoidance cost penalizes the intersection of each predicted position in $\mathbf{X}_k$ with existing obstacles in the current occupancy grid. A morphological Euclidean distance transform is applied on the occupancy grid to obtain a distance map. This computation needs only to be performed on the area which can be reached by the vehicle on the prediction horizon, starting from its current position. The user-defined borders of the zone to be explored are also considered as obstacles.

Based on the map containing the distance of any vehicle position to the nearest obstacle, the following penalty function [30] is considered

$$f_o\left(\mathbf{x}_k\right) = \frac{1 - \tanh\left(\alpha\left(d_o\left(\mathbf{x}_k\right) - \beta\right)\right)}{2} \quad (20)$$

$$\alpha = \frac{6}{d_{\text{des}} - d_{\text{sec}}} \quad (21)$$

$$\beta = \frac{1}{2}\left(d_{\text{des}} + d_{\text{sec}}\right) \quad (22)$$

where

- $d_o\left(\mathbf{x}_k\right)$ is the distance between the vehicle position at time k and the nearest obstacle.
- d_{des} is a desired distance to obstacles, beyond which they are ignored.
- d_{sec} is a safety distance that must not be reached by the vehicle, leading to full penalty.

The continuous function f_o is designed to be equal to 1 when $d_o < d_{\text{sec}}$ and to zero for $d_o > d_{\text{des}}$ (Figure 6). The obstacle avoidance cost is computed as

$$J_{\text{obs}} = \frac{1}{H_p} \sum_{i=k+1}^{k+H_p} f_o\left(\mathbf{x}_i\right). \quad (23)$$

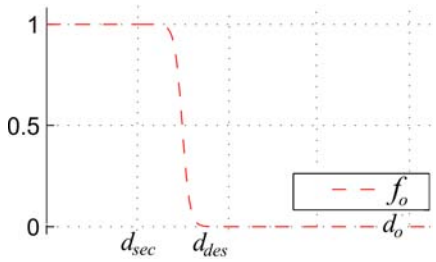


Fig. 6. Penalty function for obstacle avoidance

2) Control cost: J_u encompasses the regulation of the sequence of linear speeds to a nominal value v_0 (possibly negative) and penalization of large angular speeds, on the control horizon.

$$J_u = \frac{1}{2H_c} \sum_{i=k}^{H_c-1} \left(\frac{\omega_i^2}{\omega_{\max}^2} + \frac{w_v \left(v_i - v_0\right)^2}{\left(\|v_0\| + \|v_{\max}\|\right)^2} \right) \quad (24)$$

3) Vision-based exploration: The objective of the exploration mission is to maximize the area seen during the mission, within user-defined borders, while avoiding already explored locations to reduce the duration of the mission [12]. The definition of an exploration trajectory is tightly coupled with the characteristics of the embedded sensor: here, the vision sensor has a triangular field of view with the same heading angle as the vehicle (see Figure 7).

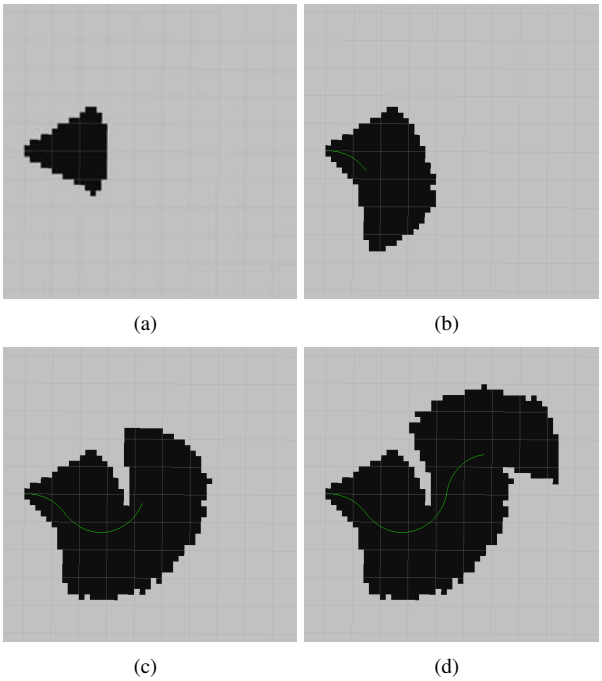


Fig. 7. Area explored (in black) for a sample trajectory (in green)

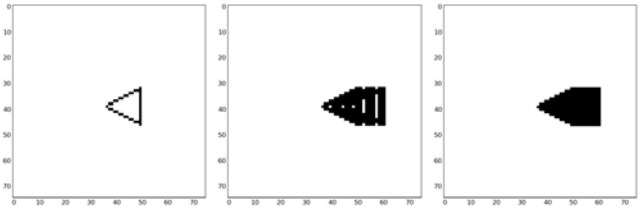
A copy of the occupancy grid is considered for recording exploration progress, obstacles being considered as explored boxes. The explored grid at timestep k is denoted by $\mathbf{G}(k)$, discretized with a uniform spatial step d_{grid} and of dimensions $l \times h$ (which could be limited to the area that can be

reached by the vehicle on the prediction horizon). Each of its components $g_{i,j}(k)$ takes the value 1 if the corresponding location has been explored and 0 otherwise, i and j being the grid coordinates.

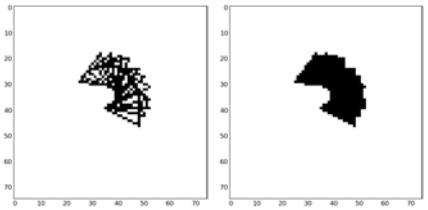
For any predicted trajectory corresponding to a control input sequence $\mathbf{U}_k$, the best exploration cost should be the one that favors the highest number of unexplored locations. To update the exploration grid, the intersection between its cells and the sensor field of view at each predicted position is computed by applying a contour detection algorithm (Bresenham line drawing) followed by a morphological closing operation on the predicted sequence (see Figure 8). The grid situation updated with a given predicted trajectory until time $k + H_p$ is denoted by $\mathbf{G}(k + H_p)$. The exploration cost to be minimized (and thus negative) is then

$$J_{\text{expl}} = \frac{d_{\text{grid}}^2}{H_p \mathcal{A}_{\text{fov}}} \sum_{i=1}^{l} \sum_{j=1}^{h} [g_{i,j}(k) - g_{i,j}(k + H_p)] \qquad (25)$$

where $\mathcal{A}_{\text{fov}}$ is the area of the sensor field of view.



(a) Two prediction steps



(b) Area covered on the prediction horizon

Fig. 8. Morphological operations for prediction of explored area, taking into account the sensor field of view

4) Navigation: The following cost is built to reach a waypoint $\mathbf{p}_w$, given the predicted positions of the robot $\mathbf{p}_i = [x_i, y_i]^{\text{T}}$,

$$J_{\text{nav}} = \frac{1}{H_p v_{\max} t_e} \sum_{i=k+1}^{k+H_p} \|\mathbf{p}_w - \mathbf{p}_i\|^2 \qquad (26)$$

If the waypoint is unreachable by the vehicle on the prediction horizon with the obstacle-free trajectory of minimal cost, i.e., $\|\mathbf{p}_w - \mathbf{p}_{k+H_p}\| > \epsilon_{\text{d}}$ with ϵ_{d} a small distance threshold, then a grid-based A* path is computed and the waypoint $\mathbf{p}_w$ is re-assigned to the farthest reachable point on this roughly planned trajectory. Another solution could consist in increasing the size of the prediction horizon to compensate for the observed distance to the waypoint.

C. Mission supervision

The MPC strategy is able to govern the exploration mission and obstacle avoidance within the reach of the prediction horizon. For large maps where the distance covered during the prediction horizon is much smaller than the size of the user-defined map to explore, a higher layer supervision process is defined.

Detection of mission ending is achieved by assessing whether the MPC cost is similar for all predicted trajectories and the optimization process results in null control inputs on successive timesteps. In this case, if there remains unexplored areas in the exploration grid, the robot switches to waypoint navigation to reach the nearest such location. Once it is reached, exploration resumes. If the entire map has been explored, then the starting point is designated as a waypoint and the mission ends when it is reached.

VI. Experiments

A. MPC implementation

As in [30], the same control input is applied on all the control horizon for computational tractability, thus the optimization problem (18) has only two variables to find, v^* and ω^*, and $\forall i \in [k; k + H_c - 1]$, $\mathbf{u}_i^* = [v^*, \omega^*]^{\text{T}}$. For steps between H_c and H_p, the linear speed remains equal to v^* while the angular speed is set to zero. Tuning parameters of the EKF and MPC algorithms are indicated in Table I.

TABLE I
MPC and EKF parameters

$w_{\text{obs}} = 40$	$t_e = 0.25$s (MPC)	$v_{0/\max} = 0.6$m/s	$\sigma_v^2 = 10^{-2}$
$w_u = 1$	$t_e = 0.05$s (EKF)	$\omega_{\max} = 0.6$rad/s	$\sigma_\omega^2 = 10^{-2}$
$w_v = 5$	$H_c = 10$	$d_{\text{grid}} = 0.2$m	$\sigma_{x/y}^2 = 10^{-2}$
$w_{\text{expl}} = 15$	$H_p = 20$	$\mathcal{A}_{\text{fov}} = 4.5\text{m}^2$	$\sigma_\theta^2 = 10^{-3}$

The deterministic global optimization algorithm DIRECT [31] was used for solving (18), using the *nlopt* package. This strategy was always able to find a result in less than 0.1s on the embedded computer, which compares favorably with the duration of the MPC timestep.

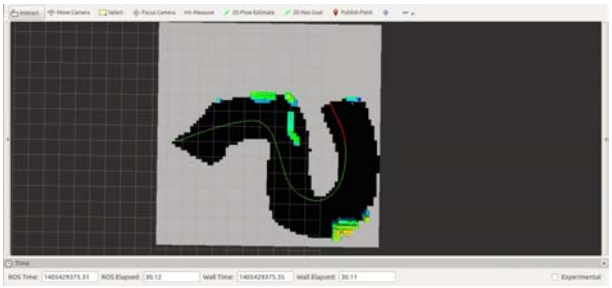
B. Preliminary Experimental results

We have experimented the proposed system in the parking of our research center. The mission consists in exploring an squared area of $15m \times 15m$.

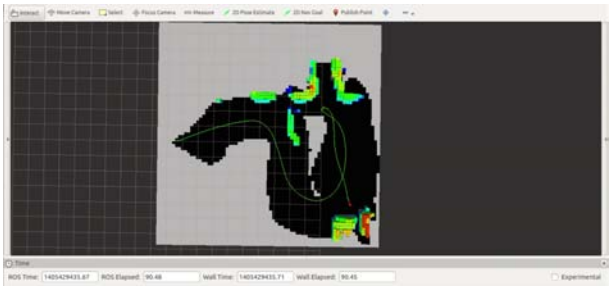
In a first step, we have validated the control functions by using only he wheel encoders. The figure 9 shows the mission progress at 4 different moments when :

- the robot goes behind an obstacle (subfigures (a) and (b)).
- the robot is blocked by an environment element (subfigure (c)). ;
- the robot selects a waypoint and switches to waypoint navigation. The waypoint is depicted by the pink point (subfigure (d)).
- the robot stops the mission (subfigure (e)).

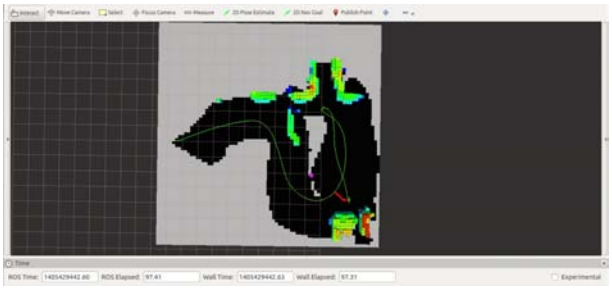
(a) Exploration behind the obstacle



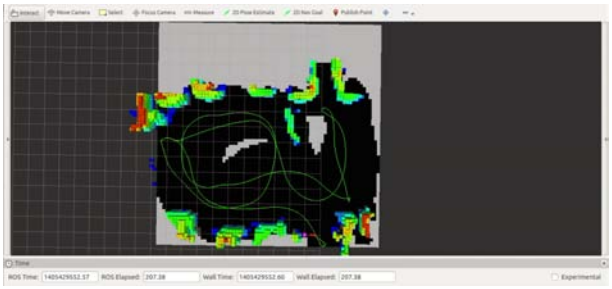
(b) Exploration behind the obstacle



(c) Robot is blocked



(d) Switch to waypoint navigation



(e) End of the mission

Fig. 9. Progress of the parking exploration at 4 different moments (Screenshot from Ros Rviz). Here, the robot localization uses only the wheel encoders.

In a second time, the robot trajectory is estimated by Visual-SLAM. The preliminary results are shown in figure 10 are equivalent to the ones obtained by wheel-odometer. In a second time, the robot trajectory is estimated by Visual-SLAM. The preliminary results are shown in figure 3 are equivalent to the ones obtained by wheel-odometer. The comparison of the second screenshot with the third one highlights a limited drift. This can be explained by a too small angular error between vision-based estimation and gyrometer-based prediction for being recovered by EFK.

VII. Conclusions and perspectives

A vision-based algorithmic architecture to tackle fully autonomous exploration missions with a mobile robot has been presented. It relies only on measurements coming from visual sensors and wheel encoders, which are fused in an EKF. The resulting state estimate is combined with visual information to build a map of the environment, which is exploited by a MPC scheme to define trajectories favouring unexplored locations without obstacle collision.

Preliminary experimental results have highlighted the interest of the approach and its fully embedded capabilities. There is room for improvement in the vision, mapping and control algorithms, which will receive further attention in the future.

In particular, the precision of the stereo visual SLAM could be improved by fusing with the IMU sensor and by updating the landmarks localization in a multi-view refinement strategy. In parallel, we are working on the environment model to increase the update rate of the 3D model.

The interaction between the visual sensors and trajectory definition could be further enhanced by selecting control inputs that lead to areas where vision-based localization would not be endangered, while here the sensor field of view was only taken into account for exploration purpose.

References

[1] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, “Vision-based autonomous mapping and

(a) (b) (c)

Fig. 10. Progress of the parking exploration at 3 different moments (Screenshot from Ros Rviz) with robot trajectory estimated by vision.

exploration using a quadrotor MAV," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal*, 2012, pp. 4557–4564.

[2] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision," in *IEEE ICRA*, vol. 4, San Francisco, USA, April 24-28 2000, pp. 3519–3524.

[3] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 3946–3952.

[4] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the International Symposium on Mixed and Augmented Reality, Nara, Japan*, 2007, pp. 225–234.

[6] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part i - the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, December 2011.

[7] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii - matching, robustness, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, June 2012.

[8] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, "State and output feedback nonlinear model predictive control: An overview," *European Journal of Control*, vol. 9, no. 2-3, pp. 190–206, 2003.

[9] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM using model predictive control and attractor based exploration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, 2006, pp. 5026–5031.

[10] T. M. Howard, C. J. Green, and A. Kelly, "Receding horizon model-predictive control for mobile robot navigation of intricate paths," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, A. Howard, K. Iagnemma, and A. Kelly, Eds. Springer Berlin Heidelberg, 2010, vol. 62, pp. 69–78.

[11] L. Bascetta, D. Cucci, G. Magnani, M. Matteucci, D. Osmankovic, and A. Tahirovic, "Towards the implementation of a MPC-based planner on an autonomous all-terrain vehicle," in *Workshop on Robot Motion Planning: Online, Reactive, and in Real-time, The IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal*, 2012, pp. 7–12.

[12] T. Gorecki, H. Piet-Lahanier, J. Marzat, and M. Balesdent, "Cooperative guidance of UAVs for area exploration with final target allocation," in *Proceedings of the 19th IFAC Symposium on Automatic Control in Aerospace, Würzburg, Germany*, 2013, pp. 260–265.

[13] M. Sanfourche, V. Vittori, and G. L. Besnerais, "eVO: A realtime embedded stereo odometry for mav applications," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, 2013, pp. 2107–2114.

[14] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jerusalem, Israel*, 1994, pp. 593–600.

[15] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision, Graz, Austria*, vol. 1, 2006, pp. 430–443.

[16] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[18] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA*, 2012, pp. 3354–3361.

[20] L. Teslić, I. Škrjanc, and G. Klančar, "EKF-based localization of a wheeled mobile robot in structured environments," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 2, pp. 187–203, 2011.

[21] A. Plyer, G. L. Besnerais, and F. Champagnat, "Real-time Lucas-Kanade optical flow estimation for real-world applications," *Journal of Real Time Image Processing*, 2014.

[22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, Canada*, 1981, pp. 674–679.

[23] G. Le Besnerais and F. Champagnat, "Dense optical flow by iterative local window registration," in *IEEE International Conference on Image Processing 2005*. IEEE, 2005, pp. I–137.

[24] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Technical report. Intel Corporation, Tech. Rep., 2001.

[25] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," *European Conference on Computer Vision*, pp. 151–158, 1994.

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011, pp. 1–4.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[28] L. Singh and J. Fuller, "Trajectory generation for a UAV in urban terrain, using nonlinear MPC," in *Proceedings of the American Control Conference, Arlington, VA, USA*, vol. 3, 2001, pp. 2301–2308.

[29] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding-horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 5, pp. 776–783, 2001.

[30] Y. Rochefort, H. Piet-Lahanier, S. Bertrand, D. Beauvois, and D. Dumur, "Model predictive control of cooperative vehicles using systematic search approach," *Control Engineering Practice*, 2014, in press.

[31] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2014)
September 18, 2014, Chicago, Illinois, USA

Planar Motion Estimation from Three-Dimensional Scenes*

Xuebo Zhang[1], Congyuan Wang[1], Yongchun Fang[1] and Kexin Xing[2]

***Abstract*— In this brief, we will present some insightful properties for pure planar motion estimation through two-view geometry. Compared with general motion, these distinct properties render a new multi-homography-based planar motion estimation method, which achieves robust fusion of the estimation results from multiple homographies on the basis of a ranking mechanism. Simulation and experimental results demonstrate the effectiveness of the proposed approach.**

I. INTRODUCTION

Wheeled mobile robots have promising applications in many fields, such as family service, automatic transportation in logistics centers, and so on. To increase the intelligence of mobile robots, some sensors are usually installed onboard the robot for environmental perception. Typically, visual sensors are more and more popular for mobile robots, due to its rich information, non-contact nature, and decreasing cost. In practical applications, such as visual navigation, visual servoing, and visual SLAM (simultaneous localization and mapping), one import component is to use visual images to estimate the motion parameters of mobile robots (as well as the scene structure sometimes) in a robust way. Since planar motion is one of the dominate motion type for most mobile robots, this paper considers the planar motion estimation problem from the two-view geometry.

Over the last several decades, one similar problem called "structure from motion (SfM)", has attracted a significant amount of attention, and many approaches are proposed in the literature [1]. Meanwhile, in the area of visual servoing, there exist two typical motion estimation methods through two-view geometry, which are homography-based and epipolar geometry-based methods, respectively [2], [3]. In the scenario of multiple planes, an approach for motion estimation is proposed in [4] and [5], wherein one interesting contribution is that three points are enough to estimated the subsequent homography if the first one is known. Yet, it should be noted that these aforementioned approaches are mainly focused on general motions other than planar motion, and little research that fully exploits the underlying planar motion constraint has been reported.

*This work is supported in part by National Natural Science Foundation of China under Grant 61203333 and 61202203, in part by Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20120031120040, in part by Tianjin Natural Science Foundation under Grant 13JCQNJC03200.

[1]Xuebo Zhang, Congyuan Wang, Yongchun Fang is with the Institute of Robotics and Automatic Information System and also Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin, 300071, China {zhangxb,wangcy,yfang@robot.nankai.edu.cn}

[2]Kexin Xing is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023 China xkx@zjut.edu.cn

To make a full use of the planar motion constraint, the work in [6] first employs the distance invariance property of rigid objects to obtain the depth distribution, and then linear least square method is further adopted to compute the motion parameters. A matrix factorization method is further proposed in [7] to use the planar motion constraint to obtain a rank 3 constraint of the measurement matrix, rather than rank 4 for general motions. Different from the static motion estimation from two-view geometry, the work in [8] introduces an novel dynamic motion estimation approach designed by using the nonlinear control theory and one-dimensional (1D) trifocal tensors. Though these methods are somehow robust to the image noise, they do not handle the outliers from mismatching of feature points.

In our previous work [9], we present an analytical homography decomposition method to obtain the motion parameters, however, feature points are required to be on a coplanar scene. Afterwards, we propose a direct motion estimation method without matrix estimation and decomposition, and it can be applied in both planar and non-planar scenes [10]; Yet, its robustness to image noise and outliers should be further investigated. Some works use the epipolar geometry for visual servoing tasks [11], yet there may exist the short-baseline degeneracy problem and it is required that the scene points should be non-planar.

In this paper, we present some insightful properties of the planar motion estimation problem. Motivated by these properties, a new muti-homography-based motion estimation methodology are proposed by using point correspondences. Different from existing approaches, the proposed approach successfully fuse the information from multiple homographies with robustness to image noises and outliers, and it can be applied for both planar or non-planar scenes. The proposed approach has the following merits: 1) compared with the traditional single homography matrix approach, the scene can be planar or nonplanar; 2) compared with the essential matrix approach, short baseline degeneracy does not exist; 3) robustness to outliers are achieved; 4) the algorithm has a good parallel structure, which makes it possible to be implemented using GPU to increase the efficiency, especially when the number of the feature points are large.

II. PROBLEM STATEMENT

In this paper, the camera undergoes a "constrained planar motion" which means that the image plane is perpendicular to the motion plane, which usually happens in vision-based mobile robotic systems [12]. For simplicity, we use "planar motion" instead of "constrained planar motion" in the rest of the paper.

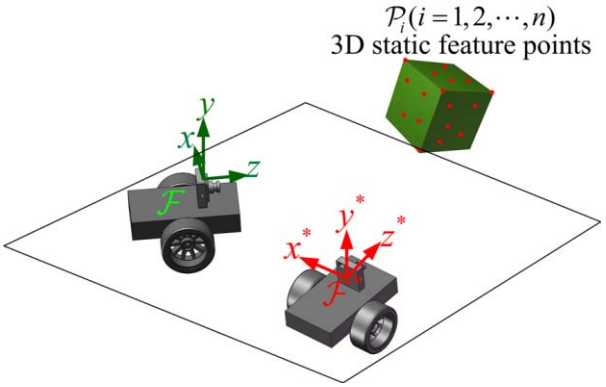


Fig. 1. Pure planar motion of an onboard camera

As shown in Figure 1, we consider two camera frames $\mathcal{F}$ and $\mathcal{F}^*$ at different poses, and a set of N feature points $\mathcal{P}_i$ $(i = 1,\ 2,\ \cdots, N)$ in the scene. These feature points may be coplanar or non-coplanar.

Let $\boldsymbol{P}_i = [X_i\ Y_i\ Z_i]^T$ and $\boldsymbol{P}_i^* = [X_i^*\ Y_i^*\ Z_i^*]^T$ denote the three dimensional camera coordinates of the point $\mathcal{P}_i$ in $\mathcal{F}$ and $\mathcal{F}^*$, with $\boldsymbol{p}_i = [u_i\ v_i\ 1]^T$ and $\boldsymbol{p}_i^* = [u_i^*\ v_i^*\ 1]^T$ being the corresponding homogeneous image pixel coordinates. Let K denote the camera internal matrix, the following relationship holds for a pin-hole camera model:

$$\boldsymbol{p}_i = \frac{1}{Z_i} K \boldsymbol{P}_i, \quad \boldsymbol{p}_i^* = \frac{1}{Z_i^*} K \boldsymbol{P}_i^*. \tag{1}$$

In addition, we introduce the normalized image coordinates $\boldsymbol{m}_i$ and $\boldsymbol{m}_i$ as follows:

$$\boldsymbol{m}_i = [x_i\ y_i\ 1]^T = \frac{1}{Z_i}\boldsymbol{P}_i, \tag{2}$$

$$\boldsymbol{m}_i^* = [x_i^*\ y_i^*\ 1]^T = \frac{1}{Z_i^*}\boldsymbol{P}_i^* \tag{3}$$

Note that the normalized image coordinates $\boldsymbol{m}_i$ and $\boldsymbol{m}_i^*$ are easy to be obtained from the image coordinates $\boldsymbol{p}_i$ and $\boldsymbol{p}_i^*$ through the camera model as follows:

$$\boldsymbol{m}_i = K^{-1}\boldsymbol{p}_i, \quad \boldsymbol{m}_i^* = K^{-1}\boldsymbol{p}_i^*. \tag{4}$$

From some geometrical analysis, the camera coordinates $\boldsymbol{P}$ and $\boldsymbol{P}^*$ are related with the motion of the robot between $\mathcal{F}$ and $\mathcal{F}^*$ as follows:

$$\boldsymbol{P}_i = R\boldsymbol{P}_i^* + \boldsymbol{T} \tag{5}$$

where the rotation matrix R and the translation vector $\boldsymbol{T}$ are

$$R = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} t_x \\ 0 \\ t_z \end{bmatrix} \tag{6}$$

with θ being the rotational angle between $\mathcal{F}$ and $\mathcal{F}^*$. For ease of subsequent analysis, we parameterize the translation vector $\boldsymbol{T}$ using the its amplitude t and the direction angle α as

$$t = \sqrt{t_x^2 + t_z^2}, \quad \alpha = \text{atan2}(t_z,\ t_x). \tag{7}$$

Then, the translation vector is rewritten as follows:

$$\boldsymbol{T} = \begin{bmatrix} t\cos\alpha \\ 0 \\ t\sin\alpha \end{bmatrix} \tag{8}$$

Two-view planar motion estimation problem: given two images captured at two different poses by a camera undergoing planar motion, the goal of the motion estimation in this paper is to recover the rotation angle and the scaled translation (translation direction) between these two poses. It can be regarded as a partial relative pose estimation problem.

Since the proposed approach is based on point features and calibrated camera model, the inputs of the algorithm include K, $\boldsymbol{p}_i$, $\boldsymbol{p}_i^*$ with $i = 1,\ 2,\ \cdots,\ N$, and the outputs of the algorithm are the rotation angle θ and the translation direction α.

III. SOME RESULTS ON PROPERTIES OF PLANAR MOTION ESTIMATION

Compared to the general motion, the pure planar motion considered in this paper has several distinct properties, which actually motivates the proposed multi-homography-based motion estimation approach in section IV. These discovered properties are actually distinct and interesting to make the motion estimation theory based on homography and epipolar geometry more complete to be applicable in more scenarios.

A. Representation of Euclidean Homography and Essential Matrix for Planar Motion

Euclidean homography. When the observed feature points are coplanar, the relationship between image coordinates of feature points in two different images can be described using homography. Meanwhile, the mapping between normalized image coordinates in $\mathcal{F}$ and $\mathcal{F}^*$ is called Euclidean homography denoted by $H \in \mathbb{R}^{3\times3}$ as

$$\boldsymbol{m}_i = \lambda_i H \boldsymbol{m}_i^*, \tag{9}$$

with λ_i being a scale factor and H being of the following form [9]:

$$H = R + \frac{\boldsymbol{T}\boldsymbol{n}^{*T}}{d^*} \tag{10}$$

where $\boldsymbol{n}^* = [n_1\ \ n_2\ \ n_3]^T \in \mathbb{R}^3$ denotes the normal vector of the planar scene expressed in $\mathcal{F}^*$, and $d^* \in \mathbb{R}$ is the distance of the origin of $\mathcal{F}^*$ to the scene plane. By substituting (6) and (8) into (11) yields that

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ 0 & 1 & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \tag{11}$$

where

$$h_{11} = \cos\theta + tn_1\cos\alpha, \tag{12}$$

$$h_{12} = tn_2\cos\alpha, \tag{13}$$

$$h_{13} = \sin\theta + tn_3\cos\alpha, \tag{14}$$

$$h_{31} = -\sin\theta + tn_1\sin\alpha, \tag{15}$$

$$h_{32} = n_2\sin\alpha t, \tag{16}$$

$$h_{33} = \cos\theta + tn_3\sin\alpha. \tag{17}$$

It is seen from (11) that the homography H describes the underlying relationship between the Euclidean homography and the motion parameters $\theta,\ t_s,\ \alpha$, as well as the parameters $n_1,\ n_2,\ n_3$ from the geometry planar constraint.

Essential matrix. The image coordinates of any scene feature obey the epipolar constraint for the two-view geometry. The essential matrix, denoted by $E \in \mathbb{R}^{3\times 3}$ is a good way to describe the epipolar constraint using the normalized image coordinates $\boldsymbol{m}_i$ and $\boldsymbol{m}^*$ as follows:

$$\boldsymbol{m}_i^T E \boldsymbol{m}_i^* = 0. \tag{18}$$

From the geometrical analysis, it is shown that the essential matrix relates to the motion parameters in the following way:

$$E = [\boldsymbol{T}]_\times R = \begin{bmatrix} 0 & e_{12} & 0 \\ e_{21} & 0 & e_{23} \\ 0 & e_{32} & 0 \end{bmatrix} \tag{19}$$

where $e_{12} = -t_z,\ e_{21} = t_z\cos\theta + t_x\sin\theta,\ e_{23} = t_z\sin\theta - t_x\cos\theta,\ e_{32} = t_x$, and $[\boldsymbol{T}]_\times$ is the skew-symmetric matrix constructed by elements of $\boldsymbol{T}$ as

$$[\boldsymbol{T}]_\times = \begin{bmatrix} 0 & -t_z & 0 \\ t_z & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}. \tag{20}$$

B. Some Results on Properties of Planar Motion Estimation

Property 1. For a calibrated camera undergoing planar motion, two-view image coordinates of three non-collinear feature points with $y_i \neq 0$ are sufficient to estimate the Euclidean homography matrix.

Proof. In the planar motion case, the Euclidean homography matrix has the form of (11), from which it is seen that a total number of six unknown parameters including $h_{11},\ h_{12},\ h_{13},\ h_{31},\ h_{32},\ h_{33}$ need to be estimated from the image coordinates.

Using the normalized image coordinates of each feature point, (9) can be adopted to obtain three equality constraints as follows:

$$x_i = \lambda_i(h_{11}x_i^* + h_{12}y_i^* + h_{13}), \tag{21}$$
$$y_i = \lambda_i y_i^*, \tag{22}$$
$$1 = \lambda_i(h_{31}x_i^* + h_{32}y_i^* + h_{33}). \tag{23}$$

After canceling the effect of the scale factor λ_i, two independent constraints can be obtained for the elements of the homography as:

$$y_i^* x_i = y_i(h_{11}x_i^* + h_{12}y_i^* + h_{13}) \tag{24}$$
$$y_i^* = y_i(h_{31}x_i^* + h_{32}y_i^* + h_{33}), \tag{25}$$

which are rewritten as

$$\begin{bmatrix} y_i x_i^* & y_i y_i^* & y_i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \end{bmatrix} = y_i^* x_i \tag{26}$$

$$\begin{bmatrix} y_i x_i^* & y_i y_i^* & y_i \end{bmatrix} \begin{bmatrix} h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = y_i^* \tag{27}$$

Therefore, it is clear that three features $\mathcal{P}_i\ (i = 1,\ 2,\ 3)$ are sufficient to estimate the the elements $h_{11},\ h_{12},\ h_{13},\ h_{31},\ h_{32},\ h_{33}$, if the following coefficient matrix is full rank:

$$A = \begin{bmatrix} y_1x_1^* & y_1y_1^* & y_1 \\ y_2x_2^* & y_2y_2^* & y_2 \\ y_3x_3^* & y_3y_3^* & y_3 \end{bmatrix} \tag{28}$$

To guarantee the rank of A to be 3, we know from $\det A \neq 0$ that

$$y_1y_2y_3(x_1^*y_2^* + y_1^*x_3^* + x_2^*y_3^* - y_2^*x_3^* - y_1^*x_2^* - x_1^*y_3^*) \neq 0 \tag{29}$$

Through the condition $y_i \neq 0$ and some transformation for the bracketed term, we have

$$(x_1^* - x_2^*)(y_2^* - y_3^*) - (y_1^* - y_2^*)(x_2^* - x_3^*) \neq 0 \tag{30}$$

which means that the normalized image point $(x_3^*,\ y_3^*)$ cannot lie on the following straight line $L(x,\ y)$ passing through the points $(x_1^*,\ y_1^*)$ and $(x_2^*,\ y_2^*)$:

$$L(x,\ y) : (x_1^* - x_2^*)(y_2^* - y) - (y_1^* - y_2^*)(x_2^* - x) = 0 \tag{31}$$

Hence, if image points of three non-collinear points are provided, then the coefficient matrix A is full rank and thus the homography can be estimated. □

Property 2. For planar motion estimation using the homography-matrix-based algorithm, the ambiguity problem only arises in the case that the scene plane is perpendicular to the motion plane.

Proof. Omitted here, please see [9]. □

Property 3. Using homography-based algorithm, analytical solution for rotation angle and scaled translation vector can be given for all cases.

Proof. In [9], it is seen that analytical solution for all cases are obtained except for the case when the scene plane is perpendicular to the motion plane. Recently, we have already found an analytical solution for this special case, which will be provided in an extended version of this paper. □

Property 4. For a calibrated camera undergoing planar motion, two-view image coordinates of three non-collinear feature points with $y_i \neq 0$ are only necessary to estimate the essential matrix up to a scale factor.

Proof. The proof proceeds in a similar way with that of *Property 1* in the framework of linear estimation through singular value decomposition (SVD). Yet it is seen that the non-collinear distribution of feature points is just a necessary yet not sufficient condition. The detail is omitted. □

Property 5. The short-baseline degeneracy problem for essential matrix estimation still exists for planar motion case.

Proof. From the expression for elements of essential matrix after (19), it is clear that all elements of essential matrix becomes zero when the baseline is zero. Hence, the motion parameters cannot be extracted from the zero matrix. This result is actually very obvious. □

C. The Motivated Idea for a New Approach

In this subsection, we will present the main idea of the newly proposed approach described in the sequel, which is motivated from the previously obtained properties for planar motion estimation.

Property 1-3 provide some interesting results on homography matrix estimation and decomposition for motion estimation. Meanwhile, *Property 4-5* present some results related to the essential matrix. By making some comparisons between *Property 1-3* and *Property 4-5*, it can be concluded that the homography-based approach presents some merits in terms of estimation, analytical decomposition, and less degeneracy problems.

However, as is well-known, one limitation is that all features are required to be in the same planar surface for a single homography matrix. Motivated by *Property 1* that three collinear points are sufficient to estimate a homography and the fact that three collinear points always constitute a plane, we aim to overcome the limitation of the planar scene through multiple homography matrices. In the case of a 3D scenario other than a 2D plane, we can find multiple three-point groups to obtain multiple homographies. By decomposition of every homography, analytical solution(s) can be obtained. To improve the estimation precision and its robustness to outliers, we propose a multi-homography-based motion estimation method to choose the best homography among multiple ones, which are further optimized to obtain more accurate results.

IV. MULTI-HOMOGRAPHY-BASED PLANAR MOTION ESTIMATION

A. Point Grouping

Whatever the scene is 2D planar or 3D, suppose that we have already obtain $N \geq 3$ pairs of matching points from two images obtained at $\mathcal{F}$ and $\mathcal{F}^*$, through some techniques such as SIFT, SURF, and so on. It is possible that some outliers exist in these matching points, which are dealt with in section IV-C.

Point grouping aims to obtain some groups of matching image points, and each group contains three non-collinear features to obtain a single homography. Currently, the point grouping is conducted by first simply computing all combinations of three points C_N^3 from N matching points, and then randomly choosing some groups among the C_N^3 groups. If the three points in some group cannot satisfy the conditions of *Property 1* (non-collinear with $y_i \neq 0$), then we just discard this group. Finally, we got a number of effective groups, denote by N_g.

Note that the point grouping techniques will be further improved in the extended version of this paper by using some criteria such as distances between points, values of y_i, and so on.

B. Motion Estimation From A Single Homography

Homograhy Estimation. For each group containing three points, the constraints for (26) and (27) can be stacked together to linearly estimate a homography matrix. Therefore, we can obtain N_f homography matrices as $H_1,\ H_2,\ \cdots,\ H_{N_f}$ for N_f groups of matching points.

Analytical Solutions to Motion Parameters. For each single homography H_k $(k = 1,\ 2,\ \cdots,\ N_f)$, we apply the techniques in [9] to obtain analytical solutions θ_k and α_k with $(k = 1,\ 2,\ \cdots,\ N_f)$. Note that the subscript $_k$ is utilized to denote the motion estimation results of θ and α from homography H_k which is estimated from realistic image coordinates with noises.

Once the estimated motion parameters θ_k and α_k are obtained for a single homography H_k, we can utilize these parameters to compute the depth-scaled translation magnitude in the following part.

Substituting (2), (3), (6) and (8) into (5), we can obtain the relationship between the normalized coordinates $\boldsymbol{m}_i = [x_i\ y_i\ 1]^T,\ \boldsymbol{m}_i^* = [x_i^*\ y_i^*\ 1]^T$ and the motion parameters as follows:

$$x_i = \frac{x_i^* \cos\theta_k + \sin\theta_k + t_{di}\cos\alpha_k}{-x_i^* \sin\theta_k + \cos\theta_k + t_{di}\sin\alpha_k} \tag{32}$$

$$y_i = \frac{y_i^*}{-x_i^* \sin\theta_k + \cos\theta_k + t_{di}\sin\alpha_k} \tag{33}$$

where

$$t_{di} = \frac{t}{Z_i^*}. \tag{34}$$

with t_{di} being the depth-scaled translation magnitude for the feature point $\mathcal{P}_i$.

Putting (32) and (33) into a form for linear estimation, we have

$$\boldsymbol{\omega} t_{di} = \boldsymbol{\eta} \tag{35}$$

where the vector $\boldsymbol{\omega}$ and $\boldsymbol{\eta}$ are

$$\boldsymbol{\omega} = \begin{bmatrix} x_i \sin\alpha_k - \cos\alpha_k \\ y_i \sin\alpha_k \end{bmatrix} \tag{36}$$

$$\boldsymbol{\eta} = \begin{bmatrix} x_i^* \cos\theta_k + \sin\theta_k - x_i\left(-x_i^* \sin\theta_k + \cos\theta_k\right) \\ y_i^* - y_i\left(-x_i^* \sin\theta_k + \cos\theta_k\right) \end{bmatrix} \tag{37}$$

Therefore, we can compute $\hat{t}_{di}^k = (\boldsymbol{w}^T\boldsymbol{w})^{-1}\boldsymbol{w}^T\boldsymbol{\eta}$ as an estimate of t_{di} for every point $\mathcal{P}_i$ $(i = 1,\ 2,\ \cdots,\ N)$, given a group of estimated parameters α_k and θ_k.

C. Outlier Removing

For each group of motion parameters θ_k and α_k from a single homography, we can use it to get the depth-scaled translation magnitude $\hat{t}_{di}^k$ for every point $\mathcal{P}_i$ $(i = 1,\ 2,\ \cdots,\ N)$. Through $\theta_k,\ \alpha_k$ and $\hat{t}_{di}^k$, we can compute the projected normalized image coordinates $(\hat{x}_i^k, \hat{y}_i^k)$ in $\mathcal{F}$ by using $\hat{t}_{di}^k$ to take the place of t_{di} as

$$\hat{x}_i^k = \frac{x_i^* \cos\theta_k + \sin\theta_k + \hat{t}_{di}^k\cos\alpha_k}{-x_i^* \sin\theta_k + \cos\theta_k + \hat{t}_{di}^k\sin\alpha_k}, \tag{38}$$

$$\hat{y}_i^k = \frac{y_i^*}{-x_i^* \sin\theta + \cos\theta + \hat{t}_{di}^k\sin\alpha_k}. \tag{39}$$

Hence, we can compute the image projection error e_i^k (in pixel) for the point $\mathcal{P}_i$ as

$$e_i^k = \|\boldsymbol{p}_i - K\hat{\boldsymbol{m}}_i^k\|_2 \tag{40}$$

with $\hat{\boldsymbol{m}}_i^k = [\hat{x}_i^k \ \hat{y}_i^k \ 1]^T$ being the estimated normalized image coordinates.

Outliers Removing Mechanism. For the motion parameters $\theta_k,\ \alpha_k$ and $\hat{t}_{di}^k$ from the single homography H_k, the matching point $(\boldsymbol{p}_i,\ \boldsymbol{p}_i^*)$ is regarded as an outlier if

$$e_i^k \geq \epsilon \tag{41}$$

with ϵ being a given threshold. To facilitate the subsequent analysis, we denote the number of inlier points by N_k for the pose $\theta_k,\ \alpha_k$, with E_k being the mean projection error of these N_k inlier points.

D. Ranking of the Results for Each Homography

After removing the outliers, it is shown that for each homography H_k, we can obtain an group of motion parameters $\theta_k,\ \alpha_k,\ \hat{t}_{di}^k$, and also the corresponding number of inliers N_k and mean projection error E_k. Then, we simply rank the motion estimation results first by N_k, with the rank being higher with a larger N_k. If the same rank is obtained with $N_{k_1} = N_{k_2}$ for some k_1 and k_2, then the one with smaller E_k ranks higher. With these ranking rules, and the final 'best' motion estimation results are selected as the one with the top rank, denoted by $\theta_{k_f},\ \alpha_{k_f}$ and $\hat{t}_{di}^{k_f}$. Meanwhile, the corresponding number of inliers and the mean projection error are N_{k_f} and E_{k_f}, respectively.

E. Optimization of Motion Parameters

To further optimize the motion estimation results of $\theta_f,\ \alpha_f$ and $\hat{t}_{di}^f$, we use nonlinear optimization techniques to minimize the overall image projection error:

$$\min \sum_{i=1}^{N_{k_f}} (e_i^{k_f})^2. \tag{42}$$

Actually, the optimization is an optional procedure, which depends on the onboard computation ability and the real-time performance requirement of the tasks.

F. Computation Issues

It is seen that for each group of feature points, the algorithms involved in the two key procedures from section IV-B to section IV-C present a good parallel structure. The homography estimation and decomposition, as well as the outlier removing algorithm, can be conducted in a parallel way for each group of feature points. Nevertheless, the GPU implementation is not completed yet in this paper.

V. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation

In the simulation, we test the performance of the proposed algorithm with respect to the image noise and internal camera calibration errors.

Twelve feature points are utilized, and their image pixel coordinates are polluted by Gaussian noise with zero mean and deviation of 0 to 5 pixels. At every noise level (the stand deviation of Gaussian noise), 100 simulations are conducted and the mean estimation errors are shown in Figure 2 and Figure 3, from which we can see that the proposed method presents some robustness with image coordinates. It is shown that the algorithm works very well if the noise level is less than 2 pixels.

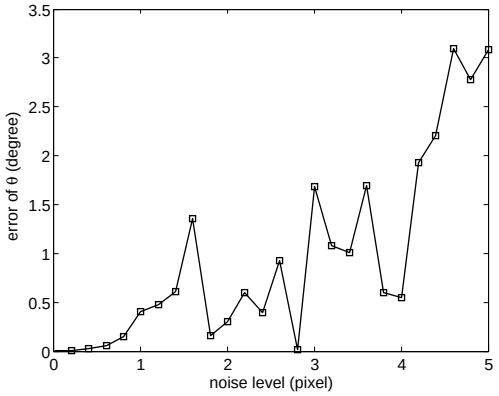


Fig. 2. The mean estimation error of θ with Gaussian image noises

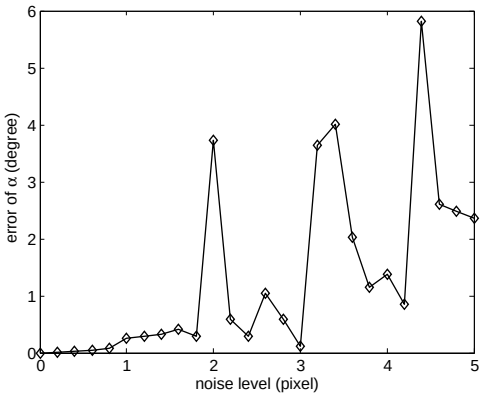


Fig. 3. The mean estimation error of α with Gaussian image noises

Figure 4 and Figure 5 show the motion estimation results in the presence of camera calibration errors. The noise level means that the corresponding percentage of the camera parameters are regarded as the amplitudes for the standard deviation of the Gaussian noise added in the internal camera parameters. In a similar way, 100 simulations are conducted for each noise level and the mean estimation error is shown in Figure 4 and Figure 5. It is seen that the effect of the calibration errors is larger than that of the image noise, and the translation direction α is more sensitive to camera calibration errors. When the calibration error is less than 2%, the motion estimation error is tolerable for general applications.

B. Experiments

To illustrate the effectiveness of the proposed method, we conduct visual servoing experiments on P3-DX robots, equipped with an onboard monocular camera. SIFT features are extracted and matched in the current and desired images,

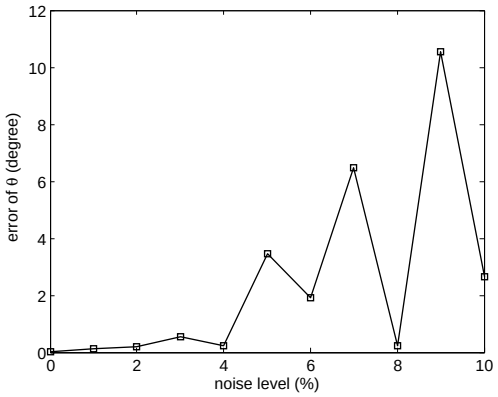


Fig. 4. The mean estimation error of θ with camera calibration errors

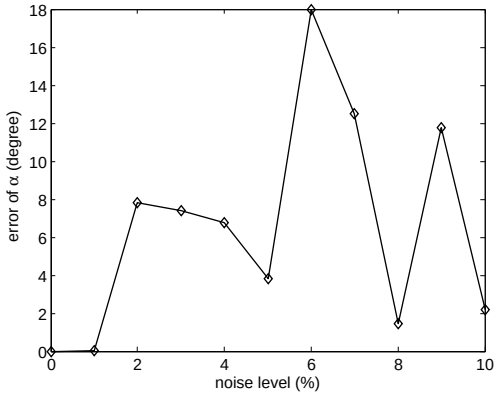


Fig. 5. The mean estimation error of α with camera calibration errors

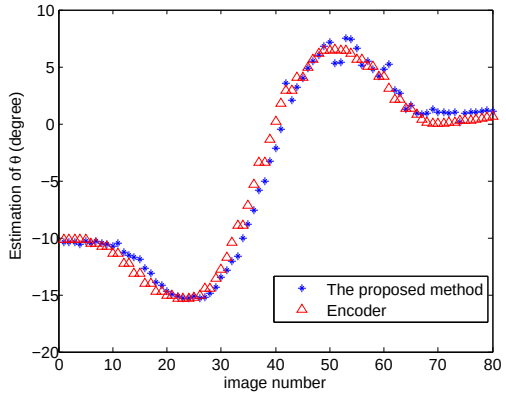


Fig. 6. The estimation of θ during the visual servoing process

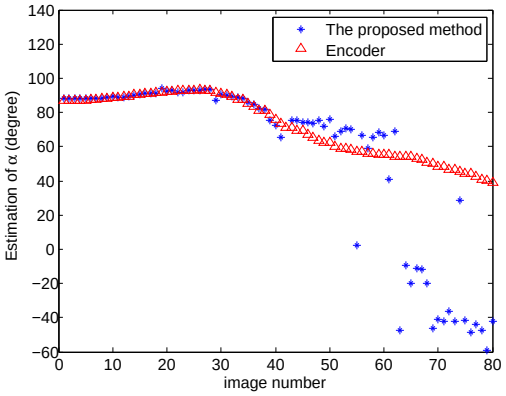


Fig. 7. The estimation of α during the visual servoing process

which are then utilized to estimate the motion parameters using the proposed method. The estimation results are shown in Figure 6 and 7, where the encoder results are regarded as the ground truth. It is seen that the estimation error of the θ is pretty well for the whole servoing process and the results for α is good for the first 50 frames. When the robot is very closed to the target pose, the estimation of α is sensitive to the image noises and calibration errors. However, it should be noted that this happens merely due to the specific angle definition of α, and the scaled translation error is actually very small, since the amplitude of the translation is very small when the robot is closed to the target.

VI. CONCLUSIONS

Some insightful properties for the planar motion estimation are presented and proved in this paper. Based on these properties, a multi-homography-based motion estimation approach is proposed, which has some merits including less limitation for scene, robustness to the image noise and outliers, and so on. Simulation and experimental results are provided to verify the performance of the proposed approach.

REFERENCES

[1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[2] Y. Fang, W. E. Dixon, D. M. Dawson. Homography-based visual servo regulation of mobile robots, *IEEE Transactions on Systems, Man, and Cybernetic: Part B: Cybernetics*, vol. 35, no. 5, 2005.

[3] G. Chesi, K. Hashimoto. A simple technique for improving camera displacement estimation in eye-in-hand visual servoing, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9): 1239-1242, 2004

[4] G. López-Nicolás, C. Sagüés, and J. J. Guerrero. Automatic matching and motion estimation from two views of a multiplane scene. *In Pattern Recognition and Image Analysis, Lecture Notes in Computer Science 3522*, pp. 69-76, June 2005.

[5] G. López-Nicolás, J. J. Guerrero, O.A. Pellejero, and C. Sagüés. Computing homographies from three lines or points in an image pair. *In International Conference on Image Analysis and Processing, Lecture Notes in Computer Science 3617*, pp. 446-453, 2005.

[6] T. N. Tan, K. D. Baker, G. D. Sullivan, 3D structure and motion estimation from 2D image sequences, *Image and Vision Computing*, vol. 11, no. 4, May 1993, pp. 203-210.

[7] J. Li, R. Chellappa. Structure from planar motion, *IEEE Transactions on Image Processing*, 2006, 15(11): 3466-3477.

[8] H. M. Becerra, C. Sagüés. Exploiting the trifocal tensor in dynamic pose estimation for visual control, *IEEE Transactions on Control Systems Technology*, vol. 21, no.5, pp. 1931-1939, Sept. 2013.

[9] X. Zhang, Y. Fang, B. Ma, X. Liu, M. Zhang, A fast homography decomposition technique for visual servo of mobile robots, *Proc. of Chinese Control Conference*, Kunming, China, pp. 404-409, 2008.

[10] X. Zhang, Y. Fang, X. Liu. Motion-estimation-based visual servoing of nonholonomic mobile robots. *IEEE Transactions on Robotics*, 2011, 27(6): 1167-1175.

[11] G. L. Mariottini, G. Oriolo, D. Prattichizzo. Image-Based Visual Servoing for Nonholonomic Mobile Robots Using Epipolar Geometry, *IEEE Trans. Robotics*, vol. 23, no. 1, pp. 87-100, February 2007.

[12] X. Zhang, Y. Fang, N. Sun. Visual servoing of mobile robots for posture stabilization: from theory to experiments. *International Journal of Robust and Nonlinear Control*, 2013, online, DOI: 10.1002/rnc.3067.