

International CBRNE Institute

ISMCR 2018

21th International Symposium on
Measurement and Control in Robotics
26-28 September 2018 – International
CBRNE Institute, MONS, Belgium

ROBOTICS FOR THE CHANGING WORLD



BEMEKO

www.imeko.org – TC 17



European Society for Defense	Hainaut Sécurité	Belgian Royal Higher Institute of Defense	Climbing and Walking Robots Association	

PROGRAM

SESSION 1. WELCOME and INTRODUCTION TO THE MOBILE ROBOTICS 26 September (Room 1)

CHAIR: Professor L.Van Biesen (VUB/BEMEKO)n Prof Dr Ir Dehombreux (FPMs) Dr Zafar Taqvi (Srt IMEKO/TC17), Professor Y.Baudoin (ICI)

10.00-10.30H	Welcome, Military Robotics, ELROB Status, ICI aims	Madame Vanwijnsberghe (Director Hainaut Sécurité), Dr F.Schneider (Fraunhofer, Germany), M.Y.Dubucq (Director ICI), Prof dr Ir H. Christensen,
10.30-11.00H	KEY-NOTE Autonomous Robot for Gas and Oil Sites (ARGOS): Total's Lessons Learnt and Next Steps	Kris KYDD Head of Robotics Total ARGOS Challenge
11.00-11.30H	KEY-NOTE Robotics and civilian emergency response : how lessons learned empower Incident command systems organization	LtCol JP Monet (BDFRD, France), Maj E.Rodriguez (BDFRD), Capt S.Mozziconacci (BDFRD), E.Dombre (Emeritus CNRS Research Director, Montpellier University
11.30-12.00H	Robotics for facing CBRNE risks	Dr J.Galatas (KEMEA, Greece, CBRN-KC)m Y.Baudoin (E-KC)
13.30-17.00H	Robotics competition and Exhibition	(ELROB, ICI)
19.30-22.00H	GET TOGETHER ISMCR at the Hotel MONS	

SESSION 2/1 IMPROVEMENT of ROBOTICS 27 September (Room 1)

CHAIR : Professor EM Y.Baudoin (ICI/RMA)

10.00-10.30H	KEY-NOTE Unmanned vehicle systems in unstructured environments : challenges and current status	Prof V.Gradetsky (IPMNET, Moscou, Russia)
10.30-10.50H	<i>IMU based gesture recognition for mobile robot control using Online Lazy Neighborhood Graph search</i>	<i>Padmaja Kulkarni (Fraunhofer-Institute, Germany), B. Illing, B. Gaspers, B. Brüggemann, D. Schulz</i>
10.50-11.10H	<i>A Novel Data Fusion architecture for Unmanned vehicles</i>	<i>I. L. Ermolov, Institute for Problems in Mechanics of the Russian Academy of Sciences, Russia.</i>
11.10-11.30H	<i>Coverage Path Planning by swarm;- of UAV by swarm of UGV for traversability analysis</i>	<i>L. Cantelli, D.C. Guastella, D. Longo, C.D. Melita, G. Muscato University of Catania</i>
11.30-11.50H	<i>Robot control based on human motion analysis with IMU measurements</i>	<i>Robin Pellois, Laura Joris and Olivier Bruls Multibody and Mechatronic Systems Laboratory University of Liege, Belgium</i>
11.50-12.10H	<i>Development of the Modular Platform for Educational Robotics.</i>	<i>A.G. Semenyaka, Moscow State Technological University "STANKIN"</i>

SESSION 2/2 MEASUREMENT and CONTROL in ROBOTICS
27 September (Room 2)

CHAIR: Professor B. Kiss (BME/Hungary)

10.00-10.30H	KEY-NOTE: JIZAI Body Design of ultra-body fit for super smart society	Prof Masahiko Inami, , Japan)
10.30-10.50H	<i>Enhancing Bodily Expression and communication Capacity of Tele-existence Robot with augmented reality</i>	<i>Yasuyuki Inoue (University of Tokyo, MHD YamenSaraiji (Keio University, Japan), Fumihiro Kato and Susumu Tachi (University of Tokyo)</i>
10.50-11.00H	<i>Semantic Grid Mapping based on Surface Classification with Supervised Learning</i>	<i>Torsten Engler (Universität der Bundeswehr München Institut für Technik Autonomer Systeme (TAS)</i>
11.00-11.20H	<i>Pre-filter to robustify the exact linearization based tracking controller of a SCARA type robot</i>	<i>Na Wang, Balint Kiss Budapest University of Technology and Economics, , Hungary</i>
11.00-11.20H	<i>Feedforward command computation of a3D flexible robot</i>	<i>Arthur Lismonde and Olivier Bruls Department of Aerospace and Mechanical Engineering, University of Liege,Belgium</i>
11.20-11.40H	<i>Effectiveness test of simulator for e-training in carrying out missions with use of tele-operated vehicles</i>	<i>Igor Ostrowski, Andrzej Masłowski NASK Governmental Research Institute Digital Mobile Robotics Department Warsaw, Poland</i>
11.40-12.00H	<i>Training of robots' operators with use of multirobot simulators</i>	<i>Marek Kacprzak, IMMSF, Warsaw, Poland</i>

SESSION 3/1 ROBOTICS for DEFENSE and SECURITY - STATUS
27 septembre (Room 1)

CHAIR: Professor Y.Baudoin (ICI-RMA) – Dr Ir F.Schneider (Fraunhofer)

13.15-13.40H	UGV-UAV Growing Market for the Defense	Prof EM Y.Baudoin (ICI/RMA)
13.40-14.10H	In-flight launch of unmanned aerial vehicles	Dr Ir Geert De Cubber (Royal Military Academy Belgium)
	Qualitative and quantitative validation of drone detection systems	
14.10-17.00H	<i>Competition/Exhibition EOD/IED trials</i>	

SESSION 3/2. CONTROL and SENSOR SYSTEMS in ROBOTICS
27 September (Room 1)

CHAIR: Prof A.Maslowski (NASK/Poland)

14.10-14.40H	KEY NOTE: Measurements for the Forthcoming Future	Dr Zafar Taqvin Scientific Secretary IMEKO TC 17 (USA)
14.40-15.00H	<i>Dedicated simulator for e-training of demining robot "Dromader" operators.</i>	<i>Igor Ostrowski, Andrzej Masłowski NASK Governmental Research Institute Digital Mobile Robotics Department Warsaw, Poland</i>
15.00-15.20H	<i>An active beacon-based Tracking System to be used for Mobile Robot Convoying</i>	<i>Stanislaw Goll, Elena Zakharova, LLC KB Avrora/Ryazan State radio Engineering University, Ryazan, Russia</i>
15.20-15.40H	<i>RADAR-based Through-Wall Mapping</i>	<i>Sedat Dogru, Lino Marques, ISR-University Coimbra, Portugal</i>
15.40-16.00H	<i>Ultrasonic Rangefinder with submillimeter resolution as part of the Rescue Robot's sensor system</i>	<i>Stanislaw Goll, Julia Maximova, LLC KB Avrora/Ryazan State radio Engineering University, Ryazan, Russia</i>

SESSION 4. TECHNICAL PRESENTATIONS and MODELING
26-27 September (Exhibition Site)
Demos in competition 24-27 Sep
IMEKO TC17 meeting
And Room 2 (schedule later adapted, depending on the scenarios)

CHAIRS: Dr F.Schneider (Fraunhofer/Germany), Dr Z.Taqvi, Y.Baudoin
In RED: exhibitors (updated 15 July)

14.00-14.20H	<i>ELROB 2018. Convoy and Mule of Team MuCAR</i>	<i>Thorsten Luettel, University of the Bunderswehr, Munich, Germany F. Ebert, P. Berthold, P. Burger, T. Engler, A. Frericks, B. C. Heinrich, J. Kallwies, M. Kusenbach, K. Metzger, M. Michaelis, B. Naujoks, A. Sticht, and H.-J. Wuensche</i>
14.20- 14.40H	<i>Standard Test Methods for Mobile Robots</i>	<i>Andreas Ciossek, Produkt Manager, TELEROB, Germany</i>
14.40-15.00H	<i>Automated Magnetic Field reproducing Stand for Debugging Algorithms Navigation of Mobile Robots.which use on board Magnetometer Sensor</i>	<i>Stanislaw Goll, Alexander Borisov, LLC KB Avrora/Ryazan State radio Engineering University, Ryazan, Russia</i>
15.00-15.20H	<i>Robsim Software for Mobile Robots modeling</i>	<i>O.P. Goidin, S.A. Sobolnikov FSUE VNIA, Moscow</i>
10.00-17.00H	The ZEUS robot	Steve Wisbey NIC Instruments Ltd
15.30H MEETING IMEKO TC17		
10.00-17.00H	The MSAS vehicle	Janusz Bedkowski, Manadal, Poland
10.00-17.00H	Telemax PRO, Hybrid, PLUS	Andreas Ciossek, Telerob, Germany
10.00-17.00H	The Mörri robot	Antti Tikanmäki, BISG Oulu ,

		University of Oulu Finland
10.00-17.00H	The SR-120D System	Patrik Bylin, Brokk AB, Sweden
10.00-17.00H	The Packbot 510 EOD/Kobra 710	Colin Weiss, ELP GmbH, Germany
10.00-17.00H	Tulf/StrAsRob, smart military vehicles	Dr Alexander Wolf, Diehl Defense GmbH, Germany
10.00-17.00H	Milrem TheMIS	Dr Alexandre Wolf, Diehl Defense GmbH&CoKG
10.00-17.00H	Patria AMV, SLO-IFV	Matti Saarikko, Patria land Systems Oy, Finland
10.00-17.00H	The Wombat Leader-Wombatt Follower	Gol Stanislav, LLC KB Avrora, Russia
10.00-17.00H	The robot LongCross	Bastian Gaspers, Fraunhofer, Germany
10.00-17.00H	TAUT	Reinhard stocker, Guenther Tratting, Austria

NOTE: a SHUTTLE BUS is foreseen during the symposium for transportation of participants to the ELROB stand (standing lunch and competition)

Enhancing Bodily Expression and Communication Capacity of Telexistence Robot with Augmented Reality

Y Inoue¹, M Y Saraiji², F Kato¹ and S Tachi¹

¹ Institute of Gerontology, The University of Tokyo, Tokyo 113-8656, Japan

² Graduate School of Media Design, Keio University, Kanagawa 223-8526, Japan

E-mail: y-inoue@tachilab.org

Abstract. This paper focuses on realization of embodied remote communication via telexistence robot with augmented reality (AR). Though a robot equipped with communication functions can realize natural conversation remotely, the robot has to reproduce bodily expressions of the robot user for achieving embodied communication. A humanoid is optimal shape for the purpose, but this kind of robot is currently expensive and difficult to popularize. By contrast, a 3DOF head-moving robot is easier to develop, but the bodily expression capacity is limited. To solve the trade-off problem, we propose an AR-based presentation system visualizing additional body-parts of head-moving robot. The developed system consists of a head-mounted display (HMD) worn by an operator, a 3DOF robot controlled by the operator's head movement, and see-through AR glasses worn by an interlocutor who faces the robot. For visualizing bodily expression, the system generates 3D-CG image of virtual avatar which copies operator's body movements, and projects the image onto both operator's HMD and interlocutor's glasses simultaneously. Consequently, proposed system provides body gesture functions to 3DOF robot, and achieves embodied remote communication.

1. Introduction

The conceptual core of telexistence [1] explains that a robot with multiple devices to substitute human communication function (i.e. camera as eye, microphone as ear, speaker as mouth, and so on) can be used for the surrogate body of an operating user. By using the robot, the user can act in a remote place spontaneously as if he/she were actually there. Also, the user can meet and communicate with people in a remote place directly via the surrogate body. This means that telexistence robot realizes natural face-to-face communication with multiple people in distant places, as if they were all in the same room. Unlike videophone, the robot is an existing object acting like human, therefore the presence of the communication partner is clearly for other meeting members and their interaction may be facilitated. That is, one major advantage of using a robot in telecommunication is increasing the sense of presence of a person inside the robot.

Recently, a number of robot-based telecommunication systems commonly referred to as "telepresence robot" such as Double2 [2] and BeamPro [3] have been produced and popularized. These kinds of robots typically provide both locomotion and communication functions, that is, they have rolling wheels to move across a room and are equipped with a computer screen to be used as the videophone. Furthermore, some other robots possess stereo camera in order to provide immersive 3D experience for the robot user (e.g. TELUBEE [4]). The two images taken from the pair of cameras are presented on the eyes through stereoscopic display device such as head-mounted display (HMD). In this case, eye contact between the robot user and the communication partner can be achieved

spontaneously because the camera's direction corresponds to user's gaze, particularly when the robot also has head rotation function. Such a robot-head moves the neck joint to change the camera direction so that the user can look around the place. That is, the robot-head copies user's head movements. This means that the robot transmits the bodily expressions to the remote place. This capability is crucial for the achievement of embodied communication, because human's spontaneous actions during conversation such as nod and glance play an important role in human communication. Through the duplicated movements of the robot, the interlocutor facing the robot can understand the implicit intentions of the user who are not actually present.

It is considered that human's body action can be effectively replicated by human-like shape, therefore a humanoid robot is the optimal to transmit embodied information. However, many of popularized robots for telepresence/telexistence purpose are not perfect humanoids. This is probably because that a whole-body humanoid is currently expensive and therefore difficult to produce and popularize. Also, bodily expression is regarded as less important information than verbal and facial ones. Nevertheless, these communication robots desirably have additional ways to show bodily expressions of the user in order to realize embodied communication with the remote interlocutor. How does the robot acquire sufficient capacity to express embodied information with the minimum effort of the implementation? To equip mechanical add-ons are not reasonable. The most practical solution to achieve the enhancement is to apply virtual-reality (VR) and augmented-reality (AR) technology. Yamen et al. [5] demonstrate that visual projection of user's hand in the remote place can be used to compensate the embodied information in the robot-based telecommunication. To show user's hand image on the surface via a projector equipped in the robot, the robot user and remote interlocutor can see same hand image to joint their visual attention. This means that the robot user can point any object in the remote environment directly by using the virtual hand as long as the object is on the projectable surface (e.g. whiteboard and table). In this case, the physical arms and hands of the robot are not necessary for social activities between them. That is, VR/AR visualization of body can be helpful to transmit embodied information. Here, our proposal system visualizing additional body-parts of robot will be described below.

2. Proposed system

2.1. Overview

Figure 1 shows the overview of proposed system. The system configuration is similar to previous study [5] besides the display of virtual body. The user can see and hear what happens at the remote place through the robot, and the interlocutor can meet the user who controls the robot. Furthermore, this system allows them to be aware of each other's bodily expressions. This system is divided into two subsystems of local side (left) and remote side (right). Both sides are connected to each other via the network.

The subsystem for local side consists of three components below: (L-1) a head-mounted display (Oculus CV2, Oculus VR) worn by the robot user to measure head movements and to present audio/visual information received from the robot; (L-2) a pair of position tracking sensor (Oculus touch, Oculus VR) held by the user's hands to obtain the hand position/attitude; (L-3) a laptop computer placed near the user to control user's equipment, record user's voice and communicate with the robot. The subsystem for the remote side consists of two components below: (R-1) a 3DOF head-moving robot with communication functions (TX-toolkit, JST-ACCEL Embodied Media Project, Figure 2) controlled by user's head movements, speaking the user's voice and sending audio/visual data recorded in the remote environment; (R-2) an optical see-through AR glasses (HoloLens, Microsoft) worn by the interlocutor to show an AR-image superimposed on the robot.

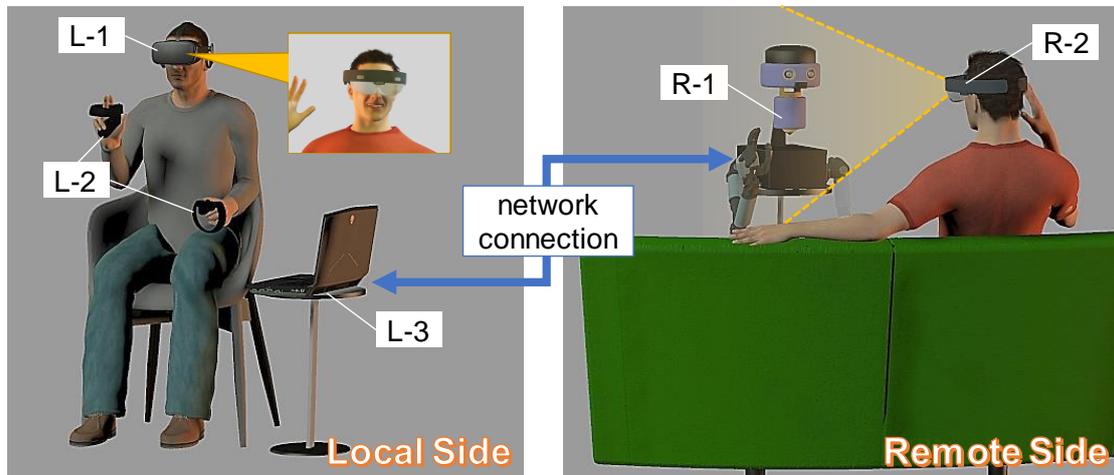


Figure 1. Overview of the proposed system.



Figure 2. “TX-toolkit”, developed by JST-ACCEL Embodied Media Project, is a 3DOF robot-head equipped with communication functions for the experimental platform of teleexistence study.

Specification:

Size	362mm x 167mm x 232mm
Weight	2.5kg
Devices	Camera x 2, Microphone x 2, Speaker x 1, USB3.0 x 2, Ethernet x 1, WiFi x 1
Motion	Yaw $\pm 90^\circ$ Pitch $\pm 20^\circ$ Roll $\pm 20^\circ$
ViewAngle	H 100° V 98°
Resolution	960 x 950 pixel/eye @ 60fps

2.2. Virtual Arm: Control

The system generates 3D-CG image of robot-arms which are controlled by user’s arm movement, and presents the image on the stereoscopic display devices of the user and the interlocutor simultaneously in different manner (explained later). The virtual arm has 7DOF, the same as human’s arm, and moves according to the tracking sensor in order to match the hand position/attitude between the user and the robot. Note that though the sensor input is insufficient to determine the posture of robot-arm uniquely, the unconstraint 1DOF of arm movement can be adjusted by using extra input (e.g. thumbstick).

2.3. Virtual Arm: Display

Figure 3 shows examples of subjective appearance of the virtual arms presented on HMD for the user (left) and AR-glasses for the interlocutor (right). For the user, the virtual arms are superimposed on the video image streaming from the eye-camera as the first-person-perspective view. Therefore the user feels to be in the place as the robot because the virtual arms are seen as if they were user’s own. For the interlocutor, the virtual arms are presented around the actual robot as if these arms were attached to the robot. Thus the interlocutor feels that he/she faces to an upper-body humanoid, despite facing to a robot-head. Consequently, even though the robot does not have actual arms or hands, the user can show his/her arm gestures to the interlocutor.

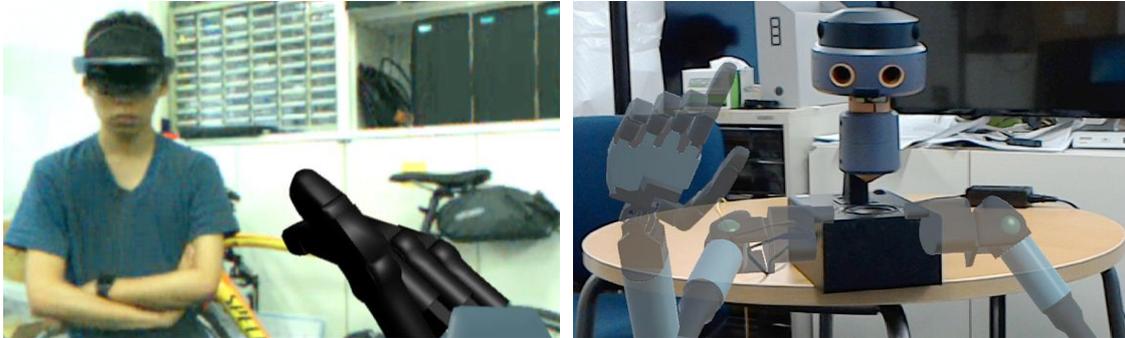


Figure 3. Examples of AR images of the user's view (left) and the interlocutor's view (right).

3. Conclusion

In this paper, we propose an AR-based enhancement system of the communication robot which has limited capacity of bodily expression. This system realizes embodied remote communication. Owing to the system's aids, persons who are actually in different places can share the same surroundings, and they have close communication through their bodily expressions. In contrast to projected hand image presented in previous study [5], the proposed system presents AR hand object on the 3D space, therefore, the robot user can point any object (not limited to be on the surface) by using the virtual hand. Because the proposed system works as an AR add-on application, this is available to any other communication robot if the robot has wireless network devices (e.g. WiFi and Bluetooth) to transmit user's body posture data to the AR glasses worn by the interlocutor. Also, by detecting spatial contacts between user's virtual hand and interlocutor's actual hand, it is possible to realize tactile communication between them such as high-five. To conclude, the combination of robotics and VR/AR has great potentials to realize new communication tools.

Acknowledgments

This research is supported by ACCEL Program from Japan Science and Technology Agency, JST.

Reference

- [1] Tachi S 2015 *Telexistence 2nd Edition* (Singapore: World Scientific)
- [2] Double Robotics, Inc. (n.d.) Double 2 – Features, Retrieved July 31, 2018 from <https://www.doublerobotics.com/double2.html>
- [3] Suitable Technologies, Inc. (n.d.) Meet BeamPro®, Retrieved July 31, 2018 from <https://www.suitabletech.com/products/beam-pro>
- [4] Laval Virtual. (February, 1, 2013) ReVolution 2013 – Ubiquitous Telexistence, Retrieved July 31, 2018 from <https://www.youtube.com/watch?v=Q5zigZDKZo8>
- [5] Saraji M Y, Fernando C L, Minamizawa K and Susumu Tachi 2015 Development of Mutual Telexistence System using Virtual Projection of Operator's Egocentric Body Images, *Proc. 25th Int. Conf. on Artificial Reality and Telexistence (Kyoto)* pp.125-132

IMU based gesture recognition for mobile robot control using Online Lazy Neighborhood Graph search

P Kulkarni, B Illing, B Gaspers, B Brüggemann, D Schulz

Fraunhofer FKIE, Fraunhoferstraße 20, 53343 Wachtberg

E-mail: padmaja.kulkarni@fkie.fraunhofer.de

Abstract. In this paper, we present and evaluate a framework for gesture recognition using four wearable IMUs to indirectly control a mobile robot. Six gestures involving different hand and arm motions are defined. A novel algorithm based on Online Lazy Neighborhood Graph (OLNG) search is used to recognize the gestures. We use this algorithm to classify the gestures online and trigger predefined behaviors. Experiments show that the framework is able to correctly detect and classify six different gestures in real-time with an average success rate of 81.6%, while keeping the false positive rate low by design and using only 126 training samples.

1. Introduction

For robots to be able to work in unstructured environments, areas dangerous to humans, or disaster sites, human intelligence is still vital. In such cases, teleoperation of robots could be one of the solutions. With recent advancements in robotics, the complexity of using robots has also increased. Despite this fact, currently used technology limits the majority of man-machine interfaces to text or GUI based interfaces and joysticks. Such types of control can become cumbersome in case of, for example, robots with a heavy control box or high degrees of freedom. Often, working in disaster areas could be stressful for an operator. Hence, alternate and intuitive control paradigms need to be developed. Gesture-based control seems particularly useful as it can be very intuitive [1].

Vision-based gesture control is well researched but the set-up time and dependency on controlled environmental conditions, like lighting, make it less suitable for teleoperation in disaster areas. On the other hand, Hoffmann et al. [2] developed an Inertial Measurement Unit (IMU) based control for a robot manipulator, which does not need any infrastructure. They transformed human arm motions into corresponding robotic manipulator motions using five IMUs attached to the sleeve of a wearable-jacket. They showed that teleoperation performed in this way is very efficient and intuitive [3]. However, to trigger some predefined manipulator motion or to trigger robot base motions this direct control method cannot be used.

This paper presents an extension of the work done by Hoffmann et al. [2] in the area of wearable IMUs. We present a framework based on OLNG search, which is able to identify dynamic gestures in real-time and can be used to trigger predefined robot motions. The main contribution of this work is an implementation and evaluation of an OLNG search based algorithm for gesture recognition and robotics application. Prior to this, OLNG search algorithm was primarily used in the area of computer graphics [4].

2. Related work

Most approaches in the field of gesture recognition are based on vision, IMU, and Electromyography (EMG) signals [5] [6].

For IMU based approaches many use glove mounted sensors. Mummadi et al. [7] use an IMU based glove for real-time sign language recognition. They use various machine learning algorithms such as Support Vector Machines, Naive Bayes, Multi Layer Perceptron, and Random Forest to classify the gestures. Wu et al. [8] use a data glove with perception and Hidden Markov Models (HMMs) to classify hand gestures. Georgi et al. [6] couple IMU based motion with EMG muscle activity to recognize hand and finger gestures. They use HMMs for the gesture recognition and obtain 74.3% in accuracy with person independent settings. These methods only classify static gestures with hand and fingers [6] [7] and need a huge database (about 1000 samples of each kind) [7]. In the domain of commercial products, the Myo armband by Thalmic Labs uses EMG signals along with IMUs to detect up to five different gestures and motions of the arm, but these gestures are pre-set.

For vision based gesture recognition the Microsoft Kinect is widely utilized. OpenNI or Kinect SDK are used for motion tracking. For gesture identification, algorithms like Dynamic Time Warping, Artificial Neural Networks, or HMMs are implemented, for an overview see [9]. Amin et al. [10] developed a vision based technique to identify hand gestures using Principal Component Analysis and Gabor representation. However, vision based approaches suffer from limitations like occlusion and are vulnerable to bad performance from ambient lighting and background changes.

3. Approach

We assume that the IMU readings are available in the form of vectors at a discrete time interval $(\dots, \vec{\alpha}_{t-2}, \vec{\alpha}_{t-1}, \vec{\alpha}_t, \dots)$, where $\vec{\alpha}$ is a vector of Euler angles and t is time. This vector is referred to as input vector in this paper. An underlying training database consists of Euler angles obtained from the four IMUs. A 12-dimensional vector forms a data point in it. For building the database, sequences of such vectors labelled with the corresponding gesture names are saved while the gestures are being performed. Every vector in the database has a unique index i , which is later used for the identification of a particular vector.

For sequence matching, a window of m input vectors is defined. The distance between the current input vector and each vector in the training database is calculated and from this set of distances the indices of the k nearest vectors (neighbors) and their spatial distances from the input vector are obtained and stored using Fast Library for Approximate Nearest Neighbors (FLANN) in real-time. A matrix is created for the k nearest neighbors in the training database and for a window of m input vectors. Considering these $k \times m$ vectors as nodes, a graph is built. For each input vector, its k corresponding graph nodes are then sorted in ascending order of the database index i . For the sequence matching, nodes are chosen in such a way that their database indices are in strictly ascending order. A path is a sequence connected from the first vector in the window to the last vector. Figure 1 shows possible sequences for three input vectors. If no valid neighbor index for a particular input vector exists, then the neighbors of the next input vector are considered for the path. In that case, an extra path cost is added for skipping one input vector. In this way all possible paths are listed and the best path amongst them is chosen based on its minimum cost. OLANG search offers extremely fast sequence matching and is suitable for real-time applications due to its linear complexity [4].

For gesture recognition the following procedure is applied. For the whole database, k nearest neighbors to the current input vector are calculated using FLANN. OLANG is then used to find a matching and valid sequence. The best path found is saved along with its cost and sequence matching length for comparison. An arm and hand movement is considered to be a gesture if the sequence matching length is more than a certain threshold value and the cost of the path is less than another threshold. Both thresholds were chosen based on initial informal experiments and were tweaked and validated during our evaluation. Each recognized gesture triggers a specific motion of the robot. During the robot motion no gestures are detected. Gesture recognition restarts when the robot signals the completion of the motion.

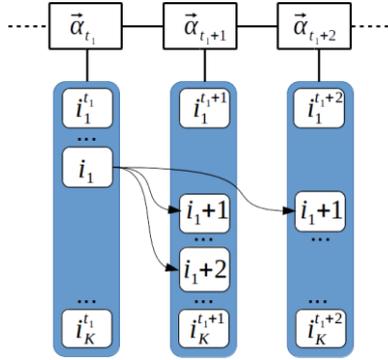


Figure 1. Path finding using OLNG search with k nearest neighbors [4].

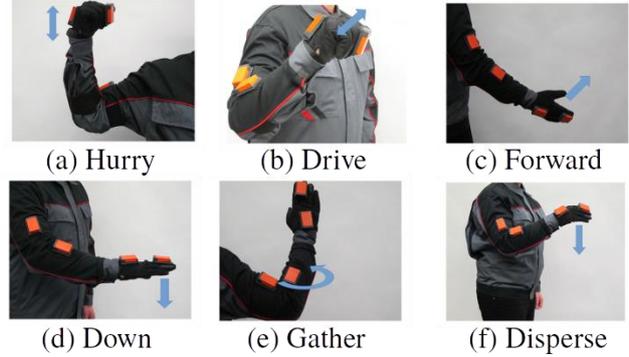


Figure 2. (a)-(f) show six different gestures defined for our experiment.

4. Experimental setup

We validated our OLNG search based algorithm by testing it for online gesture recognition. Xsens MTw sensors were used for the motion capturing and building a database. The sensor is a 9-axis IMU consisting of a 3-axis gyroscope, acceleration sensor, and magnetometer and it includes an extended Kalman filter. Internally the sensors operate at 75 Hz, but such a high rate does not offer much additional information during arm movements. Hence, we used an update frequency of 15 Hz. Four of such sensors were mounted on a jacket at humerus, radius, hand, and finger.

The training database used consisted of 21 motions of each gesture from three different users for a total of 126 samples. The test database consisted of 10 different motions of each gesture plus 10 random arm movements. To apply robot motions, a telemax manipulator from Telerob was used. The programming was done in C++ using MoveIt! motion planning framework with a Linux operating system and Robot Operating System as a middleware. For OLNG search we considered $k = 40$ nearest neighbors, a window of $m = 10$ vectors, and a path length of 9. Based on these fixed parameters the path cost was varied from 0 to 4 in steps of 0.05 squared radians.

We defined six gestures to test our algorithm. These are based on internationally used hand signals from a military context. Figure 2 shows the gestures used and the placement of the four IMUs on the jacket.

5. Results

Receiver Operating Characteristic (ROC) curves for all six gestures with varying path costs are shown in figure 3. Based on the ROC curves the threshold path cost was chosen to be 1.80 with 81.61% true-positive rate and false-positive rate of 15.12%. By including random arm movements in the test database, all thresholds were also validated to only recognize a gesture when one was performed. The confusion matrix corresponding to the chosen cost is shown in figure 4. ‘Drive’ and ‘Hurry’ were recognized the best at the chosen cost with 95.65% and 90.10% prediction rate respectively. It can be observed that the gesture ‘Disperse’ is more difficult to recognize as it partly shares similarity with the whole gesture ‘Down’. The same goes for the gestures ‘Gather’ and ‘Hurry’, which leads to the non-symmetrical confusion matrix

6. Conclusion and future work

A novel algorithm based on OLNG search was implemented and tested for the application of gesture recognition in real-time. A software framework to trigger predefined robot motions based on a detected gesture was implemented. Experiments show that we could obtain a gesture recognition rate of 81.61% while keeping the false positive rate low. In the future, we would like to expand our

training database on the fly by adding correctly recognized gestures to it. We would also like to extend our algorithm to match parts of start, middle, and end of a gesture to counter similarity between different gestures.

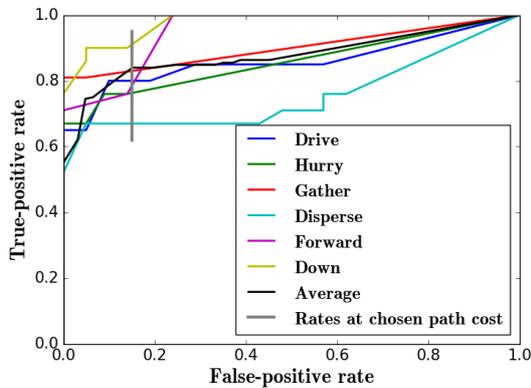


Figure 3. ROC curve for all gestures with variable threshold path cost.

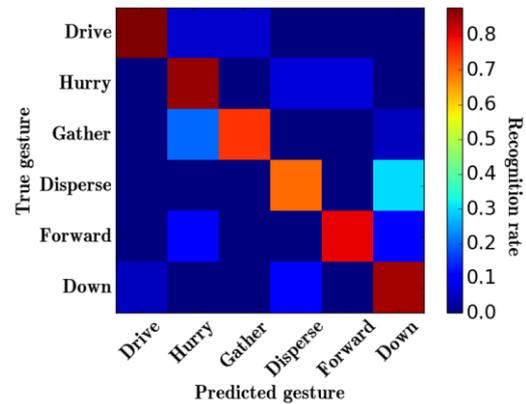


Figure 4. Confusion matrix for gestures with the chosen path cost.

References:

- [1] Coronado E, Villalobos J, Bruno B and Mastrogiovanni F 2017 Gesture-based robot control: Design challenges and evaluation with humans *IEEE Int. Conf. on Robotics and Automation (ICRA)* pp 2761–2767
- [2] Hoffmann J, Brüggemann B and Krüger B 2010 Automatic calibration of a motion capture system based on inertial sensors for tele-manipulation *Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO)* pp 121–128
- [3] Brüggemann B, Gaspers B, Ciossek A, Pellenz J and Kroll N 2013 Comparison of different control methods for mobile manipulation using standardized tests *IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)* pp 1–2
- [4] Tautges J 2012 *Reconstruction of human motions based on low-dimensional control signals* Dissertation University of Bonn
- [5] Kavarthapu D C and Mitra K 2017 Hand gesture sequence recognition using Inertial Motion Units (IMUs) *Asian Conf. on Pattern Recognition (ACPR)*
- [6] Georgi M, Amma C and Schultz T 2015 Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing *Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS)* pp 99–108
- [7] Mummadi C K, Leo F P P, Verma K D, Kasireddy S, Scholl P, Kempfle J and Laerhoven K V 2018 Real-time and embedded detection of hand gestures with an IMU-based glove *Informatics* 5 pp 28–45
- [8] Jiangqin W, Wen G, Yibo S, Wei La and Bo P 1998 A simple sign language recognition system based on data glove *IEEE Int. Conf. on Signal Processing (ICSP)* pp 1257–1260
- [9] Nandwana B, Tazi S, Trivedi S, Kumar D and Vipparthi S K 2017 A survey paper on hand gesture recognition *Int. Conf. on Communication Systems and Network Technologies (CSNT)* pp 147–152
- [10] Amin M A and Yan H 2007 Sign language finger alphabet recognition from Gabor-PCA representation of hand gestures *Int. Conf. on Machine Learning and Cybernetics (ICMLC)* pp 2218–2223

Coverage path planning by swarm of UAVs for UGV traversability analysis

L Cantelli, D C Guastella, D Longo, C D Melita and G Muscato

Dipartimento di Ingegneria Elettrica Elettronica e Informatica

Università degli Studi di Catania

Viale A. Doria 6 – 95125 CATANIA Italy

E-mail: gmuscato@dieei.unict.it

Abstract. In rough or risky environments, such as minefields, landslides or volcanic eruptions, it is extremely complex to plan safe trajectories for an Unmanned Ground Vehicle (UGV), since both robot stability and path execution feasibility must be guaranteed. In these scenarios, the adoption of a swarm of Unmanned Aerial Vehicles (UAVs) to survey the area and reconstruct 3D models of the environment can be really helpful. In this paper we will present a complete solution combining three different aspects. The first is the coverage path planning and concerns the definition of UAV trajectories for photogrammetric aerial image acquisition. When non-coverable zones are present, a suitable decomposition into subregions of the whole area to survey is performed. The second aspect is then related to the use of a swarm of UAVs to implement the coverage in a parallel way. A solution to assign the different regions among the flying vehicles will be presented, which optimises the path length of the whole swarm. The last aspect concerns the path planning of the ground vehicle, by means of a traversability analysis performed on the terrain 3D model (reconstructed from the previous aerial survey). The computed paths will be optimal in terms of difficulty of moving across the rough terrain. The results of each step of the overall approach will be shown.

1. Introduction

The problem of autonomous navigation of a UGV in outdoor environments, which are most of the times *unstructured environments*, cannot be considered fully solved in current robotics literature. In fact, the problem of path planning in such scenarios is still hard due to the difficulty of taking into account many aspects at once, such as robot kinematics and stability, terrain geometry and so on [1]. Furthermore, most of research has been carried out on structured environments, such as roads, indoor rooms, factories, where the vehicle is expected to move along clearly defined paths or regions.

A rather common approach to cope with the problem of autonomous navigation in the outdoors is to employ UAVs to provide an aerial overview of the considered environment [2,3]. In particular, photogrammetric 3D reconstruction, from aerial surveys, has gained more and more relevance over the years, thanks to the enhanced quality of results and the increased computation power available.

In this paper, an integrated strategy to solve the problem of rover path planning in unstructured environments will be presented. The three main issues faced are:

1. Coverage path planning
2. Coverage subregions assignment to a swarm of UAVs
3. Traversability costmap generation for rover optimal trajectories computation

All of these aspects will be discussed in the following sections.

2. Coverage path planning

Coverage path planning refers to a special kind of planning algorithms used for large regions surveys. Although this kind of planning can be used for any type of vehicles, namely ground, aerial or underwater vehicles, here we will focus on trajectory computation for UAVs. The first step is to define the area to survey as a 2D region, thanks to georeferenced maps. In our work we use a georeferenced Digital Elevation Model (DEM) of the environment and the area to survey is defined as a top-view over such model. A DEM is gridded representation of the environment and its cells contain the height values of the 3D structure. After that, *non-coverable zones* are defined, namely those zones where we do not want the UAV to fly above. These zones can be defined as a sequence of vertices, thus obtaining polygonal zones. At this point, the whole area is decomposed into free-to-fly subregions via a *Morse-based decomposition algorithm* [4]. A linear decomposition has been considered in this work, along either the vertical or the horizontal axis of the map frame (Figure 1). Eventually, coverage trajectories along each side of each subregion are computed. As coverage pattern the so called *back-and-forth motion* has been considered (Figure 2). Within each subregion, the optimal trajectory is chosen, which is the one with minimum number of turns. In fact, many works proved that turns are the main loss of energy and time during the execution of such kind of patterns [2-3,5]. Therefore, at the end of this step, the coverage paths for each subregion are defined. An example is shown in Figure 3.

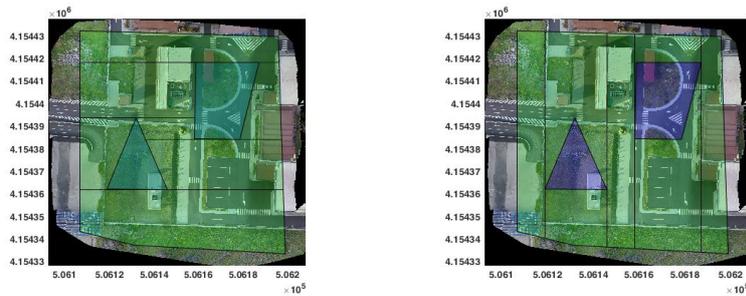


Figure 1. Example of vertical and horizontal decompositions

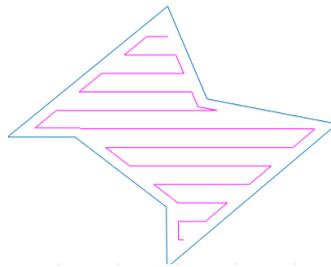


Figure 2. The back-and-forth pattern chosen as coverage trajectory within each subregion

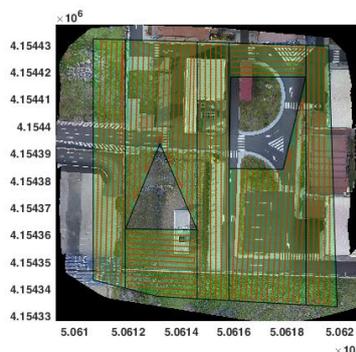


Figure 3. Example of environment decomposition and optimal coverage trajectories

These trajectories are useful in photogrammetry. In fact, by taking regularly spaced aerial pictures, if a suitable overlap is guaranteed, it is possible to derive the 3D structure of the environment, thanks to the so-called *structure from motion* approach. Nowadays, several mapping programs are available which can deliver different kinds of 3D models by processing the aerial pictures. In this work Pix4D Mapper has been used [6].

3. Coverage subregions assignment

Once subregions' coverage paths are defined, our approach exploits the use of a swarm of UAVs to parallelize the mission. This implies the definition of a rule to negotiate subregions assignment to each UAV. The strategy described in [7] has been adopted. It consists of computing the path lengths to each subregion via a 3D implementation of the A* algorithm for each UAV, from their starting positions. After that, all the possible combinations of UAVs/region are derived and, then, the combination with minimum total path length for the whole swarm is chosen. This is obtained by simply summing up the computed trajectory lengths for each UAV, for a certain combination. The enhancement in this work, with respect to [7] consists of considering both ends of the coverage trajectories as possible target positions while computing UAVs/subregion combinations. In fact, as underlined also by [2], coverage paths can be travelled indistinctly along two possible directions. In Figure 4 an example of targets assignment to a swarm of 6 UAVs is shown. It is possible to note that 3D terrain geometry is considered in trajectories computation.

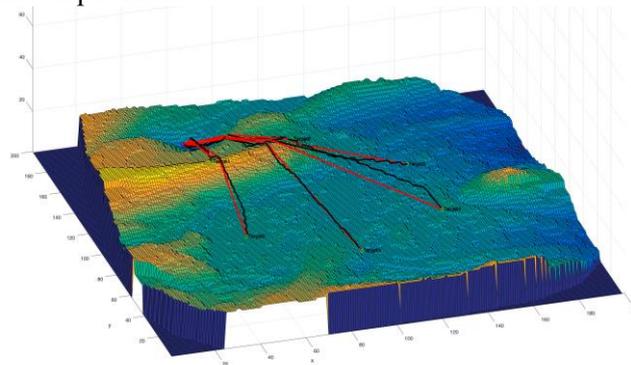


Figure 4. Terrain 3D model and UAV target assignment

4. Traversability costmap generation

Once the coverage mission is carried out by the swarm and the 3D model of the environment is computed by the mapping software, a terrain traversability analysis is performed on such model as reported in [8]. The outcome of this processing is a map including traversability costs derived by an analysis of the geometric properties of the environment. In particular, the Digital Elevation Model of the terrain is considered. In this manner the costmap can be given as heuristic to classical grid-based path planning algorithms. Thus, computed paths will result optimal in terms of crossing difficulty through the rough terrain. In this case, a D* algorithm has been used for the rover path planning. For the experimental testing the U-Go rover has been adopted [9], which is depicted in Figure 5.



Figure 5. The U-Go UGV robot adopted for the experimental tests.

5. Conclusions and future work

In this paper an integrated strategy for the efficient navigation of a rover in an outdoor unstructured environment has been presented. It exploits 3D photogrammetric reconstruction of the considered area, with the help of a swarm of UAVs to speed up the coverage mission and, therefore, the successive reconstruction.

As future development we aim at improving the subregion/UAV assignment by taking into account the coverage path lengths within each subregion, beside the distance to reach the subregion itself. Furthermore, if at least a rough 3D model of the environment is a priori known, a future enhancement would be to consider such geometry in order to keep a constant *relative* distance between the surveying UAV and the ground below.

6. References

- [1] Papadakis P 2013 Terrain traversability analysis methods for unmanned ground vehicles: A survey *Engineering Applications of Artificial Intelligence*. **26** 4 1373-1385
- [2] Torres M, Pelta D A, Verdegay J L and Torres J C 2016 Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction *Expert Systems with Applications* **55** 441-451
- [3] Di Franco C and Buttazzo G 2016 Coverage path planning for uavs photogrammetry with energy and resolution constraints *Journal of Intelligent & Robotic Systems* **83** 3-4 445-462
- [4] Acar E U, Choset H, Rizzi A A, Atkar P N, and Hull D 2002 Morse decompositions for coverage tasks *The international journal of robotics research* **21** 4 331-344
- [5] Galceran E and Carreras M 2013 A survey on coverage path planning for robotics *Robotics and Autonomous systems* **61** 12 1258-1276
- [6] Pix4D mapper, <https://pix4d.com/>
- [7] Guastella D C, Cavallaro N D, Melita C D, Savasta M and Muscato G 2018 3D path planning for UAV swarm missions *Proc. of the 2018 2nd International Conference on Mechatronics Systems and Control Engineering* 33-37
- [8] Guastella D C, Cantelli L, Melita C D and Muscato G 2017 A Global Path Planning Strategy for a UGV from Aerial Elevation Maps for Disaster Response *ICAART* **1** 335-342
- [9] Bonaccorso F, Longo D and Muscato G 2012 The U-Go Robot Proc. World Automation Congress 1-6

Acknowledgments

The presented work is in the framework of the CLARA PON project. The Project CLARA (CLOUD platform and smart underground imaging for natural Risk Assessment) is supported by MIUR under the program PON R&C SCN_00451.

Training of robots' operators with use of multirobot simulators

Communication on the work in progress

Marek Kacprzak, Institute of Mathematical Machines Scientific Foundation
Warsaw, Poland

Abstract

Training of operators of mobile devices with use of computer trainers-simulators is a widely used method nowadays. This approach is applied with reference to unmanned remote-controlled vehicles (UGVs, UAVs, USVs) as well. Typical simulator allows training of one operator at the given moment of time. But in many critical situations (like CBRNE threads, terrorist attacks, natural disasters- hurricanes, earthquakes etc) task to be done should be performed by a set of cooperating robots. Thus, training of robots' operators acting together is a must. Multirobot simulator described in the paper allows training of a group of operators cooperating at the given moment of time.

Simulators – computer trainers

History of use of simulators for training of mobile machines operators numbers about 100 years – first, mechanical simulators of airplanes for pilots' training appeared in the first decade of 1900s. Soon “electrification” of simulators ensued, and since 1960s computers have entered the air simulators field.

Now, computer simulators are in use also for training of automobile and train drivers, heavy equipment operators, for maritime training (among others for training of maritime pilots), for military training with arms and combat equipment, and many others. There are two main reasons of simulators use for training. First of them is the possibility to create extraordinary and dangerous situations in the virtual environment of simulator and repeatedly, step-by-step to train way-out. The second is the possibility of training cost reduction – even by 40% [1], [2].

The same concerns PC-based simulators for robots, which are more and more popular both for professional training and for entertainment. Majority of simulators on the market are designed for a single user. In [3] development of an software environment for simulation of UGV of different sizes “from microbots to teleoperated tanks” is presented. At present [4] simulation for training in operation of a group of UGVs is the subject of interest and work. This is exactly the same in the case of the application of robots in many critical situations (like CBRNE threads, terrorist attacks, natural disasters- hurricanes, earthquakes). Frequently tasks to be done should be performed by a set of cooperating robots. Thus, training of robots' operators acting together is a must. Multirobot simulator described in the paper allows training of a group of operators cooperating at the given moment of time.

Single robot simulators

Methodology of training based on simulators consists of two levels.

First level, introductory, is based on studies of robot's documentation, complemented with lectures, including computer presentations and video materials.

Second level is performed with use of simulators. In Fig. 1. an architecture of a typical single robot simulator is presented.

Virtual reality (or even augmented reality) and 3D technologies are applied for models of a given robot, a virtual environment and a control console. In some cases real console may be used as well. In general, training is performed on PC-class computers (desktops or laptops).

Model of a virtual robot acting in virtual environment with virtual objects is presented on a computer screen. A trainee – a robot's operator – performs his activity controlling robot with use of a console.

Only one operator may be trained on a simulator at the given moment.

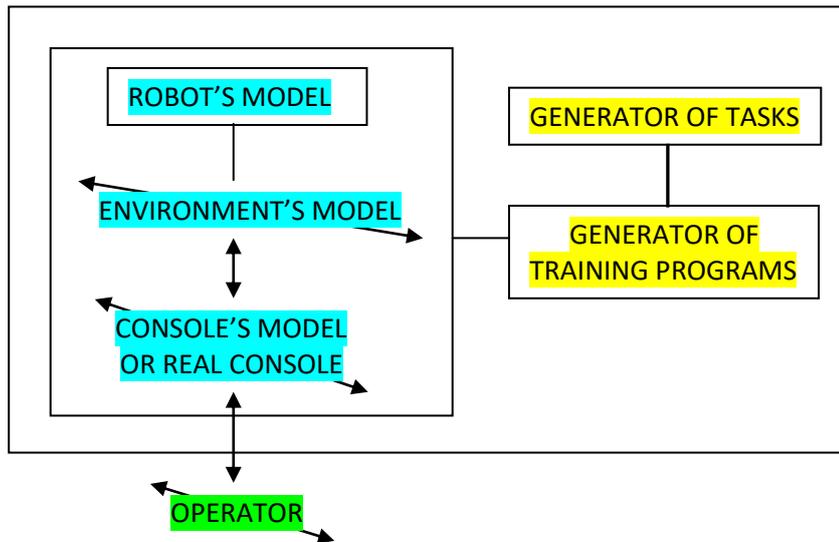


Fig. 1. Architecture of a single robot simulator

Training is organized by a trainer (an educator) – person responsible for organization, preparation, supervision and evaluation of training's results. A simulator performs an *intelligent training*:

- a trainee has to fulfill a *training program*, elaborated by a trainer, which consists of a set of *tasks* (training exercises of computer game type),
- a trainer assigns grades-points (usually different) to any task and defines a graph of a training program – a sequence of tasks to be performed by a trainee and *conditions*, which define whether to continue, repeat or finish a program (this is decided by a computer, based on results obtained by the trainee during the training),
- a trainee performs due operations controlling a robots' model with use of a control console,
- a simulator supervises the correctness of trainee's operation, evaluates the performance of a given exercise (taking into account precision, speed etc) and grants accordingly grades (points),
- at the end of a training, simulator provides a final score (in points), time taken for any task and a list of errors.

Example of a training program's graph is depicted in Fig. 2.

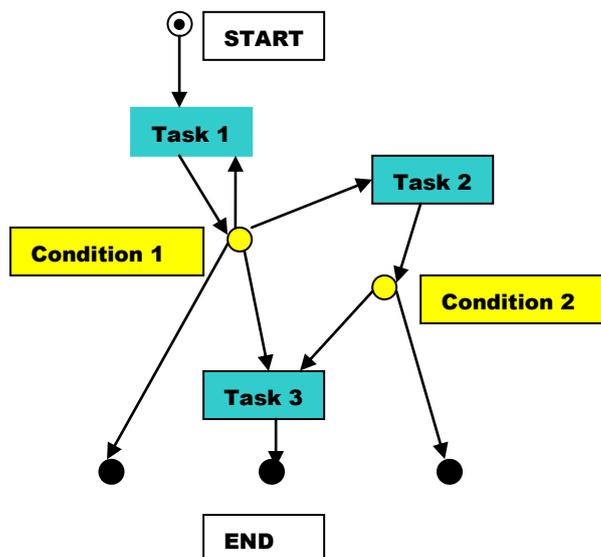


Fig. 2. Example of a training program's graph

This type of intelligent training process may be the basis for operator's certification.

In Fig. 3 there is an exemplary view of operator's console screen during a simulation run.

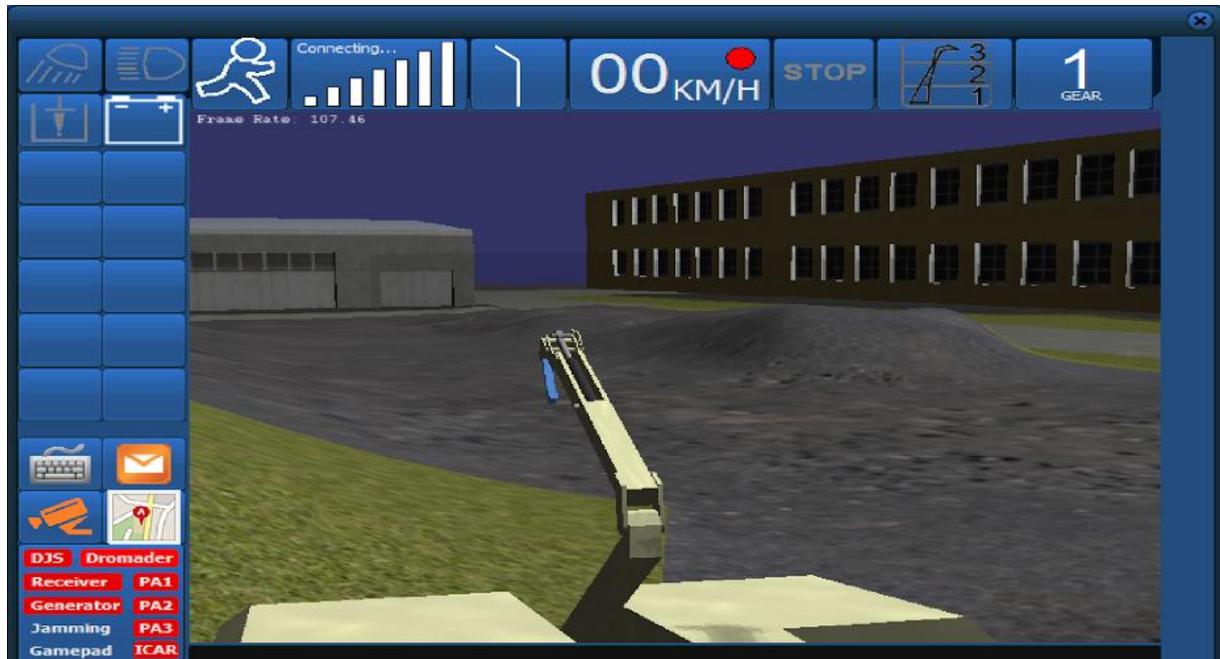


Fig. 3. Sample of a single robot's screen.

It is worth to note, that if a simulator contains virtual control console training may be performed via Internet.

Multirobot simulators

In case of tasks that have to be performed by cooperating robots, **additional, third level** of training is necessary - with use of multirobot simulators. The idea is to train simultaneously a group of trainees while performing a task that need mutual cooperation and coordination.

An architecture of a multirobot simulator elaborated within two FP7 programs [5], [6] is presented in Fig. 4. The simulator is developed in a server-client architecture on PC-class desktop. All physics computations are done on a server. As physics engine Vortex 6.2 software from CM-Labs company [7] was used. All virtual UGVs are placed in the same virtual environment. Client applications (UGV's console – cameras' viewer and joystick) are connected to a server via Ethernet. Up to ten UGVs' models may be used in training. The simulator allows interaction of several UGVs; UGV's operators can perform tasks cooperatively. Actions performed by up to ten operators are simultaneously presented on a same screen.

An intelligent training, as described for single robot simulators, is realised. Typical scenario of a training exercise in application to e.g. Humanitarian Demining activities, performed by a set of UGVs, is the following: to search a specific object, excavate it from the ground, put into a container and transport to a given place.

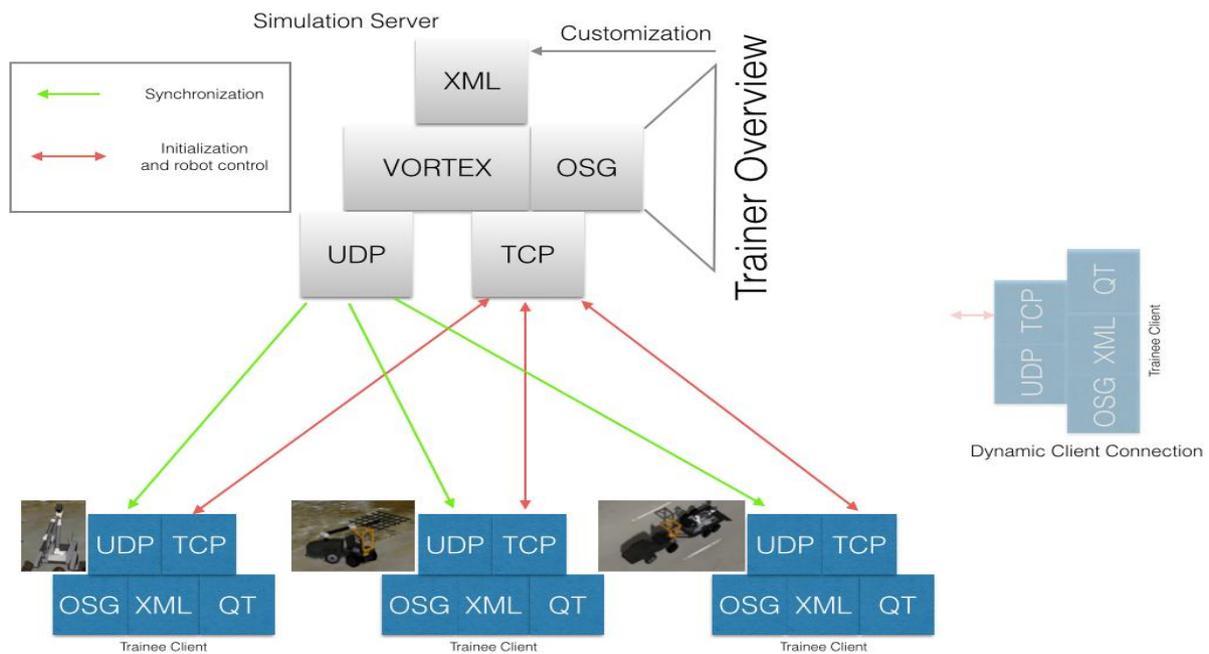


Fig. 4. Architecture of a multirobot simulator

Conclusion

Works on the multirobot simulator are continued in different areas:

- new models of different UGV's, based on robots' CAD documentation, are being elaborated, as well as of UAV's and USV's,
- new tool for support trainer's activities – a facility with expanded reporting features, is under way.

Multirobot simulator is an innovation solution, which may be a valuable tool for group training of robots' operators.

References

- [1] Kaczmarczyk A., Kacprzak M., Masłowski A., *Extension of e-learning: e-training of operators of vehicles and devices* (in Polish), *Elektronika* 12/2009.
- [2] Rizzatti L., *When to use simulation, when to use emulation*, http://s3.mentor.com/public_documents/whitepaper/resources/mentorpaper_85748.pdf.
- [3] Lif P., Hedström J., Svenmarck P., *Operating Multiple Semi-autonomous UGVs: Navigation, Strategies, and Instantaneous Performance*, 2007, <http://www.springerlink.com/content/c670125406w0r7wm/fulltext.pdf> acc. 17.05.2012.
- [4] Craighead J. et al., *Validating The Search and Rescue Game Environment As A Robot Simulator By Performing A Simulated Anomaly Detection Task*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008.
- [5] TIRAMISU – Toolbox Implementation for Removal of Anti-personnel Mines Submunitions and UXO, FP7, grant agreement no: 284747.
- [6] ICARUS – Integrated Components for Assisted Rescue and Unmanned Search Operations, FP7, grant agreement no: 285417.
- [7] CM Labs www.cm-labs.com.

Human arm motion tracking using IMU measurements in a robotic environment

Robin Pellois and Olivier Brüls

Department of Aerospace and Mechanical Engineering, University of Liège

E-mail : {robin.pellois,o.bruls}@uliege.be

Abstract. Human-robot interactions (HRI) is an emerging paradigm that aims at combining complementary skills of robot and human. The meaningful human arm motion represent an interesting way of communication to explore with robot. IMUs appear as a simple, lightweight, easy-to-use, technology for human motion tracking compared to other systems such as opto-electronic devices. However, IMUs require important data treatment to reconstruct human motion and are usually coupled with a magnetometer or even other sensors. This paper explores a method only based on IMUs (accelerometer and gyroscope) to track human motion in order to keep the simplicity and robustness of IMUs in an industrial environment with magnetic disturbances. The signal processing method presented here limit the well-known drift of the gyroscope by gravity measurement.

1 Introduction

Human-robot interactions (HRI) is a research field that aims at combining the accuracy and repeatability of a robot and the versatility of a human. One aspect of current robots is their skill-demanding programming methods. The survey [1] opposes manual to automatic programming methods. The first one involves the modification of the robot program directly by a skilled operator while, in the second one, the robot itself modifies its program according to external information. A branch of this emerging robot programming method exploits human ways of communication like upper limb motions. In this work, Inertial Measurement Units (IMUs) sensors are used to measure human arm motion. Among the available technologies, IMUs have the benefit of being easy to used, lightweight, wireless and cheap compared to light-sensitive and expensive vision-based technologies. Exoskeletons are also used but only enable to measure human joint parameters, making it difficult to map the human motion to the robot frame.

One largely used approach in human motion tracking consists in computing human joint parameters from the relative orientation of sensors attached to human segments before and after joints [2]. This method presents the same inconvenient as exoskeletons. Another approach is selected in the present work and described later.

In any case, the orientation of the IMU should be estimated based on the sensor raw data. Many different methods are reviewed in [3]. Most of them combine IMUs (accelerometer and gyroscope) with a magnetometer (noted MIMUs) measuring the Earth-magnetic north as in [4] and consist in solving the Wahba's problem. However, electromagnetic disturbances in an industrial environment may jeopardize the measurement of the Earth magnetic field. Some alternatives have been proposed which fuse the IMU's data with other sensors [2]. Other solutions with only two accelerometers is proposed in [5], for which the distance and orientation between the two sensors has to be perfectly known. But these sensors cannot be too close from each other making this technology rather bulky for a wearable device.

This work presents a new approach, suitable for an industrial context, to limit gyroscope drift with a gravity-based orientation estimation. This approach has been mentioned in a previous work [6] and is detailed here. First, the equipment is presented. Then, the strategy for arm motion tracking is shortly exposed, in order to introduce the new approach. Finally, some experimental results are given.

2 Equipment

The sensor modules used in this work have been developed by the Microsys lab from the University of Liège [7]. These wireless platforms are composed of a 3-axis IMU from Bosch (BMI160) and transmit data at the frequency

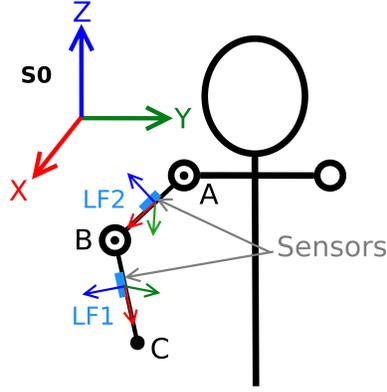


Fig. 1: Absolute segment orientation approach

of 100hz to a *Raspberry Pie 3*. The module measures 2 data sets with respect to local sensor frame (noted LF):

- The acceleration noted a^* :

$$a^* = a + g = \begin{bmatrix} a_x + g_x \\ a_y + g_y \\ a_z + g_z \end{bmatrix} \quad (1)$$

with a representing the linear acceleration of the sensor and g the Earth gravity field.

- The angular velocity ω^* :

$$\omega^* = \begin{bmatrix} \omega_x^* \\ \omega_y^* \\ \omega_z^* \end{bmatrix} \quad (2)$$

3 Trajectory computation

The problem addressed in this work is to measure the human motion in a robotic application. In a first step, the given objective is to compute the trajectory of the wrist with respect to the shoulder. The selected approach consists in computing directly the orientation of each segment of the human arm (arm and forearm) with respect to an inertial frame using two sensors as described in figure 1. The inertial lab frame noted S_0 has its Z-axis pointing vertically upwards. The sensors enable to compute the rotations ${}^{S_0}R_{LF_1}$ and ${}^{S_0}R_{LF_2}$. Thus, the trajectory $\vec{AC}_{S_0}(t)$ is computed as:

$$\vec{AC}_{S_0}(t) = {}^{S_0}R_{LF_1}(t)\vec{AB}_{LF_1} + {}^{S_0}R_{LF_2}(t)\vec{BC}_{LF_2}$$

The proposed method to compute the two rotations is based on an incremental quaternion-based estimation of the sensor orientation. At every time step n , the orientation of the local frame LF_n (either for sensor 1 or 2) with respect to the inertial frame S_0 is estimated from the previous quaternion q_{n-1} representing the rotation from the inertial frame S_0 to the local frame LF_{n-1} as follows :

$$q_n = q_{n-1} + h\left(\frac{1}{2}q_{n-1} \otimes \omega_{q_n}\right)$$

with h the time step value, ω_{q_n} is the quaternion representation of the angular velocity ω_n at time step n and ω_n is the gyroscope measurement (see Eq (2)):

$$\omega_n = \omega^*$$

After this operation, q_n is then normalized. This boils down to a direct quaternion based integration of the angular velocity, which is subject to a well-known drift over time usually overcome by extra sensors [3].

A method has been implemented in order to limit this drift without extra sensor. This method, based on gravity measurement, can be used only during slow or no motion phases such that $a^* \approx g$. Thus, at time step n , the gravity

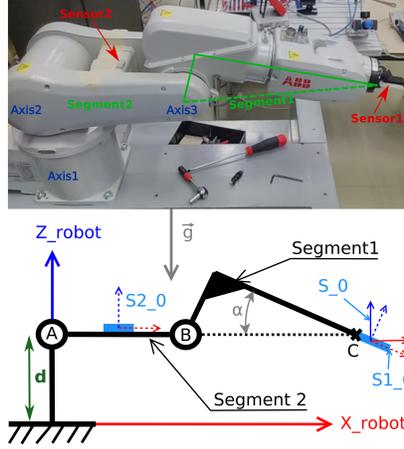


Fig. 2: Setup for comparison between robot trajectory and IMUs measured trajectory

vector g_n with respect to the local frame LF_n is close to the normalized accelerometer measurement : $g_n \approx \frac{a_n^*}{\|a_n^*\|}$. The angular velocity ω_n can be developed as follows:

$$\omega_n = (I - g_n g_n^T) \omega_n + g_n (g_n^T \omega_n) \quad (3)$$

with I is the 3-by-3 matrix identity. The first term of the equation (3) is the projection of ω_n in the plane perpendicular to g_n and approximated by:

$$(I - g_n g_n^T) \omega_n \approx \frac{1}{h} (g_n \wedge g_{n-1})$$

with h the time step value and g_{n-1} the gravity vector with respect to the sensor frame LF_{n-1} . The second term of the equation (3) is computed from the gyroscope measurement. The expression of ω_n become:

$$\underline{\omega}_n \approx \frac{1}{h} (g_n \wedge g_{n-1}) + g_n (g_n^T \omega^*)$$

In order to detect phases with negligible linear acceleration, the following criterion is implemented: if the acceleration norm is around 1 g-unit : $0,9 < \|a^*\| < 1,1$ and the norm of the gyroscope close to 0 degree/sec : $\|\omega^*\| < 1,1$ then the sensor is considered not undergoing any linear acceleration.

As the orientation along the path is computed incrementally the initial orientation q_{init} of each sensor frame with respect to the inertial common frame S_0 has to be determined. A procedure, relying on the measurement of the gravity field, was proposed and discussed in [6].

4 Results

This algorithm is tested on an *IRB 120* robot from ABB company. The robot simulates the motion of a human arm and the trajectory of its end-effector, recorded by its controller, is used as reference. Two IMU sensors have been mounted on the robot arm as described in the figure 2. The robot axes 1 and 2 represent the shoulder and the axis 3 the elbow, the segment 2 the arm and the segment 1 the forearm. The last 3 axes (4, 5 and 6) of the robot are not activated. The sensor 2 is set in a way its X-axis is aligned with the direction of the segment 2. The sensor 1 X-axis is not aligned with the direction of the segment 1, but still in the XZ-plane of the robot. This misalignment is managed by an initialization procedure. The S_0 frame, centered on A, has its Z-axis along gravity. It is assumed that the Z-axis of the robot is also aligned with the gravity vector. The X-axis of S_0 is along the \overrightarrow{AC} direction at the initial time step which is made to be parallel with the X-axis of the robot-base frame. That way, only an offset d has to be subtracted to the robot trajectory to express both trajectories with respect to the same frame. Figure 3 shows the good correlation between both trajectories of the robot end-effector measured by the robot itself and by the sensors.

3D representation of the same trajectory measured by the robot and by the sensors

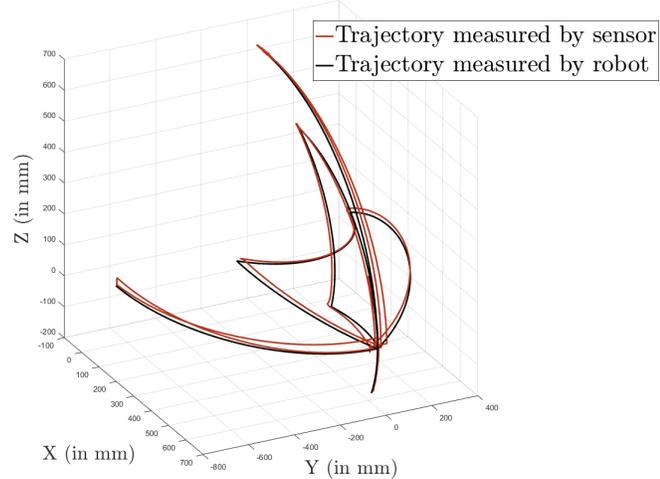


Fig. 3: Trajectory comparison between robot measurement and IMUs measurement

5 Conclusion

IMUs appear as an interesting way to measure human motion in an industrial environment. Many methods exist to compute human upper limb trajectory from IMU sensors but only a few are suitable for a robotic application. The proposed solution does not use magnetometers because of their sensitivity to electromagnetic disturbances. Only the gyroscope signal is used, completed by acceleration measurement to limit the drift from the gyroscope in slow or no motion phases. A consistent reconstruction of the trajectory is achieved but the accuracy of the measured trajectory could be further improved. The future work will consist in measuring the accuracy of the raw sensor data, of the orientation and finally of the complete trajectory in order to improve it.

Acknowledgements

This work was supported by the Interreg V programme of the Greater Region within the Robotix Academy project.

References

- [1] G. Biggs and B. Macdonald, "A survey of robot programming systems," in *in Proceedings of the Australasian Conference on Robotics and Automation, CSIRO*, 2003, p. 27.
- [2] D. Vlastic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popovic, "Practical motion capture in everyday surroundings," *ACM Transactions on Graphics (TOG)*, vol. 26, Jul. 2007.
- [3] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, "Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion," *Sensors (Basel)*, Jun. 2017.
- [4] A. M. Sabatini, "Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing," *Sensors*, 2011.
- [5] J. Lee and I. Ha, "Sensor fusion and calibration for motion captures using accelerometers," *IEEE International Conference on Robotics and Automation*, 1999.
- [6] R. Pellois and O. Bruls, "Human arm motion measurement by imus for robotic applications," *journal*, 2018.
- [7] P. Laurent, F. Dupont, S. Stoukatch, and F. Axisa, "Ultra-low power integrated microsystems," *Smart Systems Integration Conf. (SSI2015)*, Mar. 2015.

Feedforward command computation of a 3D flexible robot

Arthur Lismonde and Olivier Bruls

Department of Aerospace and Mechanical Engineering, University of Liège, {alimonde,o.bruls}@uliege.be

1 Introduction

Robotic manipulators with a lightweight structure can present some interesting features. Thanks to their reduced weight and stiffness, lightweight robots could achieve high speed tasks while being safer and more efficient than traditional rigid robots. However, when designing the controller of such systems, elastic behaviors should be accounted for in order to prevent unwanted vibrations.

In order to have a motion of the manipulator with reduced vibrations, the latter can be fed back to the controller so that proper compensation can be done [1]. By analyzing the system, appropriate feedforward inputs can also be designed in such way that the resulting motion has decreased elastic deflections. Both feedforward and feedback action can be combined to achieve robust performances [2, 3].

This work focuses on the feedforward control of 3D flexible manipulators. Based on a model of such flexible multibody systems (MBS), the inverse dynamics is solved to compute the feedforward input of the manipulator. Different methods can be used to model flexible robotic arms. Lumped mass elements models are widely used to model robotic systems [4]. Indeed, to represent the manipulator and its flexibility, this modelling technique uses a limited number of parameters and is therefore quite suitable for control purposes [5]. On the other hand, the finite element modelling approach is a general way to model MBS [6] that is able to represent distributed link flexibility. Here, the case study of a flexible joint Sawyer robot is presented. Flexibility in the joint of the robot is modelled using lumped spring and damper elements and the inverse dynamics of the system is solved using the optimization approach [7] extended to 3D problems. The computation of the feedforward input command is discussed and experimental results on the real system are presented.

2 Inverse dynamics formulation

The dynamics of multibody systems, such as robotic manipulators, can be described using rigid bodies and flexible bodies connected through kinematic joints, springs and dampers elements. The kinematics of such system is described using its generalized coordinates \mathbf{q} . In the case of a robotic manipulator, some actuators can exert some torques (or forces) \mathbf{u} on the bodies to move a specific end-effector along a given trajectory. The dynamics of the system can be governed by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{A}\mathbf{u} \quad (1)$$

$$\boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0} \quad (2)$$

$$\mathbf{y}_{eff}(\mathbf{q}) - \mathbf{y}_{presc}(t) = \mathbf{0} \quad (3)$$

where \mathbf{q} are the generalized coordinates of the system, \mathbf{M} is the system symmetric mass matrix, \mathbf{v} is the vector of nodal velocities, \mathbf{g} is the vector of internal and complementary inertia forces, \mathbf{B} is the gradient of the kinematic constraints $\boldsymbol{\Phi}$, which are used to represent the connections imposed by the kinematic joints. The matrix \mathbf{A} is a boolean matrix that applies the controls \mathbf{u} on the system. The m dimensional vector $\boldsymbol{\lambda}$ is the Lagrange multipliers related to the m kinematic constraints $\boldsymbol{\Phi}$. The last equation is called the servo constraint and fixes a part of the motion. It assures that the end-effector position \mathbf{y}_{eff} follows the prescribed trajectory $\mathbf{y}_{presc}(t)$.

In this work, the inverse dynamics problem i.e., finding the control inputs that satisfy the servo constraint, is solved by formulating a constrained optimization problem minimizing an objective function $J(\mathbf{q})$

$$\min_{\mathbf{q}} J(\mathbf{q}) \quad (4)$$

related to the configuration of the robotic manipulator under the constraint defined by the equations of motion Eqs. (1)-(3).

3 Lumped mass element model

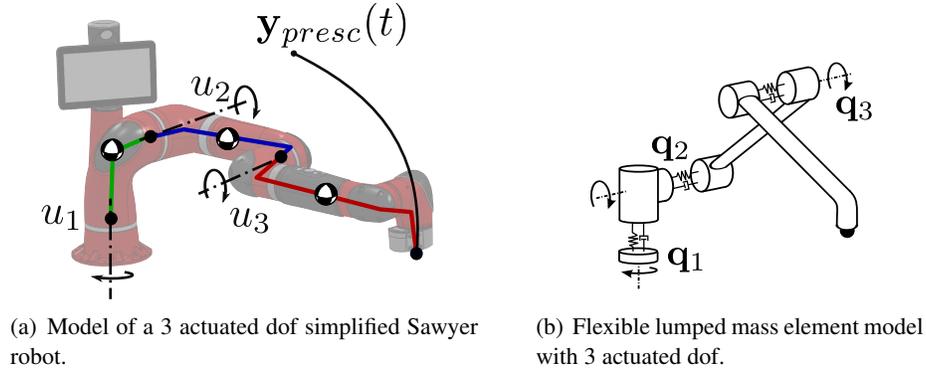


Fig. 1: Model of an elastic Sawyer robot.

To illustrate this methodology with a real case study, the feedforward action of a flexible Sawyer robot performing a trajectory tracking tasks is considered. This seven degree of freedom (dof) elastic robot is simplified to be a three actuated dof robot by locking four of its joints. The resulting model of the robot has three links and three actuated dof controlled using inputs u_i , with $i = 1, 2, 3$, as shown in Fig. 1(a). The joints of the Sawyer robot are constructed using *series elastic actuators* [8]. The flexure spring inside these actuators result in a joint with intrinsic flexibility. Here, these flexible joints are described using two angles: one describing the motor related angle $q_{M,i}$ and the other one describing the link related angle $q_{L,i}$. Flexibility is modeled using a spring-damper pair that connects $q_{L,i}$ to $q_{M,i}$ [4, 5]. Each link is modeled as a rigid body and the end effector y_{eff} is located at the tip of the third link.

Once the model is built, the inverse dynamics is solved by formulating a constrained optimization problem, where the objective is to minimize the elastic energy J inside the spring-damper pairs. Mathematically,

$$\min_{\mathbf{q}} J = \min_{\mathbf{q}} \frac{1}{2T} \sum_{i=1}^3 \int_t k_i (q_{M,i} - q_{L,i})^2 dt \quad (5)$$

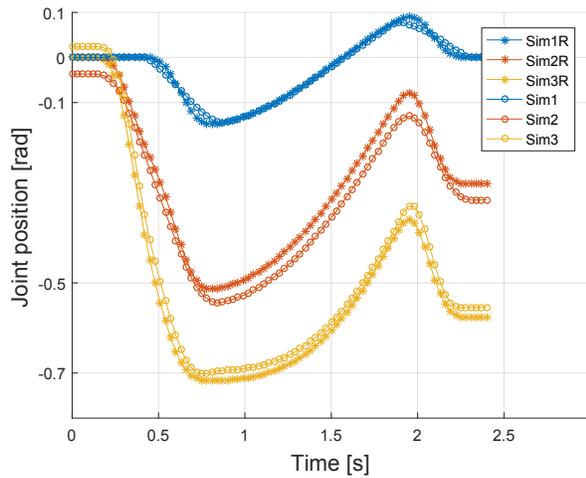
with k_i is the stiffness of spring i and T is the total duration of the motion.

4 Results and discussion

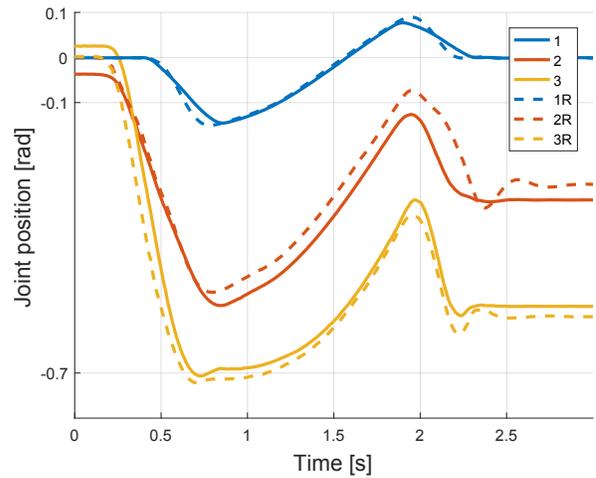
The trajectory imposed at the robot end-effector is an oscillating motion in space built by combining a seventh order polynomial with a sine function. In Fig. 2(a), the computed input motor positions $q_{M,i}$ with the flexible joint assumption are represented with the bullet lines. They are compared to the inputs that would be generated for an equivalent robot with rigid joints (star line). The inputs with the rigid joint assumptions can be computed algebraically based on the trajectory at the end-effector. Besides the visible offset due to the compensation of gravity, the flexible joint assumption generates inputs that start slightly earlier than with the rigid assumption (visible for joint 1).

These inputs are sent to the real robot at a rate of 500 Hz and the resulting link positions $q_{L,i}$ measurements are shown in Fig. 2(b). With the rigid assumptions in dashed lines, one can see that some residual vibrations are present in the joints at the end of the trajectory (around 2.25 sec). One can also observe that, the second link does not manage to follow correctly the desired trajectory and a small delay can be observed at around 2 sec. When the inputs computed with the flexible assumption are used, the link angle and the input motor angle do not have significant differences: joint deflections are better compensated and almost no vibration is visible.

It is important to note that in the above results, only the default PD feedback controller on each joint is used. No additional feedback on the end-effector state was implemented here.



(a) Computed inputs $q_{M,i}$ without and with flexible joint assumption.



(b) Experimental measurements of link angle $q_{L,i}$ with flexible and rigid joint control.

Fig. 2: Computed inputs $q_{M,i}$ and measured link angle $q_{L,i}$.

5 Conclusion

This case study is designed to validate the optimization formulation to solve the inverse dynamics of 3D flexible robot. In this method, the inverse problem is stated as a minimization of the elastic deformation energy in the system. The optimization has to satisfy some constraints defined by the dynamics of the system and the trajectory tracking task. For this experimental example of a Sawyer robot with flexible joints, it was possible to reduce the resulting vibrations of the end-effector. More complex feedback control laws could be implemented in order to further improve the results. The end-effector acceleration could be monitored using accelerometers and additional compensations could be designed. The end-effector trajectory could also be monitored using an external camera in order to measure tracking performances.

Although not shown here, experimental tests on flexible link serial robot were also performed. In that case, flexibility is modeled using beam finite elements and the same methodology is applied to solve the inverse dynamics of the system. Those results will be presented in further publications.

Acknowledgements

The first author would like to acknowledge the Belgian Fund for Research training in Industry and Agriculture for its financial support (FRIA grant).

References

- [1] R. Cannon and E. Schmitz, "Initial experiments on the end-point control of a flexible one-link robot," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 62–75, 1984.
- [2] A. De Luca, "Feedforward/feedback laws for the control of flexible robots," *In Proceedings of the IEEE International Conference on Robotics & Automation*, 2000.
- [3] J. Malzahn, M. Ruderman, A. S. Phung, F. Hoffmann., and T. Bertram, "Input shaping and strain gauge feedback vibration control of an elastic robotic arm," *In Proceedings of IEEE Conference on Control and Fault Tolerant Systems*, 2010.
- [4] S. Moberg, *Modeling and Control of Flexible Manipulators*. PhD thesis, Linköping University, 2010.
- [5] P. Stauer and H. Gatringer, "State estimation on flexible robots using accelerometers and angular rate sensors," *Mechatronics*, vol. 22, no. 8, pp. 1043 – 1049, 2012.

- [6] V. Sonneville and O. Brüls, “A formulation on the special Euclidean group for dynamic analysis of multibody systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 9, 2014.
- [7] G. J. Bastos, R. Seifried, and O. Brüls, “Inverse dynamics of serial and parallel underactuated multibody systems using a DAE optimal control approach,” *Multibody System Dynamics*, vol. 30, pp. 359–376, 2013.
- [8] G. A. Pratt and M. M. Williamson, “Series elastic actuators,” in *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Part 3 (of 3)*, 1995.

Radar Based Through-Wall Mapping

Sedat Dogru¹, Lino Marques

Institute of Systems and Robotics

Department of Electrical and Computer Engineering

University of Coimbra

3030-290 Coimbra, Portugal

E-mail: {sedat, lino}@isr.uc.pt

Abstract. Autonomous robots have been in use to construct maps of unknown areas, offering indispensable help for various emergency situations, including but not limited to hostage rescue and fire accidents. In these situations, time and safety are paramount requiring prompt and accurate action. Through-wall mapping capability, allowing robots to create a map without physically visiting all the space of interest, provides both a faster and safer way to create a map. In this paper, we show that an off-the-shelf automotive radar can be used to detect obstacles placed behind walls, and hence create a proper map for various configurations.

1. Introduction

Autonomous robots are frequently used to construct maps of unknown areas in an effective manner. These maps can later be used in search and rescue missions. Traditional mapping approaches depend on line of sight sensors, like cameras, sonar and lidar, and they require the visit of the robot to all the spaces of the environment. This requirement may not be easily met in a hostage situation, where some passages are blocked and hence the robot is not able to traverse the whole space. Additionally, the bare appearance of a robot may annoy hostage-takers endangering the hostages' life. These concerns support the research in through wall imaging, which utilizes the fact that electromagnetic waves of proper frequency can pass through building materials. However, this pass-through capability is not uniform, and depends both on the frequency of the wave and the building's material, being able to easily pass through materials like wood, plasterboard wall etc. but being considerably attenuated by reinforced concrete, which has a mesh of iron bars [1-3]. This has led researchers to try various methods to study through-wall imaging capabilities, with varying quality of results. Narayanan et al. [4], and Amin and Ahmad [5] were able to coarsely detect large objects behind walls. Aftanas and Drutarovsky [6] were able to roughly reconstruct the interiors of a wooden building, whereas Tan et al. [7] were able to roughly reconstruct a second wall behind a plasterboard wall. The best results reported in the literature up to now are constrained to simulations reported by Le and Dogaru [8], who

¹ To whom any correspondence should be addressed.

reconstruct a 2-story building. In this work, we investigate the feasibility of using an off-the-shelf Short Range Radar (SRR) working at 24GHz frequency, like the ones used in cars, for through-wall mapping of walls made of wood or plaster.

2. Method

An accurate map needs multiple measurements, collected from different positions. This can only be achieved with the aid of a proper localization system, which is able to accurately estimate the position of the measuring device. For this purpose, in this work, a mobile robot equipped with odometry, lidar and radar was used. The lidar data was fused with odometry and an improved estimation was achieved. Then the radar measurements obtained as the robot traveled through the environment were tagged with the corresponding robot positions obtained from the localization system. These tagged measurements were then used to construct an occupancy grid map by merging the measurements using a probabilistic sensor model for the detections.

The short range radar returns range and bearing information for a fixed number of obstacles in its field of view, similar to a lidar in terms of the reported data. However, in lidar mapping, all the space up to a detection is assumed to be empty, which is not a valid assumption in a through-wall mapping system. Additionally, the radar's probability of detection depends on various factors like the orientation of the target surface, its reflectivity, as well as existence of stronger reflectors in the environment. These issues require both scanning the testing area using a range of radar bearing angles and a sensor model that does not update empty space, but instead just updates detections, similar to [10], who update free space only conditionally.

The occupancy probability of a map m is calculated using all the radar measurements z and the corresponding robot positions x obtained up to time t . In order to simplify, the occupancy probability of each cell is assumed to be independent, and hence the following approximation is used

$$p(m|x_{1:t}, z_{1:t}) = \prod_i p(m_i|x_{1:t}, z_{1:t}) \quad (1)$$

where m_i represents each cell i . The probability $p(m_i|x_{1:t}, z_{1:t})$ is updated recursively using a binary Bayes filter. In order to avoid numerical issues the probability is represented in log odds form as

$$l_{t,i} = \log \left(\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})} \right) \quad (2)$$

and the following formula is used to update it recursively [9].

$$l_{t,i} = l_{t-1,i} + \log \left(\frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} - l_{0,i} \right) \quad (3)$$

The final occupancy probability of the map is later obtained by inverting the log odds ratio given in equation (2).

3. Experimental Setup



Figure 1. The differential drive robot used in the experiments. The radar can be seen on the top of the robot. **Figure 2.** The testing arena in one of the mapping configurations.

In this work, a short range radar from Cobham Ltd was utilized. The radar, reporting various information like range, bearing and signal to noise ratio of up to 10 detections in a 30 m range, was put on top of an indoor mobile robot, a Nomad Super Scout II (Figure 1), using a servo motor, which allowed changing bearing of the radar as the robot moved in the testing arena. The experiments were run in an indoor environment built out of portable wall segments, forming an enclosed arena of 4m x 4m (Figure 2). A set of obstacles were put inside the arena in various configurations, and the robot was moved around, probing the arena with its radar.

4. Results & Discussion

In the first setup (Figure 3), just an outer wall was built and two circular hollow obstacles were put inside the arena. The robot was driven around the arena, following the trajectory shown in red in figure 4. This figure shows that the robot is able to properly detect the two circular obstacles behind the wall. Fully reconstructing the fourth side of the obstacles would require scanning from the fourth side of the arena as well. The second setup (Figure 5), consisting of two layers of walls and a hollow circular obstacle was also properly reconstructed by the robot (Figure 6). In both cases, constructed maps can be seen to contain some ghost images, which are randomly distributed however with a lighter intensity and hence they can be easily filtered out for use in a mission.

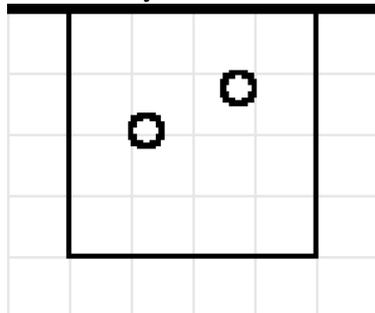


Figure 3. Setup showing two round obstacles inside a closed area.



Figure 4. The map constructed for the setup of figure 3.

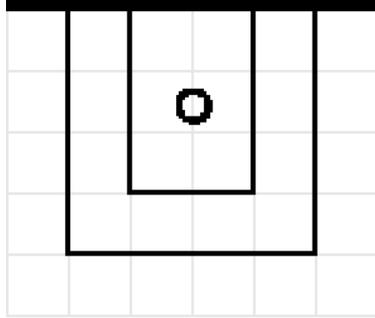


Figure 5. Setup showing one obstacle behind two layers of walls.

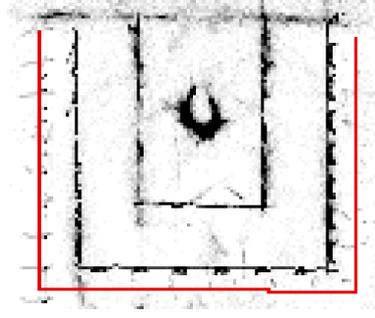


Figure 6. The map constructed for the setup of figure 5.

REFERENCES

- [1] Safaai-Jazi A, Riad S M, Muqaibel A and Bayram A 2002 Ultrawideband propagation measurements and channel modeling: Through-the-wall propagation and material characterization *DARPA NETEX Program* Arlington VA
- [2] Richalot E, Bonilla M, Wong M-F, Fouad-Hanna V, Baudrand H and Wiart J 2000 Electromagnetic propagation into reinforced-concrete walls *IEEE Transactions on Microwave Theory and Techniques* **48(3)** 357–366
- [3] Dalke R A, Holloway C L, McKenna P, Johansson M and Ali A S 2000 Effects of reinforced concrete structures on RF communications *IEEE Transactions on Electromagnetic Compatibility* **42(4)** 486–496
- [4] Narayanan R M, Xu X and Henning J A 2004 Radar penetration imaging using ultra-wideband (UWB) random noise waveforms *IEE Proceedings - Radar, Sonar and Navigation* vol. 151(3) pp 143–148
- [5] Amin M G and Ahmad F 2014 Through-the-wall radar imaging: Theory and applications *Academic Press Library in Signal Processing* Elsevier vol. 2 pp 857 – 909.
- [6] Aftanas M and Drutarovsky M 2009 Imaging of the building contours with through the wall UWB radar system *Radioengineering* **18(3)** 258–264
- [7] Tan B, Chetty K and Jamieson K 2017 ThruMapper: Through-wall building tomography with a single mapping robot *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications (New York, NY, USA: ACM)* pp 1–6
- [8] Le C and Dogaru T 2012 Synthetic aperture radar imaging of a two-story building *Radar Sensor Technology XVI* vol. 8361 p 83610J
- [9] Thrun S, Burgard W, and Fox D 2005 *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)* The MIT Press
- [10] Werber K, Rapp M, Klappstein J, Hahn M, Dickmann J, Dietmayer K and Waldschmidt C 2015 Automotive radar gridmap representations *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)* pp 1–4

Development of the modular platform for educational robotics

Semenyaka A., Poduraev Y.

Moscow State Technological University «STANKIN»

1. Abstract

This article presents a new concept in the application of educational robotics. At this stage, the development of a prototype of a modular mobile platform and its software is presented. The final goal of the project is the creation of a robotic stand that provides students with the ability to visualize group control algorithms, as well as a web interface for remote testing of algorithms used in group robotics. The main goal of this development is to expand the scope of training robotics and make it accessible not only for schoolchildren, but also for students.

2. Introduction

We live in a digital age, where science and technology are the two key concepts for understanding the fundamentals of the various devices that people face every day.

Robotization of various fields of science and technology is gaining momentum every year. For example, mobile robotics finds application in many areas of our life, such as medicine, military intelligence, service and so on. In addition to these areas, mobile robotics are also involved in education. Mobile robotics is an excellent area suitable for education, because it integrates mechanics, electronics, computer science and programming [1].

The trend of evolution of educational robotics increases every year. As a result, it is necessary to orient young people to study the foundations of the development of modern technologies.

There are many robotics construction kits on the market, such as Fischertechnik, Lego, Huna, MakeBlock and so on. But as statistics show, the majority of existing construction kits and teaching methods have age limits.

Educational robotics expands opportunities for teaching children by receiving practical skills. Most importantly, educational robotics provides an interesting and exciting learning environment because of its practical application and integration of various technologies.

Firstly, the usage of robotics in the educational process stimulates motivation of children to acquire knowledge.

Secondly, it is an evolvment of children's interest in technology and programming.

Thirdly, it is an amplification of programming skills, development of logical and algorithmic thinking.

In conditions of automation of education, we need to search new approaches to evolve of algorithmic abilities of children. The old approach to teach students programming uses only programming languages (Pascal, BASIC). It no longer meets the realities of today. Modern education requires a more active integration of robotics into educational processes [2].

In this article we present an innovative concept in the field of educational robotics, namely the process of creating a prototype of a modular mobile platform: its design, hardware and software.

3. Relevance

Despite all the diversity of educational robotics, existing construction kits have clear age limits, have closed and inflexible software. Extensibility of constructive capabilities of these construction kits is limited, and additional details to the basic kits are expensive. Therefore, the idea of creating a universal, flexible construction kit is very relevant nowadays.

4. Problem statement

Educational robotics is very popular in modern society. There is a great variety of construction kits, a lot of fields of robotics, more and more adolescents want to learn the fundamentals of robotics, learn electronics and design [3]. But again, there is the problem of creating a universal construction kit are not only for children but also for students. From here arise such issues as:

Construction kits age is limited. A very wide range of construction kits are represented on the market, but usually they are all intended for preschool children or teenagers.

Price. The cost of the basic set of the construction toy can be quite low, but the sets often don't have sufficient electrical components, the purchase of which is a necessity for the possibility of working with this construction toy.

Inflexible software. If you purchase a construction kit for a child of preschool age, then for this age category it will be the most intuitive interface and fairly simple programming basics. If you want to move to a new level of programming, you will have to buy additional software or even a new construction set.

5. Conception

We propose a new approach to educational robotics, which alleviates the main shortcoming of many construction toys, namely the age limit, allowing not only schoolchildren, but also students, to use the construction kit for educational purposes. This approach solves the problem of age restrictions and helps to prepare a suitable base for further, more serious stages of education in this direction.

The developed concept provides a training base for all ages, with different levels of training activity. It allows to acquire skills in design, programming and working with electronics.

As the result of the multi-level software, the user has the opportunity to continuously improve their knowledge in programming. The flexibility of the software consists of the fact that for each age group, there is an intuitive interface and depending on the age of the user, his knowledge. If we are talking about primary school students, it should be noted, that the interface and the submission of methodological material should be given in a game form, but the programming capabilities at this level are rather limited. Obviously, when you move to the next steps, the level of programming complexity increases. One of the advantages of our concept is the presence of a camera in sets, due to which the user can work with technical vision. Another main focus is the availability of the photogrammetric system, (MS «Kinect» is currently in use). It allows to implement centralized, group control of robots, and in case of a robotic stand, it is a necessary attribute.

The main element of this concept is the modular mobile platform. The basic configuration of the platform assumes the presence of a control unit, and 2 units with motors.

The concept is described in Table 1

Table 1

Age group	Individuals	Organizations
1-4 grade	<ul style="list-style-type: none"> • Platform in basic set • Set of sensors • Software: work with sketches • Study guide 	<ul style="list-style-type: none"> • A group of robots in the basic set • Cameras • Kinect • Software: work with sketches • Network connectivity of groups of robots • Study guide
5-7 grade	<ul style="list-style-type: none"> • Platform in basic set • Set of sensors • Software: programming in the form of algorithmic blocks, the ability to add simple blocks with elements of program code • Study guide 	<ul style="list-style-type: none"> • A group of robots in the basic set • Cameras • Kinect • Software: programming in the form of algorithmic blocks, the ability to add simple blocks with elements of program code

		<ul style="list-style-type: none"> • Network connectivity of groups of robots • Study guide
8-9 grade	<ul style="list-style-type: none"> • Platform in basic set • Set of sensors • Software: programming on Python • Study guide 	<ul style="list-style-type: none"> • A group of robots in the basic set • Cameras • Kinect • Software: programming on Python • network connectivity of groups of robots • Study guide
10-11 grade	<ul style="list-style-type: none"> • Platform in base set • Set of sensors • Camera • Software: programming on Python • Study guide 	<ul style="list-style-type: none"> • A group of robots in the base set • Cameras • Kinect • Software: programming on Python • Network connectivity of groups of robots • Study guide
Bachelor Graduate student		<ul style="list-style-type: none"> • The robotic stand with the ability to visualize group control algorithms • web interface for remote work • Ability of testing algorithms of group robotics

This project considers the initial stage of the concept implementation, namely the development of the prototype of the platform and its software.

6. Mobile platform design

The concept assumes availability of a modular platform. In this context, the word "modular" means the easy interchangeability of various blocks of the platform. Depending on goals and purposes of the platform usage, you can use module with sensors, electromagnets, a module with a camera, as well as additional modules, for example, a block with a laser module for games, can be attached.

The modules are separated by means of special fasteners, which are the electrical components of the assembly too (Figure 1)

Figure 1 shows an example of a modular mobile platform model

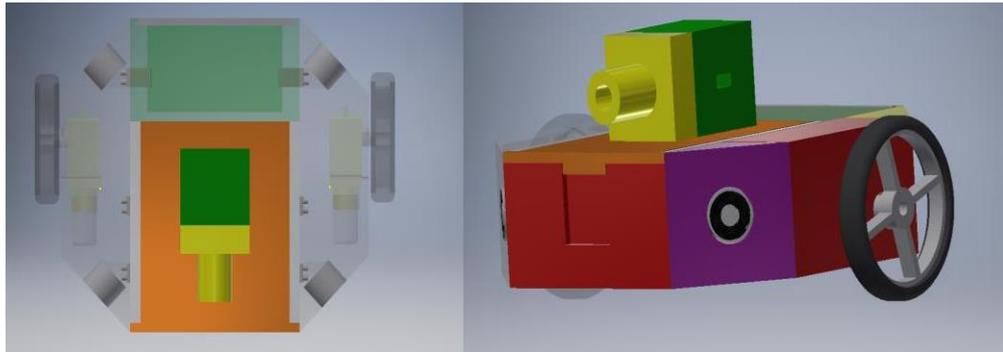


Figure 1 3D model of mobile platform

The blocks printed on the 3d printer are shown in Figure 2

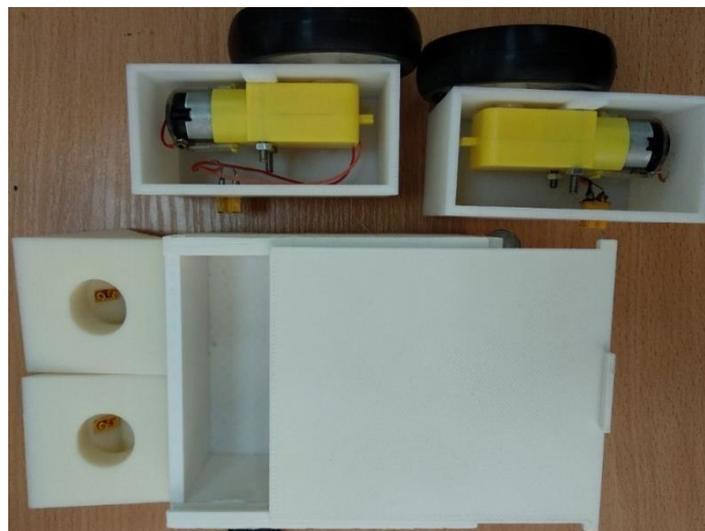


Figure 2 Blocks printed on the 3D printer

The next stage of the design will be the casting of parts in silicone molds.

7. Hardware implementation of the mobile platform

The hardware part of the platform includes two controllers Raspberry Pi Zero W (it has a WI-FI module) and Arduino Nano. Raspberry is the main controller that

transmits control signals to the Arduino via the I2C bus. Arduino sends control actions to other electrical components. A video camera can be also connected to Raspberry. The idea of using two boards is to reduce the load on Raspberry, which is a top-level controller of this scheme. At the moment, the Arduino Nano is used because of its availability. At further stages of work, it is planned to use self-made boards made on the AVR family controllers.

Figure 3 shows the hardware of the platform

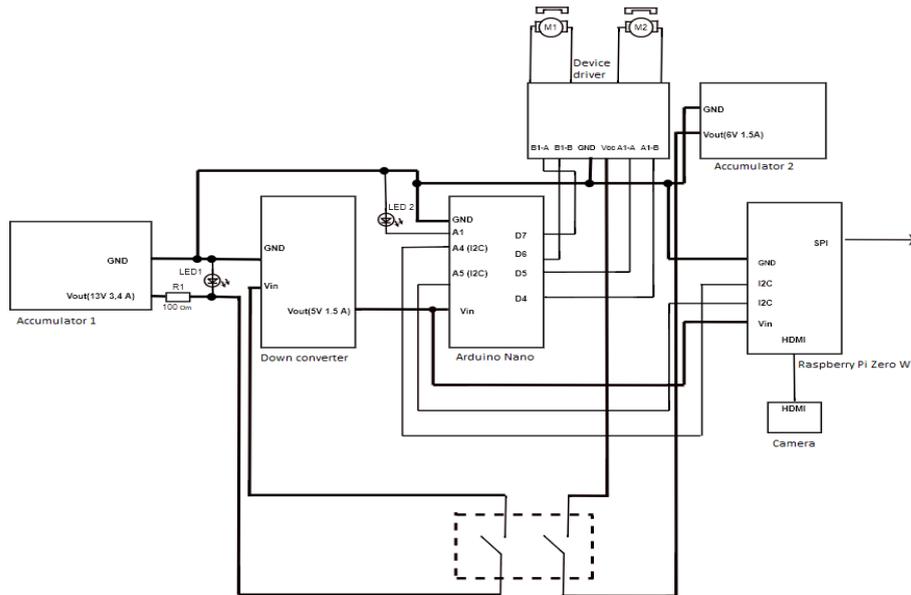


Figure 3 the hardware of the platform

8. Software of the modular mobile platform

The software implementation of the platform is presented in the following figure:

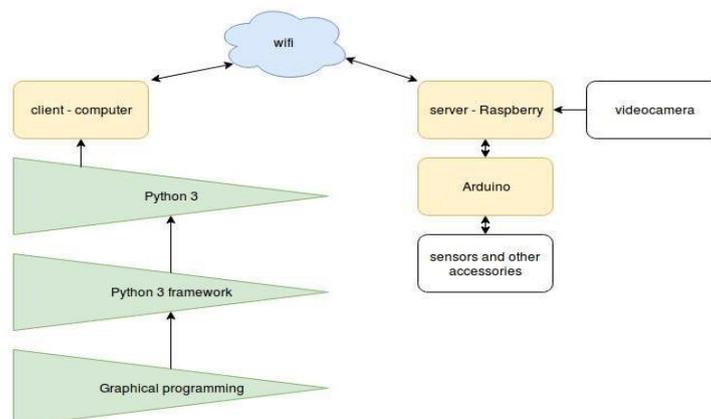


Figure 4 General scheme of software implementation of the platform

Client – computer implementation.

The implementation is built on three levels of complexity, depending on the knowledge of user programming skills

Level Python 3

At this stage, the user can add his code, he also has unlimited possibilities for working with software. This stage is the lowest level of software implementation. This level is built on the basis of web programming.

Python 3 framework

It is a class with a set of functions in Python 3, in which client-server communication is implemented, thus programming in Python 3 becomes very simple and convenient - all functions are similar to the Arduino language, which allows users with an average level of programming, opportunity to program the controller and the necessary electronics for the robot easily.

Graphical programming (Scratches)

This level is built on programming of the platform by using sketches. It is the simplest of the implementation of programs, but its capabilities are quite limited, hereby making this level ideal for schoolchildren in primary grades.

The following figure shows a simplified scheme of the framework

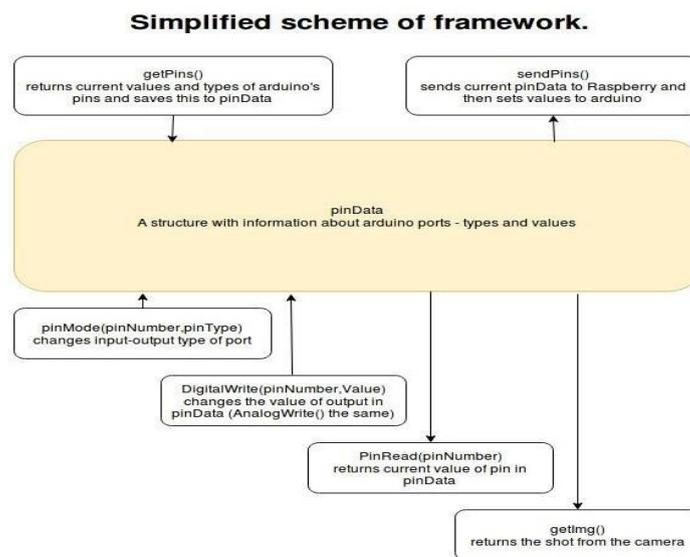


Figure 5 Work of framework

Figure 6 shows the software interface for schoolchildren studding at 5-7 grades. This age group is the most interesting for consideration, since the environment allows to add not only algorithmic blocks, but also blocks with program code. For a more understandable interaction with these blocks, there is a color identification, which makes it easier to work with the environment.

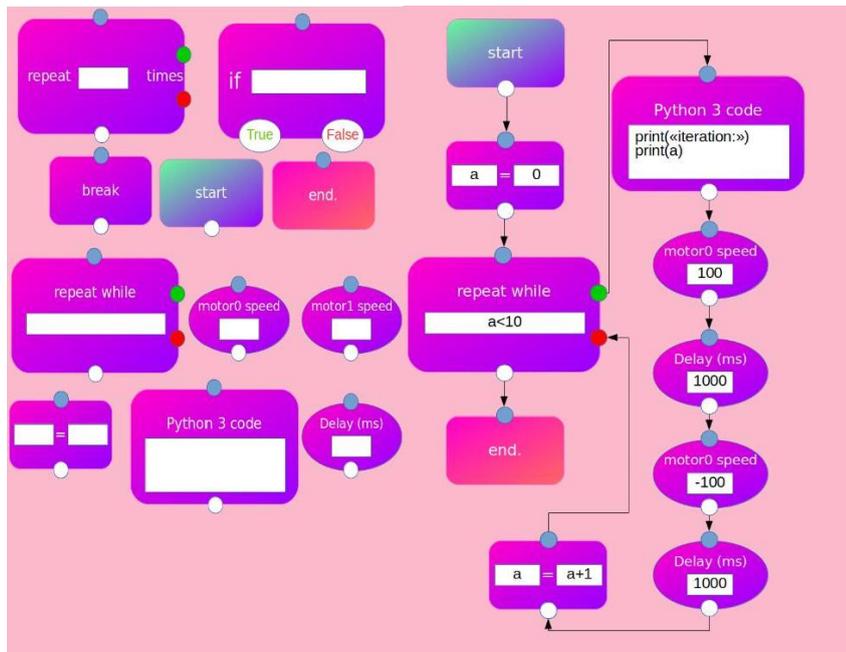


Figure 6. Example of the interface design

«Kinect» is used for the possibility of centralized management of groups of robots. To be able to register each platform as a separate device, active markers of different colors are attached to each robot. Active markers are LEDs of different colors. In future it is planned to develop its own photogrammetric system.

9. The robotic stand

The ultimate goal of the project is to create a remote, test robotic stand that will provide users with a flexible web interface for testing algorithms used in group robotics. The main focus at this development will be on supporting secure remote access to research and testing of algorithms for group robots [4]. Continuous work of the stand will be provided due to its automatic recharging of batteries of mobile robots.

The work process of the stand is planned to be realized according to the following scheme:

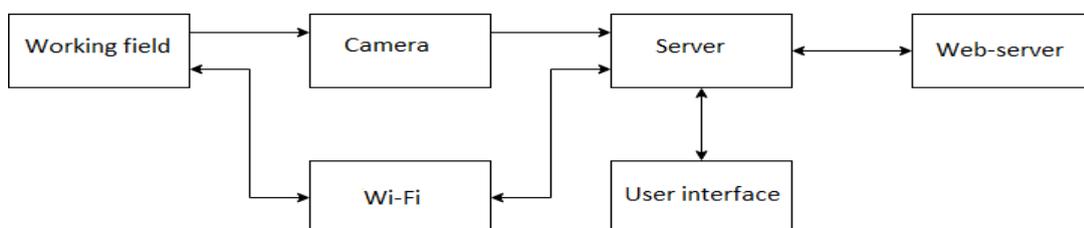


Figure 7 Structural diagram of the stand

10. Conclusion

In this project, we propose an innovative concept in the field of educational robotics. This approach to educational robotics allows students of different age categories to work with the mobile platform, to expand its functionality, depending on the training objectives. The flexibility of the software allows users with different levels of education to get programming skills. Also one of the main advantages of the platform is availability of a camera. Work with a video camera, expands the area of competence of the user, and the software allows users to learn the basics of vision systems. At the moment, the first prototype of a modular mobile platform has been produced, software is being written, teaching methods are prescribed. Monthly the project is modernized, refined and expanded.

11. REFERENCES

- [1] H. Tamada, A. Ogino, and H. Ueda, "Robot Helps Teachers for Education of the C Language Beginners," in *Human-Computer Interaction. Novel Interaction Methods and Techniques*, Springer, Berlin, Heidelberg, 2009, pp. 377–384
- [2] Yen Air Caballero González ; Ana García-Valcárcel Muñoz-Repiso «Educational robotics for the formation of programming skills and computational thinking in childish», *Computers in Education (SIIE)*, 2017 International Symposium on
- [3] Jose Etienne Bezerra Junior ; Paulo Gabriel Gadelha Queiroz ; Rommel Wladimir de Lima, «A study of the publications of educational robotics: A Systematic Review of Literature» , *IEEE Latin America Transactions (Volume: 16, Issue: 4, April 2018)*
- [4] Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt «The Robotarium: A remotely accessible swarm robotics research testbed», 2017 IEEE International Conference on Robotics and Automation (ICRA) Singapore, May 29 - June 3, 2017

ROBSIM SOFTWARE FOR MOBILE ROBOTS MODELING

FSUE VNIIA, Moscow

crer@vniia.ru

Abstract

This paper discusses issues of Unmanned Vehicles' (UV) modeling at various stages of their life-cycle. It presents software system RobSim. RobSim has a capacity to develop models of UVs of high complicity and perform modeling of their functioning. The paper describes structure of RobSim software with basic developers' tools including high-level robotic languages programming and control.

Key words: mobile robot, unmanned vehicle, modeling, software, simulator.

Introduction

Mobile robot is a complex system with sophisticated structure. Functional purpose of any mobile robot defines specific requirements for construction, chassis technical ability, attached equipment, and, simultaneously, defines environment properties and work area in which mobile robot should realize all his functional. In order to verify constructive solutions, which generated in robot development, one needs to create big amount of samples that lead to increase of robots cost.

The alternative approach is using computer modeling and 3d visualization of mobile robot and environment. The computer modeling provides estimation of mobile robot effectiveness in certain environmental conditions, defines solutions correctness, decrease amount of experiments, helps to choose optimal plan of robot technological operations execution for one robot and groups of robots. Thus, important task is applying software that provide modeling of mobile robots functioning in environment at various stages of their life-cycle. We presents software system RobSim. RobSim has a capacity to develop models of UVs of high complicity and perform modeling of their functioning.

Functionality of RobSim

Last version of RobSim provides tools for developing various types of mobile robots. The basic functions of RobSim are:

- mobile robot visual and dynamical models creation;
- models of all robot main parts development, such as actuators, sensors, etc.;
- creating environment model;
- development of mobile robots control system with using standard robots algorithms;
- parameters estimation of all subsystems in modeling process;
- modeling of robots groups;
- modeling process observation from any point of 3d scene;
- detailed logs maintenance with ability to play records of modeling process.

The Robsim structure

The Robsim software package includes a number of subsystems presented below.

1. RobSim program shell is a user interface, which provides selecting scenes and robots, participating in the modeling process, and launching the simulation. After launching, the shell loads several subsystems: control system unit, dynamics unit, visualization unit, and provides a centralized data exchange between these subsystems in the modeling process.

2. The control system unit models logic and control interface for mobile robot. The results of the unit processing are transferred to the dynamic unit through the program shell as signals to the executive system of mobile robot.

3. The dynamic unit calculates "robot-scene" state space parameters using its mathematical model. Simulated dynamic elements include bodies, hinges, approximating contact containers, linear and rotary motors, wheels with different types of suspensions, interchangeable tools, sensors, etc. As a result of the dynamics calculation, position and orientation of the scene elements are generated, which are then transferred to the visualization unit via the program shell.

4. The visualization unit provides high-quality visualization of virtual 3d scenes. The unit supports the visualization of surfaces with complex materials, including textures of reflection, transparency and relief; allows you to simulate realistic lighting with the generation of shadows based on shader technology and effects such as flame, smoke and water.

3D model creation

The main platform for creating 3D models of robots and scenes is Autodesk 3DStudio Max (3DSMax). Developer can use standard components of 3DSMax to create the model. In addition, a plug-in has been developed that has a specific set of tools. In particular, with the help of the plug-in, it became possible to model main components of the robot: chassis suspension, wheels, caterpillars, grippers, manipulators, etc.

The plug-in provides physical parameterization of robots and scenes models (mass-inertial characteristics, type of surfaces, parameters of engines, brakes, etc.). Figure 1 shows mobile robots in the working area and their models in Robsim.

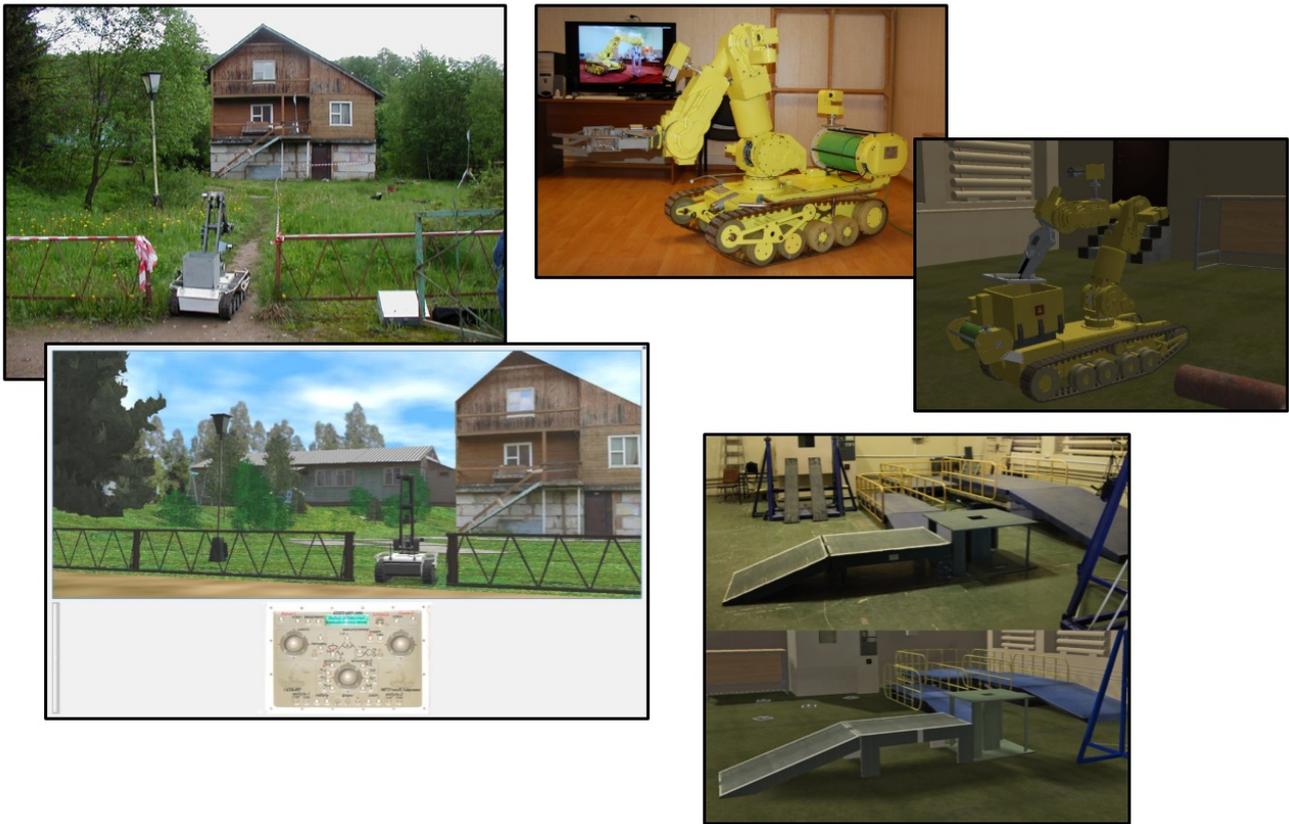


Fig.1. 3D scenes.

The modern robots field of application includes work in extreme environments, in particular: decontamination of radioactive materials, fire extinguishing, etc. In order to take into account such operations in the modeling process, Robsim implemented a number of tools:

- complex dynamic processes modeling, including a large number of complex objects interaction (blockages modeling);
- dismantling blockages modeling;
- smoke, flames, fire extinguishing systems modeling (figure 2).

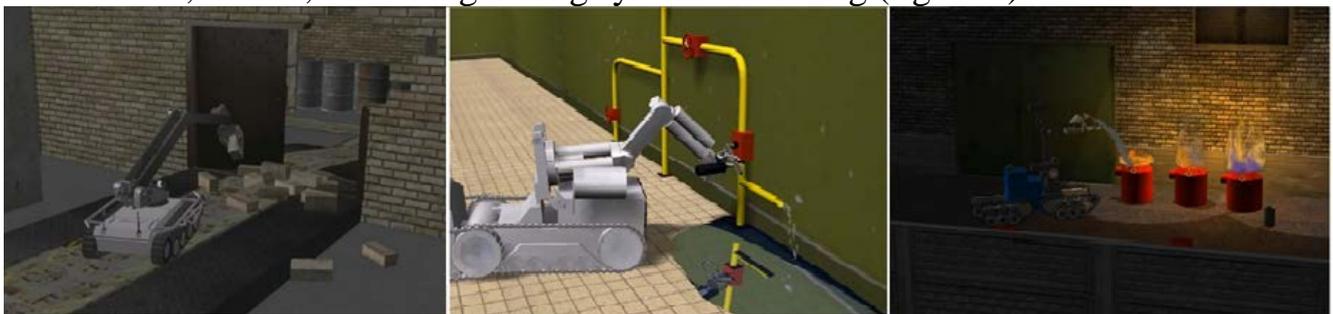


Fig. 2. Effects.

Control system modeling

A special editor CtrlPanel was developed for control system modeling. The Robsim software provides both a virtual control using control panel model and a real control via remote control desk.

The virtual robot control panel consists of two parts: interface and algorithmic. The interface part is a set of virtual controllers: joysticks, buttons, etc. The algorithmic

part embodies the control system structural diagram in form of logical connections between controllers and execution mechanisms.

RobSim realizes possibility of developing user blocks for the control system. These blocks are used in conjunction with standard blocks from the library, and allow the developer to describe the blocks logic in a high-level language. Each block connects as plug-in to the control panel editor.

Training mode

RobSim software can be used both for modeling and as a simulator and for training operators of real-world robots.

Using RobSim as simulator allows operators to:

- master the basic techniques of robot control, including work with attachments;
- improve skills of working with robots in various conditions (overcoming obstacles, working in a limited space, moving various objects, etc.);
- improve interaction between operators of various robots using the network training;
- planning tactics of robots work under certain conditions.

Conclusion

Simulation is an important tool used at different stages of the robot life cycle. The development of universal modeling systems applicable for various types of robots is an important task of robotics.

RobSim gives a powerful tool that allows simulating the functioning of different robot types in different environments when performing various tasks.

In-flight launch of unmanned aerial vehicles

Niels Nauwynck, Haris Balta, Geert De Cubber, and Hichem Sahli

Abstract—This paper considers the development of a system to enable the in-flight-launch of one aerial system by another. The paper will discuss how an optimal release mechanism was developed, taking into account the aerodynamics of one specific mother and child UAV. Furthermore, it will discuss the PID-based control concept that was introduced in order to autonomously stabilize the child UAV after being released from the mothership UAV. Finally, the paper will show how the concept of a mothership UAV + child UAV combination could be usefully taken into advantage in the context of a search and rescue operation.

Index Terms—Unmanned Aerial Vehicles, Control, Autonomous stabilization, Search and Rescue drones, Heterogeneous systems.

1 INTRODUCTION

1.1 Problem statement

As more and more unmanned aerial systems are entering our everyday lives, we also see more and more variety in the systems that are being developed, each towards a different application field. This variety should come as no surprise, as it is impossible to create one system that would fit all user needs. Heterogeneous systems, all being used at the same time are therefore the way forward. However, this also leads to new problems in terms of interoperability and the search for optimal collaboration strategies between all these different systems.

In this paper, we focus on the collaborative action between two unmanned aerial systems where one acts as a mothership / carrier / launch platform, capable of launching in-flight a smaller child system that can then be used for close-to-ground search and rescue missions.

The in-flight-launch of one aerial system by another is no easy problem and requires the careful consideration of the aerodynamics and control of the two systems. Indeed, in terms of aerodynamics and flight performance, the mothership and the child UAV impose important forces and constraints on one another that are very different when they are mechanically interlinked and from when they are separated from one another. The autonomous control concept which is implemented for this research experiment on the child UAV needs to be able to cope with these sudden changes in real-time at the moment of release in order to prevent a crash.

1.2 Previous Work

In the field of collaborative Unmanned Aerial Vehicles (UAVs), Lacroix et al. studied already in 2007 the multi-agent decision making process between the different sys-

tems in [1]. However, taking these concepts to practical applications and the reality on the field has proven to be a difficult operation, due to the complex nature of operating multiple heterogeneous platforms simultaneously. Serrano et al. have proposed in [2] an interoperability concept that enables the message-passing and collaborative control for multiple heterogeneous UAVs and applied that concept on heterogeneous systems developed within the context of the ICARUS project [3]. They put this interoperability and collaboration concept in practice in [4] in a search-and-rescue use case for the euRathlon challenge [5] where multiple heterogeneous systems (though not all airborne) were validated in a Fukushima-like response simulation scenario. While these operations entailed the use of heterogeneous UAV operations, none of the systems featured an in-flight launch capability.

The in-flight launch of one UAV by a mothership is something which has been considered mostly for military operations. Roberts et al. describe in [6] flight tests to determine the flight envelope and launch system configuration for which a small (maximum gross weight of 80 lbs), unpowered UAV glider could be safely launched from the cargo ramp of a C-130 transport aircraft. Safe separation from a C-130 aircraft was demonstrated, as well as UAV stability for successful wings deployment and fly-out. However, these tests considered a manned aircraft as a mothership and only fixed wing aircraft.

1.3 Hardware and software used

The main aim of this research work is to show the concept of the autonomous in-flight launch stabilization system on commodity hardware multi-copters, as opposed to the heavy military systems where in-flight launch systems have already been shown. Therefore, we chose to work with modest, low-cost equipment, as presented here.

The platform used for the parent UAV is a DJI Phantom 2. This ready to fly, multi-functional quad-copter is easy to fly, offers precision flight and has stable hovering without too much interaction. Throughout this research work, this system remained a closed system where the only communication was done through the included controller. The DJI Phantom 2 is a consumer product not specifically equipped

-
- N. Nauwynck and H. Sahli are with the Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussels, Belgium. E-mail: niels.nauwynck@vub.ac.be
 - H. Sahli is with the Interuniversity Microelectronics Center (IMEC), Leuven, Belgium. E-mail: hsahli@etrovub.be
 - H. Balta and G. De Cubber are with the Department of Mechanics of the Royal Military Academy, Brussels, Belgium. E-mail: geert.de.cubber@rma.ac.be



(a) DJI Phantom 2

(b) Parrot AR Drone 2.0

Fig. 1. DJI Phantom 2 and Parrot AR Drone 2.0 UAVs used as mother-UAV and child-UAV in this research work.

to carry any load but did offer the requirements for the proof of concept. By removing the pre-installed camera, the total mass of the parent UAV is 1093 g. Figure 1a displays the Phantom 2 without the camera attached.

The platform used for the child UAV is the Parrot AR Drone 2.0. This UAV is mostly conceived as a toy which makes it quite popular and affordable. This UAV has a starting mass of 501 gram. By sacrificing security and durability we are able to reduce the weight with 58 g. This however meant that no protection hull was present during crashes, bringing the lowest mass to 443 g. Figure 1b displays the Parrot AR Drone 2.0 without the protective hull.

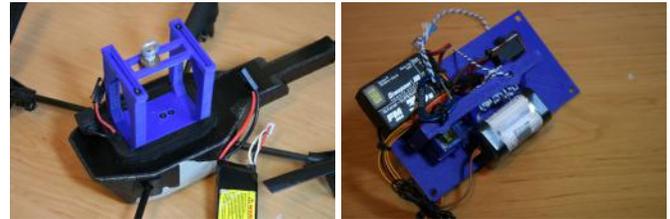
The Parrot AR Drone 2.0 is used frequently in research since it is programmable in a ROS [7] interface, making use of WiFi communication for input and output. A ROS-driver is provided to create a communication channel with the UAV. This communication driver offers a great deal of functionalities that were used for the in-flight launch software, such as:

- 3-dimensional rotation values from the X, Y & Z axis;
- magnetometer readings in three-dimensional space;
- pressure from the barometer;
- linear velocity in three-dimensional space;
- linear acceleration in three-dimensional space;
- estimated altitude;
- motor pulse width modulation values;
- forward and downward facing camera stream;
- movability through yaw, pitch and roll.

2 DESIGN OF THE RELEASE MECHANISM

As the child UAV still has a task to complete after being launched, as much weight as possible should be left on the parent UAV. This meant that a design was made where the actual launch mechanism was hanging on the parent UAV.

A major issue in the design process of developing a release mechanism on the child UAV was to prevent any unwanted rotations due to wind etc, which would cause system instability. Therefore, a child-UAV release mechanism was designed, consisting of a base plate and a locking mechanism, terminating in an O-ring where a hook can be attached. Once the design was fully made it was 3D printed. The design turned out to be 44 g. Adding the 44g to the 443g of the child UAV made sure that the child UAV now had a total mass of 487 g. Note that it is technically not possible for a DJI Phantom 2 to support such a payload, therefore it is required for the child UAV to help with lifting its own mass pre-release by spinning its rotors. Figure 2a shows



(a) Release mechanism on the child UAV

(b) Release mechanism on the mother UAV

Fig. 2. Release mechanism on the child UAV.

the result of this design: a lightweight, stern and rotation resistant component capable of carrying the child UAV.

As discussed above, the child UAV can be carried through an o-ring. This was specifically done to create an easy to use launch mechanism on the parent UAV. The major difficulty on the parent-side was to include a mechanism that can increase or decrease the distance between the parent and child UAV. Indeed, due to turbulence effects under the mother aircraft, it is required to release the child UAV at a reasonable distance from the mother UAV, sufficiently away from the turbulence zone. This so-called "downwash" area can be modeled or experimentally measured [8]. In our case, as we lacked the input of the necessary modelling parameters, an experimental study was required. We therefore needed to experiment with different release altitudes (measured between the mother and child UAV) in order to study these effects. Therefore, a winch system was developed, consisting of a PCB-controlled servo motor. Once 3D printed, the base plate extension creates a functional winch system as seen in Figure 2b. The parent UAV now has the possibility to lower the UAV to any desired launch height from a remote site. The final design of the parent UAV release mechanism has a mass of 245g, bringing the total mass of the parent UAV to 1338g.

3 AUTONOMOUS STABILIZATION

In order to be platform independent a new PID controller is created that takes over the default hovering function embedded in the used devices, taking into account the constant turbulence by the parent UAV. Since we wanted a platform-independent solution, we did not rely for this on the built-in stabilization method that also makes use of the downward facing camera. For the creation of the PID controller, a custom package was created that subscribed to the navigation data and odometry. In return, it could publish to the necessary yaw, pitch and roll values, calculated as control commands to stabilize the UAV.

In the implementation the maximum reference speed of the UAV is limited to 0.6 which prevents it from performing jerky movements. The velocity error is calculated by the difference of the navigation commands of yaw, pitch and roll and the incoming odometry values. This value is assigned to the proportional gain. The integral gain is calculated with the previous integral gain and the proportional gain. By using the proportional gain we are able to determine the integral gain seen on Figure 3, based on a set limit, the current situation of the error (new_err) and the previous

integral gain (i_term). Lastly, the derivate gain is calculated by filtering the incoming odometry data.

```
def FilterVelocity(self, velocity):
    result = 0.0

    self.m_input_buffer[0] = velocity

    for i in range(0, 30):
        result += self.m_input_buffer[i] * self.m_coefs[i]

    for x in range(0, 30):
        self.m_input_buffer[x] = self.m_input_buffer[x - 1];

    return result

def ITermIncrease(self, i_term, new_err, cap):
    result = 0.0
    if new_err < 0 and i_term > 0:
        result = max(0.0, i_term + 2.5 * new_err)
    elif new_err > 0 and i_term < 0:
        result = min(0.0, i_term + 2.5 * new_err)
    else:
        result = i_term + new_err

    if i_term > cap:
        result = cap
    if i_term < -cap:
        result = -cap

    return result
```

Fig. 3. Calculation of the PID controller values

After a successful series of static tests (manually pushing the UAV from its stable position, using the proposed PID-based stabilization method to prevent a crash), in-flight launch tests were performed, with a different separation distance between the both UAVs: 140cm, 100cm and 60cm.

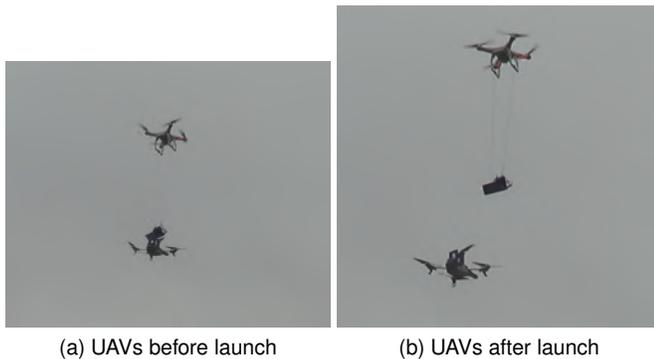


Fig. 4. In-flight launch on the child UAV by the mother UAV.

Using a **140cm launch distance** (<https://youtu.be/hvxIr1gvgtc>), the PID controller does not need to change the yaw, pitch or roll values. Its only task is increasing the power on all four motors to counteracting the descent. This is a fairly easy task and the release goes therefore quite smooth.

Using a **100cm launch distance** (<https://youtu.be/-HsyfGzBpow>), the behaviour is in most cases similar to the previous case (140cm). However, sometimes we observe that the child UAV needs to compensate pre-release already for the extra downward forces induced by the downwash of the mother UAV. The result is that the PID controller acquires the correct height by lifting its own weight, not relying on the strength of the parent UAV. Once the child

UAV is released, it does not need to adjust anymore to any turbulence anymore, just like in the previous experiments and the release goes smooth.

Using a **60cm launch distance** (<https://youtu.be/3Xvp1fMt6tg>), the PID controller is no longer capable of recovering the turbulence induced by the rotors of the parent UAV and the child UAV always crashes upon release. In all of the four runs made, the release was never possible because the parent UAV created turbulence on the child UAV. This turbulence interfered with the spinning propellers of the child UAV which made it move all over the place. Because of the moment of the child UAV, the parent UAV also started to wiggle which only increased the movement on the child UAV, repeating this pattern until a crash occurred. Obviously, this means that here we have reached the limits of what was possible with the given platforms and the proposed control and stabilization paradigm.

4 VALIDATION OF THE CONCEPT IN A SEARCH & RESCUE USE CASE

In order to present a meaningful use case for the validation of the proposed system, the field of search and rescue was chosen. This specific domain was not chosen by accident, as the specific requirements of the search and rescue workers [9] often demand for multiple heterogeneous robotic tools to be deployed. Indeed, large fixed wing systems are required to have a permanent eye in the sky and to create a map of the area, whereas rotorcraft are generally more suited for outdoor victim search or dropping rescue kits, whereas small rotorcraft are excellent for indoor victim search. In this context, we envision a search and rescue operation where a large UAV launches a smaller one at a specific site, such that this small UAV can go and search for victims.

A necessary requirement for using a UAV for victim search is the capability to detect human survivors in a totally unstructured environment. For scene analysis, using the on-board camera, the UAV has to detect and classify the objects seen by the camera. For this purpose a deep neural network is used to achieve semantic segmentation, assigning a class label to every pixel. A deep neural network is another form of an artificial neural network which has shown spectacular accuracy on datasets with large feature and solution space. Since deeper networks often have more vanishing gradient problems and exploding gradient problems, they are harder to train than other networks.

For this application, we will use the on the ENet semantic segmentation algorithm [10], which uses a deep neural network architecture to provide real-time semantic segmentation for self-driving vehicles. By requiring 75 times less FLOPs and 79 less parameters it functions eighteen times faster than existing models by early down-sampling, nonlinear operations, changing the decoder size, regularization and much more.

To train from a dataset a modified version of Caffe [11] was used which supported all the necessary layers for ENet. This requires a training and testing set where first the encoder is trained with pre-labeled objects from the data set [12]. After about 75 000 iterations, we noticed convergence with a minimum of 80% training accuracy. After finishing

training the decoder, the encoder was further trained to obtain also a 80% training accuracy.

After launching the child UAV from the parent UAV, the ENET semantic segmentation algorithm was activated on the images of the Parrot AR Drone 2.0 front-facing camera, which has a resolution of 1280x720 at 30 fps. The first test, shown on Figure 5, shows an example of how the output on a small access road to a building to mimic the idea of a small road in open country side.

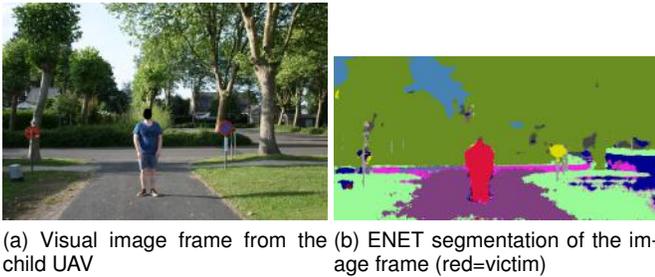


Fig. 5. ENet's semantic segmentation input image of a lost person on small road in open country side.

The second experiment set can be seen on Figure 6 and displays the detection possibilities in front of tunnels and shows that while inside a dark tunnel, person detection becomes less obvious.

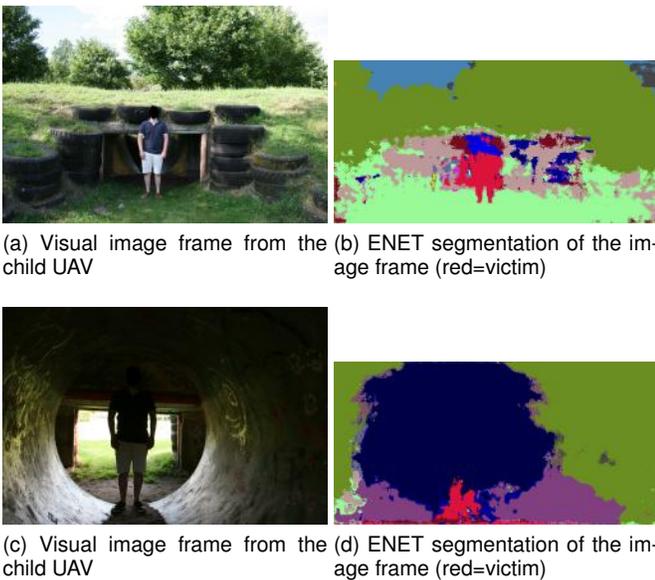


Fig. 6. ENet's semantic segmentation output image of a lost person in front and inside of a tunnel.

5 CONCLUSION

Within this paper, an in-flight launch concept has been proposed for a child rotorcraft UAV by a parent rotorcraft UAV. The solution developed not only in theory, but also in practice, by the design of a release mechanism and a control concept in order to stabilize the child UAV after the launch procedure. The system was extensively validated by multiple launch experiments, evaluating the limits of the control concept. Furthermore, a practical use case was elaborated where this concept could be put into practice:

search and rescue. Therefore, a deep neural network was implemented in order to perform a semantic segmentation of the video data of the child UAV (after being released in a disaster area by the parent UAV), enabling autonomous victim search operations.

It must be stressed that the objective of this research work was to provide a proof of concept, using cheap hardware. Future work will thus mainly focus on porting this concept to more performing hardware platforms, such that real use cases can be performed.

REFERENCES

- [1] S. Lacroix, R. Alami, T. Lemaire, G. Hattenberger, and J. Gancet, "Decision making in multi-UAVs systems: Architecture and algorithms," in *Multiple Unmanned Aerial Vehicles*, ser. Tracts on Advanced Robotics, A. Ollero and I. Maza, Eds. Springer, 2007.
- [2] D. S. López, G. Moreno, J. Cordero, J. Sanchez, S. Govindaraj, M. M. Marques, V. Lobo, S. Fioravanti, A. Grati, K. Rudin *et al.*, "Interoperability in a heterogeneous team of search and rescue robots," in *Search and Rescue Robotics-From Theory to Practice*. In-Tech, 2017.
- [3] G. De Cubber, D. Serrano, K. Berns, K. Chintamani, R. Sabino, S. Ourevitch, D. Doroftei, C. Armbrust, T. Flamma, and Y. Baudoin, "Search and rescue robots developed by the european icarus project," in *7th Int. Workshop on Robotics for Risky Environments*, 2013.
- [4] M. M. Marques, R. Parreira, V. Lobo, A. Martins, A. Matos, N. Cruz, J. M. Almeida, J. C. Alves, E. Silva, J. Bdkowski, K. Majek, M. Peka, P. Musialik, H. Ferreira, A. Dias, B. Ferreira, G. Amaral, A. Figueiredo, R. Almeida, F. Silva, D. Serrano, G. Moreno, G. D. Cubber, H. Balta, and H. Beglerovi, "Use of multi-domain robots in search and rescue operations contributions of the icarus team to the eurathlon 2015 challenge," in *OCEANS 2016 - Shanghai*, April 2016, pp. 1–7.
- [5] A. F. T. Winfield, M. P. Franco, B. Brüggemann, A. Castro, M. C. Limon, G. Ferri, F. Ferreira, X. Liu, Y. R. Petillot, J. Röning, F. E. Schneider, E. Stengler, D. Sosa, and A. Viguria, "eurathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots," in *TAROS*, ser. Lecture Notes in Computer Science, vol. 9716. Springer, 2016, pp. 351–363.
- [6] D. W. Roberts and A. D. Judy, "Separation Flight Tests of a Small Unmanned Air Vehicle from a C-130 Transport Aircraft," *NATO STO Meeting*, vol. MP-SCI-162-19, 2012. [Online]. Available: <https://www.sto.nato.int//publications//STOMeetingProceedings//RTO-MP-SCI-162//MP-SCI-162-19.pdf>
- [7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [8] M. Sjöholm, N. Angelou, P. Hansen, K. H. Hansen, T. Mikkelsen, S. Haga, J. A. Silgjerd, and N. Starsmore, "Two-dimensional rotorcraft downwash flow field measurements by lidar-based wind scanners with agile beam steering," *Journal of Atmospheric and Oceanic Technology*, vol. 31, no. 4, pp. 930–937, 2014. [Online]. Available: <https://doi.org/10.1175/JTECH-D-13-00010.1>
- [9] D. Doroftei, A. Matos, and G. De Cubber, "Designing search and rescue robots towards realistic user requirements," in *Advanced Concepts on Mechanical Engineering (ACME)*, 2014.
- [10] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: <http://arxiv.org/abs/1606.02147>
- [11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *ArXiv e-prints*, Jun. 2014.
- [12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Qualitative and quantitative validation of drone detection systems

Daniela Doroftei, and Geert De Cubber

Abstract—As drones are more and more entering our world, so comes the need to regulate the access to airspace for these systems. A necessary tool in order to do this is a means of detecting these drones. Numerous commercial and non-commercial parties have started the development of such drone detection systems. A big problem with these systems is that the evaluation of the performance of drone detection systems is a difficult operation, which requires the careful consideration of all technical and non-technical aspects of the system under test. Indeed, weather conditions and small variations in the appearance of the targets can have a huge difference on the performance of the systems. In order to provide a fair evaluation and an honest comparison between systems, it is therefore paramount that a stringent validation procedure is followed. Moreover, the validation methodology needs to find a compromise between the often contrasting requirements of end users (who want tests to be performed in operational conditions) and platform developers (who want tests to be performed that are statistically relevant). Therefore, we propose in this paper a qualitative and quantitative validation methodology for drone detection systems. The proposed validation methodology seeks to find this compromise between operationally relevant benchmarking (by providing qualitative benchmarking under varying environmental conditions) and statistically relevant evaluation (by providing quantitative score sheets under strictly described conditions).

Index Terms—Unmanned Aerial Vehicles, Drones, Detection systems, Drone detection, Test and evaluation methods.

◆

1 INTRODUCTION

1.1 Problem statement

CONSUMER drones are more and more becoming commodity items in our modern world. This is a positive evolution, as these tools have many positive use cases and the affordability of the current systems means that all new business opportunities pop up. However, we cannot be blind as well to the negative aspects these novel tools may induce into our society. Indeed, next to the many airspace infringements, where uneducated hobbyists enter potentially dangerous airspace (e.g. near airports, close to manned aviation, ...) inadvertently, we also see an increasing use of drone technology by criminals [1], [2]. In most countries, rules for access to airspace by unmanned aerial vehicles / drones / Remotely Piloted Aircraft Systems (RPAS) have been created. The challenge is now to enforce these rules, as the police services lack the means to automatically detect airspace infringements. Indeed, something like a car traffic speed camera for the air does not really exist yet, but it is dearly needed.

1.2 Previous work on drone detection

Numerous commercial and non-commercial parties have noted this gap in the market and have started the development of drone detection systems.

There are in general two main difficulties related to the detection of drones. First, the cross section / detection baseline for these systems is in general very limited, whatever sensing technology is used. Indeed, drones have a small

RADAR cross section, a small acoustic signature (from a relevant distance), a small visual / infrared signature, they use common radio signal frequencies, etc. Of course, it would be possible to make the detection methodologies extremely sensitive, but this then leads to the second difficulty: how to avoid false positives? Indeed, the signature of many drones is quite close to the one of birds, so it is really difficult to filter out these false positives [3].

Sensing modalities that can be used to solve the drone detection problem are typically RADAR [4], acoustics [5], visual [6], IR [7] (thermal and short-wave), sensing of the radio spectrum [8], LIDAR [9], etc. However, as the problem is so difficult to solve in realistic operating conditions, most of the existing solutions rely on a mix of different sensing methodologies in order to solve the drone detection problem [2] and use a mix of traditional detection and tracking methodologies [10], [11] originating from computer vision to achieve multi-sensor tracking.

1.3 Previous work on quantitative operational validation

The problem with the evaluation of drone detection systems is twofold:

- 1) Drone detection systems most often rely on complex data fusion & processing of sensor data, which means that it is required to carefully control the test conditions in order to single out the limits of the system under test.
- 2) Drone detection systems need to be operational 24/7 and under all weather conditions, meaning that it is required to assess their performance within a wide range of conditions.

Clearly, both of these constraints are somewhat in contradiction with one another and it is not evident to seek a

• *D. Doroftei and G. De Cubber are with the Department of Mechanics of the Royal Military Academy, Brussels, Belgium.
E-mail: daniela.doroftei@rma.ac.be*

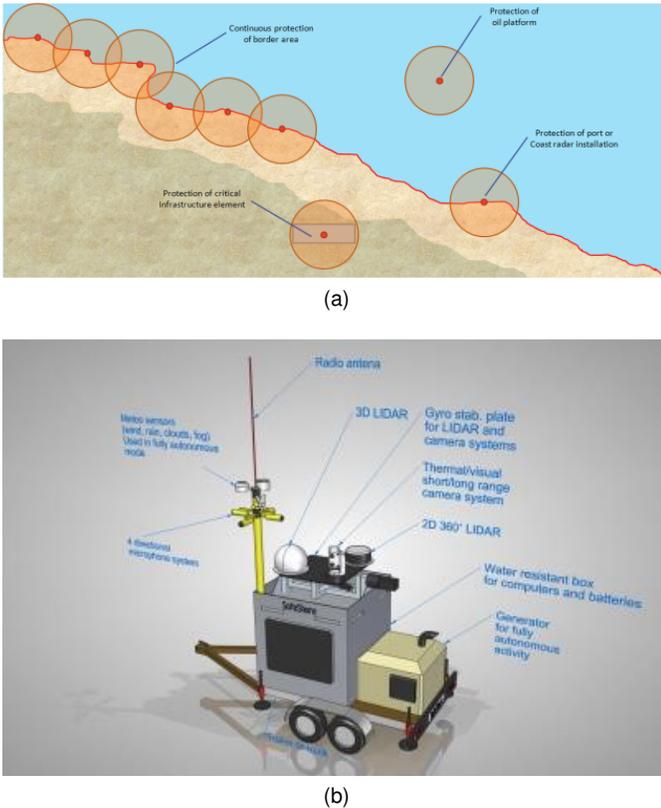


Fig. 1. SafeShore concept sketch.

compromise between these two types of requirements. The objective is therefore is to find a validation methodology that satisfies both the request of the end-users towards a qualitative operational validation of the system and the platform developers of a quantitative statistically relevant validation.

Such qualitative and quantitative validation methodologies have been proposed before, e.g. by the U.S. National Institute of Standards and Technology (NIST) in the field of robotics [12]. In [13], a qualitative and quantitative validation methodology was proposed, based on the work performed at NIST and this technique was validated in [14]. In this research work, we will elaborate on this methodology and port it from the realm of search and rescue robotics to the field of drone detection.

1.4 Introduction to the SafeShore project

The use case that was chosen in the scope of this research work was the validation of a drone detection system, developed within the scope of the EU-H2020-SafeShore project [15].

The main objective of the SafeShore project is to cover existing gaps in coastal border surveillance, increasing internal security by preventing cross-border crime such trafficking in human beings and the smuggling of drugs. It is designed to be integrated with existing systems and create a continuous detection line along the border.

The SafeShore solution for detecting small targets that are flying in low attitude is to use a 3D LIDAR that scans the sky and creates above the protected area a virtual dome

shield. In order to improve the detection, SafeShore integrated the 3D LIDAR with passive acoustic sensors, passive radio detection and video analytics. All those technologies can be considered as low cost and green technologies (compared to the traditional RADAR systems). It is expected that a combination of orthogonal technologies such as LIDAR, passive radio and acoustic and video analytics will become mandatory for future border control systems in environmentally sensitive areas.

The SafeShore objective is to demonstrate the detection capabilities in the missing detection gaps of other existing systems such as coastal radars, thereby demonstrating the capability to detect mini-RPAS along the shore and the sea or departing from civilian boats.

Another important SafeShore goal is to ensure fusion of information and increasing the situational awareness and better implementation of the European Maritime Security Strategy based on the information exchange frameworks while ensuring the privacy of the data and conformity to internationally recognized ethical issues concerning the safety of the information and the equipment subject of the project.

2 PROPOSED METHODOLOGY

2.1 Requirements gathering methodology

A first step in the development of the validation framework was the requirements analysis, which followed a step-wise approach:

- The end user community was approached via market studies and targeted interviews
- An early draft methodology proposal was compiled
- This draft document was extensively discussed with both end users (in this specific case: maritime border management agencies) at relevant events and with platform developers in order to come to target performance levels which are both operationally realistic from and end-user point of view and also realistic from a platform developer point of view in terms of required effort, resources and state-of-the-art and physical constraints.
- As SafeShore focuses on drone detection for the protection of maritime borders, a number of operational validation scenarios were proposed in order to address major issues the maritime border security community is facing today.
- For each of the validation scenarios, target performance levels were proposed in discussion with end users and platform developers.

2.2 Concept overview

Two crucial aspects of obtaining realistic results from validation scenarios are that the scenarios should be as close as possible to operational reality and that the validation tests should be repeated enough to ensure statistical relevance. These two considerations are often in conflict with one another, as operational testing requires uncontrolled environments, whereas statistical relevance of results can only be obtained in controlled settings.

Within SafeShore, we have aimed to strike a balance between both aspects, by providing a qualitative and quantitative assessment of the SafeShore system capabilities and by having multiple repeated experiments in realistic environments, following scenarios which are described by end users, based upon their needs and their practical maritime border security problems of today.

The different components of the SafeShore validation concept are:

- A traceability matrix which indicates clearly what are for each validation scenario the relevant user requirements which are tested, allowing to identify how (by which validation scenario) each system requirement will be validated. This is important in order to keep track of the different user requirements and to make sure that for each of the requirements, there is a validation scenario in place that makes sure that the attainment of the requirement can be verified.
- A number of detailed scenarios, each related to maritime border security and safety. In total, SafeShore considers 14 validation scenarios: 5 to be executed in Belgium, 3 in Israel and 6 in Romania. In this paper, we will focus on those executed in Belgium. Each of these scenarios contains:
 - A capability score sheet, allowing for a qualitative assessment of the validation of the target performance levels. These capability score sheets allow to make a binary assessment (YES / NO) whether one of the user or system requirements has been attained by the system or not.
 - Template forms to be filled in during the validation tests, providing standardised information on the threat agents and the detection results. These template forms contain valuable environmental information, such as weather conditions, sea state, etc. They also provide crucial information on the drones used as test agents: their visual / infrared / radio-frequency / acoustic / LIDAR signature, including ground truth timestamped GPS tracks, which allows for a full quantitative evaluation of the precision of the detection results. These evaluation forms also provide a means to evaluate the human-machine interface, as they gather information on the sample sizes for human verification, the detection resolution and video framerates, etc.
 - A score sheet for the different metrics (Key Performance Indicators or KPI's), allowing for a quantitative assessment of the validation of the target performance levels.
 - Detailed target performance levels for each of the measured metrics. For each of the KPI's, 3 different levels of scoring were assessed in collaboration with the end users:
 - * Minimum Acceptance level: Performance below this level is not acceptable by the end users in operational conditions. Anything above is considered workable.
 - * Goal level: This is the performance level hoped for by the end-users.
 - * Breakthrough level: This is a performance level

beyond initial expectations that end users would like one day to have.

3 VALIDATION OF THE METHODOLOGY

3.1 Trial concept & execution

As discussed above, five different trial scenarios related to maritime border security and safety were validated during the SafeShore trial in Belgium, which was the first in a series of 3 trial events of the project where this validation methodology was applied.

For this operational field test, 11 different drone platforms (rotary wing, fixed wing, systems made of different materials, very fast drones and slow ones, etc) were deployed during a 2-week measurement campaign, in order to grasp different kinds of system capabilities and meteorological and operational conditions.

Figure 2 shows the SafeShore prototype as it was installed on the beach in Belgium for a period of 2 weeks, while detecting numerous types of drones.

3.2 Trial results

As this was the first out of a series of 3 successive test campaigns, it was to be expected that the system was going to have some quirks and child diseases. The performance validation methodology was therefore essential in order to identify these issues and to give indications on the causes for these problems.

Thanks to the proposed validation, at the end of each validation day it was possible to provide an overview of the performance of the system, both from a qualitative as from a quantitative point of view. As a result of this, daily debriefings between SafeShore developers and SafeShore end users could be held in order to discuss the possibilities and deficiencies of the system. As such, an action plan could be set up on a daily basis in order to improve the performance of the system. Due to this iterative review of the system, the performance of the SafeShore system improved on a daily basis.

At the end of the trial, the proposed validation methodology enabled to provide a full overview of the performance of the system for all 5 scenarios, both from a qualitative as from a quantitative point of view. However, as this was the very first trial, it was not possible to sort out all problems with the system by the end of the trial period. Based upon the result of the validation method, a new action plan was therefore elaborated between end-users and developers in order to improve the performance of the system during the next trials in Israel and Romania.

4 CONCLUSIONS

In this paper, a validation methodology was proposed for evaluating complex systems that aims to strike a balance between the rigorous, scientifically correct and statistically relevant evaluation methodologies requested by platform developers in the iterative design stage on one hand and the requirements of the end users on the other hand, who require field tests in operational conditions in order to evaluate the real-life performance of the system. The proposed methodology reaches this objective by incorporating



Fig. 2. SafeShore system as installed on the beach in Belgium. ©Daniel Orban.

and integrating qualitative and quantitative aspects in the validation process. The proposed methodology was tested on a drone detection system in the context of the EU-H2020 SafeShore project and allowed the project participants (a heterogeneous mix of end users and platform developers) to improve the performance of the system on a daily basis during operational field tests of the system, thereby proving the value of the proposed methodology.

ACKNOWLEDGEMENTS

This work was supported by the European Unions Horizon 2020 research and innovation programme under grant agreement N700643 (SafeShore).

REFERENCES

- [1] U. Franke, "Drone proliferation: A cause for concern?" *ISN Articles*, November 2014.
- [2] M. Buric and G. De Cubber, "Counter remotely piloted aircraft systems," *MTA Review*, vol. 27, no. 1, 2017.
- [3] A. Coluccia, M. Ghenescu, T. Piatrik, G. De Cubber, A. Schumann, L. Sommer, J. Klatte, T. Schuchert, J. Beyerer, M. Farhadi *et al.*, "Drone-vs-bird detection challenge at 2017 IEEE AVSS," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–6.
- [4] C. J. Li and H. Ling, "An investigation on the radar signatures of small consumer drones," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 649–652, 2017.
- [5] J. Mezei and A. Molnr, "Drone sound detection by correlation," in *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, May 2016, pp. 509–518.
- [6] A. Rozantsev, "Vision-based detection of aircrafts and uavs," Master's thesis, EPFL, Lausanne, 2017.
- [7] P. Andrai, T. Raddi, M. Mutra, and J. Ivoeivi, "Night-time detection of uavs using thermal infrared camera," *Transportation Research Procedia*, vol. 28, pp. 183 – 190, 2017, iNAIR 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2352146517311043>
- [8] Y. L. Sit, B. Nuss, S. Basak, M. Orzol, W. Wiesbeck, and T. Zwick, "Real-time 2d+velocity localization measurement of a simultaneous-transmit ofdm mimo radar using software defined radios," in *2016 European Radar Conference (EuRAD)*, Oct 2016, pp. 21–24.
- [9] M. U. de Haag, C. G. Bartone, and M. S. Braasch, "Flight-test evaluation of small form-factor lidar and radar sensors for suavs detect-and-avoid applications," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sept 2016, pp. 1–11.
- [10] G. De Cubber, S. A. Berrabah, and H. Sahli, "Color-based visual servoing under varying illumination conditions," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 225–249, 2004.
- [11] V. Enescu, G. De Cubber, K. Cauwerts, H. Sahli, E. Demeester, D. Vanhooydonck, and M. Nuttin, "Active stereo vision-based mobile robot navigation for person tracking," *Integrated Computer-Aided Engineering*, vol. 13, no. 3, pp. 203–222, 2006.
- [12] A. Jacoff, *Guide for Evaluating, Purchasing, and Training with Response Robots Using DHS-NIST-ASTM International Standard Test Methods*, standard test methods for response robots ed. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2014.
- [13] D. Doroftei, A. Matos, E. Silva, V. Lobo, R. Wagemans, and G. De Cubber, "Operational validation of robots for risky environments," in *8th IARP Workshop on Robotics for Risky Environments*, 2015.
- [14] G. De Cubber, D. Doroftei, H. Balta, A. Matos, E. Silva, D. Serrano, S. Govindaraj, R. Roda, V. Lobo, M. Marques *et al.*, "Operational validation of search and rescue robots," in *Search and Rescue Robotics-From Theory to Practice*. InTech, 2017.
- [15] G. De Cubber, R. Shalom, A. Coluccia, O. Borcan, R. Chamrád, T. Radulescu, E. Izquierdo, and Z. Gagov, "The safeshore system for the detection of threat agents in a maritime border environment," in *IARP Workshop on Risky Interventions and Environmental Surveillance*. IARP, 2017.