IARIA

ICAS 2016

The Twelfth International Conference on Autonomic and Autonomous Systems

ISBN: 978-1-61208-483-1

June 26 - 30, 2016

Lisbon, Portugal

ICAS 2016 Editors

Mark J. Balas, Embry-Riddle Aeronautical University in Daytona Beach, USA

Mario Freire, University of Beira Interior, Portugal

ICAS 2016

Foreword

The Twelfth International Conference on Autonomic and Autonomous Systems (ICAS 2016), held between June 26 - 30, 2016 - Lisbon, Portugal, was a multi-track event covering related topics on theory and practice on systems automation, autonomous systems and autonomic computing.

The main tracks referred to the general concepts of systems automation, and methodologies and techniques for designing, implementing and deploying autonomous systems. The next tracks developed around design and deployment of context-aware networks, services and applications, and the design and management of self-behavioral networks and services. We also considered monitoring, control, and management of autonomous self-aware and context-aware systems and topics dedicated to specific autonomous entities, namely, satellite systems, nomadic code systems, mobile networks, and robots. It has been recognized that modeling (in all forms this activity is known) is the fundamental for autonomous subsystems, as both managed and management entities must communicate and understand each other. Small-scale and large-scale virtualization and model-driven architecture, as well as management challenges in such architectures are considered. Autonomic features and autonomy requires a fundamental theory behind and solid control mechanisms. These topics gave credit to specific advanced practical and theoretical aspects that allow subsystem to expose complex behavior. We aimed to expose specific advancements on theory and tool in supporting advanced autonomous systems. Domain case studies (policy, mobility, survivability, privacy, etc.) and specific technology (wireless, wireline, optical, e-commerce, banking, etc.) case studies were targeted. A special track on mobile environments was indented to cover examples and aspects from mobile systems, networks, codes, and robotics.

Pervasive services and mobile computing are emerging as the next computing paradigm in which infrastructure and services are seamlessly available anywhere, anytime, and in any format. This move to a mobile and pervasive environment raises new opportunities and demands on the underlying systems. In particular, they need to be adaptive, self-adaptive, and context-aware.

Adaptive and self-management context-aware systems are difficult to create, they must be able to understand context information and dynamically change their behavior at runtime according to the context. Context information can include the user location, his preferences, his activities, the environmental conditions and the availability of computing and communication resources. Dynamic reconfiguration of the context-aware systems can generate inconsistencies as well as integrity problems, and combinatorial explosion of possible variants of these systems with a high degree of variability can introduce great complexity.

Traditionally, user interface design is a knowledge-intensive task complying with specific domains, yet being user friendly. Besides operational requirements, design recommendations refer to standards of the application domain or corporate guidelines.

Commonly, there is a set of general user interface guidelines; the challenge is due to a need for cross-team expertise. Required knowledge differs from one application domain to another, and the core knowledge is subject to constant changes and to individual perception and skills.

Passive approaches allow designers to initiate the search for information in a knowledge-database to make accessible the design information for designers during the design process. Active approaches, e.g., constraints and critics, have been also developed and tested. These mechanisms deliver information (critics) or restrict the design space (constraints) actively, according to the rules and guidelines. Active and passive approaches are usually combined to capture a useful user interface design.

We take here the opportunity to warmly thank all the members of the ICAS 2016 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICAS 2016. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICAS 2016 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICAS 2016 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the fields of autonomic and autonomous systems.

We are convinced that the participants found the event useful and communications very open. We also hope that Lisbon provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ICAS 2016 Chairs:

Michael Bauer, The University of Western Ontario - London, Canada
Radu Calinescu, University of York, UK
Michael Grottke, University of Erlangen-Nuremberg, Germany
Bruno Dillenseger, Orange Labs, France
Mark Balas, Embry-Riddle Aeronautical University, USA
Alex Galis, University College London, UK
Antonio Liotta, Eindhoven University of Technology, The Netherlands
Jacques Malenfant, Université Pierre et Marie Curie, France
Mark Perry, University of New England in Armidale, Australia
Wendy Powley, Queen's University - Kingston, Canada
Nikola Serbedzija, Fraunhofer FOKUS, Germany

ICAS 2016

Committee

ICAS Advisory Committee

Michael Bauer, The University of Western Ontario - London, Canada
Radu Calinescu, University of York, UK
Michael Grottke, University of Erlangen-Nuremberg, Germany
Bruno Dillenseger, Orange Labs, France
Mark Balas, Embry-Riddle Aeronautical University, USA
Alex Galis, University College London, UK
Antonio Liotta, Eindhoven University of Technology, The Netherlands
Jacques Malenfant, Université Pierre et Marie Curie, France
Mark Perry, University of New England in Armidale, Australia
Wendy Powley, Queen's University - Kingston, Canada
Nikola Serbedzija, Fraunhofer FOKUS, Germany

ICAS 2016 Technical Program Committee

Jemal H. Abawajy, Deakin University, Australia
António Abelha, Universidade do Minho - Braga, Portugal
Nouara Achour, USTHB University, Algeria
Carl Adams, University of Portsmouth, UK
Jose Aguilar, Universidad de Los Andes, Venezuela
Adel Al-Jumaily, University of Technology, Sydney, Australia
Javier Alonso, Duke University, USA
Alba Amato, Second University of Naples, Italy
Cesar Analide, Universidade do Minho, Portugal
Razvan Andonie, Central Washington University - Ellensburg, USA
Richard Anthony, University of Greenwich, UK
Eva Ibarrola Armendariz, Escuela Técnica Superior de Ingeniería de Bilbao, Spain
Markus Bader, Vienna University of Technology, Austria
Senén Barro, University of Santiago de Compostela, Spain
Faiz Ben Amar, Institut des Systèmes Intelligents et de Robotique - Université Pierre et Marie Curie, France
Ismailcem Budak Arpinar, University of Georgia - Athens, USA
Tsz-Chiu Au, Ulsan National Institute of Science and Technology (UNIST), Korea
Mark Balas, Embry-Riddle Aeronautical University, USA
Michael Bauer, The University of Western Ontario -London, Canada
Matthias Becker, University Hannover, Germany
Janusz Bedkowski, Institute of Mathematical Machines / Warsaw University of Technology, Poland
Julita Bermejo-Alonso, Universidad Politécnica de Madrid, Spain
Karsten Berns, University of Kaiserslautern, Germany
Philippe Besnard, IRIT - CNRS /Universite Paul Sabatier - Toulouse, France
Ateet Bhalla, Independent Consultant, India
Karsten Böhm, Fachhochschule Kufstein, Austria

Sheryl Brahnam, Missouri State University, USA
Stainam Brandao, COPPE/Federal University of Rio de Janeiro, Brazil
Ruth Breu, University of Innsbruck, Austria
Daniela Briola, University of Genova, Italy
David W Bustard, University of Ulster, UK
Giacomo Cabri, Universita' di Modena e Reggio Emilia, Italy
Radu Calinescu, University of York, UK
Paolo Campegiani, University of Roma Tor Vergata, Italy
Valerie Camps, IRIT - Universite Paul Sabatier, France
Giuseppe Carbone, University of Cassino and South Latium, Italy
Dominic Carr, National College of Ireland (NCI), Ireland
Sara Casolari, Università di Modena e Reggio Emilia, Italy
Fernando Cerdan, Universidad Politecnica de Cartagena, Spain
Michal Certicky, Czech Technical University in Prague, Czech Republic
Lei Chen, Sam Houston State University, USA
Mu-Song Chen, Da-Yeh University, Taiwan
Albert M. K. Cheng, University of Houston, USA
Ryszard S. Choras, University of Technology & Life Sciences, Poland
Feng Chu, University of Evry Val d'Essonne, France
Paolo Ciancarini, University of Bologna, Italy
Stéphanie Combettes, Université Paul Sabatier, IRIT, France
Peng Dai, Google Inc., USA
Prithviraj (Raj) Dasgupta, University of Nebraska at Omaha, USA
Alessandra De Benedictis, University of Naples Federico II, Italy
Carlos Roberto de Oliveira Junior, Instituto Federal do Rio de Janeiro, Brazil
Noel de Palma, University Joseph Fourier, France
Marina De Vos, University of Bath, UK
Angel P. del Pobil, Universitat Jaume I, Spain
Mingcong Deng, Tokyo University of Agriculture and Technology, Japan
Sotirios Ch. Diamantas, Athens Information Technology, Greece
M. Bernardine Dias, Carnegie Mellon University, USA
Tadashi Dohi, Hiroshima University, Japan
Changyu Dong, University of Strathclyde, UK
Wenjie Dong, University of Texas - Rio Grande Valley, USA
Richard J. Duro, Universidade da Coruña, Spain
Larbi Esmahi, Athabasca University, Canada
Thaddeus Eze, University of Chester, UK
Christophe Farges, Bordeaux University, France
Pedro Felipe do Prado, University of São Paulo, Brazil
Ziny Flikop, Scientist, USA
Naoki Fukuta, Shizuoka University, Japan
Alex Galis, University College London, UK
Rodrigo Garcia Carmona, Universidad Politecnica de Madrid (DIT-UPM), Spain
Fabio Gasparetti, Roma Tre University, Italy
Benoit Gaudou, University Toulouse 1 Capitole, France
Ryszard Gessing, Institute of Automatic Control - Silesian University of Technology, Poland
Andrzej M. Goscinski, Deakin University - Geelong, Australia
Dominic Greenwood, Whitestein, Switzerland

Teodor Lucian Grigorie, University of Craiova, Romania
William Grosky, University of Michigan - Dearborn, USA
Michael Grottke, University of Erlangen-Nuremberg, Germany
Jordi Guitart, Universitat Politècnica de Catalunya - Barcelona Tech (UPC), Spain
Maki K. Habib, The American University in Cairo, Egypt
Mustapha Hamerlain, CDTA, Algeria
Takako Hashimoto, Chiba University of Commerce, Japan
Gerold Hoelzl, University of Passau, Germany
Wladyslaw Homenda, Warsaw University of Technology, Poland
Wei-Chiang Hong, Oriental Institute of Technology, Taiwan
Marc-Philippe Huget, Polytech Annecy-Chambery-LISTIC | University of Savoie, France
Tim Hussein, University of Duisburg-Essen, Germany
Yoshiro Imai, Kagawa University, Japan
Luis Iribarne, University of Almeria, Spain
Michael Jenkin, York University, Canada
Yiming Ji, University of South Carolina - Beaufort, USA
Richard Jiang, Northumbria University, UK
Yanguo Jing, London Metropolitan University, UK
María José Ibáñez, RIAM I+L Lab (GNOSS), Spain
Imed Kacem, Université de Lorraine (UdL), France
Kashif Kifayat, Liverpool John Moores University, UK
Won-jong Kim, Texas A&M University, USA
Fernando Koch, SAMSUNG Research Institute, Brazil
Ah-Lian Kor, Leeds Beckett University, UK
Boris Kovalerchuk, Central Washington University - Ellensburg, USA
Satoshi Kurihara, University of Osaka, Japan
Helge Langseth, NTNU, Norway
Jian-Wei Li, Chaoyang University of Technology, Taiwan
Jingpeng Li, The University of Nottingham - Ningbo, China
Fidel Liberal Malaina, University of the Basque Country, Spain
Antonio Liotta, Eindhoven University of Technology, The Netherlands
Hai-Bin Liu, China Aerospace Engineering Consultation Center, China
Angela Locoro, University of Genova, Italy
Noel Lopes, Polytechnic of Guarda, Portugal
Daniela López De Luise, CI2S Labs, Argentina
Hanan Lutfiyya, The University of Western Ontario - London, Canada
José Machado, University of Minho, Portugal
Isabel Machado Alexandre, Iscte-IUL - Instituto Universitário de Lisboa & Instituto de Telecomunicações, Portugal
Prabhat Mahanti, University of New Brunswick, Canada
Sayyed Majid Esmailifar, Sharif University of Technology -Tehran, Iran
Jacques Malenfant, Université Pierre et Marie Curie, France
Zhihong Man, Swinburne University of Technology, Australia
Elisa Marengo, Università degli Studi di Torino, Italy
Mauricio Marin, Universidad de Santiago and Yahoo! Labs Santiago, Chile
Konrad Andrzej Markowski, Warsaw University of Technology, Poland
Goreti Marreiros, Polytechnic of Porto, Portugal
Rajat Mehrotra, Desert Research Institute, USA

Márcio Mendonça, Universidade Tecnológica Federal do Paraná, Brazil
Yasser F. O. Mohammad, Assiut University, Egypt / Kyoto University, Japan
Thierry Monteil, LAAS-CNRS, INSA de Toulouse, Toulouse, France
José Moreira, University of Aveiro, Portugal
Masayuki Murata, Osaka University, Japan
Adnan Abou Nabout, University of Wuppertal, Germany
Nicol Naidoo, University of KwaZulu-Natal in Durban, South Africa
Roberto Nardone, University of Naples Federico II, Italy
José Neves, Universidade do Minho - Braga, Portugal
Andreas Oberweis, Karlsruhe Institute of Technology (KIT), Germany
Jonice Oliveira, Federal University of Rio de Janeiro, Brazil
Rafael Oliveira Vasconcelos, Pontifical Catholic University of Rio de Janeiro (PUC-Rio) / University Tiradentes (UNIT), Brazil
Michael O'Mahony, University College Dublin, Ireland
Chiemela Onunka, University of KwaZulu-Natal, South Africa
Jose Oscar Fajardo, University of the Basque Country, Spain
David Ostrowski, Ford Motor Company / University of Michigan - Dearborn, USA
Maurice Pagnucco, University of New South Wales, Australia
Umberto Panniello, Politecnico di Bari, Italy
Nandan Parameswaran, University of New South Wales - Sydney, Australia
Pushparaj Mani Pathak, Indian Institute of Technology, Roorkee, India
Luis Paulo Reis, University of Minho, Portugal
Loris Penserini, Informatica e Società Digitale - IES, Italy
Mark Perry, University of New England in Armidale, Australia
Maria Silvia Pini, University of Padova, Italy
Agostino Poggi, Università degli Studi di Parma, Italy
S. G. Ponnambalam, Monash University Sunway Campus, Malaysia
Wendy Powley, Queen's University - Kingston, Canada
Rosario Pugliese, Universita' di Firenze, Italy
Mariachiara Puviani, DIEF - University of Modena and Reggio Emilia, Italy
Francesco Quaglia, Sapienza Università di Roma, Italy
Ahmad B. Rad, Simon Fraser University, Canada
José Ragot, Université de Lorraine, France
Kanagasabai Rajaraman, Institute for Infocomm Research, Singapore
Alejandro Ramirez-Serrano, University of Calgary - Alberta, Canada
Martin Randles, Liverpool John Moores University, UK
Marek Reformat, University of Alberta, Canada
Wolfgang Reif, Institute for Software & Systems Engineering - University of Augsburg, Germany
Douglas Rodrigues, University of Sao Paulo, Brazil
Paolo Romano, INESC-ID Lisbon, Portugal
Juha Röning, University of Oulu, Finland
Rosaldo Rossetti, University of Porto, Portugal
Lakhdar Sais, Université Lille Nord de France, France
Ricardo Sanz, Universidad Politecnica de Madrid, Spain
Jagannathan Sarangapani, Missouri University of Science and Technology, USA
Munehiko Sasajima, Osaka University, Japan
Jurek Z. Sasiadek, Carleton University, Canada
Mariano Saura, Polytechnic University of Cartagena, Spain

Christoph Schommer, University Luxemburg, Luxemburg
Paulo Jorge Sequeira Gonçalves, Polytechnic Institute of Castelo Branco, Portugal
Nikola Serbedzija, Fraunhofer FOKUS, Germany
Madhavan Shanmugavel, Monash University, Malaysia
Mohamed Shehab, University of North Carolina at Charlotte, USA
Maxim Shevertalov, Drexel University, USA
Fábio Silva, University of Minho, Portugal
Arnab Sinha, INRIA, France
David Šišlák, Czech Technical University in Prague, Czech Republic
Petr Skobelev, Samara State Aerospace University / Smart Solutions, Russia
Flavio Soares Correa da Silva, University of Sao Paulo, Brazil
Nisheeth Srivastava, University of California, USA
Claudius Stern, Opsolution GmbH, Germany
Vinitha Hannah Subburaj, Baldwin Wallace University, Berea - Ohio, USA
Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland
Yuqing Tang, Carnegie Mellon University, USA
Patricia Tedesco, Federal University of Pernambuco, Brazil
J. A. Tenreiro Machado, Institute of Engineering (ISEP) - Polytechnic of Porto, Portugal
Daniel Thalmann, EPFL, Switzerland
Dan Tiba, Institute of Mathematics - Romanian Academy, Romania
Ingo J. Timm, Universität Trier, Germany
Manachai Toahchoodee, The University of the Thai Chamber of Commerce, Bangkok, Thailand
Irina Topalova, Technical University of Sofia, Bulgaria
José Manuel Torres, University Fernando Pessoa - Porto, Portugal
Viviane Torres da Silva, Fluminense Federal University, Brazil
Davide Tosi, Università dell'Insubria – Como, Italy
Raquel Trillo Lado, University of Zaragoza, Spain
Nikos Tsagarakis, Istituto Italiano di Tecnologia (IIT-Genova), Italy
Ali Emre Turgut, Middle East Technical University, Ankara, Turkey
Bartlomiej Ufnalski, Warsaw University of Technology, Poland
Egon L. van den Broek, University of Twente, Enschede, Karakter University Center, Radboud University Medical Center Nijmegen, Nijmegen, The Netherlands
Stavros Vassos, Sapienza University of Rome, Italy
Pierluigi Vellucci, Sapienza Università di Roma, Italy
Ramon Vilanova i Arbos, Escola d'Enginyeria - UAB, Spain
Phan Cong Vinh, NTT University, Vietnam
Ljubo Vlacic, Griffith University, Australia
Vladimir Vlassov, KTH Royal Institute of Technology, Sweden
Sebastian von Mammen, Universität Augsburg - Institut für Angewandte Informatik, Germany
Stefanos Vrochidis, Centre for Research and Technology Hellas - Thermi-Thessaloniki, Greece
Thomas Walsh, Massachusetts Institute of Technology, USA
Yufeng Wang, Nanjing University of Posts and Telecommunications, China
Bozena Wozna-Szczesniak, Jan Dlugosz University, Institute of Mathematics and Computer Science – Czestochowa, Poland
Nanjian Wu, Chinese Academy of Sciences, China
Reuven Yagel, Ben-Gurion University, Israel
Linda Yang, University of Portsmouth, UK
Xin-She Yang, Middlesex University London, UK

Copyright Information

Table of Contents

Semi-Autonoumous Modular Robot for Maintenance and Inspection

Yuji Sato
Dept. of Electrical and Electronics Engineering
Hosei University
Tokyo, Japan
e-mail: yuji.sato.5b@stu.hosei.ac.jp

Kazuyuki Ito
Dept. of Electrical and Electronics Engineering
Hosei University
Tokyo, Japan
e-mail: ito@hosei.ac.jp

***Abstract*—In this research, we developed a modular robot by improving our previous rescue robot. The previous robot has many serially connected crawlers to realize high mobility. However, the number of the crawler units was fixed and the operator could not customize the robot for the given task. In this research, we modularized our previous robot to solve this problem. We conducted experiments and demonstrated that the proposed robot can be applied to various search tasks by changing its formation.**

Keywords-snake-like robot; rescue robot; module; passive mechanism.

I. INTRODUCTION

Maintenance and inspection of buildings are among the important tasks for robots, and these robot can be diverted to search and rescue missions when disaster occurs [1]. The serially connected robot is a possible candidate for such robots [2-5]. It can overcome dents, bumps, steps, and rubble by utilizing its many crawlers. In addition, it can enter small spaces because its shape is long and thin.

However, these types of conventional robots have problems in operation [1-3]. Usually, these robots have many actuators and complex autonomous control or manual operation is required to operate in real complex environments such as rubble. In our conventional works [4, 5], to solve this problem, we proposed a serially connected crawler robot that has passive joints. Because the passive joints adapt to rubble without operation, the proposed robot can overcome rubble easily without complex control.

Unfortunately, the number of links of this previous robot was fixed, as was the search function of the robot. Thus, the operator could not customize the robot for the given task and given environment.

In this research, to solve this problem, we developed a modular robot by improving our previous serially connected crawler robot. Experiments were conducted and we demonstrated that the proposed robot can be applied to various search tasks by changing its formation.

II. PROPOSED ROBOT

We modularized each link of the previous robot. Figure 1 shows the basic structure of a module.

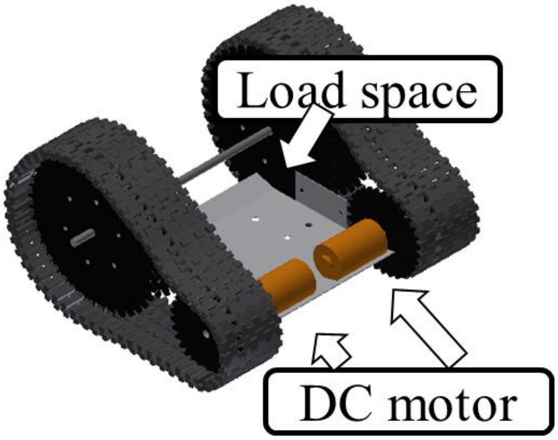


Figure 1. Basic structure.

As shown in figure 1. The basic module has a crawler on each of its two sides, and each crawler is rotated independently by its own DC motor. The basic module has load space, and we extend the function of the module by installing various devices as shown in Figure 2 and Table 1.

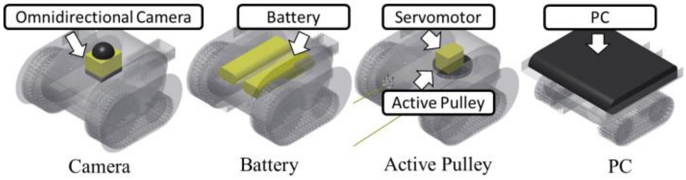


Figure 2. Various modules.

TABLE I. SPECIFICATIONS OF THE ROBOT

	Camera	Pulley	Battery	PC
Length [cm]	25	25	25	25
Height [cm]	15.5	15	15	12.5
Width [cm]	23	23	23	24.5
Weight [kg]	2.2	2.2	2.6	2.6

For example, by installing a battery, the basic module alone can be operated. It can turn to the desired direction by changing the rotating speed of the crawlers. These crawlers are operated by the user interface of a typical radio-controlled car.

ISBN: 978-1-61208-483-1

By connecting various modules, some abilities of the robot can be enhanced. Figure 3 shows an example of three-module robot. Each module has a flexible link to connect another module.

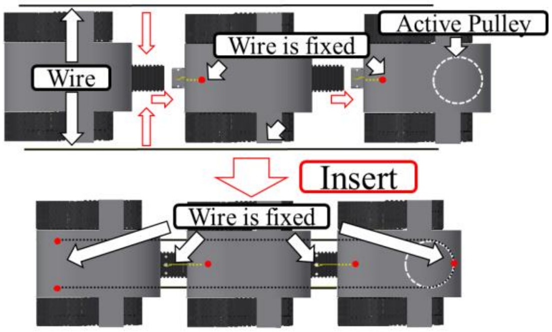


Figure 3. Muti-modules robot.

As shown in Figure 3. By connecting many modules, the mobility to overcome rubble can be enhanced. We can also install many functions on the robot. For example, by installing sensors and a PC, this robot can be controlled autonomously.

In case of many connected modules, the robot can be controlled by the mechanism proposed in [4, 5] as shown in Figure 4.

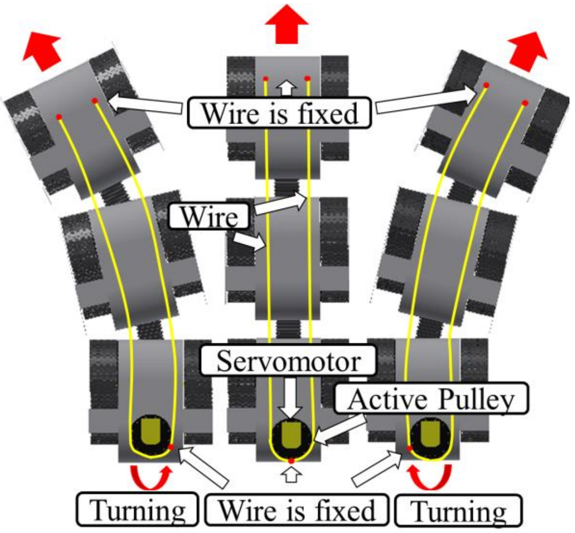


Figure 4. Turning mechanism.

We installed two wires on both sides of the robot as shown in Figure 3 and 4. These wires are pulled by an active pulley. Because all the joints are flexible, when the right (left) side of the wire is pulled, the body twists to the right (left), and the robot moves to the right (left). Because the joints move passively, the robot also adapts to complex environments and can avoid obstacles by utilizing the reactive force from contacting obstacles. Details of this mechanism are available in [4, 5].

Figure 5 shows the previous robot and proposed four-module robot.

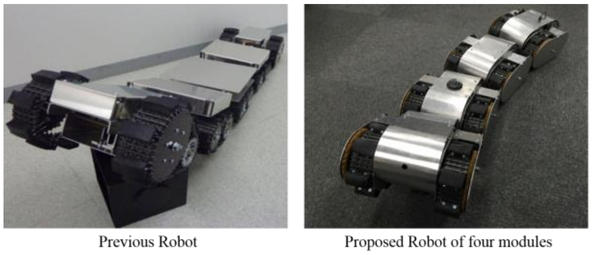


Figure 5. Previous robot and Proposed robot (four modules)

As shown in Figure 5, a robot similar to the previous robot can be realized by connecting Battery module, Camera module, and Pully module.

III. EXPERIMENT

We conducted experiments to confirm its mobility. Table 2 lists best results for each configuration.

TABLE II. EXPERIMENTAL RESULTS

	One module	Two modules	Three modules	Four modules
Bump [cm]	5	15	20	25
Dent [cm]	15	20	25	25
Minimum turning radius [cm]	0	60	90	120

From Table 2, we confirmed that the small number of modules robots have higher turning abilities, and the large number of modules robots have higher mobility to overcome bump and dent.

Next, we conducted experiments to confirm performance of the proposed module mechanism. First, we applied a one-module robot to the task of inspection in a ceiling space as shown in Figure 6.

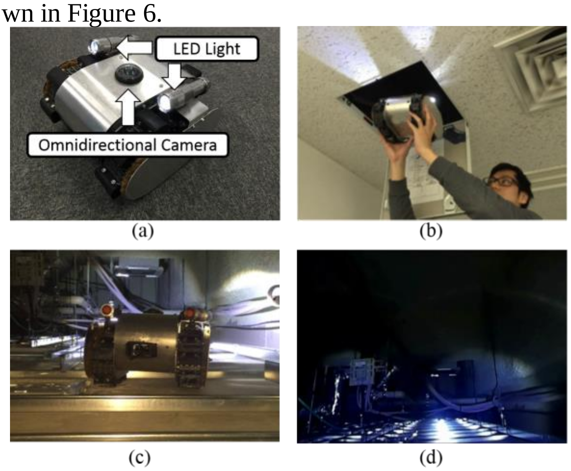
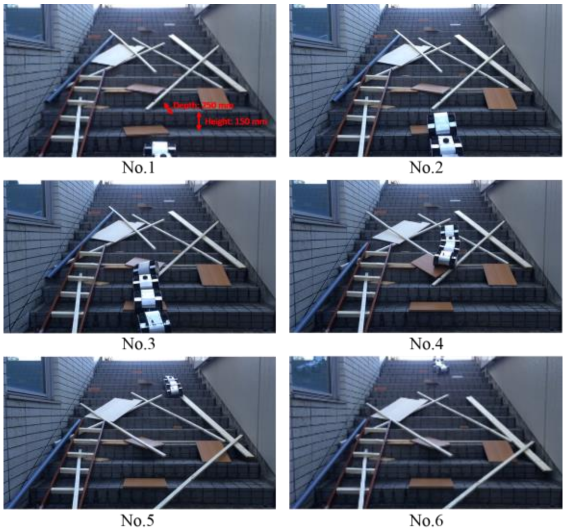


Figure 6. Inspection in a ceiling space : (a) One-module robot (b) Entrance of the celing space (c) Imagine from the entrance (d) image from the onborad camera

The module has two LED lights and one wireless omnidirectional camera. By exploiting its small size, the robot could enter the ceiling space.

Secondly, we applied four-modules robot to stairs with rubbles in order to confirm its mobility. Figure 7 shows experimental result.



No.1 No.2

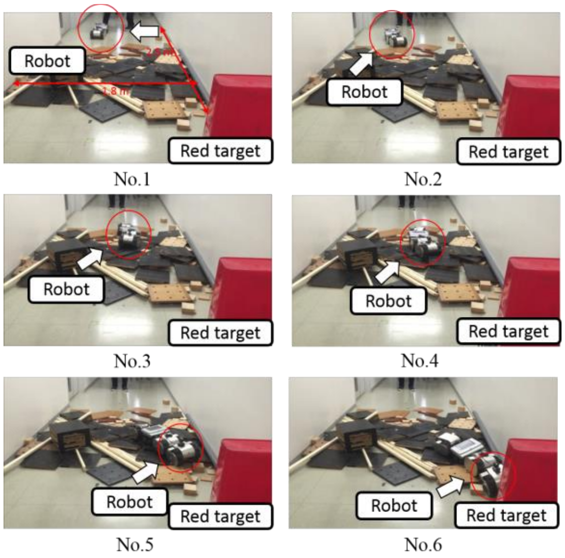
No.3 No.4

No.5 No.6

Figure 7. Experiment result of stairs

We can confirm that the robot has high mobility. In addition, as the flexible joints moves passively to adopt environment, operator did not have to control each joint. It means that proposed robot can be operated very easily and has high mobility.

Thirdly, we applied a four-modules autonomous robot to a rubble environment in order to confirm its autnomyy. This robot has a camera and a PC, and it chases a red target. Figure 8 shows the experimental result.



No.1 No.2

No.3 No.4

No.5 No.6

Figure 8. Experimental result of rubble

We confirmed that the robot can chase the red target autonoumously with overcoming the rubbles.

IV. CONCLUSION

In this research, we developed a modular robot by improving our previous serially connected crawler robot. With the proposed mechanism, the operator can customize the robot for a given task and given environment. Thus, the applicable tasks are drastically widened compared with the previous robot.

To demonstrate the effectiveness of the proposed robot, a prototype robot was developed. The results of experiments confirmed the advantages of both a small robot and a long-shaped robot can be realized by changing its formation.

ACKNOWLEDGMENT

This research was partially supported by the Japan Society for the Promotion of Science through the Grant-in-Aid for Scientific Research (C) 24500181.

REFERENCES

[1] R. R. Murphy et al., “Search and rescue robotics,” in Springer Handbook of Robotics, B. Sciliannopp et al., Eds. 1151–1173, 2008.

[2] L. Shao, **B.** Guo, Y. Wang, “An overview on theory and implementation of snake-like robots,” IEEE International Conference on Mechatronics and Automation (ICMA 2015), Aug. 2015, pp. 70-75, ISSN: 2152-7431

[3] S. Murata, H. Kurosawa, “Self-Reconfigurable Robot,” IEEE Robotics & Automation Magazine Vol. 14 , Dec. 2007, pp. 71-78, ISSN: 1070-9932

[4] K. Ito and H. Maruyama, “Semi-autonomous serially connected multi-crawler robot for search and rescue,” [Online]. Available from: http:/www.trandfonline.com/doi/pdf/10.1080/01691864.2015.1122553/2016.01.08

[5] M. Mizutani, H. Maruyama, and K. Ito, “Development of autonomous snake-like robot for use in rubble,” Proc. IEEE Int. Conf. Safety, Security, and Rescue Robotics, IEEE Press, Nov. 2012, pp. 1-7, doi: 10.1109/SSRR.2012.652388

[6] Y. Yokokohji, “Interface design for rescue robot operation-introduction of research outcomes from the human-interface group of the DDT project,” J. Robot. Soc. Japan, vol. 22, no. 5, 2004, pp. 566–569. 2012, pp. 1–7, 2012.

QoS-Aware Scale Up on IaaS Clouds

Luis Fernando Orleans
Computer Science Department
Universidade Federal Rural do Rio de Janeiro
Rio de Janeiro, Brazil
Email: lforleans@ufrrj.br

Geraldo Zimbrão da Silva
Computer Science Department
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
Email: zimbrao@cos.ufrj.br

***Abstract*—For systems hosted in IaaS clouds that target profit, like blogs and e-commerce systems, the final revenue should be the most important metric. Balancing the QoS experienced by clients as a manner to avoid their drop-out against the cost related to leasing instances forms the basis of a good instance management for those scenarios. In this paper, we demonstrate the feasibility of using a Fuzzy Logic Inference System as a tool for maintaining the QoS. Furthermore, it was created a method for reducing the overall cost related to instances leasing using an incremental algorithm, acquiring one computational unit at a time. Finally, we used hooks to handle unpredictable burst of requests, called here as noises. Our experiments evidenced that those methods can keep the response time of all requests below a deadline, avoiding customer dissatisfaction. At the same time, the total cost of servers lease is reduced, even when the cost-benefit among different configurations is not linear.**

***Index Terms*—scale up;PROFUSE;IaaS clouds**

I. Introduction

Elasticity is one of the key foundations of Cloud Computing [1] [2]. The ability to rapidly increase the number of resources without the need of service stopping/restarting plus its pay-per-usage nature opened room for a great number of proposals for minimizing both requests response time and costs with instance leasing [3] [4] [5] [6].

Elasticity became particularly important when Quality of Service (QoS) has turned into a crucial requirement for internet-based business models. If a customer is willing to purchase a product or a service, he or she expects the best treatment possible, which can be partially translated as not having to wait too long for his/her requests to be processed. In fact, [7] states that customers' patience lasts 4 seconds in average for each request. According to that study, when requests takes longer than 4 seconds to be processed a phenomena called customer drop-out emerges [8], i.e., clients begin to abandon the system unsatisfied. Guaranteeing QoS is important for avoiding customer frustration and the consequent revenue loss. Several studies have been done in that direction [5] [6] [9] [10], most of them using some mathematical modeling and targeting efficient workflow execution or minimizing the number of available instances for reducing the power consumption.

In this paper, we propose a novel *elasticity model* (em), called PROFUSE, that uses a Fuzzy Logic Inference System to calculate the necessary computing power needed to keep the QoS for requests processing.

A. Problem characterization

A cloud provider (*cp*) offers virtualized instances for e-commerce system providers (*sp*) to lease. Those instances can be of one out of a total of *t* different configurations, having each configuration a specific cost per hour of rental. Also, client *c* uses the system maintained by *sp*.

Instances already leased by *sp* form its *instances array*. The cloud provider can limit the maximum size of the array, hence forcing the client who wants to increase overall system computational power to acquire more expensive types of instances. The strategy a client uses to (re)lease an instance forms its elasticity model *em*. Also, *cp* provides an API that *sp* can use for acquiring new instances, opening room for a dedicated middleware that automates *em*.

The system maintained by *sp* works as follows: while surfing through the application, *c* issues several requests (product search, other customers comments about a product, providing credit card details, etc.). An incoming request is processed on an idle instance. If all instances are busy upon its arrival, the request is sent to a First Come First Serve (FCFS) queue. Also, requests response times should be kept under a threshold (a *deadline*) otherwise *c* becomes unsatisfied and leaves the system.

The Client Conversion Rate *ccr* is known and represents the percentage of system visits that actually become purchases and is calculated as $\frac{\#purchases}{\#visits}$. On a visit, *c* issues *n* requests in average until he or she leaves the system, regardless of whether a purchase was made. In average, each purchase generates a revenue *rb*. Hence, the mean value *v* of a request can be computed as

$$\overline{v} = \begin{cases} \overline{rb} \times \overline{ccr} \times \frac{1}{n} & \text{if deadline not reached} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Another way to calculate $\overline{v}$ is

$$\overline{v} = \overline{rb} \times \overline{ccr} \times \frac{1}{n} \times q \quad (2)$$

where q is the probability of missing the deadline for that request.

Consider that the system has h visits per day in average and each visit contains in average r requests, the mean day revenue can be computed as

$$\overline{rb}_{day} = \overline{h} \times \overline{r} \times \overline{v}. \tag{3}$$

Another variable that might affect daily revenue is the cost i related to instances leasing per hour, known as instance-hour. The relationship between q and i occurs as to increase the probability of processing requests within the deadline is sometimes necessary to increase the amount of computational power (instances) in the array. Hence, our goal is to minimize i without affecting q. A possible solution relies on Queueing Theory [11] where the following metrics are important: mean requests arrival rate (λ), mean requests processing time (β) and mean requests queue size (δ). The first two can be used to compute the mean system utilization (ρ) as

$$\rho = \lambda \times \beta. \tag{4}$$

Note that $1/\beta$ is the mean number of processed request per time unit. In this paper, β is the mean time a request takes to be processed by 1 instance core. Thus, (4) becomes

$$\rho = \lambda \times \beta \times \frac{1}{s} \tag{5}$$

s being the total number of cores contained in the instances array. An instance may have 1, 2 or 4 cores each and each configuration has its own price. The computational unit cost is achieved by dividing the instance hour price by the respective number of cores (the computational unit used in this work) of the configuration. A *non-linear* cost-benefit occurs when the computational unit cost varies for different configurations.

Finally, with probability *z*, an unpredictable burst of requests can increase λ in Z_p percent. Those bursts last $\overline{t}$ seconds in average and are referred to in this work as *noises*.

B. Contributions

The main contributions of this paper are:

1) A detailed guide for building a Fuzzy Logic-based Inference System that can predict workload changes and detect variations;
2) An efficient strategy for lease instances that aims to minimize the related cost;
3) A strategy that rapidly detects noises on the workload preventing deadline miss rate (DMR) increases.

C. Paper organization

The remainder of this paper is structured as follows: Section II lists the related work, while Sections III, IV and V present the several proposals of this paper. Section VI lists the experimental setup, the obtained results and discusses them thoroughly. Finally, Section VII lists the conclusions and points for future directions.

II. Related Work

Due to its novelty, scale up automation for systems hosted on IaaS clouds is an open problem, having several researches been done in that field.

In [3], authors consider geographically distributed datacenters and each hosted system having its own SLA – which establishes a maximum percentile of unattended deadlines. A dynamic ranking algorithm that identifies the most valuable requests, a gi-FIFO scheduling system and a heuristic-based task placement algorithm are presented.

Also, [12] modeled the problem of scale up as a predictive stochastic problem. The proposed approach explores the trade-off between QoS and servers lease cost by categorizing instances according to their configurations/costs and creating a cost function that is minimized using a customized Convex Optimization Solver algorithm, invoked periodically. Meanwhile, the research done in [13] proposes a scheduling algorithm based on jobs hierarchy. Such algorithm differentiates requests tied to an SLA from requests that do not have time-constraints and prioritize the processing of the former kind.

IBM presented the SmartScale tool in [4]. That work proposes a combination of horizontal and vertical scale up flavors to ensure the system is using the most affordable configuration and idle resources. It uses Decision Trees to periodically determine what have to be altered in the array and keep QoS constant.

Compared to this work, none of the previous mentioned researches comprises workload noises, neither did they used Fuzzy Logic to predict workload changes. Also, we focused solely on horizontal scale up as both [1] and [14] states that vertical scale up causes a momentary performance loss.

III. Predictable scale up

For predicting workload changes, a Fuzzy Logic Inference System (FLIS) [15] was created. We chose a FLIS over other inference mechanisms because it uses a set of IF-THEN rules which allows adjustments made by specialists. Prior to FLIS creation, a requests arrival histogram (RAH) is needed - it can be an estimate for systems that are not in production yet. Such histogram describes the number of requests that arrives per time unit (in this work we used *hour* as time unit, though it could be minute, second, etc.). The number of bars presented on the histogram represents the *time window* (a day, a week, a month, a year, or even longer periods). After RAH creation, the difference of workload (DoW) can be easily calculated from a time unit to the next through the simple formula:

$$DoW_h = DoW_{h+1} - DoW_h \tag{6}$$

where *h* is the current time bar on the histogram and *h+1* is the next one.

A. Linguistic variables definition

In order to determine the linguistic variables should be used in the FLIS, we used the following process: the RAH was used as input for various rounds of simulations, where some system variables were periodically logged: *requests arrival rate* (RAR), *requests processing rate* (RPR), *mean queue waiting time* (MQWT), *queue size* (QS), *system utilization* (SU), *deadline miss rate* (DMR). The log file served as input for an attribute selection/reduction analysis, performed on the WEKA software [16], which is a tool used by Data Mining professionals mostly because it has many Machine Learning algorithms implemented in it. Afterwards, the following attributes remained: SU, DMR and QS. Note that those variables can provide the FLIS its *reactive* behaviour only. As the *proactive* characteristic of the FLIS, we added two more variables: DoW and *time to next interval* (TTN), where the last stands for the number of time units remaining until the next bar of the histogram is reached.

The output variable was defined as the *number of computational power units* (NCPU). In this work we considered a computational power unit as a computing core, i.e. the number of cores denotes the computing capacity of an instance.

Finally, also using the WEKA software, we performed a Cluster Analysis [17] to determine the initial number fuzzy regions for each linguistic variable, a similar step to that performed by Google to define task placement strategies [18]. Near clusters were combined to reduce the total number of rules.

IV. Unpredictable scale up

The FLIS inside PROFUSE mechanism predicts and reacts well to workloads that are similar to previous ones, notably those that were used to build the RAH. However, web systems can incur into some situations where the difference on the expected number of requests and the actual number of requests is very high. In these scenarios, FLIS react speed may not be fast enough for prevent increasing on the deadline miss rate. As an example, consider a promotion widely spread on social networks made by a sales web system. The requests arrival rate explodes as the promotion announcement gets deeper into the social networks and people get interested on it. The miss rate increase is faster than FLIS feedback and should be detected separately. In this work, those abrupt and unexpected changes on workloads are denoted as *noises* and the noise detection system is called *hook*. Please note that noises are unusual and unpredictable events which severely affect workloads. However they do not last long, which means that once they are gone workloads return to previous states and FLIS use is effective again.

Essentially, a hook is a monitoring component and keeps critical units (e.g. miss rate, queue size, etc.) under close surveillance. Whenever one of those units behaves unexpectedly the expansion routine is called and the system is put back into a consistent state. The algorithm described on Figure 1 details how PROFUSE uses hooks.

V. Leasing policy

From a revenue-centric perspective, only instances with the cheapest configuration should fill in the array. However, such a strategy limits the computational power when cloud providers limit the number of instances and causes DMR to increase as the requests arrival rate increases. In order to assess the DMR impact, the cost of processing a request on each server configuration (t_x) should be added to (2).

```
procedure HOOK_MONITORING
    for all incoming request do
        $su$ ←current_system_utilization()
        $qs$ ←current_queue_size()
        $mr$ ←current_miss_rate()
        if $su$, $qs$, $mr$ exceeds threshold then
            $expand()$
        end if
    end for
end procedure
```

Fig. 1. Hooks monitoring algorithm

Consider ih_x as the instance-hour of a configuration *t* which can process *tr* requests per second. Hence, t_x can be calculated as:

$$t_x = \frac{ih_x}{tr \times 3600} \tag{7}$$

Equation (7) divides the instance-hour cost of configuration *t* by the total number of requests that *t* can handle in one hour. Therefore, equation 2 can be rewritten taking into consideration the cost of processing a request using configuration *t*

$$\overline{v} = \overline{rb} \times \overline{ccr} \times \frac{1}{n} \times q - t_x \tag{8}$$

Equation (8) computes the aggregated value of a request being processed on an instance with configuration *t* and client remains on the system with probability *q*.

Finally, in order to determine whether is more profitable to process the incoming request on an instance with configuration x_0 or with configuration x_1, where x_1 is more expensive and has twice computing power than x_0, the outcome of $v_{x1} \geq v_{x0}$ should be evaluated. Thus, when

$$q \geq \left(\frac{2 \times [c_{x1} - c_{x0}]}{\overline{rb} \times \overline{ccr} \times \frac{1}{n}} \right) \tag{9}$$

is worth the swapping. Note that i_1 has twice the computing power of i_0, hence the "2×" on (9)

Focusing on system provider financial loss reduction, PROFUSE's instance allocation works as follows: consider an IaaS cloud provider that limits the number of instances *sp* can lease on *MAX* instances. Each instance is of a configuration *c* and each configuration has an associated cost per hour. Therefore, each instance is represented as i_{nk}, where *n* is the position in the array ($1 \geq n \geq MAX$) and *k* is its configuration ($1 \geq k \geq t$). Hence, the initial array of instances can be represented as

$$a = \{i_{11}, i_{21}, ..., i_{n1}\} \tag{10}$$

```
procedure EXPANSION
    Start with an instance array with the minimal configuration ($k = 1$)
    While size(array) <max - 1 acquire instances with minimal configuration
    From that point onwards, use formula 9 to decide whether or not to lease an instance with a superior configuration. In case of swapping:
    Start an immediately superior configuration instance.
    Release an instance with the current configuration using 9
    If the array contains only instances of the configuration $k = t$, start a new instance of configuration $t$
end procedure
```

Fig. 2. Expansion algorithm

```
procedure INSTANCE_RELEASE($k$)
    for all instances $i$ in array do
        if $i$ is of type $k$ then
            Compute remaining time $rt$ until next instance-hour
        end if
    end for
    Release instance with the least $rt$
end procedure
```

Fig. 3. Instance release algorithm

In our approach, the initial array has only instances of the most basic type. For each positive outcome Δ computed by either FLIS or Hooks monitoring system, the leasing module keeps acquiring instances of type $k = 1$ until size of array reaches MAX - 1. Note that in case of *cp* does not impose an array size threshold, PROFUSE will lease only instances of that configuration. The remaining spot in the array is used for swapping instances when additional computing power is needed. Instance swapping consists on leasing an instance with a superior configuration and releasing a smaller instance – needed for future swaps.

Worth mentioning instance swapping only occurs when the outcome of (9) is positive, which indicates the $DMR \times profit$ balance was unfavorable. All instance swaps keep a free spot on the array except on the case the array contains only instances of configuration $k = t$ – when another instance of type *c* is acquired. Figure 2 describes the expansion algorithm.

Similarly, Figure 3 details release algorithm, taking into consideration the *release opportunity*, i.e. the instance closest to increase its cost.

In contrast to expansion algorithm, on the shrink algorithm (Figure 4) an instance of a more expensive configuration is released prior acquiring an instance with a simpler configuration.

```
procedure SHRINK
    if array is full then                ▷ All instances are of type $t$
        Release an instance using algorithm ( 3)
    else
        Find configuration $conf = MAX(k)$
        if $conf = 0$ then ▷ Only small instances in array
            Release an instance using algorithm ( 3)
        else
            Start an instance with configuration $conf - 1$
            Release an instance using algorithm ( 3)
        end if
    end if
end procedure
```

Fig. 4. Array shrinking algorithm

VI. EXPERIMENTS

In order to assess the robustness of PROFUSE, we conducted an exhaustive set of experiments using a simulator that was built for easily switch among a plethora of environments.

A. Workload types

For defining workloads shapes we used the study presented in [10]. According to the authors, four kinds of workloads are typical for systems hosted in IaaS clouds: (i) *stable*, where requests arrival rate is almost linear; (ii) *normal*, presenting the occurrence of peak situations; (iii) *growing*, where the number of incoming requests does not decrease over time; and (iv) *on-and-off*, representing some background, administrative tasks such as log archiving and compacting. Note that those workloads shapes can be combined and represent different epochs of the same system through time. Since the objective of this work is to find a new elasticity model capable of handle expected and unexpected burst of requests, the experiments were conducted using normal and growing workload shapes. The former was extracted from a real system whereas the later is a synthetic workload.

B. Parameters

Other parameters used on our experiments were extracted from [19] and are shown in Table I (comma-separated values indicates more than 1 value was used).

C. Environments

The experiments were conducted starting from the most basic scenario and then introducing limitations one at a time. To facilitate referencing the environments, letters were assigned as follows:

(A) Unlimited instances, workload without noises and hooks system deactivated;
(B) Limited number of instances, no noises and hooks system deactivated;
(C) Limited number of instances, noises and hooks system deactivated;
(D) Limited number of instances, noises and hooks system activated.

TABLE I
SIMULATOR PARAMETERS

first	second
Initial instances array size	5
Instance leasing mean time	97s
Instance releasing mean time	8s
Instances array max size	∞, 20
Single-core instance-hour cost	\$0.02
Dual-core instance-hour cost	\$0.34
Quad-core instance-hour cost	\$2.00
Deadline	4s
Client conversion rate	0.01, 1
Request value	\$100, \$0.001
Request-to-revenue probability	0.05, 1
Mean-time between FLIS feedback	120s

TABLE II
DMR PROFUSE WITH NORMAL WORLOAD

Environment	DMR
A	0.00014
B	0.00012
C	8.58581
D	0.00000

Note that environment (A) reflects a linear cost-benefit between instances configuration, since all computational units have the same cost.

PROFUSE's performance was compared against a basic elasticity model, that aims to keep the system utilization constant. Such EM is called Fixed Utilization and works by periodically (using same feedback interval used by PROFUSE) gathering data and (re)leasing instances in order to keep system utilization 0.7. Finally, all presented results are the mean value obtained out of 10 simulation rounds.

D. Results

Tables II and III compare PROFUSE's performances for all environments. Note that environment (B) has a slightly better performance over environment (A) because the boot time of new instances – the more instances leased, greater is the time needed to make them available. At environment (B) when the computational capacity should be increased in 4 units, a quad-core instance can be leased. On the other hand, at environment (A) the system has to wait 4 instances to boot up, with small fluctuations on their boot time.

When environment (C) is used there is a strong performance drop: approximately 8.6% and 10.8% of DMR for normal and growing workloads, respectively. Such a poor performance was expected since a FLIS is incapable of detecting noises and react to them efficiently. However, when hooks are turned on (environment(D)), PROFUSE presents an acceptable performance – there were no deadlines misses with the normal workload and only 0.53% of misses with the growing workload. Worth mention that PROFUSE's FLIS was using an *aggressive* configuration, trying to keep a system utilization of 0.8 in average. When we changed to a *conservative* approach and targeted system utilization to 0.6 in the FLIS, there were no misses at all (last line of Table III).

TABLE III
DMR PROFUSE WITH GROWING WORLOAD

Environment	DMR
A	0.01928
B	0.01691
C	10.79001
D	0.52795
D Conservative	0.00000

TABLE IV
ELASTICITY MODELS COMPARISON

EM	Env.A	Env.C	Env.D
FU	2.57830	97.98622	5.47310
PROFUSE	0.00014	8.58581	0,00000

For comparisons purpose, Table IV shows DMR of Fixed Utilization and PROFUSE elasticity models when submitted to the normal workload on environments (A), (C) and (D). As expected, PROFUSE outperforms FU model on all scenarios, being a more secure choice for guaranteeing QoS constraints.

Using equation 3, the total revenue loss can be calculated. Table V presents the results for both scenarios considered here: e-commerce systems and blogs (see section I-A). PROFUSE's robustness is confirmed on those results, such as the usefulness of hooks to handle workload noises.

Client conversion rate was set to 1% which, as suggested in [20]. Also, the minimum number of clicks needed to purchase an item is 7 (initial, item search, add to cart, initiate check-out, provide username and password or register, insert payment data, confirm purchase). To simulate a more real scenario, where users search other items, read opinions, etc., we assumed purchases are done after 20 requests in average. Finally, the mean value for each purchase was \$100.00. Revenue loss on a blog can be calculated in a simpler way, as all requests generate an income (\$0.001).

PROFUSE's instance rental efficiency was compared against both the cheapest case (20 single-core instances, fixed number) and the most expensive case (20 quad-core instances, fixed number), comprising lower and upper bounds. Table VI shows the cumulative cost (revenue loss plus cost with instances rental) for each strategy, where SC, QC, PL and PNL stands for Single-Core, Quad-Core, PROFUSE-Linear and PROFUSE-Non-Linear, respectively. From the 10th hour onwards, the single-core only strategy is incapable of maintain the agreed QoS becoming the most expensive configuration. From the results, it becomes clear that PROFUSE is a cheaper alternative than resource overprovisioning, here denoted as the quad-core only configuration.

VII. CONCLUSIONS AND FUTURE WORKS

IaaS cloud hosted systems administrators usually face the problem of deciding the computational power needed to ac-

TABLE V
REVENUE LOSS

Environment	E-Commerce	Blog
A	$0.21	$0.00
B	$0.18	$0.00
C	$18,546.68	$370.93
D	$0.00	$0.00

TABLE VI
CUMULATIVE COST

Time	SC	QC	PL	PNL
2	$0.80	$80.00	$0.16	$0.28
4	$3.60	$160.00	$0.74	$4.56
6	$8.00	$240.00	$1.32	$8.84
8	$14.00	$320.00	$2.54	$13.76
10	$20,397.31	$400.00	$3.98	$22.18
12	$61,533.23	$480.00	$6.00	$34.94
14	$102,678.31	$560.00	$8.42	$56.34
16	$143,894.43	$640.00	$11.12	$82.28
18	$185,064.57	$720.00	$14.06	$109.44
20	$226,253.46	$800.00	$18.04	$156.60
22	$267,323.94	$880.00	$22.38	$207.22
24	$308,412.46	$960.00	$27.82	$267.12

complish the QoS concerns needed to guarantee users satisfaction. Finding the cheapest combination among the number of instances, their configurations and prices is not an easy task. Also, as workloads varies through time resource overprovisioning can be a very expensive strategy – particularly for cases when the number of clients is small.

This paper presented a novel elasticity model called PROFUSE which computes the necessary computing power needed for keeping requests response times below a threshold. PROFUSE has a Fuzzy Logic Inference System that predicts workload changes. The processes used for determining FLIS variables, their fuzzy regions and initial IF-THEN rules were described in great detail. Also, for handling unpredictable huge workload changes PROFUSE provides a monitoring mechanism, called hooks, that keep crucial system metrics under close surveillance. Whenever an outlier is detected on one of those metrics, the computing power expansion routine is called. Finally, PROFUSE also provides a set of algorithms to lease and release instances using a revenue-centric approach where computational unit prices for each instance configuration are taken into consideration.

The experiments were conducted using two types of workloads that are typical for web system and four possible environments, covering from the simplest to the most complete scenario. Analyzing experiments results, PROFUSE's robustness is clear with it being able to keep the QoS even for the most stressful case. Finally, we showed that FLIS feedback time, called latency, is crucial for a good PROFUSE performance and should be set according to the mean time for leasing a new instance from the cloud provider.

A. Future works

As future works, we intend to investigate a way to identify workload patterns at runtime, giving PROFUSE the ability to handle different workloads with different FLIS and compare the approaches (single FLIS PROFUSE x multiple FLIS PROFUSE). In the same direction, we intend to investigate a method for automatize FLIS feedback times. Finally, we intend to build an incremental version of PROFUSE, without the need of historical data to create the FLIS.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, p. 5058, Apr. 2010.

[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[3] K. Boloor, R. Chirkova, Y. Viniotis, and T. Salo, "Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, Dec. 2010, pp. 464 –472.

[4] S. Dutta, S. Gera, A. Verma, and B. Viswanathan, "SmartScale: automatic application scaling in enterprise clouds," in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, Jun. 2012, pp. 221 –228.

[5] A. L. Freitas, N. Parlavantzas, and J.-L. Pazat, "An integrated approach for specifying and enforcing SLAs for cloud services," in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, Jun. 2012, pp. 376 –383.

[6] P. Leitner, W. Hummer, B. Satzger, C. Inzinger, and S. Dustdar, "Cost-efficient and application SLA-Aware client side request scheduling in an infrastructure-as-a-service cloud," in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, Jun. 2012, pp. 213 –220.

[7] G. McGovern. Selfish, mean, impatient customers: New thinking: Gerry McGovern. [Online]. Available: http://www.gerrymcgovern.com/nt/2008/nt-2008-07-14-selfish.htm

[8] M. Mazzucco, D. Dyachuk, and M. Dikaiakos, "Profit-aware server allocation for green internet services," *arXiv:1102.3059*, Feb. 2011, 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2010, pp 277-284.

[9] I. Goiri, F. Julií, J. O. Fitó, M. Macías, and J. Guitart, "Supporting cpu-based guarantees in cloud slas via resource-level qos metrics," *Future Gener. Comput. Syst.*, vol. 28, no. 8, pp. 1295–1302, Oct. 2012.

[10] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, Nov. 2011, pp. 1 –12.

[11] P. Bocharov, C. D'Apice, A. Pechinkin, and S. Salerno, *Queueing Theory*. Walter de Gruyter, 2004.

[12] H. Ghanbari, B. Simmons, M. Litoiu, C. Barna, and G. Iszlai, "Optimal autoscaling in a IaaS cloud," in *Proceedings of the 9th international conference on Autonomic computing*, ser. ICAC '12. New York, NY, USA: ACM, 2012, p. 173178. [Online]. Available: http://doi.acm.org/10.1145/2371536.2371567

[13] R. Rajavel and T. Mala, "Achieving service level agreement in cloud environment using job prioritization in hierarchical scheduling," ser. Advances in Intelligent and Soft Computing, S. Satapathy, P. Avadhani, and A. Abraham, Eds. Springer Berlin / Heidelberg, 2012, vol. 132, pp. 547–554.

[14] C. A. Ardagna, E. Damiani, F. Frati, D. Rebeccani, and M. Ughetti, "Scalability patterns for platform-as-a-service," in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, Jun. 2012, pp. 718 –725.

[15] R. C. Berkan and S. L. Trubatch, *Fuzzy systems design principles: building Fuzzy IF-THEN rule bases*. IEEE Press, Apr. 1997.

[16] U. of Waikato, "WEKA," http://www.cs.waikato.ac.nz/ml/weka, 2016, [Online; accessed 31-May-2016].

[17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.

[18] B. Sharma, V. Chudnovsky, J. L. Hellerstein, R. Rifaat, and C. R. Das, “Modeling and synthesizing task placement constraints in google compute clusters,” in *Proceedings of the 2nd ACM Symposium on Cloud Computing*. ACM, 2011, p. 3.

[19] M. Mao and M. Humphrey, “A performance study on the VM startup time in the cloud,” in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, Jun. 2012, pp. 423 –430.

[20] D. Felipini, “Plano de negcios para empresas da internet,” Jun. 2003.

Adaptive Control for Persistent Disturbance Rejection in Linear Infinite Dimensional Systems

Mark J. Balas
Embry-Riddle Aeronautical University
Daytona Beach, FL, USA
email: balasm@erau.edu

Susan A. Frost
NASA Ames Research Center
Moffet Field, CA, USA
emai: susan.frost@nasa.gov

***Abstract*—Given a linear continuous-time infinite-dimensional plant on a Hilbert space and persistent disturbances of known waveform but unknown amplitude and phase, we show that there exists a stabilizing direct model reference adaptive control law with disturbance rejection and robustness properties. The plant is described by a closed, densely defined linear operator that generates a continuous semigroup of bounded operators on the Hilbert space of states. There is no state or disturbance estimation used in this adaptive approach. Our results are illustrated by adaptive control of general linear diffusion systems.**

***Keywords*- Hilbert space; persistent disturbances; general linear diffusion systems; adaptive control**

I. Introduction

Many control systems are inherently infinite dimensional when they are described by partial differential equations. Currently there is renewed interest in the control of these kinds of systems especially in flexible aerospace structures and the quantum control field [1][2]. In this paper, we want to consider how to make a linear infinite-dimensional system regulate its output to zero in the presence of persistent disturbances.

In our previous work [3]-[6] we have accomplished direct model reference adaptive control and disturbance rejection with very low order adaptive gain laws for MIMO finite dimensional systems. When systems are subjected to an unknown internal delay, these systems are also infinite dimensional in nature. Direct adaptive control theory can be modified to handle this time delay situation for infinite dimensional spaces [7]. However, this approach does not handle the situation when partial differential equations (PDEs) describe the open loop system.

This paper addresses the effect of infinite dimensionality on the adaptive control approach of [4]-[6]. We will show that the adaptively controlled system is globally asymptotically stable using a new Barbalat-Lyapunov result. We apply this controller to linear PDEs with analytic semigroup generators and compact resolvent which model general linear diffusion systems.

II. Adaptive Regulation with Disturbance Rejection

Let X be an infinite dimensional separable Hilbert space with *inner product* (x,y) and corresponding norm $\|x\| \equiv \sqrt{(x,x)}$. Consider the Linear Infinite Dimensional Plant with *Persistent Disturbances*:

$$\begin{cases} \dfrac{\partial}{\partial t}x(t) = Ax(t) + Bu(t) + \Gamma\, u_D(t) \\ x(0) \equiv x_0 \in D(A) \subseteq X \\ Bu \equiv \sum\limits_{i=1}^{m} b_i u_i \\ y(t) = Cx(t) + Eu_D(t) \\ y_i \equiv (c_i, x(t)), i = 1...m \end{cases} \tag{1}$$

where $x \in D(A)$ is the plant state, $b_i \in D(A)$ are actuator influence functions, $c_i \in D(A)$ are sensor influence functions, $u, y \in \Re^m$ are the control input and plant output m-vectors respectively, u_D is a disturbance with known basis functions ϕ_D . The persistent disturbances u_D will enter the plant through the state channels Γ and the output channels E .

In order to accomplish disturbance rejection in a direct adaptive scheme, we will make use of a definition, given in [4][7], for persistent disturbances:

Definition 2: *A disturbance vector $u_D \in R^q$ is said to be* ***persistent*** *if it it satisfies the* ***disturbance generator equations****:*

$$\begin{cases} u_D(t) = \theta z_D(t) \\ \dot{z}_D(t) = F z_D(t) \end{cases} \text{or} \begin{cases} u_D(t) = \theta z_D(t) \\ z_D(t) = L\phi_D(t) \end{cases} \tag{2}$$

where F is a marginally stable matrix and $\phi_D(t)$ is a vector of known functions forming a basis for all the possible disturbances. This is known as “a disturbance with known waveform but unknown amplitudes”. We can easily show that an operator L exists to relate the persistent disturbances to a known basis vector $\phi_D(t)$, but the adaptive controller will not need to know the actual L .

The *objective of control* in this paper will be to cause the output *y(t)* of the plant to regulate asymptotically:

$$y \xrightarrow[t\to\infty]{} 0 \tag{3}$$

and this control objective will be accomplished by a *Direct Adaptive Control Law* of the form:

$$u = G_e y + G_D \phi_D \tag{4a}$$

The *direct adaptive controller* will have adaptive gains given by:

$$\begin{cases} \dot{G}_e = -yy^* \gamma_e ; \gamma_e > 0 \\ \dot{G}_D = -y\phi_D^* \gamma_D ; \gamma_D > 0 \end{cases} \tag{4b}$$

Note that the output feedback gains are directly adapted and no estimation or identification of plant information is used in the control law.

III. Ideal Trajectories

We define the *Ideal Trajectories* for (1) in the following way:

$$\begin{cases} x_* = S_1 z_D \\ u_* = S_2 z_D \end{cases} \text{with } z_D \in \Re^{N_D} \tag{5}$$

where the *ideal trajectory* $x_*(t)$ is generated by *the ideal control* $u_*(t)$ from

$$\begin{cases} \dfrac{\partial x_*}{\partial t} = Ax_* + Bu_* + \Gamma u_D \\ y_* = Cx_* + Eu_D = 0 \end{cases} \tag{6}$$

If such ideal trajectories exist, they will be linear combinations of disturbance state, and they will produce exact output tracking in a disturbance-free plant (8).

By substitution of (5) into (6), we obtain the *Model Matching Conditions:*

$$\begin{cases} AS_1 + BS_2 = S_1 F + \underbrace{H_1}_{\Gamma\theta} \\ CS_1 = H_2 = -E\theta \end{cases} \tag{7}$$

where $S_1 : \Re^{N_D} \to D(A) \subset X , S_2 : \Re^{N_D} \to \Re^M$.

Because (S_1, S_2) are both of finite rank, they are bounded linear operators on their respective domains.

A Special Case occurs when *E=0* and $Range(\Gamma) \subseteq Range(B)$. Then there exists *S₂* such that $BS_2 + \Gamma\theta = 0$ and *S₁=0.* In this case the full system state x becomes disturbance-free, but in general we really only want to make the output y disturbance-free.

IV. Normal Form

We need two lemmae:

Lemma 1: If CB is nonsingular then $P_1 \equiv B(CB)^{-1}C$ is a (non-orthogonal) bounded projection onto the *range of B*, R(B), along the *null space of C*,N(C) with $P_2 \equiv I - P_1$ the complementary bounded projection, and $X = R(B) \oplus N(C)$, as well as $D(A) = R(B) \oplus [N(C) \cap D(A)]$.

Proof of Lemma 1: See [17].

Now for the above pair of projections $(P_1 . P_2)$ we have

$$\begin{cases} \dfrac{\partial P_1 x}{\partial t} = P_1 \dfrac{\partial x}{\partial t} = (\underbrace{P_1 A P_1}_{A_{11}}) P_1 x + (\underbrace{P_1 A P_2}_{A_{12}}) P_2 x + (\underbrace{P_1 B}_{B}) u \\ \dfrac{\partial P_2 x}{\partial t} = P_2 \dfrac{\partial x}{\partial t} = (\underbrace{P_2 A P_1}_{A_{21}}) P_1 x + (\underbrace{P_2 A P_2}_{A_{22}}) P_2 x + (\underbrace{P_2 B}_{=0}) u \\ y = (\underbrace{CP_1}_{C}) P_1 x + (\underbrace{CP_2}_{=0}) P_2 x \end{cases}$$

which implies $\begin{cases} \dfrac{\partial P_1 x}{\partial t} = A_{11} P_1 x + A_{12} P_2 x + Bu \\ \dfrac{\partial P_2 x}{\partial t} = A_{21} P_1 x + A_{22} P_2 x \\ y = CP_1 x = Cx \end{cases}$

because $y = Cx = C(B(CB)^{-1}C)x = CP_1 x$

and $P_1 x = B(CB)^{-1}Cx = B(CB)^{-1}y.$

and $CP_2 = C - CB(CB)^{-1}C = 0$

and $P_2 B = B - B(CB)^{-1}CB = 0.$

Lemma 2: If CB is nonsingular, then there exists and invertible, bounded linear operator

$$W \equiv \begin{bmatrix} C \\ W_2 P_2 \end{bmatrix} : X \to \tilde{X} \equiv R(B) x l_2$$

such that

$$\bar{B} \equiv WB = \begin{bmatrix} CB \\ 0 \end{bmatrix}, \bar{C} \equiv CW^{-1} = \begin{bmatrix} I_m & 0 \end{bmatrix}, \text{and } \bar{A} \equiv WAW^{-1} \cdot$$

This coordinate transformation puts (1) into *normal form:*

$$\begin{cases} \dot{y} = \bar{A}_{11} y + \bar{A}_{12} z_2 + CBu \\ \dfrac{\partial z_2}{\partial t} = \bar{A}_{21} y + \bar{A}_{22} z_2 \end{cases} \tag{8}$$

where the subsystem: $(\bar{A}_{22}, \bar{A}_{12}, \bar{A}_{21})$ is called the *zero dynamics* of (1) and

$$\bar{A}_{11} \equiv CA_{11}B(CB)^{-1} = CAB(CB)^{-1}; \bar{A}_{12} \equiv CAW_2^*;$$
$$\bar{A}_{21} \equiv W_2 A_{21} B(CB)^{-1}; \bar{A}_{22} \equiv W_2 A_{22} W_2^*$$

and $W_2 : X \to l_2$ by $W_2 x \equiv \begin{bmatrix} (\theta_1, P_2 x) \\ (\theta_2, P_2 x) \\ (\theta_3, P_2 x) \\ \cdots \\ \end{bmatrix}$ is an isometry from $N(C)$ into l_2.

Proof of Lemma 2: See [17].

Now we can prove the following theorem about the *Existence of Ideal Trajectories*:

Theorem 1: Assume CB is nonsingular. Then

$$\sigma(F) = \sigma_p(F) \subset \rho(\bar{A}_{22})$$
$$\equiv \{\lambda \in C / (\lambda I - \bar{A}_{22})^{-1} : l_2 \to l_2 \text{ is a bounded linear operator}\}$$

(or $\sigma_p(F) \cap \sigma(\bar{A}_{22}) = \varphi$ where $\sigma(\bar{A}_{22}) \equiv [\rho(\bar{A}_{22})]^c$)

if and only there exist unique bounded linear operator solutions (S_1, S_2) satisfying the Matching Conditions (7).

Proof: Define $\bar{S}_1 \equiv W^{-1} S_1 = \begin{bmatrix} \bar{S}_a \\ \bar{S}_b \end{bmatrix}$ and $\bar{H}_1 \equiv WH_1 = \begin{bmatrix} \bar{H}_a \\ \bar{H}_b \end{bmatrix}$. From (7), we obtain

$$\begin{cases} \bar{A}\bar{S}_1 + \bar{B}S_2 = \bar{S}_1 L_m + \bar{H}_1 \\ \bar{C}\bar{S}_1 = H_2 \end{cases}$$

where $(\bar{A}, \bar{B}, \bar{C})$ is the Normal Form (8). From this we obtain:

$$\begin{cases} \bar{S}_a = H_2 \\ S_2 = (CB)^{-1}[H_2 L_m + \bar{H}_a - (\bar{A}_{11} H_2 + \bar{A}_{12} \bar{S}_b)] \\ \bar{A}_{22} \bar{S}_b - \bar{S}_b F = \bar{H}_b - \bar{A}_{21} H_2 \end{cases}$$

We can rewrite the last of these equations as

$(\lambda I - \bar{A}_{22})\bar{S}_b - \bar{S}_b(\lambda I - F) = \bar{A}_{21} H_2 - \bar{H}_b \equiv \bar{H}$ for all complex λ. Now assume that F is simple and therefore provides a basis of eigenvectors $\{\phi_k\}_{k=1}^{N_D}$ for $\Re^{N_D}$. This is not essential but will make this part of the proof easier to understand. The proof can be re-done with generalized eigenvectors and the Jordan form. So we have

$$(\lambda_k I - \bar{A}_{22})\bar{S}_b \varphi_k - \bar{S}_b \underbrace{(\lambda_k I - F)\varphi_k}_{=0} = \bar{A}_{21} H_2 - \bar{H}_b \equiv \bar{H}$$

which implies

$$\bar{S}_b \varphi_k = (\lambda_k I - \bar{A}_{22})^{-1} \bar{H} \varphi_k \text{ because } \lambda_k \in \sigma(F) \subset \rho(\bar{A}_{22})$$

Thus we have

$$\bar{S}_b z = \sum_{k=1}^{L} \alpha_k (\lambda_k I - \bar{A}_{22})^{-1} \bar{H} \phi_k \forall z = \sum_{k=1}^{L} \alpha_k \phi_k \in \Re^L .$$

Since $\lambda_k \in \sigma(F) \subset \rho(\bar{A}_{22})$, all $(\lambda_k I - \bar{A}_{22})^{-1}$ are bounded operators. Also $\bar{H} \equiv \bar{A}_{21} H_2 - \bar{H}_b$ is a bounded operator on $\Re^{N_D}$. Therefore $\bar{S}_b$ is a bounded linear operator, and this leads to S_1 also bounded linear.

If we look at the converse statement and let $\lambda_* \in \sigma(F) \cap \sigma(\bar{A}_{22}) = \phi$.

Then there exists $\varphi_* \neq 0$ such that

$$(\lambda_* I - \bar{A}_{22})\bar{S}_b \varphi_* - \bar{S}_b \underbrace{(\lambda_* I - F)\varphi_*}_{=0} = (\lambda_* I - \bar{A}_{22})\bar{S}_b \varphi_*$$
$$= \bar{H}.$$

In this case 3 things can happen when $\lambda_* \in \sigma(\bar{A}_{22})$: $(\lambda_* I - \bar{A}_{22})$ can fail to be 1-1 so multiple solutions of $\bar{S}_b$ will exist, $R(\lambda_* I - \bar{A}_{22})$ can fail to be all of *X* so no solutions $\bar{S}_b$ may occur, or $(\lambda_* I - \bar{A}_{22})^{-1}$ can fail to be a bounded operator so solutions $\bar{S}_b$ may be unbounded. In all cases these 3 alternatives lead to a lack of unique bounded operator solutions for S_1.

And the proof of Theo. 1 is complete.

It is possible to relate the point spectrum $\sigma_p(\bar{A}_{22}) \equiv \{\lambda / \lambda I - \bar{A}_{22} \text{ not } 1\text{-}1\}$ to the set Z of *transmission (or blocking) zeros* of (A, B, C).

Similar to the finite-dimensional case [16], we can see that

$$Z \equiv \left\{ \begin{array}{l} \lambda / V(\lambda) \equiv \begin{bmatrix} \lambda I - A & B \\ C & 0 \end{bmatrix} : \\ D(A) x \Re^m \to X x \Re^m \text{ linear operator is not 1-1} \end{array} \right\}$$

Lemma 3: $Z = \sigma_p(\bar{A}_{22}) \equiv \{\lambda / \lambda I - \bar{A}_{22} \text{ is not } 1\text{-}1\}$ is called the *point spectrum* of $\bar{A}_{22}$. *So the transmission zeros of the infinite-dimensional open-loop plant* (A, B, C) *are the eigenvalues of its zero dynamics* $(\bar{A}_{22}, \bar{A}_{12}, \bar{A}_{21})$.

Proof of Lemma 3:
From

$$\bar{V}(\lambda)=\begin{bmatrix}\lambda I-\bar{A} & \bar{B}\\ \bar{C} & 0\end{bmatrix}$$

$$=\begin{bmatrix}W^{-1} & 0\\ 0 & I\end{bmatrix}\underbrace{\begin{bmatrix}\lambda I-A & B\\ C & 0\end{bmatrix}}_{V(\lambda)}\begin{bmatrix}W & 0\\ 0 & I\end{bmatrix}$$

we obtain $\begin{bmatrix}\lambda I-\bar{A} & \bar{B}\\ \bar{C} & 0\end{bmatrix}$ not 1-1 if and only if $\begin{bmatrix}\lambda I-A & B\\ C & 0\end{bmatrix}$ not 1-1.

But, using normal form from Lemma 2,

$$\bar{V}(\lambda)\equiv\begin{bmatrix}\lambda I-\bar{A} & \bar{B}\\ \bar{C} & 0\end{bmatrix}=\begin{bmatrix}\lambda I-\bar{A}_{11} & -\bar{A}_{12} & CB\\ -\bar{A}_{21} & \lambda I-\bar{A}_{22} & 0\\ I_m & 0 & 0\end{bmatrix}$$

And therefore

$$0=\bar{V}(\lambda)h=\bar{V}(\lambda)\begin{bmatrix}h_1\\ h_2\\ h_3\end{bmatrix}\text{ if and only if}$$

$$h_1=0;h_3=(CB)^{-1}\bar{A}_{12}h_2;(\lambda I-\bar{A}_{22})h_2=0.$$

So $h\neq 0$. if and only if $h_2\neq 0$ Therefore $\begin{bmatrix}sI-\bar{A} & \bar{B}\\ \bar{C} & 0\end{bmatrix}$ not 1-1 if and only if $\lambda\in\sigma_p\ (\bar{A}_{22})$.
This completes the proof of Lemma 3.

Using Lemma 3 and Theo. 1, we have the following *Internal Model Principle*:

Corollary 1: Assume *CB* is nonsingular and $\sigma(\bar{A}_{22})=\sigma_p(\bar{A}_{22})\ =\sigma_p(P_2AP_2)$ where $\bar{A}_{22}\equiv W_2^*P_2AP_2W_2$. There exist unique bounded linear operator solutions (S_1,S_2) satisfying the Matching Conditions (10) if and only if $\sigma(F)\cap Z=\varphi,$ i.e., no eigenvalues of *F* can be zeros of the open-loop plant *(A,B,C)*.

Note: $\lambda I-\bar{A}_{22}$ is not 1-1 if and only if there exists $x\neq 0$ such that $P_2x\neq 0$ and

$$\begin{aligned}0&=(\lambda I-\bar{A}_{22})W_2P_2x\\ &=(\lambda\underbrace{W_2W_2^*}_{I}-W_2PAP_2W_2^*)W_2P_2x\\ &=[W_2(\lambda I-P_2AP_2)W_2^*]W_2P_2x\end{aligned}$$

if and only if

$W_2(\lambda I-P_2AP_2)W_2^*$ is not 1-1 on *N(C).*

But W_2 is an isometry on *N(C).*

Therefore $\sigma_p(\bar{A}_{22})=\sigma_p(P_2AP_2)$.

V. Stability of the Error System

The error system can be found from (1), (2) and (6): Define $e\equiv x-x_*$and $\Delta u\equiv u-u_*$ this implies

$$\begin{cases}\dfrac{\partial e}{\partial t}=Ae+B\Delta u\\ y=y-0=\Delta y\equiv y-y_*=Ce\end{cases}\tag{9}$$

Now we consider the definition of Strict Dissipativity for infinite-dimensional systems and the general form of the "adaptive error system" to prove stability. The main theorem of this section will later be utilized to assess the convergence and stability of the adaptive controller with disturbance rejection for linear diffusion systems.

Noting that there can be some ambiguity in the literature with the definition of strictly dissipative systems, we modify the suggestion of Wen in [8] for finite dimensional systems and expand it to include infinite dimensional systems.

Definition 1*: The triple* (A_c, *B, C*) *is said to be* ***Strictly Dissipative (SD)*** *if* A_c *is a densely defined ,closed operator on* $D(A_c)\subseteq X$ *a complex Hilbert space with inner product* (x,y) *and corresponding norm* $\|x\|\equiv\sqrt{(x,x)}$ *and generates a* C_0 *semigroup of bounded operators* $U(t)$ *,and* (B,C) *are bounded finite rank input/output operators with rank M where* $B:R^m\rightarrow X$ and $C:X\rightarrow R^m$ *. In addition there exist symmetric positive bounded operator P and Q* on *X such that*

$$0\leq p_{\min}\|e\|^2\leq(Pe,e)\leq p_{\max}\|e\|^2;0\leq q_{\min}\|e\|^2\leq(Qe,e)\leq q_{\max}\|e\|^2$$

i.e. P,Q are bounded and coercive, and

$$\begin{cases}\mathrm{Re}(PA_ce,e)\equiv\dfrac{1}{2}[(PA_ce,e)+\overline{(PA_ce,e)}]\\ =\dfrac{1}{2}[(PA_ce,e)+(e,PA_ce)]\\ =-(Qe,e)\leq-q_{\min}\|e\|^2;e\in D(A_c)\\ PB=C^*\end{cases}\tag{10}$$

where $W^*$is the adjoint of the operator *W*.

We also say that (A, B, C) is *Almost Strictly Dissipative (ASD)* when there exists G_*mxm gain such that (A_c,*B*,*C*) is SD with $A_c\equiv A+BG_*C$ **.** Note that if P=I in (5a), by the Lumer-Phillips Theorem [10], p 405, we would have $\|U_c(t)\|\leq e^{-\sigma t};t\geq 0\ ;\sigma\equiv q_{\min}>0$.

Henceforth, we will make the following set of assumptions:
Hypothesis 1: *Assume the following:*

i.) *There exists a gain,* G_e^* *such that the triple* $(A_C \equiv A + BG_e^*C, B, C)$ *is SD, i.e.* (A,B,C) is ASD,

ii.) *A is a densely defined ,closed operator on* $D(A) \subseteq X$ *and generates a* C_0 *semigroup of bounded operators* $U(t)$,

iii.) ϕ_D is bounded

From (5), we have $u_* = S_2 z_D$ and using (4a), we obtain:

$$\Delta u \equiv u - u_* = (G_e y + G_D \varphi_D) - (S_2 \underbrace{z_D}_{L\varphi_D})$$

$$= G_e^* y + \Delta G_e y + \Delta G_D \varphi_D = G_e^* e_y + \Delta G \eta \qquad (11)$$

where

$\Delta G \equiv G - G_*; G \equiv \begin{bmatrix} G_e & G_D \end{bmatrix}; G_* \equiv \begin{bmatrix} G_e^* & S_2 L \end{bmatrix};$

$G_D^* \equiv S_2 L;$ and $\eta \equiv \begin{bmatrix} y \\ \varphi_D \end{bmatrix}$

From (4), (9) and (11), the *Error System* becomes

$$\begin{cases} \dfrac{\partial e}{\partial t} = \underbrace{(A + BG_e^*C)}_{A_c} e + B\Delta G\eta = A_c e + B\rho; \\ e \in D(A); \rho \equiv \Delta G\eta \\ e_y = Ce \\ \Delta\dot{G} = \dot{G} - \dot{G}_* = \dot{G} = -e_y \eta^* \gamma \end{cases} \qquad (12)$$

where $\gamma \equiv \begin{bmatrix} \gamma_e & 0 \\ 0 & \gamma_D \end{bmatrix} > 0$

Since *B,C* are finite rank operators, so is BG_e^*C. Therefore

$A_c \equiv A + BG_e^*C$ which has $D(A_c) = D(A)$ and generates a C_0 semigroup $U_c(t)$ because *A* does (see [9] Theo 2.1 p 497). Furthermore, by Theo 8.10 p 157 in [11], *x(t)* remains in *D(A)* and is differentiable there for all $t \geq 0$ .This is because $F(t) \equiv B\rho = B\Delta G\eta$ is continuously differentiable in *D(A).*

We see that (12) is the *feedback interconnection* of an infinite-dimensional linear subsystem with $e \in D(A) \subseteq X$ and a finite-dimensional subsystem with $\Delta G \in \Re^{mxm}$. This can be written in the following form using $w \equiv \begin{bmatrix} e \\ \Delta G \end{bmatrix} \in D \equiv D(A)x\Re^{mxm} \subseteq \bar{X} \equiv Xx\Re^{mxm}$:

$$\begin{cases} \dfrac{\partial w}{\partial t} = w_t = f(t,w) \equiv \begin{bmatrix} A_c e + B\rho(t) \\ -e_y \eta^* \gamma \end{bmatrix} \\ w(t_0) = w_0 \in D \text{ dense in } \bar{X} \equiv Xx\Re^{mxm} \end{cases} \qquad (13)$$

The inner product on $\bar{X} \equiv Xx\Re^{mxm}$ can be defined as

$$(w_1, w_2) \equiv \left(\begin{bmatrix} x_1 \\ \Delta G_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ \Delta G_2 \end{bmatrix} \right) \equiv (x_1, x_2) + tr(\Delta G_2 \Delta G_1^*)$$

which will make it a Hilbert space also.

Now we present a new version of Barbalat-Lyapunov for systems on an infinite dimensional Hilbert space:

Theorem 2 (Lyapunov-Barbalat): Let $w(t) = w(t, t_0, w_0) \in D$ and $V(t,w)$ satisfy:

$$\begin{cases} \alpha \|w\|^2 \leq V(t,w) \leq \beta \|w\|^2 \\ \dot{V}(t,w) \equiv \dfrac{\partial V(t,w)}{\partial t} + \dfrac{\partial V(t,w)}{\partial w} f(t,w) \leq -S(w) \leq 0 \end{cases}$$

for all $w \in D$. Then $w(t)$ is bounded in $\bar{X}$. Furthermore, if the following are true:

a) $S(w) \geq \mu \|\aleph w\|^2 \; \forall w \in D; \mu > 0;$ with $\aleph$ a bounded operator on $D \subseteq \bar{X} \equiv Xx\Re^{mxm} \rightarrow X$ such that $(\aleph w)_t = \aleph w_t$

b) $\mathrm{Re}\left(\aleph w, \aleph f(t,w)\right)$ is bounded on bounded sets of $w \in D$.

Then $\aleph w(t) \xrightarrow[t \to \infty]{} 0.$

Proof: See Appendix I in [17].

For this proof, we will need the following version of Barbalat's Lemma; see [15] pp210-211:

Lemma 4: We say *f(t)* is a *uniformly continuous* function on (0,∞) when for all $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$ such that $\left| f(t_2) - f(t_1) \right| < \varepsilon \;\; \forall \left| t_2 - t_1 \right| < \delta$. If *f(t)* is a real, *uniformly continuous* function on (0,∞) with $\int_0^\infty f(t)dt < \infty$, then $f(t) \xrightarrow[t \to \infty]{} 0$.

Now we can prove the stability and convergence of the direct adaptively controlled error system (15):

Theorem 3: Under Hypothesis 1and Re $(A_c e, e)$ bounded on bounded sets of $e \in D(A)$ we will have state and output tracking of the reference model: $e \xrightarrow[t \to \infty]{} 0,$ and since *C* is a bounded linear operator: $e_y = y - y_m = Ce \xrightarrow[t \to \infty]{} 0$ with bounded adaptive gains $G \equiv \begin{bmatrix} G_e & G_m & G_u & G_D \end{bmatrix} = G_* + \Delta G$

Proof: See Appendix II in [17].

VI. Application: Adaptive Control of Unstable Diffusion Equations

We will apply the above direct adaptive controller on the following single-input/single-output Cauchy problem which represents a *general linear diffusion problem*:

$$\begin{cases} \dfrac{\partial x}{\partial t} = Ax + b(u + u_D), x(0) \equiv x_0 \in D(A) \\ y = (c, x), \text{ with } b = c \in D(A) \end{cases} \qquad (14)$$

where *A* has compact resolvent and generates and analytic *C₀* semigroup.

From the compact resolvent property, we know that $\sigma(A) = \sigma_p(A)$ and by the analyticity requirement there will only be a finite number of unstable eigenvalues $\lambda_k \in \sigma_p(A)$.

Consequently, there exists G_* such that $A_c \equiv A + BG_*C$ satisfies

$\mathrm{Re}\,\lambda_k \leq -\mu < 0 \forall \lambda_k \in \sigma_p(A_c)$ which implies that

$$\mathrm{Re}(A_c x, x) \equiv \frac{1}{2}[(A_c x, x) + \overline{(A_c x, x)}] = \frac{1}{2}[(A_c x, x) + (x, A_c x)]$$
$$= -(Qx, x) \leq -\mu \|x\|^2 ; x \in D(A_c)$$

Also, since *b*=c we have $C^* = B$. Therefore we have that (A, B, C) is ASD with $P = I$.

From $\mathrm{Re}(A_c x, x) \leq -\mu \|x\|^2 \forall x \in D(A)$ we clearly have $\mathrm{Re}(A_c x, x)$ bounded on bounded sets of $x \in D(A)$.

For this application we will *assume the disturbances are sinusoidal with frequency 1 rad/sec* (but this is not a restriction as long as φ_D is bounded:

o

$$\begin{cases} u_D = \begin{bmatrix} 1 & 0 \end{bmatrix} z_D \\ \dot{z}_D = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} z_D \end{cases}$$

implies that

$$F = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}; \theta_D = \begin{bmatrix} 1 & 0 \end{bmatrix}; \varphi_D \equiv \begin{bmatrix} \sin t \\ \cos t \end{bmatrix}$$

implies that

$$u = G_e y + G_D \varphi_D \text{ with } \begin{cases} \dot{G}_e = -yy^* \gamma_e \\ \dot{G}_D = -y\varphi_D^* \gamma_D \end{cases}$$

So, since $B = \Gamma$, there is a gain $S_2 = -\theta$ such that $BS_2 + \Gamma\theta = B(-\theta + \theta) = 0$ which implies that $S_1 = 0$ and this is the special case of (7). Finally E=0 and the eigenvalues of F are $\pm j$ but the zeros of *(A,B,C)* are real; so the matching conditions are satisfied and ideal trajectories exist. Therefore we satisfy the hypothesis of Theo. 3 and we have, via the direct adaptive controller, state regulation $x \xrightarrow[t \to \infty]{} 0$ and output regulation $y \xrightarrow[t \to \infty]{} 0$ with bounded adaptive gains $G \equiv \begin{bmatrix} G_e & G_D \end{bmatrix}$ in the presence of sinusoidal persistent disturbances.

VII. Conclusions

In Theorem 1, we showed conditions under which ideal trajectories exist for a linear infinit-dimensional system to be capable of rejecting a persistent disturbance in the the output of the plant. In Theorem 3 we used an extension of Barbalat-Lyapunov result for linear dynamic systems on infinite-dimensional Hilbert spaces under the hypothesis of almost strict dissipativity for infinite dimensional systems, to show that direct adaptive control can regulate the state and the output of a linear infinite-dimensional system in the presence of persistent disturbances without using any kind of state or parameter estimation.. We applied these results to a general linear diffusion problem with sinusoidal disturbances using a single actuator and sensor and direct adaptive output feedback.

These results do not require deep knowledge of specific properties or parameters of the system to accomplish model tracking. And they do not require that the disturbance enter through the same channels as the control.

References

[1] A. Pazy, Semigroups of Linear Operators and Applications to partial Differential Equations, Springer 1983.

[2] D. D'Alessandro, Introduction to Quantum Control and Dynamics, Chapman & Hall, 2008.

[3] M. Balas, R.S. Erwin, and R. Fuentes, "Adaptive control of persistent disturbances for aerospace structures", AIAA GNC, Denver, 2000.

[4] R. Fuentes and M. Balas, "Direct Adaptive Rejection of Persistent Disturbances", Journal of Mathematical Analysis and Applications, Vol 251, pp. 28-39, 2000

[5] R. Fuentes and M. Balas, "Disturbance accommodation for a class of tracking control systems", Proceedings of AIAA GNC, Denver, Colorado, 2000.

[6] R. Fuentes and M. Balas, "Robust Model Reference Adaptive Control with Disturbance Rejection", Proc. ACC, 2002.

[7] M. Balas, S. Gajendar, and L. Robertson, "Adaptive Tracking Control of Linear Systems with Unknown Delays and Persistent Disturbances (or Who You Callin' Retarded?)", Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL, Aug 2009.

[8] J. Wen, "Time domain and frequency domain conditions for strict positive realness", IEEE Trans Automat. Contr., vol. 33, no. 10, pp. 988-992, 1988.

[9] T. Kato, Perturbation Theory for Linear Operators, Springer, 1980.

[10] M. Renardy and R. Rogers, An Introduction to Partial Differential Equations, Springer, 1993

[11] R. Curtain and A. Pritchard, Functional Analysis in Modern Applied Mathematics, Academic Press, 1977.

[12] M. Balas, "Trends in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams", IEEE Trans Automatic Control, AC-27, No. 3, 1982.

[13] M. Balas and R. Fuentes, "A Non-Orthogonal Projection Approach to Characterization of Almost Positive Real Systems with an Application to Adaptive Control", Proc of American Control Conference, 2004.

[14] P. Antsaklis and A. Michel, A Linear Systems Primer, Birkhauser, 2007.

[15] V.M. Popov, Hyperstability of Control Systems, Springer, Berlin, 1973.

[16] T. Kailath, Linear Systems, Prentice-Hall, pp. 448-449, 1980.

[17] M. Balas and S. Frost, "Robust Adaptive Model Tracking for Distributed Parameter Control of Linear Infinite-dimensional Systems in Hilbert Space", Acta Automatica Sinica, 1(3), 92-96, 2014.

A Preflight Planner for Succesful Missions of Unmanned Aerial Vehicles

Carlo Di Benedetto, Domenico Pascarella, Gabriella Gigante, Salvatore Luongo, Angela Vozella
Integrated Software, Verification and Validation Laboratory
CIRA (Italian Aerospace Research Centre)
Capua, Italy
e-mail: c.dibenedetto@cira.it, d.pascarella@cira.it, g.gigante@cira.it, s.luongo@cira.it, a.vozella@cira.it

Francesco Martone
Software Development and Virtual Reality Lab
CIRA (Italian Aerospace Research Centre)
Capua, Italy
e-mail: f.martone@cira.it

***Abstract*—This paper investigates the flight planning for unmanned aerial vehicles. It proposes a prototype of preflight planner for different models of unmanned aircrafts. The planner is able to take into account several constraints (e.g., the vehicle dynamics, the no-fly zones, the endurance, the feasibility of the mission objectives, the terrain separation, etc.). It also provides a quantitative estimation of the air data link coverage and of the National Imagery Interpretability Rating Scale (NIIRS) index for the images quality. An overview of the prototype is reported and some significant test results are discussed in order to show its features.**

Keywords-UAV; flight planner; payload management; NIIRS.

I. INTRODUCTION

An Unmanned Aerial Vehicle (UAV) is an aircraft with no human pilot onboard. It is the central element of an Unmanned Aerial System (UAS), which is the set of the aircraft and all the other elements supporting its service. Recent advances in UAVs' technology allowed the emergence of a wide range of applications, such as military operations [1], disaster management [2], and urban terrain surveillance [3]. Without the need of an onboard pilot, a vehicle may be designed to accomplish the D-cube (dull, dangerous and dirty) missions [4]. Nowadays, UAVs are mostly Remotely Piloted Vehicles (RPVs) since their operations are performed by large teams of human operators, who remotely pilot the aircraft and control its actions. For RPVs, ground operators must be endowed with the proper expertise and this represents a substantial constraint, especially concerning costs. Dull missions particularly stress the training requirements. Nevertheless, tedious and repetitive tasks relating also to mission preflight operation (such as the design of the flight path) could relieve the remote pilots if they were autonomously performed and could provide a formal guarantee of the mission success.

Indeed, an integral part of UAV operation is the design of a flight path that attains the mission objectives. Flight planning shall ensure that the UAV operates in a safe and efficient way. Moreover, the mission effectiveness shall be ensured by verifying that all the required objectives are fulfilled by means of the designed route.

This work deals with an offline flight planner, named PreFlight Planner (PFP), wherein the mission objectives concern the proximal sensing of geographical targets. The PFP is a Java software prototype, which is in charge of the 4D flight planning for different samples of UAVs. The 4D flight planning problem is concerned with finding a path that links a specified initial state and several goal states. These states are four-dimensional (three spatial and one time dimension). It is also a constrained problem. Indeed, the proposed PFP is able to take into account various mission constraints for the planning, such as the vehicle dynamics, the no-fly zones, the endurance, the data link coverage, the feasibility of the mission objectives, the terrain separation, etc. The proposed software is an innovative UAV flight planner since it permits: a planning that is jointly based on the mission targets and the payloads; an integrated insertion of emergency and termination routes; the verification of the performances and the constraints for the achievability of the waypoints and the mission objectives.

In the following sections, the background, an overview of the prototype and some significant test results are discussed.

II. BACKGROUND

An UAV mission may be divided in two main parts: the flight and the fulfillment of the assigned objectives. Objectives are reached by means of onboard payloads. A typical UAV mission starts with the assignment of the objectives, goes on with the definition of the flight plan to reach them and the execution and control of the flight from take-off to landing, and it ends with the post flight analysis of collected data. All such phases are supported by different types of software, that may be categorized in:

1. **UAV Activities Management –** Software to manage the different activities of UAV fleets and related projects at business level, maintenance plans and pilots work. Different platforms providing such services are going to be developed in Europe.
2. **Flight Management –** Software allowing the execution of the flight from take-off to landing. Such class includes both Ground Control Station (GCS) software and onboard guidance, navigation and control software (autopilot). The autopilot works according to the flight plan and by means of sensing and actuating. The typical UAV ground control software receives telemetry data from UAV and sends telecommands to it. It allows the aircraft operator to communicate the flight plan to onboard autopilot and/or to remotely control the UAV.

It may support First-Person View (FPV) equipment to enhance the situational awareness of the remote pilot. In these fields, much research effort has been focusing on relevant aspects such as the perceptual and cognitive issues related to the interface of the UAV operator, including the application of multimodal technologies to compensate for the dearth of available sensory information. GCS software products usually allow to manage one UAV and they are combined to the UAV autopilot. For example, APM is the GCS of all UAVs with Ardupilot, a 3D robotics autopilot. Paparazzi GCS is the software employed in projects using the UAV Paparazzi platform [5]. It allows the design of the flight plan as well as the system configuration by means of a TCP-IP aircraft server. DJI provides a PC ground station for multi-rotor UAVs and manages the no-fly zones by means of a global list with a safety margin of 8 km [6]. The KopterTool is the ground software for the platform MikroKopter [7], whereas OpenPilot is an open platform [8]. Currently, it is possible to find commercial GCSs for multi-UAV systems ranging from the advanced proprietary and closed solution by Boeing for the X-45, Parrot SDK systems of PrecisionHawk, Draganfly, and Aeryon to open source solutions as QGroundControl Station and others [9]-[19].

3. **UAV Payload Management** – Software enabling the management of the onboard payloads during the flight. This class allows the fulfillment of the assigned mission objectives. Payload management products may be integrated into ground control software or not. They strictly depend on the payload model and type. The payload usually provides its own control software.
4. **UAV Post Flight Analysis** – Software producing evidences on the basis of data collected by the UAV during the flight. In the photogrammetry domain, companies such as Erdas or Inpho have been proposing solutions for UAV. APS from Menci Software has been one of the first platforms for UAV in Italy. It provides some additional functionalities, such as StereoCAD and Terrain Tools to elaborate the cartographic data, and APSCheck for the check of the UAV shoots. It also allows to validate and classify the collected data [20]. Pix4D from Pix4D Switzerland (a spin-off of Swiss university, born in 2011) provides Pix4Dmapper Capture App, which allows to display on tablets or smartphones the images from commercial UAVs, like the DJI Phantom. ENSOMosaic Suite and PIEneering ([21],[22]) offer different and integrated solutions from flight planning software to post flight photogrammetric analysis, including 3D models. The PhotoScan platform from Agisoft proposes the SFM (Structure For Motion) innovative approach. PhotoScan Professional and Standard Edition products are cheap and are open enough to accomplish the growing needs from applications [23]. Cloud services for UAV (like REDcatch GmbH [24], Agribotix [25], and the Maps Made Easy project [26]) may support UAV not only for planning, but especially for post flight elaboration of geo data. Additionally, a transversal category may be considered regarding the 3D modeling and vision digitalizing to realize 3D model and advanced visualization applications.
5. **UAV Flight Planning** – Software implementing: the strategic planning, which occurs before take-off and takes a priori information about the environment and the mission goals to construct an optimal path for the given objectives; the tactical planning, which involves re-evaluation of the flight plan during flight.

In this paper, we will refer to the strategic planning allowing the mission controller to plan (edit), validate and then upload the flight plan to the UAV. Research has focused on the identification of approaches and optimization algorithms obtaining the best route to guarantee the feasibility according to the vehicle performances, the compliance with the safety objectives, the endurance, the ability to return to base, and the terrain profile. Such software enables each UAV to properly flight followed by its own GCS, but two point seems to need further studies:

- to guarantee a successful mission, what about the flight plan and the clear sight of the targets associated to mission objectives?
- to guarantee the UAV flight according to airworthiness requirements, which ground station will cover the UAV?

A careful study of the market and of the existing products shows that very few products combine these aspects. The purpose of this work is to extend the capabilities of a UAV mission planner by proposing a solution of an offline flight plan validated against aspects related to mission objectives and data link coverage.

A. Images Quality Metrics

In any application where proximal sensing on a specific target is required, a variable that plays an important role is the quality of the set of pictures. Many image quality metrics have been proposed in the recent years [27]. The quality of images is expressed by several technical parameters, such as ground sampling distance (GSD), modulation transfer function (MTF), signal to noise ratio (SNR) and National Imagery Interpretability Rating Scale (NIIRS). However, these parameters may partially address interpretability. GSD is related to the spatial resolution of images and is probably the most popular parameter. This is not the ultimate parameter to describe quality of images. For example, images with a same GSD may have very different interpretability. MTF and SNR may specify some aspects of image quality. For this reason, the NIIRS index has been proposed as a measure of image quality in terms of interpretability criteria. It has been applied with multiple types of imagery and offers a robust approach to developing a scale. It was formerly defined for intelligence and military use and extended to civilian use later on. The general approach is to use image exploitation tasks to indicate the level of interpretability for imagery basing on the detection of the object. The scale is defined so that when more information may be extracted from the image, the NIIRS rating increases. A set of standard image exploitation tasks or “criteria” defines the levels of the scale. The NIIRS consists of 10 graduated levels (0 to 9), with several interpretation tasks or criteria forming each level. These criteria indicate the level of information that may be extracted from an image of a given interpretability level. All NIIRS rating levels are described in Table I.

Because of different types of imagery support different types of interpretation tasks, individual NIIRS indexes have been developed for four major imaging types: Visible, Radar, Infrared, and Multispectral. It provides a simple, yet powerful, tool for assessing and communicating image quality and sensor system requirements and it has been used for our purposes to provide a direct criterion to validate the waypoint and the relative legs associated to mission targets objectives. In literature, many tools to predict NIIRS have been proposed. This work addresses a possible approach for a quantitative assessment of the NIIRS index.

TABLE I. NIIRS LEVELS [28]

Rating Level	Description
0	Interpretability of the imagery is precluded by obscuration, degradation or very poor resolution.
1	It is possible to: distinguish between major land use classes; detect a medium-sized port facility; distinguish between runways and taxiways at a large airport; identify large area drainage patterns by type.
2	It is possible to: identify large fields; detect large buildings; identify major road patterns; detect ice-breaker tracks; detect the wake from large ships.
3	It is possible to: detect large area contour ploughing, individual houses in residential areas, trains or strings of rolling stock; identify inland waterways navigable by barges; distinguish between natural forest and orchards.
4	It is possible to: identify farm buildings as barns, silos or residences; detect basketball or tennis courts in urban areas; identify individual tracks, rail pairs and control towers; detect jeep trails through grassland.
5	It is possible to: identify individual rail wagons by type; detect open bay doors of storage buildings; identify tents at recreational camping areas; distinguish between coniferous and deciduous trees during leaf-off conditions; detect large animals in grasslands.
6	It is possible to: identify cars as saloon or estate types; identify individual electricity or telephone posts in residential areas; detect footpaths through barren areas; distinguish between grain crops and row crops.
7	It is possible to: identify individual railway sleepers; detect individual steps on a stairway; detect tree-stumps and rocks in forest clearings and meadows.
8	It is possible to: identify vehicle grille detailing and/or the license plate on a truck; identify individual water lilies on a pond; identify the windscreen wipers on a vehicle; count individual lambs.
9	It is possible to: identify individual barbs on a barbed-wire fence; detect individual grain heads on small grain crops; identify an ear tag on livestock.

III. THE UAV PREFLIGHT PLANNER

The PFP is a Java software prototype that allows to plan a mission of a UAS, namely, to identify the mission objectives and to design the mission path to observe them. Furthermore, the PFP ensures the success of the planned mission. The success assurance of the mission is attained by guaranteeing the following properties for the designed plan:

- the dynamic feasibility from a 4D point of view by means of the selected vehicle;
- the terrain separation;
- the compliance with the no-fly zones, i.e., the 3D regions that shall not be entered by the UAV;
- the compliance with the safe zones, i.e., the 3D regions that are reserved for the UAV flight and that shall not be left by the UAV;
- the endurance, which requires that the boarded fuel level is enough to accomplish the mission;
- the air data link coverage at any point of the route;
- the visibility of the targets at the related route points.

The preflight verification of these properties is necessary to avoid potential and expensive mission aborts due to neglected offline checks. In particular, the visibility check of the targets is profitable in order to avoid online changes of the UAV flight plan for the achievement of the mission objectives. In this way, the PFP provides a flight plan that is entirely verified and approved to guarantee the success of the designed mission.

A. Software Architecture

The PFP operation has been structured in three main phases: the setup phase, which allows for the configuration of all the mission parameters that are required for the planning; the planning phase, which is in charge of the route design by means of the waypoints positioning; analysis, which allows for the necessary checks in order to approve the designed plan.

The software structure of the PFP is split into five modules: User Database (setup phase); Mission Data (setup phase); Route Planner (planning phase); Analysis (analysis phase); Export. The data flow diagram of the PFP is shown in Figure 1.

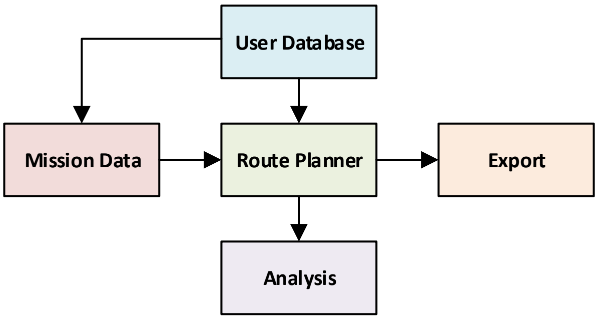


Figure 1. Data flow diagram of the PreFlight Planner.

The User Database is the module for the management of the database of objects that may be applied for different missions (e.g., vehicles, no-fly zones, safe areas, etc.). These objects may be defined and reused without modifications in order to simplify the operator throughout the generation of a mission plan. Some of the reusable entities are: aircrafts; airports; payloads that may be boarded; point targets, i.e., mission objectives without a significant size; area targets, i.e., mission objectives with a significant size; user waypoints, which are defined by the user; standard waypoints, which are standard aeronautic waypoints; air data links, i.e., the transmission/reception instruments that may be boarded; no-fly zones and safe zones; patterns, i.e., waypoint sequences that define remarkable route segments; contingency routes, i.e., standard routes that may be reused in case of failure to the air data link. The managed waypoints are compliant with the ARINC (Aeronautical Radio INCorporated) 424 standard.

The Mission Data carries out the management and the insertion of the set of data that characterize a given mission throughout the planning phase. The module is invoked both for the creation and for the change of a mission. It collects the following data from the user: the mission vehicle, payloads and air data links; the fuel level; the start time; the safe zone; the ground control stations that are active. The Mission Data receives the list of user entities from the User Database and sends its own data to the Route Planner.

The Route Planner is the module that accomplishes the planning phase. Moreover, it performs the following functions by means of the interaction with a 2D map: insertion of a new waypoint; change of a previously inserted waypoint; removal of a previously inserted waypoint. Each waypoint may be related to one or more targets, which shall be observable (i.e., shall exhibit a minimum specified NIIRS) along the route section between two consecutive waypoints. The user may request that a target is observable by means of one or more payloads within the set of boarded payloads.

Besides, every waypoint may be optionally related to one or two contingency routes, that shall be selected within the User Database. One contingency route may be defined as emergency route, whereas the other may represent a termination route: the former is the route to follow if the air data link is lost along the course starting from the chosen waypoint, while the system is waiting for the link recovery; the latter is the route to follow if the air data link is lost along the course starting from the chosen waypoint and it cannot be recovered. Hence, the match between a waypoint and the contingency routes is static.

During the insertion and the change/removal of the waypoints, the Route Planner executes some validity checks in order to ensure that the following conditions always hold: the vehicle is able to perform the necessary manoeuvres to reach the waypoints; there are no ground impacts. If one of these conditions is violated, the system does not agree to the modification of the route. The module also handles a 3D view of the Earth, that may be invoked anytime.

In more detail, the Route Planner carries out the computations of the flight plan for the specific aircraft. It employs the performance model of the aircraft in order to ensure the realistic and optimized route. The performance model includes some well-known characteristic parameters, such as cruise airspeed, climb rate, roll rate, etc. The route is modeled by means of a sequence of curves and the state of the vehicle may be analytically computer at any given time. Moreover, this module provides a software geometry engine that accurately illustrates dynamic objects.

The Analysis module is in charge of the analysis of the flight plans as a function of the mission objectives. It is examined in depth in the following section.

The Export module exports one or more planned missions in order to upload them in the Flight Management System (FMS) of the reference UAV. The interchange format is based on XML (eXtensible Markup Language) and has been implemented by a configurable XML schema.

B. *Flight Plan Verification*

The Analysis module verifies that all the mission constraints are fulfilled and ensures the success of the plan. In detail, the following properties are checked:

- the vehicle never leaves the coverage region of the air data links, which is computed by taking into account the positions of the GCSs and the land orography;
- the targets are always visible along the route sections, by taking into account the boarded payloads and the land orography and by envisaging a minimum level of quality of the captured image; if some variable confocal optics are boarded, the visibility check is carried out with four different focal lengths, namely, minimum, 1/3 of the maximum, 2/3 of the maximum and maximum;
- the vehicle never leaves the safe zone, if this is included in the mission planning;
- there are no ground impacts; a minimum distance with the terrain is guaranteed for each point of the route along vertical, frontal and lateral directions;
- the boarded fuel is enough for the accomplishment of the whole flight plan.

Furthermore, the coverage limit of the air data link is computed starting from the link budget equation, i.e.

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} \, , \qquad (1)$$

wherein P_{RX} is the power of the signal that arrives at the receiver, P_{TX} is the transmitted power, G_{TX} is the gain of the transmitter antenna, L_{TX} is the transmitter loss, L_{FS} is the loss due to the signal propagation in space, L_M is the safety link margin, and G_{RX} is the gain of the receiver antenna. All these parameters are known and are stored as data of the air data links in the User Database, except L_{FS}. The latter depends on the distance R that is covered by the wave and the wave length λ, which is derivable from the frequency of the transmission channel (also stored in the User Database). In detail, the relation between L_{FS}, R and λ is

$$L_{FS} = 20 \, \ln \tfrac{4 \, \pi \, R}{\lambda} P_{RX} \, . \qquad (2)$$

In order to receive a signal, the condition $P_{RX} > 0$ must hold. This condition is equivalent to

$$20 \, \ln \tfrac{4 \, \pi \, R}{\lambda} < P_{TX} + G_{TX} - L_{TX} - L_M + G_{RX} = \alpha \, . \qquad (3)$$

Hence, the maximum coverage distance R_{MAX} is

$$R_{MAX} = \frac{\lambda}{4\pi} e^{\frac{\alpha}{20}} \,. \tag{4}$$

As regards the NIIRS quantitative assessment, the first step is the computation of the GSD, which is the dimension of the ground projection of a sensor pixel. If we assume the pixels to be square with dimension d and the acquisition to occur with an elevation angle that is different from $\pi/2$, the ground projection of the pixel is distorted in a rectangle. Starting from Figure 2, the following equations hold

$$x = \frac{d \cdot r}{f} \,, \tag{5}$$

$$y = \frac{d \cdot r}{f \cdot \sin elev} \,, \tag{6}$$

$$\mathrm{GSD} = \sqrt{x \cdot y} = \frac{d \cdot r}{f \cdot \sqrt{\sin elev}} \,. \tag{7}$$

The expected NIIRS may be computed as

$$\mathrm{NIIRS} = A + B \cdot \log_{10}\mathrm{GSD}, \tag{8}$$

wherein A and B are two constants, whose values have been set as $A = 10.251$ and $B = -3.32$.

The structure of eq. (8) and the values of A and B are coherent with the General Image Quality Equation (GIQE). The GIQE is an empirical formula for calculating the image quality that is expected for a given optical system [29]. It is a model that was developed using statistical analysis of imagery analyst responses.

The coefficients A and B and the logarithmic structure were obtained by regression to fit the results of an image evaluation study. In detail, the logarithmic structure of eq. (8) embodies the notion that NIIRS changes by 1.0 each factor of two in the spatial resolution is equivalent to one unit on the NIIRS scale, namely, a change of ±1 of the NIIRS is equivalent to halving or doubling the distance between the sensor and the observation point. This relationship was confirmed by visual observations [29].

More broadly, the GIQE predicts the NIIRS value as a function of other parameters in addition to the GSD (which is directly related to the spatial resolution). These supplementary parameters are: the Relative Edge Response (RER), that is indirectly associated to the point spread function and that estimates the effective slope of the imaging system's edge response; the SNR and the system post-processing noise gain, which quantify the noise in the post-processed imagery; the system post-processing edge overshoot factor, that measures the amount of edge ringing resulting from post-processing. Within this work, we consider only the spatial resolution (i.e., the GSD) as a parameter for the NIIRS estimation, whereas the other criteria are not considered since they are related to the post-processing phase and the aperture configuration.

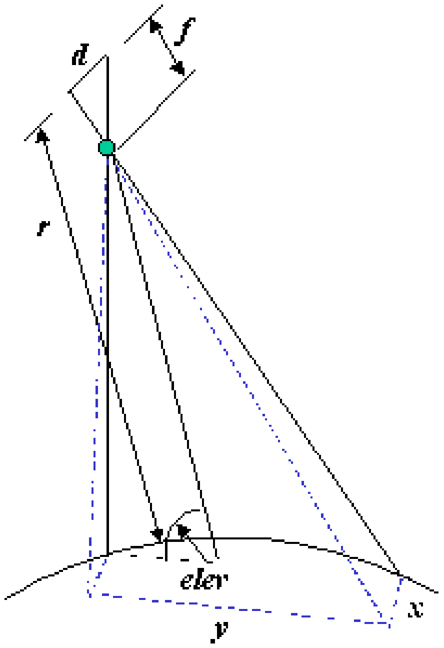


Figure 2. NIIRS quantitative estimation in the PreFlight Planner.

C. *Test Results*

We have conducted a series of tests to verify the correct implementation of the software. The main entities have been tested by creating, modifying and deleting records in different databases and also checking their correct visualization during the planning process. The verification of the analysis has required the creation of a number of flight plans to test the software behavior on different situations. In the following, two test cases are reported.

The first test and the related check results are depicted in Figure 3. The flight takes place in a segregated area (the azure line), the route (the yellow line) consists of eight waypoints, three of which are loiter. The no-fly zone is reported in red. There is a single GCS, but the link coverage is not visible because the area of operations is much less extensive. Two targets are associated to loiter waypoints. As shown by the right side of Figure 3, the flight plan validation fails on two aspects: the targets visibility and the boundaries overcome of segregated flight zone. The PFP analysis module is able to provide other graphic evidences: the non compliance with safety objectives, the issues on target visibility (highlighted red path) and the report on the fuel consumption.

In the second test, the flight plan of the first test has been modified in order to violate the data link coverage, the fuel consumption and the terrain obstacles on a linear target. The outcomes of the analysis are shown in Figure 4, which provides: the evidence that the flight plan is not feasible due to the overcoming of all the considered constraints; finally, the evidence of the link coverage analysis, the problems of visibility on the linear target (a river).

It may be noted that the previous test cases have been discussed in order to highlight the verification and the analysis capabilities of the PFP. Indeed, the checking phase of the PFP is able to formally verify the compliance of the computed flight plan with all the reference constraints and to guarantee the success of the designed mission. However, some of these constraints are previously taken into account by the Route Planner, which processes the actual flight plan in order to reach the prescribed waypoints by means of the selected aircraft (i.e., the related dynamic model). Clearly, the other constraints are not considered in the planning phase since they do not directly involve the trajectory elaboration. Thus, they may be only evaluated by means of the PFP checks.

IV. Conclusion and Future Work

This work proposes some new perspectives on UAV preflight panning by pursuing the idea that a flight plan should not only guarantee a successful flight, but also a successful mission. It analyses the typical UAV missions where proximal sensing is requested and their main requirements. Here, the quality of images is a critical aspect and an approach for its measurement is implemented in the PFP as a criterion to validate the flight plan.

In addition, the verification of data link coverage encourages future enhancements by considering a fleet of UAVs with different GCSs. Other possible improvements cannot overlook the research issues concerning the waypoints scheduling optimization.

References

[1] C. Schumacher, P. R. Chandler, M. Pachter, and L. S. Pachter, “Optimization of Air Vehicles Operations Using Mixed-Integer Linear Programming”, CMU/SEI-95-TR-021 ESC-TR-95-021, Air Force Research Laboratory, 2006.

[2] M. Quaritsch, “Agent-oriented programming”, Elektrotechnik & Informationstechnik, vol. 127, no. 3, 2010, pp. 56–63.

[3] D. Gross, S. Rasmussen, P. Chandler, and Greg Feitshans. “Cooperative operations in urban terrain (COUNTER)”, Proceedings of Society of Photo-Optical Instrumentation Engineers Conference, Vol. 6249, 2006, pp. 1-11.

[4] L. A., Ingham, “Considerations for a roadmap for the operations of Unmanned Aerial Vehicles (UAV) in South African Airspace”, PhD thesis, Electrical and Electronic Engineering, Universiteit Stellenbosch University, 2008.

[5] P. Brisset, A. Drouin, M. Gorraz, P. S. Huard, and J. Tyler, “The Paparazzi Solution”, Proceedings of MAV (Micro Air Vehicles) 2006, pp. xxxx.

[6] DJI, Fly Safe, [Online], Available from http://www.dji.com/fly-safe [retrieved: 05, 2016].

[7] MikroKopter, MikroKopter Tool, [Online], Available from http://www.mikrokopter.de/ucwiki/MikroKopterTool [retrieved: 05, 2016].

[8] OpenPilot, [Online], Available from https://www.openpilot.org [retrieved: 05, 2016].

[9] Easy Map UAV, Specification, [Online], Available from http://www.easymapuav.com/specification [retrieved: 05, 2016].

[10] APM, Mission Planner Home, [Online], Available from http://planner.ardupilot.com [retrieved: 05, 2016].

[11] 3DR, Free Groun Station Application, [Online], Available from http://3dr.com/download_software [retrieved: 05, 2016].

[12] mdCockpit, UAV Control, [Online], Available from http://www.microdrones.com/en/products/software/mdcockpit/flight-planning [retrieved: 05, 2016].

[13] UAV Navigation, Visionair GCS Software, [Online], Available from http://www.uavnavigation.com/products/visionair-ground-control-station-software [retrieved: 05, 2016].

[14] UAV-EA, Mission Planner Autopilot Software, [Online], Available from http://uaveastafrica.wordpress.com/mission-planner-autopilot-software [retrieved: 05, 2016].

[15] Orbit Logic, UAV Planner, [Online], Available from http://www.orbitlogic.com/products/uav.php [retrieved: 05, 2016].

[16] MAVinci, MAVinci Desktop, [Online], Available from http://www.mavinci.de/en/completesys/desktop [retrieved: 05, 2016].

[17] FAS, Tactical Control Station (TCS), [Online], Available from http://fas.org/irp/program/collect/uav_tcs.htm [retrieved: 05, 2016].

[18] Micropilot, Micropilot Ground Control Station, [Online], Available from http://www.micropilot.com [retrieved: 05, 2016].

[19] SAGEM, SAGEM Mission Planning Systems, [Online], Available from http://www.sagem.com/aerospace/military-aircraft/mission-planning-systems [retrieved: 05, 2016].

[20] Menci Software, Photogrammetry Software, [Online], Available from http://www.menci.com [retrieved: 05, 2016].

[21] MoasicMill, EnsoMOSAIC photogrammetry software and hardware for aerial image processing, [Online], Available from http://www.ensomosaic.com [retrieved: 05, 2016].

[22] PIEnnering, Parallel Image Engineering, [Online], Available from http://www.pieneering.fi [retrieved: 05, 2016].

[23] Agisoft, Ahisoft PhotoScan, [Online], Available from http://www.agisoft.com [retrieved: 05, 2016].

[24] REDcatch, Photogrammetric Cloud Service, [Online], Available from http://www.redcatch.it [retrieved: 05, 2016].

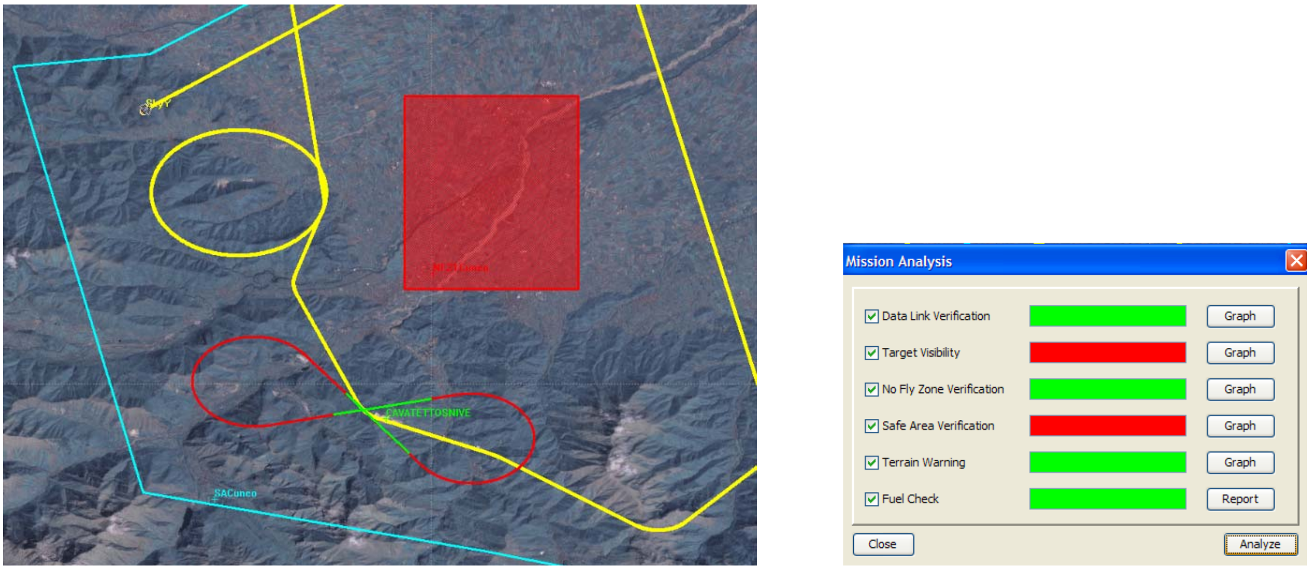
[25] Agribotix, Agricultural Intelligence Drone-Enabled, [Online], Available from http://www.agribotix.com [retrieved: 05, 2016].

[26] Maps Made Easy, Aerial Map Processing & Hosting, [Online], Available from http://www.mapsmadeeasy.com [retrieved: 05, 2016].

[27] R. Reulke, and A. Eckardt, “Image quality and image resolution”, In 2013 Seventh International Conference on Sensing Technology (ICST), 2013, pp. 682-68.

[28] The National Imagery Interpretability Rating Scale (NIIRS), [Online], Available from http://ncap.org.uk/sites/default/files/NIIRS.pdf [retrieved: 05, 2016].

[29] Thurman, S. T., and Fienup, J. R., “Analysis of the general image quality equation”, Proceedings of SPIE, Vol. 6978, Visual Information Processing XVII, 2008.



Mission Analysis

☑ Data Link Verification 🟩 Graph

☑ Target Visibility 🟥 Graph

☑ No Fly Zone Verification 🟩 Graph

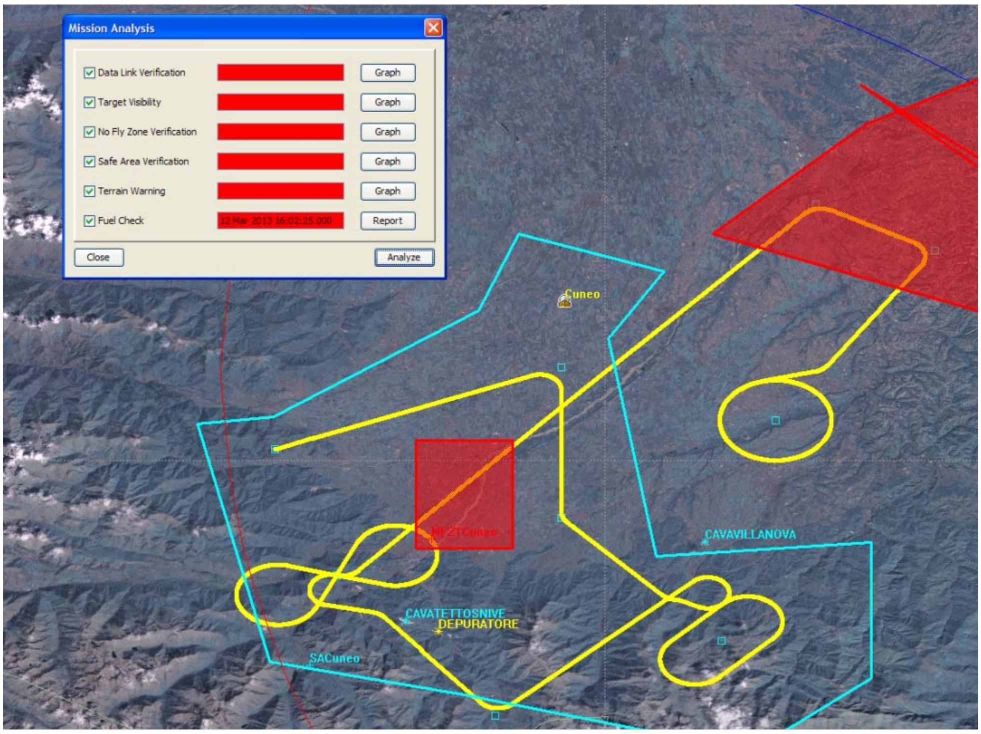
☑ Safe Area Verification 🟥 Graph

☑ Terrain Warning 🟩 Graph

☑ Fuel Check 🟩 Report

Close Analyze

Figure 3. Results of the first test.



Mission Analysis

☑ Data Link Verification 🟥 Graph

☑ Target Visibility 🟥 Graph

☑ No Fly Zone Verification 🟥 Graph

☑ Safe Area Verification 🟥 Graph

☑ Terrain Warning 🟥 Graph

☑ Fuel Check 12 Mar 2013 16:02:25.000 Report

Close Analyze

Figure 4. Results of the second test.

ISBN: 978-1-61208-483-1

Threat Evaluation Based on Automatic Sensor Signal Characterisation and Anomaly Detection

Anatolij Bezemskij, Richard John Anthony, Diane Gan, George Loukas
Department of Computing & Information Systems
University Of Greenwich
Email: {a.bezemskij, r.j.anthony, d.gan, g.loukas}@gre.ac.uk

Abstract—**Autonomous cyber physical systems are increasingly common in a wide variety of application domains, with a correspondingly wide range of functionalities and types of sensing and actuation. At the same time, the variety and frequency of cyber attacks is increasing in correspondence with the increasing popularity and functionality of these systems, from in-vehicle driver assistance to smart city infrastructure and robotics. These technologies rely on a variety of sensors, actuating nodes and control communications. Each sensor adds context by which the autonomous system can better understand its environment, but each sensor also provides opportunities for attack, as has been observed in a variety of attacks on different systems. In this paper, we introduce a model to observe signal characteristics, including noise level patterns, on sensor data streams and incorporate this information to differentiate between normal or abnormal behaviour of a robotic vehicle. This model forms the basis of an automated threat detection scheme, which we test using a purpose-built testbed. Experiments are conducted in a controlled environment using stochastic elements to introduce certain levels of randomness during the experiment. The results indicate that the system is able to distinguish the behaviour of a robotic vehicle under different levels of environmental volatility and is able to identify a sensory channel attack against it.**

Keywords–Anomaly detection; Autonomous behaviour; Threat; Cyber-Security; Signature.

I. Introduction

Detection of cyber threats is an expanding area of study in the embedded systems domain. The need for cyber security has increased significantly and there are many researchers currently working towards cyber-physical security of such systems, such as the decision tree-based approach in [1] using decision trees for anomaly detection, and the behaviour rule specification in [2]. In this paper, we evaluate our robotic testbed system behaviour by monitoring components with instrumentation installed on the system.

Several different attack vectors can apply to cyber-physical systems. We divide these into cyber-physical and physical-cyber. Cyber-physical attacks are attacks in cyberspace that adversely affect the physical space. For instance, an attacker can target the communication between the system and the operator to disrupt normal system operation. In an autonomous system, a system's own autonomy can be used against it to take over control over the autonomous system. Conversely, physical-cyber attacks are the ones performed in physical space to adversely affect cyberspace [3]. A trivial example would be physical damage that would make the network unavailable. A non-trivial example would be an attack consisting of custom laser beams targeting an autonomous vehicle's LiDAR [4], or externally generated noise targeting ultrasonic sensors so as to confuse the vehicle's spatial awareness. Such attacks that manipulate the input to sensor systems with the purpose to affect the operation of a system that depends on them are often referred to as sensory channel attacks.

Previously, there had been little or no consideration for cyber security during the design of safety-critical systems, but this is changing since the practical cyber-physical attacks against vehicles were showcased for the first time a few years ago [5][6]. Ten years ago, the threat level was significantly lower, but now with the availability of electronic devices such as Arduino kits, a variety of sensors that can be used for educational purposes, consumer products and industrial applications are wide spread. With increasing knowledge in this area the threat to such systems increases. An attacker may not necessarily have the intention of disrupting the system; motives can vary and the outcomes can range from small value fluctuations to possible lethal injuries [7]. This shows that there is a need to secure cyber-physical systems.

In this paper, we focus on robotic vehicles, but we believe that our model can be extended for use in unmanned aerial vehicles, other cyber-physical systems where erratic sensing can be the target or an indicator of a sensory channel attack. In Section I-A, a reader will find the discussion on the current state of a research in the cyber-security domain for robotic vehicles. Later in Section II robotic vehicle testbed design is discussed in detail covering its functionality, design specifics and the experiment environment discussed in Section III. The behaviour profile that we use in our methodology is discussed in Section IV, explaining how sensor unique characteristics are formed for the behavioural profile and its format. The methodology itself is explained in Section V using readings from a single data source during an attack. Overall (using all data sources) the robotic testbed methodology performance is discussed in Section VI, followed by the methodology evaluation and conclusion in Section VII.

A. Related Work

Previous research in cyber attack resilience for such systems, has focussed on detection using a variety of techniques such as anomaly detection based on rule specification[7] where state is being defined using pre-defined system functionality. The approach by Vuong et al. [1] shows that it is highly beneficial to monitor not only cyber but also physical metrics to identify cyber attacks [8], for instance to reduce the false positive rate of detection [9]. Various voting algorithms [2] where system nodes are interacting with each other to identify an attack based on behaviour rule specifications have also been proposed. A similar approach has been used by [10] where robotic multi-agents have a reputation based on their observations and try to reach consensus regarding misbehaving robotic agents. Most researchers agree that a cyber-physical system's security has to be improved at the design stage, and for this reason propose the use of more secure communication [11] or the integration of gateway firewalls [6].

Another point of view is to evaluate mission success threats based on the risk of a failure. For instance, Orojloo et al.

have developed a method for evaluation of the security of cyber-physical systems [12] by evaluating the mean time to system security failure with regards to system components and different types of cyber-attacks. Majed et al. [13] have proposed a framework for evaluating cyber threat exposure for energy smart-grids by using attack trees and attack-graphs. A variety of reliability [14] and survivability [15] models have also been proposed for cyber threat evaluation.

Yampolskiy et al. proposed a language describing attacks on cyber-physical systems [16]. This language would enable the impact of certain attacks applicable to specific systems to be described. When it comes to threat analysis, there is little research done on quantification of threats. One example from Sandia National Laboratories [17] uses a threat driven approach for cyber security evaluation of organisations. Some aspects of their findings can be taken into account when a cyber-physical system is evaluated. The majority of these approaches and frameworks take into account an attack based on methods, conditions and impacts.

In most research presented above, researchers have taken into account attack characteristics as input to identify anomalous behaviour. In other words, the type of attack is predefined. Our view is that this limits the practicality and likely effectiveness of a protection mechanism to attacks that have already occurred and are known to the system at hand. Here, we attempt to detect attacks on which we have not already trained our system. Our proposal is to monitor sensor noise data accompanied with a system's knowledge about itself, as input for anomaly detection and to evaluate a possible threat to the system. Such approach will treat attacks as generic entities, therefore introducing dynamic anomaly detection approach, which will continue being applicable as new attack vectors are introduced and the sophistication (and/or number) of attacks increases over time.

II. Robotic Testbed System Design

To facilitate detailed investigation of autonomous techniques to detect cyber-physical attacks, we have built a richly-instrumented robotic vehicle testbed which is shown in Figure 3, with a variety of different sensor types. The control system of the testbed comprises an integrated set of modular embedded systems. It uses a variety of communication protocols that are used in the industry, such as CAN, RS-485, WiFi and ZigBee. This system was intentionally built integrating technologies that are used by the industry so as to be representative of a large subset of deployed systems. We conduct a variety of real-world relevant experiments and evaluate the system within the cyber security domain.

TABLE I. ROBOTIC TESTBED INSTALLED EQUIPMENT

Feature	Purpose
CAN bus	Internal communication
ZigBee	External communication
WiFi	Media streaming
Compass Bearing	Navigation correction
DC Motors	Movement
Ultrasonic Rangers	Collision avoidance

System components produce signals and feedback that is used by other system components to change overall system behaviour. Several components that are mentioned in Table I produce instrumentation data which is used as cyber or physical domain indicators. The combination of such indicators can produce additional meta-data that can be used to identify a particular behaviour of a system, as we describe later. All sensor data is generalised and is treated as a data source. Processing is distributed across the various embedded processors on the testbed platform.

One type of processing node is an AVR-CAN development board, several of which are used to host specific sensors. If such a node is responsible for navigational tasks, the node will listen for data sources with related data and act appropriately. A variety of sensing or actuating components share their processing node. For instance, a single node is responsible for processing bearing, pitch and roll sensors. Overall the system contains six processing nodes, five of which are AVR-CAN development boards clocked at 16 MHz, and one STK300 Kanda board powered by Atmel ATMega1281 chip clocked at 8 MHz.

System components allow the robotic vehicle testbed to undertake a variety of autonomic tasks, such as navigation based on the logical mission layer that represents a sequence of steps given to the testbed. Sensors allow the vehicle to navigate autonomously in an environment using the compass bearing to keep track of the direction, ultrasonic rangers for collision detection and avoidance, and pitch and roll sensors to make direction corrections and inform the system of environment volatility. Also, the system uses an informative meta-data sensor that measures the temperature of the heat sink connected to the on-board voltage regulators which supply power for the camera and robotic arm. In this way, the system is able to determine if these heavy-current-drawing system components are in use. These sensors and additional meta-data extraction allow automatic characterisation of the behaviour of a robotic testbed vehicle whilst in operation.

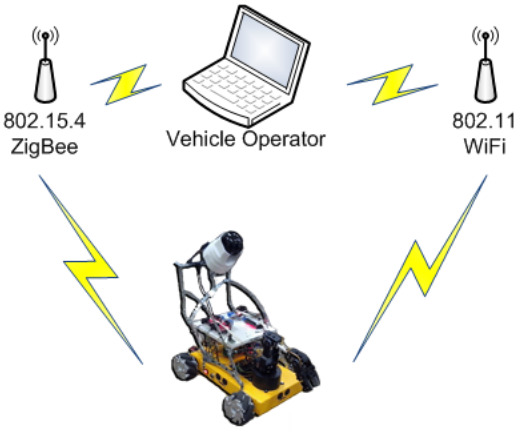


Figure 1. High-level communication

To gather the data for off-line analysis, we use an external workstation. Sensor data from the testbed is collected and stored in a knowledge base. Communication between the workstation and the robotic testbed vehicle is achieved using a dedicated ZigBee network. The ZigBee connection also enables us to transmit commands to the testbed (e.g. to initiate missions). The camera is a self-contained unit; its audio and video feeds are streamed using a standard WiFi protocol. An overview of high-level communication architecture between workstation and robotic testbed vehicle can be seen in Figure 1.

A variety of commands can be sent to the robotic vehicle as simple navigation commands, camera or robotic arm control commands. Additionally, the vehicle supports complex mission task uploads. The command transmission is one-way communication functionality; commands are only executed if they are received from verified ZigBee network nodes and the command is in the correct format. The robotic vehicle testbed does not send any commands to any external nodes within the ZigBee network. The testbed will only periodically report its instrumentation data to a verified connected workstation. The instrumentation report periodicity is one second, due to the low bandwidth ZigBee protocol and unique ZigBee ZE10 module behaviour. Therefore higher-rate sample aggregation is performed on-platform on the sensor hosting nodes.

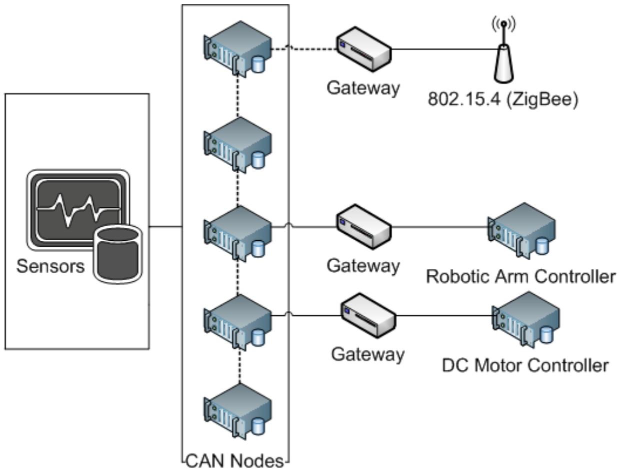


Figure 2. Internal Communication: gateways connect different subsystems

For communication between system components, the testbed uses a CAN bus. This bus is used to share overall sensor data from data sources, including additional metadata extracted during data analysis by the processing nodes. Internal communication architecture is shown in Figure 2. This data is retransmitted to other nodes through gateways and is collected at the reporting node which will transmit data to the workstation when appropriate.

The software structure of the robotic vehicle testbed uses a layered architecture, which separates the different levels of reasoning from the lowest physical sensor level, represented by individual embedded nodes performing analog to digital conversions interpreting signals into an understandable software language. The next-higher level is the classification layer where data is analysed using statistical analysis approaches, such as exponential smoothing to determine the trends in the data. A level higher, we have an autonomous module controller layer which controls actuating capabilities based on the data received from the lower layers of the model. The autonomic module controller layer is a set of autonomic controllers that are carrying out their defined tasks, such as robotic arm movement or navigational control. A mission layer then collects knowledge from autonomic controllers and evaluates if the expected mission goal has been achieved. The layered software approach improves flexibility and maintainability in terms of a robotic vehicle testbed programming, as all these layers are implemented as a set of libraries that can be extended further.

In a real-world environment, there are a variety of physical threats to the system that can be caused by unknown factors, such as rain which affects the grip on a road, windy weather that can affect vehicle movement etc. An issue can arise when an attacker targets a specific sensor to disrupt its activity during the learning process, as it will affect overall operation of the vehicle at later operational stages. One of the examples would be to disrupt a compass intentionally during learning so that the robotic vehicle will learn the disrupted pattern as being ’normal’. We eliminate this risk by securing our vehicle from attacks during the learning process.

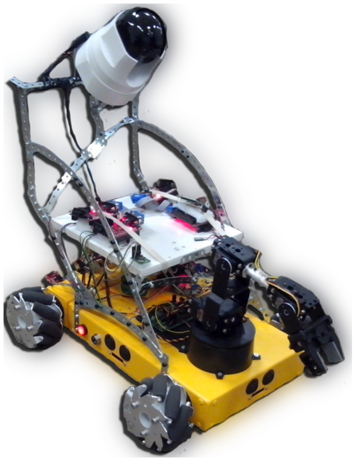


Figure 3. Robotic vehicle testbed

To summarise, our robotic testbed vehicle has been designed to facilitate a variety of experiments targeting different data sources and identifying behavioural abnormalities. Our goal is to develop a methodology that will improve robustness of autonomous vehicles using a sensor-agnostic learning approach where the type of data source does not matter, as the requirement is to learn the “normal” signal characteristics, including noise characteristics, generated by the data sources. This robotic vehicle testbed has been built to conduct experiments for a variety of navigational tasks combined with robotic arm actuation. Additional sensors can be added to extend evaluation of the behavioural model.

III. Experiment Environment

Initial experiments that were used for behaviour definition are conducted in the Queen Mary Building at Greenwich University. The environment provides an area with stochastic elements for the data sources, such as old uneven stone flooring with an irregular surface as can be seen in Figure 4. The space is a controlled environment that will not change overtime. The flooring has a variety of dents and lumps that affect the testbed movement throughout the experiment and introduce a stochastic randomness that is used to learn normal deviations. The distance between walls is constant. This allows us to identify the behavioural profile of an environment based on the data source information. The corridor has a set of inset door openings on either side which allows observation of periodic behaviour in the ultrasonic distance sensor signals as the vehicle passes by.

The corridor is 28 m long and the distance from wall to wall is 2 m. The experiments were repeated five times to ensure that the collected data set is representative and these

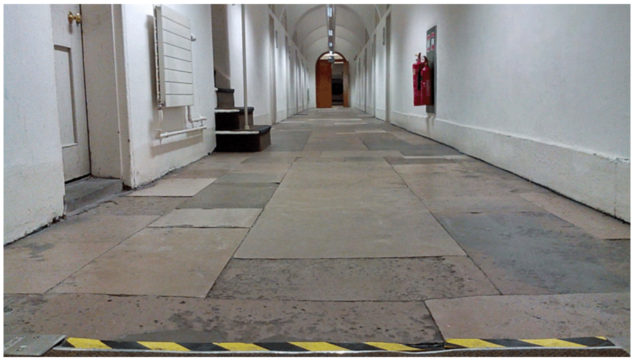


Figure 4. Experiment environment

were used for the creation of the behavioural profile. Two further experiments were used for evaluation of the behavioural profile. The behavioural profile is built using patterns of the variation and background noise in data sources; mainly we are looking at the spikiness of the data variations and the variety of deviations. The experimental environment facilitates repeatability and contains static elements that can be used as guideline features during analysis of gained data, but it also introduces significant stochastic elements which are essential for understanding the normal levels of noise and variability in sensor signals.

The experimental scenario evaluated in this paper is a mission in which the robotic vehicle testbed has to reach the end of the corridor using its own sensing capabilities. The complexity of such a mission is not obvious. The uniqueness of the flooring surface disrupts direction of the vehicle, forcing it to continuously adapt the speed of its motors and its direction and ensure that it maintains a safe distance from the walls during operation. The scenario was chosen due to the structural uniqueness of the vehicle, and as such the scenario exercises all sensor capabilities. The experiment is organised in two steps. The first is a training step, where over several runs we collect a learning data set that will allow us to create a "normal" behavioural profile. The second step is being conducted to evaluate the recognition of "normal" behaviour profile, as well as we evaluate the representational normality value of a signature that was obtained during the learning step. This value will be used to monitor normality at the higher level observation, if an anomalous representational value has been identified, the system will examine the amount of anomalies and relationship between them, thus reducing computational power requirement of the system.

IV. Normal Behaviour Definition

Our behavioural model uses a sensor-independent approach, in the sense that the sensor-signal characterisation is performed without any additional contextual information to indicate the type of the sensor. Each different type of sensor has its unique output, but we are not interested in determining the type of the sensor, but instead we are interested in learning the signal characteristics under normal operating conditions and thus being able to automatically determine when an anomalous condition occurs by monitoring data source signature.

A compass sensor provides a valuable example: due to the limited speed at which the vehicle can turn, there is a corresponding limit to the rate at which the compass bearing can be expected to change. The compass bearing will also contain a certain amount of noise as the vehicle travels over non-perfect surfaces and does not track in a perfect straight line (there is a detectable "wobble" of typically one to two degrees). These characteristics can be learnt by examining the signal over a series of test missions, without having to explicitly know that the sensor is a compass. For simplicity, we demonstrate the impact on the compass of a cyber-physical disruption using a magnet-based sensory channel attack. By placing a magnet in the vicinity of the sensor, we cause a variable disruption of the vehicle's navigational ability. By so doing, the data stream from the sensor is affected in two detectable ways. Firstly the sharp change in bearing when the magnet is applied (or removed), and secondly, in the reduced noise levels since the magnet causes the sensor to read near-static values (which are anomalous because they are suspiciously "clean"). The proposed approach enables attack detection without prior knowledge of the attack type. The compass example is a part of the experimental set used in our evaluation.

We represent the characteristics of sensor signals in a signature format that can be used to compare expected and actual behaviour in order to detect anomalous events. The signature comprises a number of metrics whose values are learnt during the mission experiments described earlier. The metrics describe characteristics such as the signal-to-noise levels, maximum and minimum sensor readings detected, size and frequency of spike values and rate of change of sensor values. The signature approach facilitates evaluation of the enviroconsistency of a particular trace. For example, the system may learn that a particular data source generates data values distributed in the range 100 to 400 with a mean of 200 during normal operation. The new trace can be compared against the expected behaviour based on these specific characteristics. There is no need to compare the raw data directly. The model will determine whether a particular trace represents normal or abnormal behaviour based on the distance between the trace characteristics and the corresponding values in the signature.

TABLE II. SIGNATURE CHARACTERISTICS

Value Type	Characteristic
Raw	Minimum Maximum
Exponential Smoothing	Minimum Maximum Lowest Difference Highest Difference
Deviation	Standard Deviation
Spike Areas	50% - 100% 100% - 150% 150% - 200% Over 200%

Our signature format contains various characteristics as shown in Table II. Values are exponentially smoothed to provide a basis for comparing instantaneous values with the recent trend, thus detecting noise levels and abrupt changes in values which are short lived are categorised as spikes. Such concept has been used in a dynamic system in [18].

V. Identification of anomalous signals and behaviour

The signatures are constructed during the learning stage to define normal behaviour on a per-sensor signal stream basis. To capture the range of normal behaviour the experiments were repeated five times to identify the domain of values where the data sources operate and their normal deviations. Using such an approach it is possible to classify normality when the system operates within the normal experiment environment. Currently, we evaluate the results of test runs off-line after each run, however the learnt-signature based approach has the potential to be used in real-time, for self-protection of the autonomous vehicle. In this publication, we review multiple results from seven experiments and different scenarios which are: learning stage, evaluation stage, and physical-cyber compass attack scenario.

To smooth data we are using exponential smoothing in our model as it enables dynamic smoothing to be more reactive or passive by changing the α value. It is a simple and efficient means by which to follow an unfolding trend in sample values. The technique if very efficient in regards to memory and processing and so is well suited for use in embedded systems. Each element of a signature comprises of characteristics that may indicate an anomaly. An operational signature is applied to a learnt "normal" signature, and this facilitates observation of a data source anomalous behaviour and reasoning about component behaviour at the higher layers of our software stack and evaluate the deviations from normality. By observing deviation coefficients (from the learnt normal characteristics), we form a dynamic behaviour score for the data source.

The behavioural score (from the signature) can be used to evaluate the level of threat to the system i.e. the higher the deviation from the learnt normality, the higher the likelihood of an attack. Table III shows an example analysis of data from a compass bearing sensor. By comparing signature elements we identify those elements which indicate that an attack might be present. The deviation extents are weighted and combined to determine the likelihood of an actual attack, i.e. co-deviation on multiple elements reinforces the attack risk.

TABLE III. COMPASS BEARING BEHAVIOUR SIGNATURE DATA

		Deviation Coefficient		
Characteristic	Value	Learnt	Test	Attack
Min	167.8	0.0292	0.0148	0.4388(A)
Max	194.7	0.0151	0.0128	0.3180(A)
Exp. Min.	170.9	0.0145	0.0035	0.0995(A)
Exp. Max.	188.9	0.0083	0.0171(A)	0.3269(A)
Exp. Diff. Min.	0.0	0.0000	0.0000	0.0000
Exp. Diff. Max.	10.1	0.2362	0.1289	>1.0000(A)
Std. Deviation.	5.3	0.1094	0.0582	>1.0000(A)
Spikes >50%	55	0.2435	0.3043(A)	0.9348(A)
Spikes >100%	25	0.5632	0.4079	0.8684(A)
Spikes >150%	10	0.5576	0.5455	0.8485(A)
Spikes >200%	5	0.4462	0.2308	0.6154(A)
Threat	**Summary**	2.2231	1.7237	17.921

Table III shows the deviations from a variety of data sets which are a **Learning scenario**, **Test scenario** and **Compass bearing attack scenario**. By learning we mean that the vehicle is operated in a series of known missions which exercise the sensor signals across their normal value ranges. For example, following the earlier discussion concerning the compass sensor, the vehicle can be operated moving over various types of surfaces to determine the levels of noise in the compass sensor signal, and also can be made to turn at various angular rates-of-change in compass sensor values.

By test scenario, we mean that the vehicle is operated (post learning) in a variety of normal scenarios, with the objective of testing the vehicle's ability to detect the abnormalities solely on the basis of finding anomalous conditions where the sensor signals do not conform to expected learnt behaviour.

By attack scenario, we mean that the vehicle is operated using the same conditions as in the learning and test scenarios, but during the experiment we place the magnet near the compass sensor for forty seconds. This is to validate the behavioural profile approach and determine if the vehicle is able to identify anomalous conditions.

To calculate the deviation coefficient reference for each element in the signature, we have used results of five learning stage experiments. The data set is normalised in the following way, firstly for each signature characteristic the deviation of its current value from the mean is calculated. This deviation is then divided by the mean value of that corresponding signature characteristic, the result is an absolute value. Then the knowledge base is updated with the highest deviations from all learning stage experiment data sets and forms the "Learnt" knowledge which will be used as a reference for anomaly detection. Test scenario is then compared to evaluate the normality behavioural profile and identify the quantity of allowed anomalies. To demonstrate the ability to identify anomalous behaviour the data set of an "Attack" scenario was used.

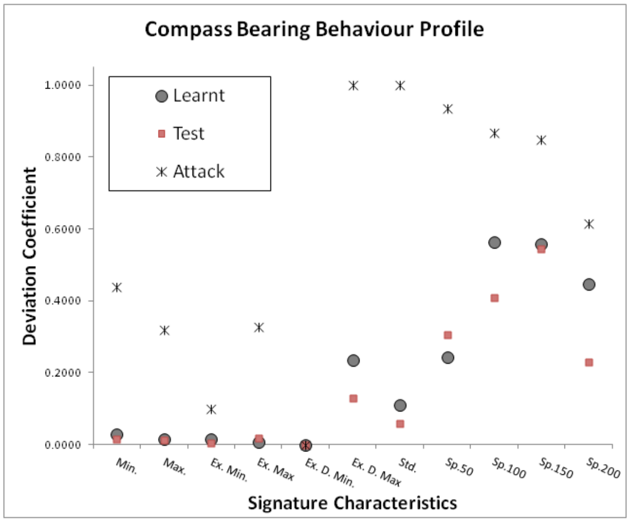


Figure 5. Compass bearing behaviour analysis

We detect an anomaly in terms of sensor signal values as a situation where the signal deviates significantly from expected (learnt) mean behaviour, as held in the particular sensor's signature. We investigate the automated detection of anomalies, based on our signature approach using, initially a single sensor. The corresponding data values are shown in Table III for one data source using two different scenarios. The learnt behaviour signature is based on running the identical normal scenario experiment five times. The absolute value of the registered maximums of all learning data sets is used as the anomaly limit. To identify the number of acceptable anomalies we have used our test scenario experiment runs. The data set from these runs is evaluated in regards to the "Learnt" knowledge to identify the amount of anomalies that exceed the learnt threshold. This procedure has shown that two anomalies have been observed and the threat summary score has not exceed the learnt score, therefore these anomalies were classified as acceptable.

The number of anomalies identifiable from an attack scenario data are shown (A) in the Table III. The summary score (behavioural score) is a sum of all deviation coefficients of signature characteristics, and is used as an indicator of a threat to the system. Such score explains the deviation of the data source saying; the higher the score, the higher the deviation, therefore the higher threat risk to the system. This is also shown graphically in the Figure 5.

Initially all signature characteristics have equal weights, and due to the weighing scheme we use, signature elements are significantly out of line with expected values when an attack occurs and so the robot is able to autonomously identify an attack based on the detection of anomalies in the sensor data using our methodology. At a level higher, if we will take into account other data sources we can form a system behavioural profile and identify if the system is exhibiting normal or abnormal behaviour.

VI. Evaluation

Earlier, we have demonstrated how a single data source is analyzed producing the behavioural score that can be used at the level higher for surface analysis of the data source. If the behavioural score is exceeding the normality score, the system will investigate the lower layer and will identify what is the cause of such a high score. All system data source signatures operate in the same data domain allowing the system to produce a behavioural score by combining these signatures together.

In this publication, we have reviewed a single data source from multiple experiment scenarios. At the level higher observation, the system uses the overall behavioural score produced by all available data sources. Such approach can decrease the computational power requirement, however potentially a situation can arise where multiple signature characteristic readings are abnormal, but cancel each other out. Leading to a threat summary score which does not indicate a threat. This could mask an actual threat. This can be avoided by an occasional low-level analysis and generating an interrupt-based procedures when anomaly has been identified within the signature.

As for the system's final evaluation, it combines the threat summary scores of all data sources to classify the behaviour profile of a vehicle. The system has access to 17 instrumentation channels from the internal components, which include physical and cyber metrics, such as internal communication utilisation, or sensing the physical environment such as a compass bearing. For each data source a signature is automatically generated. These signatures can be used in isolation or in combination to determine the presence of anomalies and thus determine the level of threat.

To summarize our experimental setup, we combine signatures together to form a behavioural profile score of the system which can represent level of threat to the system during a mission. All data sources have equal weights when combined together producing a sum of all available data source signature scores. For current experimental setup that is reviewed in this publication, we have learnt that the overall behavioural score from the learning scenarios was **46.323**. This score has been produced by a combination of signature summary scores from all available data sources during the learning stage. In this we publication, the key aspect was made to demonstrate the conceptual idea of a data source signature approach and it would be thoroughly explained, therefore the higher level of anomaly detection approach will be investigated further and published in the future. The test scenario produced a score of **54.237**, we can notice that the score for the test scenario is higher than the "Learnt" behavioural score which was learnt using the learning scenarios, through a thorough investigation of the results we have identified that the amount of allowed anomalies has not been exceeded, and overall higher behavioural score was produced by accumulated anomalies that were classified as allowed, resulting in a higher behavioural score. The attack scenario has produced a score of **113.6568**, which is considerably higher than the score produced by the learnt and test scenarios. This shows that the deviations from normality were highly exceeding the threshold allowance on multiple data source signatures.

Further improvements have to be made to increase robustness of the described methodology. Currently, we are using sensor characteristic weights that are equally distributed, thus affecting an overall threat score of a sensor and system itself. Also to make system more robust it would be necessary to investigate how correlation affects an overall behaviour score, as some data sources may have dependencies and these dependencies would result in an anomalous accumulative behaviour score that was described earlier. Weighing system has to be enhanced on a data source and signature characteristic level. One of the solutions is the examination of the spikiness level that can be used to implement dynamic weighing. One of the examples would be that the values from a sensor that are continually volatile (e.g., an accelerometer reading when travelling over a bumpy surface). In such cases, a lower weight would be assigned to the particular sensor characteristic or a signature characteristic. In this way the system can adapt to changing environmental contexts. It is less sensitive to noise or spikes when the ambient noise level or spike frequency is higher.

VII. Conclusion

The work presented here forms part of a wider project to develop techniques for autonomous systems to self-detect attacks. In this paper we have presented a sensor-agnostic learning technique in which a set of sensor-signal characteristics are collectively represented in a signature for each particular sensor. In terms of detecting attacks, the system need not know the type of the sensor, but instead looks at characteristics such as the typical noise levels, the range of data values, the rate of change of data values, the occurrence of spike values, etc.

The initial signatures are generated in experimental mission scenarios but in the absence of attacks, the data signals from sensors are therefore realistic in terms of data values, noise levels, etc. An attack is subsequently detected by observing significant deviations in one or more signature elements for a specific sensor or across several sensors. This approach lends itself to dynamic adaptation which enables the anomaly detection thresholds to be adjusted in line with the environmental volatility, although to date, we have only addressed this step at the concept level (using static signatures for detection).

The main strengths of our approach are that it can be applied universally across a wide range of sensor types without needing manual configuration and that multiple signatures can be used to enhance the attack discrimination accuracy (facilitated by the standardised signature representation). In addition it has the potential to operate in a continuous learning mode in which it will adapt to its environmental conditions over short to medium time spans, but it will always be sensitive to abrupt changes.

We have developed a custom testbed vehicle in order to evaluate the approach. The experimental method and some initial results are presented above and illustrate how the vehicle was able to successfully identify anomalous events which were part of an attack. Our current findings are very encouraging. Due to the weighting scheme we use, the effects of significant differences between expected and actual sensor data are amplified and thus we have achieved a high true-positive rate and simultaneously a low false-positive rate.

The current implementation requires a training phase, during which it builds up behavioural signatures based on the sensed data signals. These signatures then form the basis on which reasoning is performed at several layers in our software stack. The first layer is concerned with anomaly detection at the level of a sensor, whereas at higher levels it is possible to gain a picture of the attack status across the entire vehicle.

Further work includes dynamic adjustment of the anomaly threshold, as discussed above with the intention of removing the need to retrain the vehicle for use in different environments, as well as further evaluation on the training algorithm itself to understand the optimal level of training and to avoid over-training or under-training issues.

Our cyber-security approach is to consider the robotic system from the perspective that the system initially has no knowledge about itself i.e. a box-in-a-box concept where the perception of the robotic system is stored in a box with several doors and the outer box represents the operating environment. The robotic system only observes values coming in or out and is not able to directly observe the true outside environment. In such a case the robotic system's perception has to make sense to itself, without knowing what is outside and is entirely based on sensor data and patterns within. In such a scenario the autonomous system is sensitive to manipulated sensor data and therefore the signature based approach has been devised specifically to facilitate discrimination between normal and abnormal situations, using a combination of learnt mean behaviour, current data signals and trends in data signals.

Acknowledgement

This research has been funded and supported by the Defence Science and Technology Laboratory. We thank Robert Sayers for his invaluable assistance and feedback.

References

[1] T. Vuong, A. Filippoupolitis, G. Loukas, and D. Gan, "Physical indicators of cyber attacks against a rescue robot," in IEEE International Conference on Pervasive Computing and Communications. IEEE, 2014, pp. 338–343.

[2] R. Mitchell and I.-R. Chen, "Adaptive Intrusion Detection of Malicious Unmanned Air Vehicles Using Behavior Rule Specifications," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. PP, no. 99, 2013, p. 1.

[3] G. Loukas, Cyber-Physical Attacks: A Growing Invisible Threat. Butterworth-Heinemann, 2015.

[4] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," Intelligent Transportation Systems, IEEE Transactions on, vol. 16, no. 2, 2015, pp. 546–556.

[5] K. Koscher et al., "Experimental security analysis of a modern automobile," in Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010, pp. 447–462.

[6] M. Wolf, A. Weimerskirch, and C. Paar, "Secure in-vehicle communication," in Embedded Security in Cars. Springer, 2006, pp. 95–109.

[7] R. Mitchell and I.-R. Chen, "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems," Dependable and Secure Computing, IEEE Transactions on, vol. 12, no. 1, 2015, pp. 16–30.

[8] T. P. Vuong, G. Loukas, D. Gan, and A. Bezemskij, "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles," in Information Forensics and Security (WIFS), 2015 IEEE International Workshop on. IEEE, 2015, pp. 1–6.

[9] T. P. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on. IEEE, 2015, pp. 2106–2113.

[10] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, "Consensus-based distributed intrusion detection for multi-robot systems," in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008, pp. 120–127.

[11] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck, "To make a robot secure: an experimental analysis of cyber security threats against teleoperated surgical robots," arXiv preprint arXiv:1504.04339, 2015.

[12] H. Orojloo and M. A. Azgomi, "A method for modeling and evaluation of the security of cyber-physical systems," in Information Security and Cryptology (ISCISC), 2014 11th International ISC Conference on. IEEE, 2014, pp. 131–136.

[13] S. Majed, S. Ibrahim, and M. Shaaban, "Energy smart grid cyber-threat exposure analysis and evaluation framework," in Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services. ACM, 2014, pp. 163–169.

[14] A. Hahn and M. Govindarasu, "Cyber attack exposure evaluation framework for the smart grid," Smart Grid, IEEE Transactions on, vol. 2, no. 4, 2011, pp. 835–843.

[15] R. Mitchell and I.-R. Chen, "On survivability of mobile cyber physical systems with intrusion detection," Wireless Personal Communications, vol. 68, 2013, pp. 1377–1391.

[16] M. Yampolskiy, P. Horváth, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "A language for describing attacks on cyber-physical systems," International Journal of Critical Infrastructure Protection, vol. 8, 2015, pp. 40–52.

[17] M. Mateski et al., Cyber threat metrics. Sandia National Laboratories, 2012.

[18] R. J. Anthony, "Load sharing in loosely-coupled distributed systems: A rich-information approach," Ph.D. dissertation, University of York, 2000.