The 15th International Conference on Advanced Robotics
Tallinn University of Technology
Tallinn, Estonia, June 20-23, 2011

[VOILES | SAILS]: a modular architecture for a fast parallel development in an international multidisciplinary project.

David St-Onge Nicolas Reeves Clément Gosselin

***Abstract*— The [VOILES | SAILS] project was born from architect and artist Nicolas Reeves' will to evoke the age-old myth of an architecture freed from the law of gravity. Many challenges had to be solved in order to achieve this result, and many skills and expertises were required to implement the different modules of the project, from mechatronics to high-level behavior programming. Standing at the crossroads between Art, Architecture and Science, this highly interdisciplinary project brings together researchers from the artistic, technological and scientific realms in an international academic collaboration. Accordingly, the structure, software, mechatronics and artistic concepts were developed in order to allow rapid implementation of new behaviors and potentials for the robots. This paper presents an overview of the development required to create this robotic platform. It is shown that the modular architecture of both the mechatronics and the software was essential for this international and multidisciplinary initiative.**

I. INTRODUCTION

To create novel approaches and even paradigm shifts in robotics research, engineers may find inspiration in other realms such as biology or art. The specificity of the research-creation platform discussed in this paper comes from its original source. The robotic development described here was initiated by the will of creating a novel and ambitious media artwork. The idea was so appealing that it managed to gather many specialists from different countries and different fields.

Professor Nicolas Reeves, a Canadian architect and artist, started to share his vision of the [VOILES | SAILS] concept about ten years ago. Based on the idea of creating flying objects whose shape would be in strong contradiction with the idea of flight, these structures would constitute an architectural statement by themselves. They would materialize the old and mythical dream of an architecture freed from the law of gravity [1]. Professor Reeves' first installation presenting this idea was shown in 1999, in Moncton, Canada. A cubical structure, made from expanded polystyrene and covered by thin plastic films was filled with helium, and suspended over the Peticodiac coastal river. A wooden cubic structure was installed on the shore, in which a chime that was triggered by the passage of a tidal bore on the river.

During the same year, while presenting an artwork in Lausanne, Switzerland, professor Reeves met with Alcherio Martinoli, then student and now professor at the École Polytechnique Fédérale de Lausanne, who was working on robot swarming and assembling behaviors [7]. They started discussing about future projects, which marked the beginning of the instantiation of professor Reeves' vision. It was in 2004 that the [VOILES | SAILS] project was officially launched [2], through a collaboration between professor Martinoli at EPFL, professor Guy Theraulaz at U. Paul Sabatier (Toulouse), and professor Alan Winfield at the U. of the West of England in Bristol. At this time, very few people had the expertise needed to develop this project, and the required equipment, which had to reach very high efficiency-to-weight ratios, was hardly existing. This situation provided an opportunity for the team to create a unique working environment, in which artists, researchers and students in different countries could collaborate under optimal conditions.

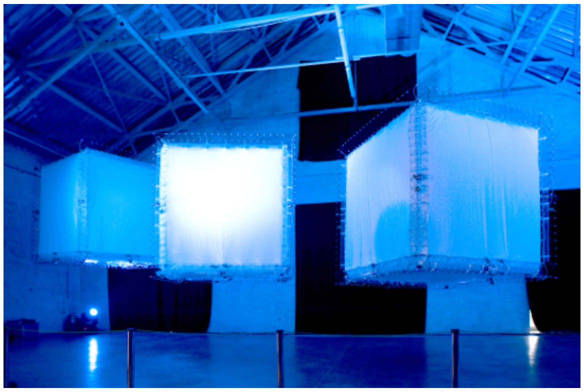


Fig. 1. The Geometric Butterflies installation, Winzavod gallery for Contemporary Arts, ScienceAndArt Fest, Moscow, 2009. Three flying cubes t225c "Tryphon" from the [VOILES | SAILS] project were evolving autonomously according to two basic behaviors: to fly away of light, to avoid collisions.

A. Collaboration context

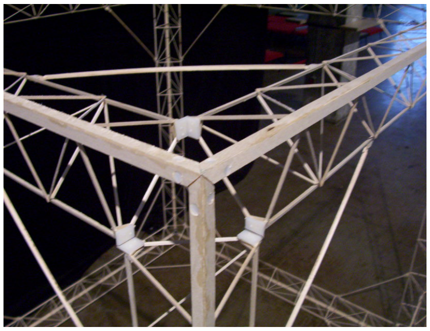
The challenges faced by a multidisciplinary and international research program require an appropriate collaboration framework. A clear communication protocol and the availability of tools to enhance cross-disciplinary learning (as discussed by Fruchter and Emery in [8]) are the keys to a successful venture. Through student training, in-situ teamwork and frequent meetings, the [VOILES | SAILS] platform confirmed the potential of such multidisciplinary collaborations.

All sub-systems used to create the robots were specifically developed or adapted to this project from various sources, with a strong preference for open platforms whenever they were available; this framework is perfectly suited for such

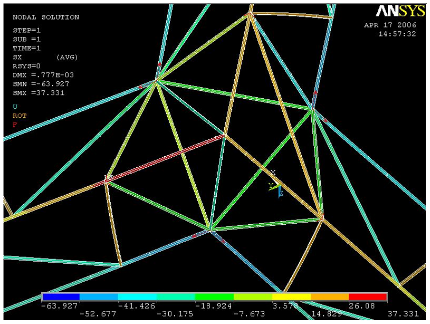
D. St-Onge and C. Gosselin are from the Robotics Laboratory of the Department of Mechanical Engineering, Université Laval, Québec, Canada `david.st-onge.2@ulaval.ca`

N. Reeves is from the NXI Gestatio Design Lab of the School of Design, University of Quebec in Montreal, Montréal, Canada `reeves.nicolas@uqam.ca`

978-1-4577-1159-6/11/$26.00 ©2011 Crown



(a) First basswood design



(b) Analysis for the scalable carbon fiber structure

Fig. 2. Structure corner

a complex collaboration. It nicely resolves several issues related to ownership, and it provides from the beginning a common set of standards and methodologies. Furthermore, as will be shown in this paper, the modular approach was essential to both the engineering research and the artistic creation. This approach consists in focusing on a versatility criterion for each design step and on the ease of adding extensions. Each of the following sections describes the technical and software-related issues that were addressed over the past years.

II. STRUCTURE

The first challenge that had to be solved concerned the structure of the aerobots, which had to be simultaneously extremely light and extremely rigid. For this particular module, Guillaume Credoz, a multi-talented independent architect from Grenoble (France) worked together with Nicolas Reeves to test different truss configurations and materials, and ended up with a truss design that was equilateral in cross-section, and used basswood as a primary material (Fig. 2(a)). After this first functional structure was built, it was equipped with force sensors to analyze the distribution of the tension and compression forces. After several finite-element analyses based on empirical stress data, Canadian engineer David St-Onge managed to produce an optimized structure that was easy to assemble and disassemble in a short time (Fig. 2(b)) [11]. Assembling the whole structure of the latest prototype (a 2.25 metre-edge cube christened Tryphon) and inflating the blimp can now be done in less than two hours. The current design of the trusses uses carbon fiber elements assembled with resin connectors. It does not require any screws since the beams fit into one another trough tubular connectors. The structure keeps its coherence thanks to the forces generated by the inflated helium bladder.

One of the main concerns in the latter analysis was to create a scalable structure. A parametric design was created, based on the surface to stretch. It can easily be scaled to create cubic aerobots of different sizes. A 1.6 meter-edge and a 2.25 meter-edge aerobots were built with this model. The collaboration between professor Reeves, D. St-Onge and G. Credoz established the communication grounds for the next phases of the project, which were primarily concerned with mechatronics and programming.

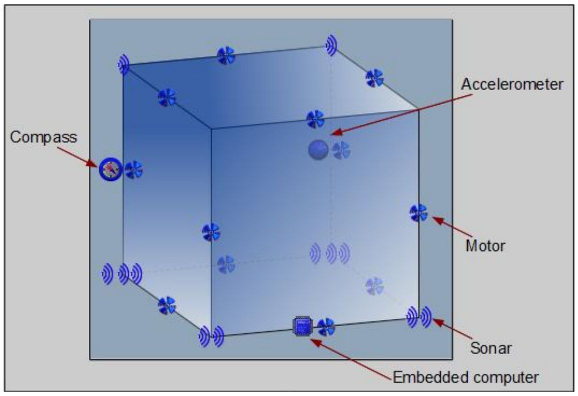


Fig. 3. Embedded hardware - basic configuration

III. HARDWARE

The hardware architecture and its components were originally selected following a comprehensive study by Paola Bernasconi [15], then a Master's student in Electrical Engineering at EPFL and Éric Poncet, a French electronics engineer, who had just returned from the Silicon Valley. After she finished her courses in Switzerland, Mrs Bernasconi moved to Canada for a year in order to work in-situ with the [VOILES | SAILS] team. At that time, the expertise needed to model and develop autonomous aerobots was almost exclusive to EPFL. It is worth mentioning that the lab she came from — recently split into the Laboratory for Intelligent Systems and the Distributed Intelligent Systems and Algorithms Laboratory — is still very active and making major contributions to the field [9].

Since this first version of the mechatronic system of the aerobots, the embedded components have constantly evolved following the need of each performance or artistic installation. The current basic equipment configuration, designed to enhance the robustness of flight control of the aerobot, appears in Fig.3.

A. Electronic brain

The [VOILES | SAILS] robots' brain is a 85mm by 57mm board, called 'Korebot' developed by the K-Team company, located in Yverdon(Switzerland)[1]. The weight of the board is about 40g. It works on a Linux-based operating system

---

[1] http://www.k-team.com

(Angstrom) and provides the required processing power; it also has I2C communication capability. The role of the on-board computer is to gather and interpret the data from all the sensors on the I2C bus and following its internal state model and its central control algorithm to produce the proper motor commands. A wireless connection is always maintained with a ground computer to allow an operator to monitor the different parameters of the robot.

Until 2010, the contributions from this project's team to the above developments was mainly useful to K-Team and its customers. However, the last K-Team board, the KorebotII, is based on the OpenEmbedded environment, which means that much more people and projects can now benefit from this team's contributions. Obviously, it also means that a large number of projects with compatible code can help this team's research.

For instance, for our purposes, a reliable connection between the KorebotII board and the SAILS wireless network is critical; we modified the flash card wireless driver (*libertas*) so that it reconnects automatically and immediately if a problem occurs. We also modified the new I2C drivers (kernel 2.6) to resolve bugs experienced on the new cards. All these works were published on K-Team forums, and will be soon made available to the OpenEmbedded community.

B. Motors

The motors and their controllers have undergone considerable evolution. The first models we used were low-cost GWS Electric Ducted fans (EDF-50mm), mainly intended for toys. The availability of these cheap fans and of their replacement parts, combined with the simplicity of control of brushed DC motors justified this choice at the time. Each motor was controlled by a custom made H-Bridge based on a PIC chip, in order to manage some low-level routines and to interface with the I2C bus.

Unfortunately this strategy, which proved adequate for prototyping, was neither robust and reliable enough for multiple presentations, nor powerful enough for installations in galleries or theaters, where the aerobots often need to work against air streams that can empty their batteries within a few minutes. We then switched to Alfa EDF-60 high-reliability carbon fiber ducted fans (for 16/7/5 Mega motors), whose design ensures longer lifetimes and better reliability. The use of brushless motors is a major factor for the extended lifetime, because of the reduced internal friction. In 2010, we found a German company developing their Electronic Speed Controller (ESC) through open source code[2]. This allowed us to easily adapt our software; among other things, we were able to allow forward and backward rotation selection, and to send battery monitoring informations over I2C communication - a major improvement in our circuit design. The updated software was added to their development branches and discussion threads were started with their international developer community, which is quite dynamic.

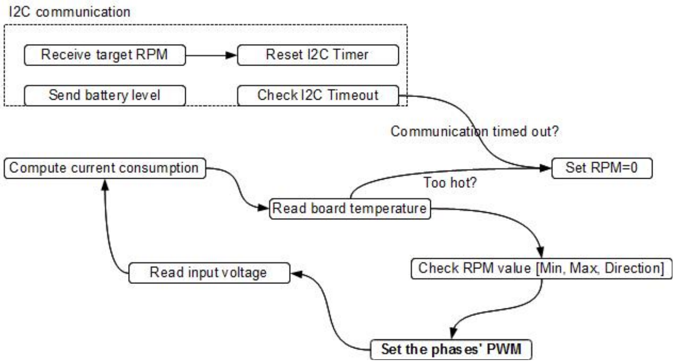


Fig. 4. ESC software diagram

The functional diagram (Fig.4) shows the overall software design embedded on each ESC. One of the major changes we were able to introduce thanks to the open architecture of these devices was to set up limits for temperature and current levels through analog inputs on the chip. We also wrote new routines to handle I2C communication, and to ensure that a motor will stop if the corresponding ESC does not receive instructions from the main computer for too long a time. Also, the motor bases are made of resin, and can be clipped inside the triangular structure, in the middle of each edge. Even if six motors are sufficient to control all the degrees of freedom of the aerobot, as many as twelve can be installed. This redundancy is both a safety feature and a way to maximize the potential thrust. Just like the sensors, the motors are connected to hubs, so that the replacement of a broken or noisy motor can be done in a minute.

The motors were the subject of a specific study in order to monitor real-time:

- The relationship between the speed commands and power consumption.
- Their battery voltage for different configurations.

Indeed separate batteries are used for the logic (sensors and computer) and actuators. The latter may share batteries depending on their location and number. This greatly impacts the computation of the battery level.

C. Sensors, bus and hubs

The communication bus used the I2C protocol, mainly selected for its low number of wires (two plus digital ground; we also use these wires to carry a 5V powering line) and for the range it offers (up to 10 meters without amplification). The hubs allow easy connection and reconfiguration of the sensors plugged in with robust RJ9 connectors.

Many different sensor sets have been implemented on the aerobots. The design was based on biomimetic principles, and was strongly influenced by studies on social insects. We were particularly interested in working from the knowledge of the perception and communication among bees and ants, and professor Theraulaz, a specialist in animal ethology at U. Paul Sabatier in Toulouse, helped to sketch the architecture and configuration of our robots' perception devices. This bio-inspired swarming development context did not require extensive sensing abilities and only sonars (ultrasound) were implemented in the first attempt to focus on the knowledge

---

[2] http://www.mikrokopter.de

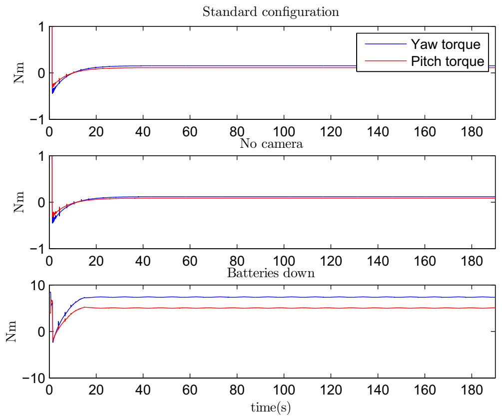


Fig. 5. Simulations of different equipment configurations

the robots can gather from their collaboration. Other robotic platforms have been based only on ultrasound information [12], even indoor blimps [6]. Lately, other participants coming from the realm of performing art entered the project. The potential of the robots as autonomous performers, or co-performers, required extended sensing abilities.

A compass, an altimeter and a 3-axis accelerometer were added this year to the aerobots' spectrum of sensors. They allow a more general and more reliable pseudo-absolute positioning (or pseudo-dead-reckoning) than what can be obtained from sonars only, especially since the accelerometer is a real, full proprioceptive sensor. Moreover, the new sensors will greatly expand the range of environments in which the aerobots can fly, since they will free them from the necessity to evolve in closed areas with flat surfaces on which ultrasonic waves can echo. Comparable studies have been realized on outdoor blimps, but mostly relying also on a GPS [5].

D. Configuration tests

To enhance the understanding of this specific shape of blimp, a study of its dynamics was conducted, which is summarized in [10]. The results were implemented in a numerical Matlab model and optimized following empirical data from the existing prototypes. This model can be used to predict the behavior of the robots, to test controllers or to test different equipment configurations. Indeed, the electronics allows easy reconfigurations of the equipment on the robot, by connecting the sensors or motors to any of the available hubs. Even if the controller is able to compensate for perturbations of the initial equilibrium of the cubes, the location of different devices may affect the energy required to fulfill its tasks.

For instance, we compared the three following standard configurations in order to better understand these effects:

- 1. A battery in each corner, the camera opposite to the main brain for balancing.
- 2. Same as 1. but without the camera.
- 3. Same as 1. but installing the 8 batteries on the lower crown (2 per corner).

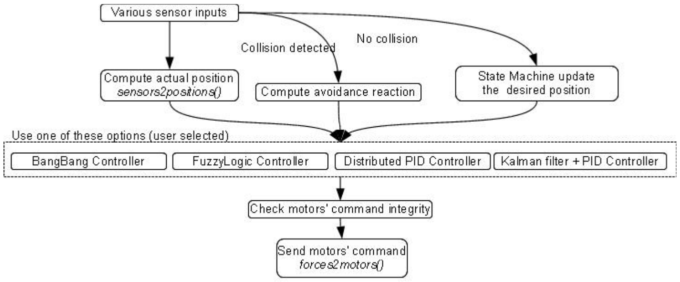


Fig. 6. Main control software architecture

All the above mentioned configurations are using 12 motors, 12 sonars, an accelerometer, a compass, as presented in Fig. 3. In simulation as well as with the real robots, the controller succeeded in stabilizing all configurations. For the first and last configurations the inertia tensor computed by the model is the same:

$$J_{1,3} = \begin{bmatrix} 15.97 & -0.07 & 0 \\ -0.07 & 16.10 & 0.03 \\ 0 & 0.03 & 15.95 \end{bmatrix} (kg.m^2) \tag{1}$$

while removing the camera slightly changes the inertia tensor to:

$$J_2 = \begin{bmatrix} 15.95 & -0.09 & 0.08 \\ -0.09 & 16.00 & 0.03 \\ 0.08 & 0.03 & 15.98 \end{bmatrix} (kg.m^2) \tag{2}$$

Swapping the batteries does not produce any noticeable effect due to the body-fixed reference frame located at the centroid. On the other hand, the height of the center of mass is impacted:

$$CM_1 = 0.025m \quad CM_2 = 0.019m \quad CM_3 = 0.1335m \tag{3}$$

The tests were conducted with the same position commands (from null initial state): $u_{x,y,z,\theta,\phi,\psi} = \{0,0,0,30,20,0\}$. The resulting forces and moments using a PD controller are shown in Fig. 5.

The global energy spent after 200 seconds of simulation, computed using the forces and displacements, is represented by:

$$E_1 = 0.4249 \quad E_2 = 0.4269 \quad E_3 = 0.7550 \tag{4}$$

A comparison between the first two tests shows that the robot has good robustness to sensor scheme modification. The camera was selected because it is the heaviest sensor on board. The last test shows that the users should be careful while positioning the main components of the mass, namely the batteries and the motors. Even if the robot will manage to stabilize, its autonomy will be greatly impacted.

IV. SOFTWARE

The main software is completely embedded on the Korebot card. It was originally developed by É. Poncet and Julien Nembrini, a post-doc in electrical engineering, who was working under the joint supervision of professor Martinoli at EPFL and professor Winfield at UWE - Bristol. The software was based on reactive behavior patterns. Following our biomimetic principles, it was meant to ensure exchanges between the aerobots about their limited knowledge of the environment, as in a community of social insects collaborating through swarm intelligence protocols. An iterative development approach allowed all the protagonists (professors Reeves, Theraulaz, Martinoli and Winfield, E. Poncet and J. Nembrini) to have direct and on-line meetings to insure a close follow-up of the development process. Just as P. Bernasconi, J. Nembrini worked in Montreal for one year to complete his task.

The simulation software Webots[3] was used for testing the in-flight control algorithms of the aerobots. Because of its 3D-visualization capabilities and the realistic simulation it allows, all team members, even those who were not acquainted with programming, could understand the different phases of this key work, and contribute to its evolution. It also greatly facilitated the communication between the Montreal and Lausanne teams, since all professors were able to visualize a realistic model of the aerobots behavior.

As for the hardware and structure, they evolved to a more modular architecture, in order to ensure easy implementation of new behaviors, controllers or sensor drivers.

A. General architecture

The current architecture of the main program of the [VOILES | SAILS] robots is represented on Fig.6. It was developed by D. St-Onge and S. Bracher, another Swiss student who was pursuing a Master's degree in Mechanical Engineering at the École Polytechnique de Montréal. Four different controller-software implementations currently coexist in the main program. They are still the object of a lot of tests and experimentations, and each of them can be selected at any time by the user (dynamically through the SAILS●Ji presented below). They allow us to gather data about their respective performance in various situations. For instance until now the most robust controller turns out to be the fuzzy logic controller as it can be seen from the test made on an earlier version of the aerobots (Fig. 7).

Unfortunately, its adaptation to a new environment requires a good knowledge of the parameters. A simple PID, together with a properly calibrated Kalman filter, is being developed to ensure a robust and easy to adapt controller.

Some key features of the first version of the software by J. Nembrini and E. Poncet proved very efficient, and were kept in the most recent versions:

- Multi-threading. Each sense is controlled in a separate thread. The motor control and the communication control are also on threads. They all read from, and write to, structured global variables.
- State Machine. The main loop is a state machine as described in the next section.

Since the Tryphons are meant to be autonomous and easily reconfigurable robots, their initialization process is a critical routine. At this stage of their startup procedure, the program scans the I2C bus to figure out the number, nature and status of all the connected pieces of hardware, and configures itself accordingly. There's no information on where physically the sensors are connected, but it is rather seen as a reactive approach; by changing the location of a sensor one changes the direction toward which the robot is looking. The last sensors configuration allows a better knowledge of the environment, but also makes use of redundant information: for example, if the altimeter is not connected, the aerobot will automatically decide to use the bottom sensors to compute altitude.

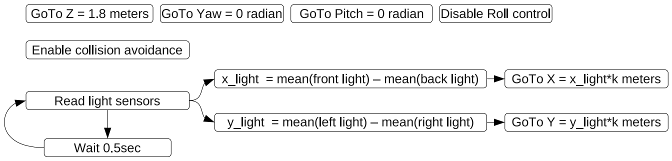


Fig. 8. State example - algorithm behind the Geometric Butterfly installation developed for the ScienceAndArt Fest (see Fig.1).

B. State machine

The main loop of the software is timed to run the state machine twice a minute, which proved to be sufficient, considering the slow dynamics of the robots. The state machine was written on multiple code files, which are only loaded when needed.

By dynamically loading the required state scripts, it is possible to develop, change or modify states while the aerobot is running, as long as the file the user is writing into is not used at this moment. The behaviors are either entirely scripted such as the 'NERVOUS' state, in which the Tryphon reaches the ceiling of the room and then makes a maximum of noise with its motors for a timed period before coming back to its 'IDLE' state, or dynamically change the desired position (command) according to a specific context (Fig. 8).

C. SAILS●Ji

In order to allow easy monitoring of the aerobots state and hardware, we developed a Java interface, named SAILS●Ji. This module is led by D. St-Onge and Romanian student Oldrin Barbulescu, who is currently pursuing a Master's degree in Computer Science at Université de Montréal, Canada.

The SAILS●Ji command application is a Java graphic interface, which uses the Swing graphic library. Java has been selected for its high level of portability across different platforms. It is based on a widely used MVC (Model-View-Controller) architecture for user interfaces [14], each element having a precise role in the interface.

- The Model is divided into two parts: configuration and connection. The first one controls the configuration of the interface (multilingual display, management of configuration files, etc). Two languages are currently implemented (English and French), but all the displayed text is recorded in an external resource file (*.xml),

---

[3] http://www.cyberbotics.com

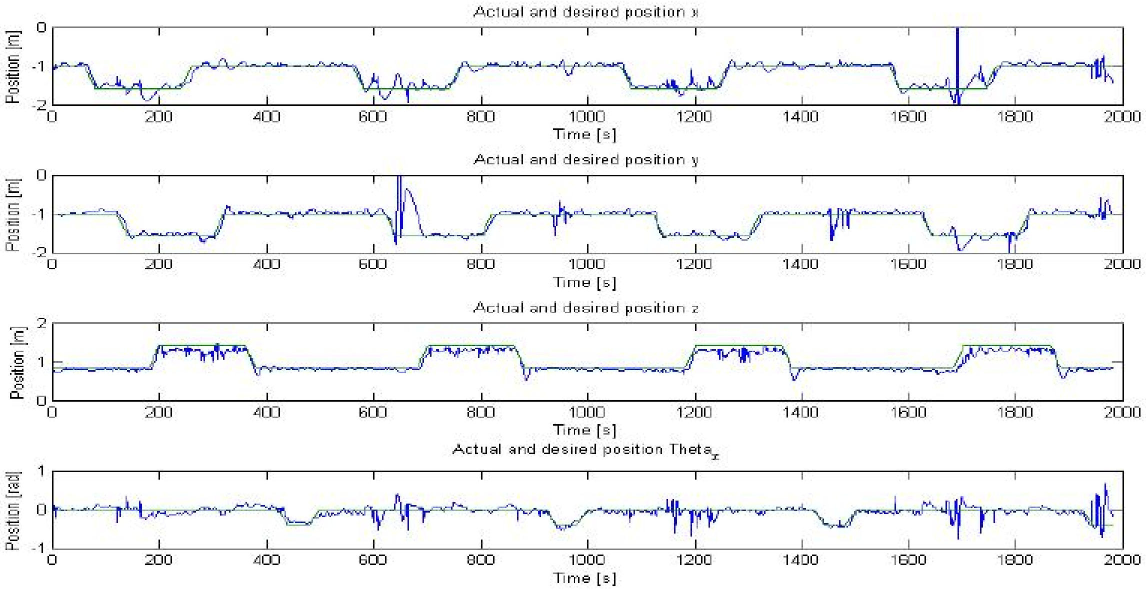


Fig. 7. Fuzzy logic controller tests

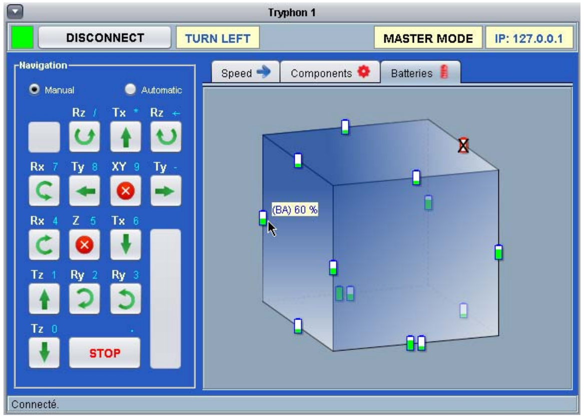
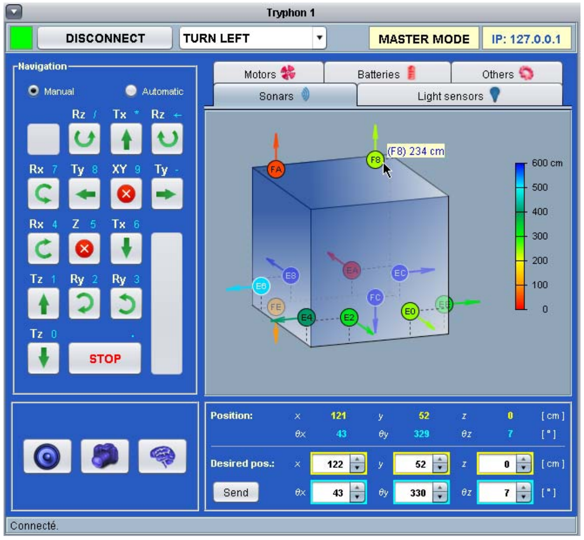


Fig. 9. [VOILES | SAILS] Java Interface in Basic mode



Position: x 121 y 52 z 0 [cm]
θx 43 θy 329 θz 7 [°]
Desired pos.: x 122 y 52 z 0 [cm]
θx 43 θy 330 θz 7 [°]

Fig. 10. [VOILES | SAILS] Java Interface in Expert mode

which allows easy addition of other languages without modifying the code. The second part of the model is in charge of managing the files that contain the connection parameters and the location of the different components. This part also initializes the TCP connection with the server on the Tryphon robot, prepares the commands that will be transmitted, and prepares to display the received data.

- The Controller receives all events from the user, and calls the appropriate mode to implement the required tasks. It is split into several classes in order to efficiently manage events coming from different modules of the application.
- The View is the graphic interface that allows interaction with the user. It does not execute any task or processing. It only displays the results that are sent by the two components of the Model, and send to the controller the data resulting from user inputs. For each connected aerobot a display window is created, which is organized in several sections: a title bar, a connection button and an aerobot identification display; a panel that includes the navigation commands; a graphic panel who uses Java 2D API to display all information pieces collected from the different sensors and components; an "expert mode" section with a panel for the display of speeds and positions of the aerobots, and another that allow to open new windows for the display of supplementary informations.

The TCP client is a thread that uses methods from the Socket class of the java.net package. A server side for the communication with a Matlab client has also been implemented, since it proved efficient to remotely control the robot in this environment for controller software design. Once the connection is established, the commands are queued to be then sent one by one. For this purpose, a 15ms loop listens to the server, in order to send/receive the data. A proper synchronization is essential when several threads are used simultaneously; the Java Swing components are not thread safe, and must be manipulated from one single thread, 'Event Dispatching Thread' (EDT). The 'invokeLater' method from the SwingUtilities class is used to insure that all data is transferred to the interface through EDT.

V. CONCLUSION

In this paper, we described how the aerobots of the [VOILES | SAILS] project, from their artistic origin [3] and throughout all the artistic requirements that directed their technological evolution, have become spokesmen for science and technology for the widest audiences. This is mainly due to the international collaborative network that we managed to develop, which demonstrates the potential of interdisciplinary, international and transcultural teams to create fascinating and beautiful art pieces and performances that can cross borders to become objects of sophisticated technological and scientific researches. The modular implementation of the sensors, motors and their software drivers proved to be a successful strategy for rapidly managing prototypes for different behaviors or human interactions. By choosing an open source approach for the software, the project has benefited from communities of developers and made contributions to them. These exchanges showed their potential to speed up the development of such platforms.

VI. ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chair Program, the Quebec Ministry of Culture (MCCCF), the Quebec Council for Arts and Letters (CALQ), the Canadian Council for the Arts, the Quebec Council for Research in Society and Culture (FQRSC), the Hexagram Institute, the Interuniversity Center for media Arts (CIAM), as well as of the UQAM fund for support to research and creation (PAFARC).

REFERENCES

[1] Cooke, C., and al., 1990. *Architectural Drawings of the Russian Avant-Garde, See in particular Krutikovs flying cities*. Editions of The Museum of Modern Art, New York, USA.

[2] St-Onge, D., and Reeves, N. [VOILES SAILS] project website. URL http://www.voiles-sails.org.

[3] N. Reeves, J. Nembrini, A. Martinoli and A. Winfield, Voiles;Sails : Self-Assembling Intelligent Lighter than air Structures, *Generatifs Arts Conference*, Milano, December, 2005.

[4] N. Reeves, J. Nembrini, E. Poncet, A. Martinoli and A. Winfield, MASCARILLONS: Flying swarm intelligence for architectural research, *IEEE Swarm Intelligence Symposium 2005*, June, 2005.

[5] J. Bijker, W. Steyn. Kalman filter configurations for a low-cost loosely integrated inertial navigation system on an airship, *Control Engineering Practice*, Vol. 16, pg. 1509 1518, 2008.

[6] Mller, A. Rottmann, L.M. Reindl, and W. Burgard. A probabilistic sonar sensor model for robust localization of a small-size blimp in indoor environments using a particle filter. *In Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3589-3594, Kobe, Japan, 2009.

[7] A. Martinoli, K. Easton, and W. Agassounon, Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation, Int. Journal of Robotics Research, vol. 23, no. 4, pp. 415-436, Special Issue on Experimental Robotics, B. Siciliano, 2004.

[8] R. Fruchter and K. Emery. Teamwork: assessing cross-disciplinary learning. *CSCL '99 Proceedings of the 1999 conference on Computer support for collaborative learning. No. 19. International Society of the Learning Sciences*. 1999.

[9] J.-C. Zufferey. *Bio-inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*, EPFL Press, 2008.

[10] D. St-Onge, and C. Gosselin and N. Reeves, "Dynamic modelling of a cubic flying robot", *Proceedings of 2010 ASME International Design Engineering Technical Conferences (IDETC)*, August, Montreal, Canada, 2010.

[11] D. St-Onge, *Analyse de la structure du Mascarillon*,École Polytechnique de Montréal, End of studies project report, Montreal, Canada, 2006.

[12] L. Moreno J. M. Armingol S. Garrido, A. de la Escalera and M. A. Salichs. A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors, *Journal of Intelligent & Robotic Systems*, Springer Netherlands, pg. 135-154, vol. 34, 2002.

[13] S. Tejada and A. Cristina and P. Goodwyne and E. Normand and R. O'Hara and S. Tarapore, Virtual Synergy: A Human-Robot Interface for Urban Search and Rescue, AAAI Technical Report, 2003

[14] SEARIS 2008. Software engineering and architectures for real-time interactive systems on http://www.searis.net.

[15] Paola Bernasconi. *Robotisation des Mascarillons*, Technical report, EPFL, 2004.