

Proposal for a Rapid Prototyping Environment for Algorithms Intended for Autonomus Mobile Robot Control

Andrzej BURGHARDT

Department of Applied Mechanics and Robotics, Rzeszow University of Technology

Received (18 October 2007)

Revised (25 November 2007)

Accepted (30 January 2008)

In the paper, the author provides a proposal of a quick prototyping environment for algorithms controlling homogenous, autonomous mobile robots. The solution proposed enables monitoring of the operation of a group of mobile robots consisting of the recording of motion parameters and signals from distance detectors as well as of controlling the drive units. Furthermore, the solution is flexible, enabling work with various numbers of robots as well as modification of their settings, and conforms to the real time system requirements assumed. In the course of the development process, AmigoBot mobile robots manufactured by ActiveMedia Robotics were used. The control and measuring environment proposed has been verified based on the sample behavioural task of "running in the middle of a free space".

Keywords: Autonomous mobile robots, multi-robot, rapid prototyping, robot control

1. Introduction

Tasks assigned to mobile robots constitute the main driving force behind the science of robotics. The factors decisive for successfully achieving the capabilities of a robot include the inventiveness of the designers, the variety of demands and the current status of knowledge. The range of tasks that can be performed expands when a group of robots are used. Such an expansion of the potential tasks brings certain benefits, including the option of performing new tasks, although it also entails significant problems. The latter are primarily related to the aspects of control, analysis of multiple measurement signals, data transfer, real time control, etc.

Under the solution proposed, a control and measuring environment has been developed to manage the operation of a group of n robots, the requirements for which have been defined as follows:

- ensured autonomous control,
- performance of control and measuring processes for n mobile robots,

- system compliance with the real time system requirements,
- measurement and recording of the following parameters:
 - angular velocities of driving wheels,
 - angular displacement of driving wheels,
 - distance detector measurements,
 - battery voltage measurement,
 - measurement of wheel driving motor current,
- ensured control of a mobile robot's drive units in an open feedback loop,
- options for configuring the settings of the Amigo-SH programme from the managing programme level,
- development of software for managing mobile robots as a library for the Simulink (Toolboxes) package.

The solution proposed enables the number of robots to be changed without the necessity of interfering with the software, the possibility of data sharing, and the control and management of any peripherals that each robot may be equipped with.

The wireless communication system used, based on a Wi-Fi wireless computer network, provides a range of opportunities that includes making use of the latest achievements in this technology, increasing the operational range due to the global network of the internet, data coding, and availability of proxy devices.

2. AmigoBot Mobile Robot

The studies described by this paper involved the use of two-wheel AmigoBot mobile robots. They are manufactured by the ActiveMedia Robotics company, which has been designing and manufacturing mobile robots for many years.



Figure 1 AmigoBot mobile robot

The selected mobile robot functions as an autonomous system. Tab. 1 [4] lists some of the parameters of the AmigoBot robot.

The AmigoBot mobile robot is provided with a PID programme controller, a current protection system monitoring drive unit operation, a unit switching the

weight	3.6 [kg]
maximum velocity	1 [m/s]
operational time	4 [h]
distance detectors	8
digitalinputs/outputs	8
analogue inputs	3
motor transmission ratio	19.5:1
number of encoder impulses per rotation	39,000
Flash	128 [kB]
RAM	32 [kB]

Table 1 Basic technical parameters of the AmigoBot robot

robot off in the event of a break in communications with the managing computer, as well as battery status monitoring units.

3. ARIA and AmigoSH Software

Control of the AmigoBot mobile robot is achieved using two computer programmes, Amigo-SH and ARIA, both delivered by the manufacturer with the robot.

The Amigo-SH software enables the AmigoBot mobile robot to be configured. Software updates as well as reference default parameter values are provided by the manufacturer on its website. The software and the manufacturer's default parameters are stored in the FLASH memory. Modification of the settings as well as updating of the Amigo-SH software can be performed by means of the AmigoSHcf application. In order to introduce modifications, it is also necessary to start the AmigoBot controller in a special mode. This can be achieved:

- automatically on robot start-up (parameter settings are retrieved via the microcontroller's COM1 interface, and if none have been defined, the default values are retrieved from the EPROM memory),
- manually (using a combination of the robot's external MOTOR/TEST/RESET keys),
- automatically from the ARIA programme level (once, during start-up).

Selected parameters of the AmigoBot robot include:

- information package cycle duration time,
- PID controller settings for rotations and translation,
- maximum translation and rotational velocity,
- maximum translation and rotational acceleration,
- maximum translation and rotational delay,
- client-server transmission velocity,

- serial port external transmission velocity,
- distance detector operational cycle time.

Besides the parameters described above, the Amigo-SH programme provides an option to define another series of parameters, thus enabling the user to comprehensively determine the manner of robot operation.

```

C:\WINDOWS\system32\cmd.exe - demo -rh 192.168.1.13
You are in 'position' mode currently.
Mode is one of two values:
  both: heading and move can happen simultaneously
  either: only heading or move is active (move holds heading)

  up arrow: forward 1.0 meter(s)
  down arrow: backward 1.0 meter(s)
  page up: increase distance by 1/2 meter
  page down: decrease distance by 1/2 meter
  left arrow: turn left 90 degrees
  right arrow: turn right 90 degrees
  space bar: stop
  'r' or 'R': reset position to <0, 0, 0>
  'x' or 'X': switch heading/velocity mode

Position mode shows the position stats on a robot.
  x      y      th      comp      volts      mpacs      mode
-335    -930    -121.73    0      12.40      191      both

```

Figure 2 Appearance of the ARIA software user interface

The ARIA software (ActivMedia Robotics Interface for Application) is delivered by the robot’s manufacturer. Moreover, the manufacturer provides a very extensive library of sub-programmes written in C++ which may be utilised in the course of developing applications managing the mobile robot’s operation and in configuring its parameters. The AriaDemo application delivered by the manufacturer enables parameter control for one or several robots by multiple running of the application. By selecting one of the available operation modes, for instance, the user is able to perform robot movements in the “teleop mode”. The user interface in this operational mode is shown in Fig. 2.

The software discussed above, delivered by the robot manufacturer, neither supports the function of robot formation control nor enables recording of measurement data, and the interface is not user-friendly as it requires knowledge of a series of commands. Based on the ARIA source codes and the Matlab/Simulink applications, a far more user-friendly solution has been proposed, supporting the control of a group of robots.

4. Communication with a Group of Mobile Robots

AmigoBot mobile robots communicate with the managing computer by means of a Wi-Fi wireless network. For that purpose, they use WiBox serial servers enabling quick and easy connection of asynchronous devices to a radio network based on the 802.11b standard. WiBox serial servers (Fig. 3) cooperate with the Lantronix Com Port Redirector software handling communication between computers in the MS Windows operating environment.

Wireless WiBox serial servers are provided with two RS232/422/485 (DB-9) serial ports, 10Base-T/100Base-TX network terminal, their own reliable and verified

operating system and support for the function of sending information concerning emergency (alarm) situations via electronic mail. Moreover, they are provided with a set of various protective systems including 128-bit WEP/WPA coding, internal WWW server using the HTML standard, JAVA applets or scripts enabling communication with the given device by means of standard internet browsers (Explorer, Communicator, etc.) [5].



Figure 3 Wireless WiBox serial server

Functional properties:

- wireless communication interface based on the 802.11b standard,
- built-in website server,
- complete, stable and verified TCP/IP stack,
- simple configuration of settings via a website interface,
- e-mail management,
- 128-bit WEP/WPA coding (upgrade),
- high data transfer capacity,
- comprehensive communication management (without the necessity of installing additional software),
- reliable and verified software updated via TFTP or a serial interface,
- serial communication conversion into wireless communication based on the 802.11b standard.

Application of the Wi-Fi communication system enables wireless control of a mobile robot by means of information transfer between the managing computer and the wheeled mobile robot's microcontroller.

5. Software Architecture

The Matlab/Simulink programme was chosen due to its flexibility, the simulation based studies conducted so far, familiarity and availability of the solution, and for the sake of the measuring environment management. It integrates well with external programmes using various sub-programmes and tools, including:

- MEX files (Matlab EXecutable),
- Windows dynamic libraries (DLL),
- DDE mode,
- ODBC/JDBC mode,
- Matlab's notebook,
- ActiveX technology,
- Matlab's engine,
- Matlab Web Server.

For the purposes of the present studies, MEX files were selected after analysing all the pros and cons as well as the aforementioned capabilities of connecting the Matlab environment with external programmes. These are Matlab sub-programmes of a specific structure which, after compilation, function as sub-programmes attached as Matlab software fragments. In the S-function structure, they can cooperate with Simulink. The use of MEX files is a very efficient solution; hence they are used if [3]:

- it is desirable to use the existing software library in C languages,
- it is necessary to increase the calculation velocity,
- the MATLAB environment is to cooperate in real time with units linked with the given computer, such as: sensors, microphones, video cameras, robots, etc.

MEX files are an efficient solution for real time system applications as well as for the processing of information stored in various formats.

The software architecture layout has been based on Matlab's help files and the solution proposed therein has been adopted. The software architecture layout is shown in Fig. 4.

Linking of the above programmes is possible due to one of the sub-programmes (ArMatlab.c), whose structure has been designed in a manner enabling data transfer between Simulink and external programmes. It is a MEX files based data transfer. To compile the programme, the external Visual Studio NET 2003 Enterprise Architect compiler was used.

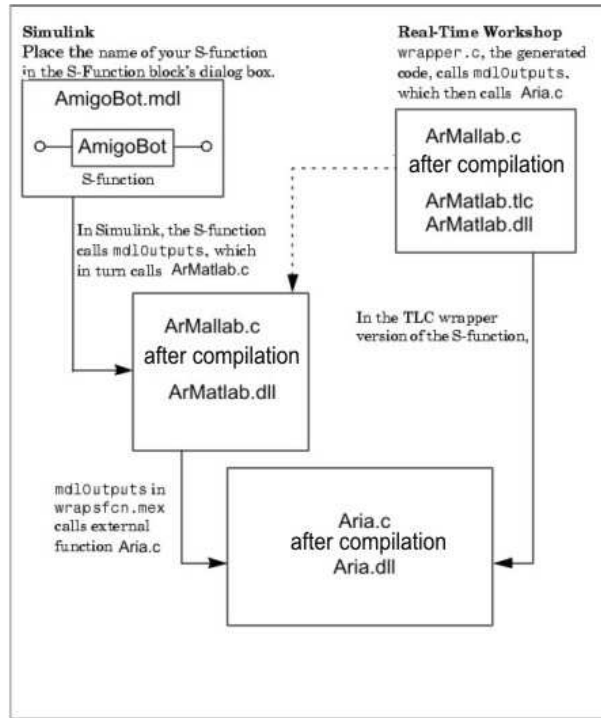


Figure 4 Software architecture layout

6. Development of a Quick Prototyping Environment

A quick prototyping environment managing the operations of a group of n mobile robots has been developed as a compilation of selected sections of the ARIA software provided by the manufacturer and the Simulink S-function, using a wireless computer network for communication purposes. In the Simulink programme, a library has been created containing a "Robot" block, the application of which enables connection of the control algorithms being developed with a real object.

The general diagram of the control and measuring environment development is shown in Fig. 5.

To simplify modification of the AmigoBot robot's settings, a special configuration window has been created in the sub-system with its source code located in the functional m-file (Fig. 6). This enables modification of the robot's parameters, including:

- robot number,
- robot *IP* address,
- values predefined in the Amigo-SH programme,
- transmission velocity for the selected serial port.

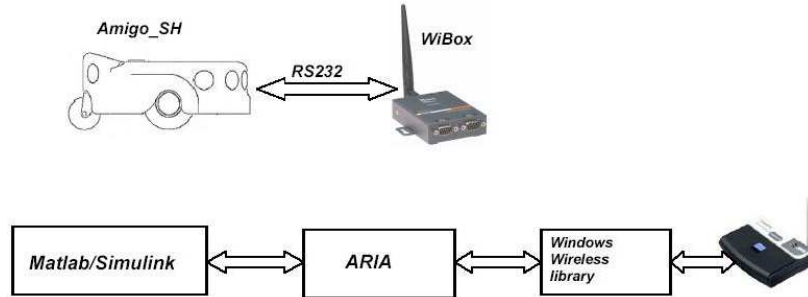


Figure 5 Quick environment diagram

Moreover, it enables definition of the manner of drive unit operation control. Two options are available: control of the driving wheel velocities or control of the supplementary wheel velocity and the frame turning angular velocity. The layout of the configuration window of the sub-system developed in the Matlab/Simulink programme is shown in Fig. 6.

Parameter	Value	Unit
No:	0	
IP:	192.168.1.13	
FLASH:	Default	
SONARCYCLE:	40	[ms]
TRANSELTOP:	1000	[mm/s]
TRANSELMAX:	500	[mm/s]
TRANSACCTOP:	500	[mm/s ²]
TRANSACC:	300	[mm/s ²]
TRANSDECEL:	300	[mm/s ²]
TRANSP:	40	
TRANSPV:	30	
TRANSPKI:	0	
HOSTBAUD:	115200	[bps]
AUXBAUD2:	9600	[bps]
Input:	<input checked="" type="checkbox"/> α_1, α_2 <input type="checkbox"/> α_2, β	
Output:	<input checked="" type="checkbox"/> IO Analog 1 <input checked="" type="checkbox"/> IO Analog 2 <input checked="" type="checkbox"/> IO Analog 3 <input checked="" type="checkbox"/> IO Digital <input checked="" type="checkbox"/> Sonar X,Y <input checked="" type="checkbox"/> Battery	
ROTVELTOP:	180	[deg/s]
ROTVELMAX:	50	[deg/s]
ROTACCTOP:	180	[deg/s ²]
ROTACC:	50	[deg/s ²]
ROTDECEL:	100	[deg/s ²]
ROTKP:	40	
ROTKV:	20	
ROTKI:	0	
AUXBAUD1:	9600	[bps]
AUXBAUD3:	9600	[bps]

Figure 6 The configuration window of the AmigoBot sub-system

Furthermore, the configuration window enables the defining of the vector of any parameter to be scanned into the mobile robot. The available options are:

- driving wheel velocities,
- supplementary wheel velocity and the robot frame angular velocity,

- three external analogue inputs,
- digital inputs/outputs,
- measurements from eight distance detectors,
- battery voltage measurement.

For the control and measuring environment proposed, some sample behavioural tasks were conducted.

7. Verification of the solution proposed

Verification of the solution proposed has been conducted based on performance of the sample behavioural task. The kinematics of the mobile robot has been defined by means of equation (1) [2]:

$$\begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_{A \max} & \dot{\beta}_{\max} l_1 \\ v_{A \max} & -\dot{\beta}_{\max} l_1 \end{bmatrix} \begin{bmatrix} u_V \\ u_\beta \end{bmatrix} \quad (1)$$

Values $V_{A \max}$, $\dot{\beta}_{\max}$ present in relationship (1) represent the maximum velocity of point A and the mobile robot frame velocity correspondingly.

The robot used in the studies was equipped with eight distance detectors: two at the rear (not used in the current studies), and six at the front. The distance detectors located in the robot's circumference (Fig. 7) have been divided into groups marked: d_R , d_L , d_F where: $d_L = \min(s_1, s_2, s_3)$, $d_F = \min(s_3, s_4)$, $d_R = \min(s_4, s_5, s_6)$. Values from the sensors are limited by their scope of operation and contained within the range $d_{\min} \leq d_{(\cdot)} \leq d_{\max}$.

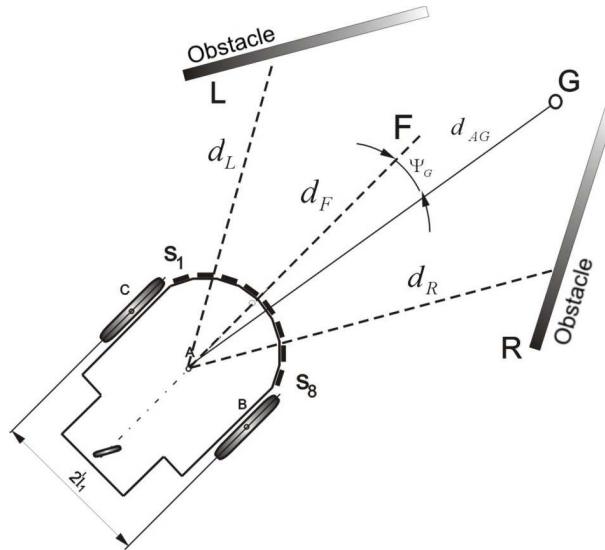


Figure 7 Lab robot diagram

Signals from sensors have been normalised in the following manner: distance measurement made by the right detector: $d_R^N = d_R(d_R + d_L)^{-1}$, left detector: $d_L^N = d_L(d_R + d_L)^{-1}$ and middle detector: $d_F^N = d_F\rho^{-1}$, where: ρ – sensor’s maximum measuring range.

The elementary behavioural action of “running in the middle of a free space” has been performed by application of a modified Braitenberg algorithm [1] in accordance with the relationship:

$$\begin{bmatrix} u_V \\ u_\beta \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} d_L^N \\ d_R^N \\ d_F^N \end{bmatrix} \quad (2)$$

where: w_{ij} corresponds to the weights of the neural network chosen experimentally assuming values from the $(-1, 1)$ range. In the algorithm discussed, the values for the weights were chosen in a manner ensuring correct performance of the elementary behaviour. The control signals attained based on relationship (2), after being introduced into relationship (1), enabled the generation of velocity at the given mobile robot driving wheels. The robot moved in the prepared environment, where it was assumed that the verification was complete if $x_A \geq 1.4 [m] \wedge y_A \in (1.2, 1.8) [m]$. The following initial conditions were assumed: $x_A = 0.2 [m]$, $y_A = 0.41 [m]$, $\beta = 0 [rad]$. Fig. 8. – 11. illustrate the results achieved in the course of the test.

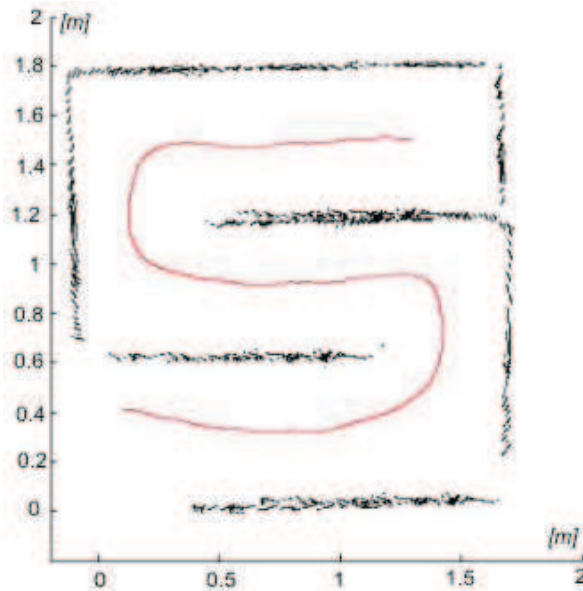


Figure 8 Re-creation of the operational environment based on signals received from sensors and the trajectory of point A

The control and measuring environment proposed enabled the correct performance of the task of “running in the middle of a free space” as well as recording of the motion parameters.

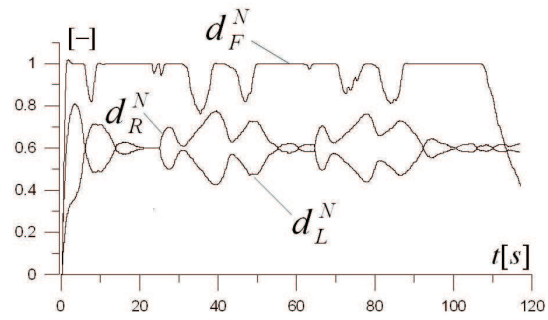


Figure 9 Normalised distances from obstacles

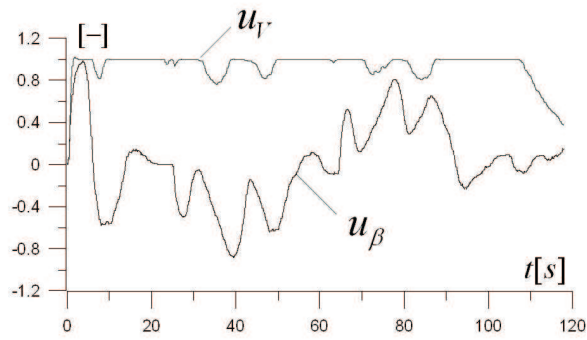


Figure 10 Control signals u_V , u_β

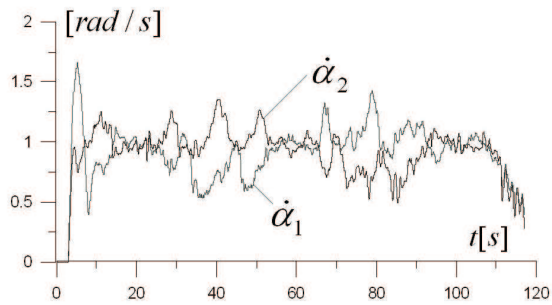


Figure 11 Driving wheel angular velocities $\dot{\alpha}_1$, $\dot{\alpha}_2$

8. Conclusions

The software proposed enables control of drive unit operations as well as the recording and reading of data received from the sensory systems of the AmigoBot mobile robot. This can be applied to several robots simultaneously and it enables modification of a robot's settings from the control programme level. It ensures autonomous control of a group of robots in conformity with the assumptions of a real time system.

References

- [1] **Burghardt, A.**: Sterowanie behawioralne mobilnym robotem koowym, *WBMiL*, Politechnika Rzeszowska, Rzeszów, **2005**.
- [2] **Giergiel, M., Hendzel, Z. and Żylski, W.**: Modelowanie i sterowanie mobilnych robotów koowych, *PWN*, Warszawa, **2002**.
- [3] **Mrozek, B. and Mrozek, Z.**, Matlab i Simulink. Poradnik użytkownika, *Helion*, **2005**.
- [4] Robot manufacturer's instructions at robots.mobilerobots.com/amigobot/amigofree/AmigoGuide.pdf
- [5] Wibox manufacturer's instructions at <http://www.lantronix.com/device-networking/external-device-servers/wibox.html>