WIRELESS REAL-TIME SYSTEM FOR OMNIDIRECTIONAL MOBILE ROBOTS CONTROL

Tiago Trindade Ribeiro*, Jessivaldo Santos*, André Gustavo S. Conceição*, Augusto L. da Costa*

*LaR - Robotics Laboratory
Department of Electrical Engineering
Federal University of Bahia
Salvador, Bahia Brazil

Emails: ttrindade.ee@gmail.com, jessivaldojr@hotmail.com, andre.gustavo@ufba.br, augusto.loureiro@ufba.br

Abstract— This paper presents a wireless real-time system developed for omnidirectional mobile robots control. The system uses a strategy of control in cascade form composed by two levels, one embedded on mobile robot and other implemented on PC. The communication between the robot and the supervision/decision system is accomplished through a Zigbee platform with latency requirements explained. Real results of real-time trajectory tracking are provided to demonstrate the performance of the developed system.

Keywords— Mobile robots, Wireless communications, Real-time Control Systems, Feedback systems.

1 Introduction

The scenario of continuous technological progress currently seen in various sectors of society has required the development of systems with increasingly stringent requirements of efficiency, reliability and robustness, among others. Advances in electronic and automatic control systems further increases this stringency in such a way that the development of alternative systems that make it possible to weigh the aforementioned factors in a manner convenient to a particular application is quite attractive.

To this end, it is necessary to design integrated hardware and optimized software that satisfy all the requirements of project seamlessly to the system's overall performance. Particularly in mobile robotics, where verifies up big level of interdisciplinary, will be possible to provide autonomy near of idealized. Such autonomy is possible when defining the objective of not only certain effect or result, but also on how long you should get them, which inserts fully the automatic systems in the context of real-time systems.

Historically, great concern is shown by the academic community in looking for solutions to various problems in robotics from the standpoint of real-time systems, for example, in (Borenstein and Koren, 1990) to avoid obstacles during navigation of mobile robots and (Shaffer and Herb, 1990) for the prevention of collisions in the workspace of robotic arms. But investigations of real-time systems are not restricted to a few areas of knowledge, which provides the continuous expansion of a broad spectrum of applications, as can be checked for biomedical area in (Ruha et al., 1997), for telecommunications in (Lima et al., 2006), and for detection of flooding in (Oprea et al., 2010), all these works endowed with large background and immediate social counterpart.

In general, applications of robotics do not require communication with high data traffics (Wan et al., 2009) and this makes it possible to use protocols whose primary concerns are accuracy and efficiency, such as those implemented by the Zigbee standard. This standard is used in low power consumption, Wireless Personal Area Network (WPAN) and nowadays is widely used in various applications.

The *Zigbee* standard is applied in robotics, for instance, in point to point communications, where the main benefit is to provide remote communication without wires, in designing a sensor network that modularizes the control system and improves the computational capacity, as in (Ru et al., 2008), or even to design networks of cooperative robots where you can monitor all actions of each agent through a remote point (Jung et al., 2010) and send new coordinates of a given trajectory (Johansson et al., 2008).

This paper presents a low cost wireless system applied to real-time control of an omnidirectional mobile robot. A *ZigBee* platform is used to implement the wireless communication. The cascade control strategy used in (Conceicao et al., 2007) was applied in this work and real results shows the effectiveness of the developed system providing a reliable experimental platform.

In the following, Section 2 describes the developed system, details some elements of the system and presents the wireless communication platform. Section 3 presents the models of the system and section 4 presents the two level control strategy. In section 5, real results are presented showing the system's performance. Finally, the conclusions are drawn in section 6.

2 System description

The real-time system developed is basically composed of two main modules, one is embedded in a mobile robot, responsible for the system instrumentation and the generation of the time base for the design of a real-time clock. Other module is developed in a personal computer (PC) and consists of the high-level layer of controllers and decision/supervision system. These modules communicates in real-time through a *Zigbee* platform.

The mobile base used in this work is made of fiberglass, and comprises two main parts that fits together. The robot is equipped with three omnidirectional wheels connected to geared motors, arranged in a space of 120 degrees. The top part is poured in the region between the motors, generating an internal space in the structure that is used to accommodate the modules rechargeable batteries. The wheels have bearings around its perimeter that allows a movement in the transverse direction of rotation. More information about this robot and your embedded system are provided in (Conceicao et al., 2011)

2.1 Wireless communication

Among the main criteria for determining the wireless communication technology to be used in mobile robotic systems are those that determine the requirements for reliability and data transfer. In addition to a good compromise between these characteristics, one should analyze the energy consumption required by the communication system, since it is a global trend to design devices with optimized power consumption.

A major challenge arises from the definition of consistent weights for each one of these characteristics for the determination the ideal technology of communications for real-time mobile robotic systems, because guaranteeing a good performance for one can be in total detriment to another. After brief review of existing technologies, a *Zigbee* platform implemented with *Xbee* modules was chosen.

Among the reasons for selecting these modules can be mentioned the ease of programming as well as aesthetic characteristics compatible with the project developed. As previously mentioned, communication rate is not a critical factor to this development, but some features of communication platform should be highlighted.

This standard has a maximum rate of communication in the *RF* link of *250Kbps*, but as a typical message has a header, with about 25 bytes when performed 64-bit addressing, the rate drops significantly, so this feature must be considered on implementation. Moreover, as the firmware *Xbee* works with a clock around 10MHz, communications at high baudrates (more than 57600 bps) becomes very inaccurate so this should be avoided.

Thus, one can't guarantee real-time without a quantification of all these factors affect the rate of communication. Another feature that may compromise system performance, is that the module *Xbee* only implements the medium access control technique CSMA-CA, which is inherently stochastic, generating naturally random delays. Given these circumstances, defines up the platform to communicate only two *ZigBee* devices aiming to determine all of the time delays inherent in the development of wireless real-time system. The latency characteristics (data delivery time), strongly affected by the factors mentioned above must be analyzed in detail, because intend up on future add other end devices to form a network for real-time control of a multi-robot system. Initially is considered to be major time spent on communication as the transmission time of data packets over the air T_{air} and the random time spent by the CSMA-CA and the reentries.

The physical layer of 802.15.4 standard allows a maximum of 127 bytes per packet. Due to the size of the header, the *Xbee* module can send a maximum of 100 bytes. The 2.4GHz physical layer specifies a transmission rate of the RF channel of $250Kbps$, which is spending $4\mu s$ per bit. This provides enough information to calculate the time spent in ms for send a packet of B bytes when deployed 16-bit address:

$$T_{air}(B) = 0.416 + 0.032B \quad (1)$$

Thus, as in this application is transmitted to a maximum of 25 bytes in both directions of the link, and considering the start and stop bits, the time spent airborne transmission is defined as $1.42ms$ for send.

The random delay generated to implement the CSMA-CA and reentries is more difficult to calculate, but the *Maxstream* provides a way to estimate of the best and worst cases for the total transmission time, as given by the following expressions:

$$T_{Total}^{Best}(B) = 0.544 + 0.032B \quad (2)$$

$$T_{Total}^{Worst}(B) = 40.096 + 0.128B \quad (3)$$

For this application it was defined as being $1.54ms$ and $44.1ms$ the best and worst cases, respectively, and these values were lower than the period of $50ms$ defined for data acquisition by the supervision/decision system.

3 Modeling Robot

To implement the control strategy proposed in this paper it was considered only the dynamic characteristics of the control loop of the low level composed of three DC motors, being still that, for the complete robot movement uses the base's kinematic model. The following are the models used for the design of two layers of controllers.

3.1 Models motors

How it is not known the operating conditions that motors used in this study were previously submitted, uses the semi-graphical method for the analysis of their dynamic models. Thus, considering that tests with the motors provided a response time in open loop of $210ms$, the model motor DC motors obtained after discretization with a sampling period of $10ms$ is given by:

$$\frac{u_i(z)}{\omega_i(z)} = \frac{1,5163}{z - 0,8253} \tag{4}$$

Where, $u_i(z)$ the voltage in V and ω_i is the speed in rad/s of motor i.

Thus, the developed system has one sample time for low level layer and other for the high level layer.

3.2 Kinematic model

To determine the kinematic model of the base, any influence of forces and only geometric characteristics is disregarded are used. In Figure 1 has illustrated the arrangement of the various velocity vectors and also the coordinate systems used.

Analyzing the compositions of motion the following matrix equations are obtained:

$$\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & L \\ cos\delta & sen\delta & L \\ -cos\delta & sen\delta & L \end{bmatrix}}_{B} \begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} v \\ v_n \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & L \\ cos\delta & sen\delta & L \\ -cos\delta & sen\delta & L \end{bmatrix}^{-1}}_{G} \begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} \tag{6}$$

Thus, 5 determines the inverse kinematic model of the base, ie, given the speed of the center of mass, the velocities of the wheels are obtained, yet 6 determines the direct kinematic model of the base, or that is, given the wheels speeds one may calculated the velocities of robot's center of mass.

The relationship between linear and angular speeds is obtained through the wheels radius matrix, R and the matrix of reduction factors on motors, l, as follow:

$$\begin{bmatrix} v_{m1} \\ v_{m2} \\ v_{m3} \end{bmatrix} = Rl^{-1} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix} \tag{7}$$

3.3 Real-time system

According to (Cheng, 2002) to design systems truly in real-time, it should be, depending on the specific application, succinctly defines the types of tasks to be performed and the criteria for their scheduling and characterize the platform where they will be executed. As an immediate result of uses one method to meet the needs mentioned above, there is the definition of computational time.

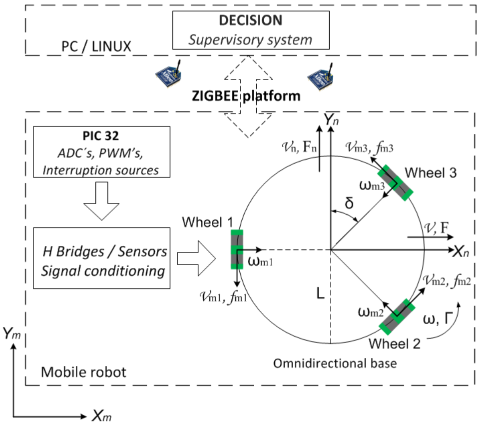


Figure 1: Main elements of wireless real-time system

In this paper, uses as criteria for determining the computational time required, those that ensure that it is possible to instrument the layer of low-level control already that appears in this the faster dynamics. As the main objective is to design a system that enables the design and evaluation of various control strategies, defines use a PC to implement the system of decision/supervision.

To make this possible, it should be ensured that the operating system used is able to prioritize care tasks in the system within feasible time limits, which became possible with the use of the operational system *(OS) Linux* and the whole system development decision/supervision done through software *Lazarus IDE.*.

As it needs to transmit 25 *bytes*, it follows that the time spent in nominal transmission is only 2.17 ms aditioned to the time spent by platform communication. This last time is estimated as being 6 ms. Considering the activities of the *SO* for the implementation of the system, majora the total value for $15ms$. Thus one can define this value as a lower limit for the sampling period, and for the control of the aforementioned dynamic dominant defined that period as $50ms$ aiming to enable the implementation of varied control strategies.

The Figure 1 shows a diagram with global system described.

4 Real-time controllers

The control scheme is based on the cascade structure, as shown in Figure 2. The inner-loop is intended to control the robot dynamics by tracking the reference velocities given by the outer-loop controller.

In this work we propose the use of discrete *PID* controllers in the internal loop (low level)

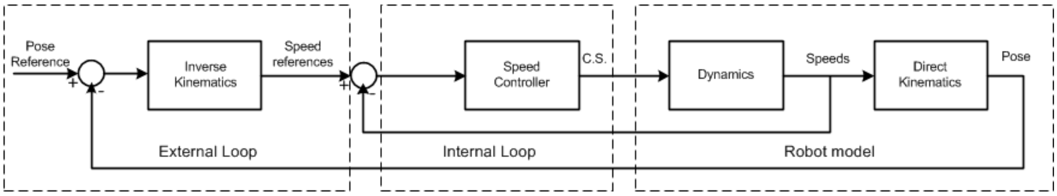


Figure 2: Block diagram of cascade form control strategy

which controls the wheel speeds. The external loop (high level) consists of a kinematic controller. The inherent simplicity of this architecture allows for the validation of algorithms to task planning and trajectory tracking in an objective manner.

4.1 Low level controller

The discrete *PID* controllers for each motor are implemented on mobile robot firmware through the version called speed algorithm, ie, the plant sends up the rate of change of control signal. The justifications for this form of implementation are derived from the nature of plant and facilitations offered to development. Another reason for implementation of this algorithm is that protections wind-up are easier to implement.

Considering the following objectives for speed control of motors, $Overshoot = 0$, response time in closed loop smaller than the open-loop, zero position error and not to exceed the voltage levels supplied to the motors, and given the characteristics of the motors, it is known that the use of *PI* controller is enough, but the derivative action is also implemented in case there is a need to change the specifications of both modules of drives or the motors themselves.

The real-time system designed was able to provide data acquisition that would enable the use of the analytical method of tuning, or set up values for gains K_c and K_i by examining the root-locus of system, these parameters are obtained for 0.239 and 0.051, respectively.

The transfer function for the *PI* controller obtained is given as follows:

$$\frac{e_i(z)}{u_i(z)} = \frac{z - 0,808}{z - 1} \tag{8}$$

Where, $e_i(z)$ is the error signal calculated from speeds, of reference and actual of motor i.

4.2 High level controller

In this level, the external loop is closed by the inverse kinematics block, which generates velocity references to the internal controller, from the measurement of the robot position.

The poses of reference is defined as a set of points in the world frame (OXY), see Figure 3: $W(k+j) = [x_{ref}(k+j)\ y_{ref}(k+j)\ \theta_{ref}(k+j)]^T$, $j = 0, 1, ..., N$. The vector of velocity references $\bar{W}(k+j) = [\bar{v}(k+j)\quad \bar{v_n}(k+j)\quad \bar{w}(k+j)]^T$, $j = 0, 1, ..., N$, where j is an step prediction of the robot velocity made at instant k, is given by

$$\begin{bmatrix} \bar{v}(k+j) \\ \bar{v_n}(k+j) \\ \bar{w}(k+j) \end{bmatrix} = \begin{bmatrix} cos(\theta(k)) & sin(\theta(k)) & 0 \\ -sin(\theta(k)) & cos(\theta(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix}, \tag{9}$$

with

$$\begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix} = \begin{bmatrix} v_{nav} cos(\varphi) \\ v_{nav} sin(\varphi) \\ \bar{\theta}(k+j) - \theta(k) \end{bmatrix}, \tag{10}$$

and

$$\varphi = atan2\left(\bar{y_r}(k+j) - y_r(k), \bar{x_r}(k+j) - x_r(k)\right). \tag{11}$$

$\xi(k) = [x_r(k)\quad y_r(k)\quad \theta(k)]^T$ is the robot pose at time step k, N is the number of trajectory points, and v_{nav} is the linear velocity of navigation of the robot, which is a design parameter.

5 Results

Described the most important steps for the development of wireless real-time system, perform up a set of laboratory tests to be able to assess project performance. Initially it was presented the results obtained by analyzing the system's performance aiming to obtain a characterization that allows safely proceed with the implementation of control strategies needed.

5.1 Accuracy and limitations of real-time system

The components used to system's conception provides benefits of critical importance to the development of all activities such as support of a *OS* reliable, compatible hardware and software and reduced development time.

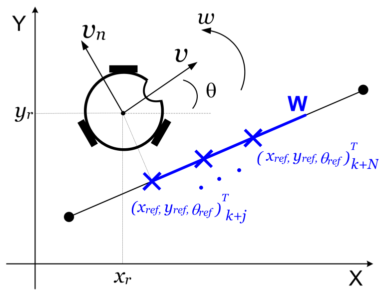


Figure 3: Evolution of mobile robots poses

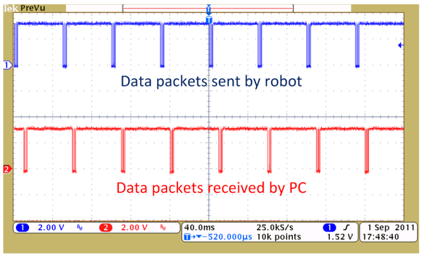


Figure 4: Packets traffic from robot to PC

It is possible to cite, for example, the processing time it takes to boot functions *I/O* of decision system. This system works with *Lazarus* installed in *Windows* for this task requires a processing time of about $20ms$, whereas when running on *Linux* has an average value for this magnitude of $5ms$. However, because the environment is not a priority in its operational platform, this time varies greatly with the characteristics and current conditions of processing of *CPU*. Thus, this study confirms the great importance of generating external time base for the system, so it has a limitation to the processing time previously set at $50ms$.

This sample time is sufficient for meet the requirements necessary for development of real-time environment and also provides a time period sufficient for the implementation of various control strategies. Figure 4 shows results for the transmission of packets over *ZigBee* platform from robot to PC, with the system in full operation. There was a delay of less than $6ms$, a value much lower than $50ms$ defined for data acquisition by the supervisory system and that $44.1ms$ estimated for transmission on worst case over *ZigBee* platform. But the most important in this analysis is that this period remains always fixed at both ends of the communication system and in both directions so that have guarantees that the system operates in true real-time.

5.2 Pose controllers

Setting the speed control strategy and characterizing the odometric system performances of mobile robot, results were obtained for the control of pose through experiments for the trajectory tracking.

The trajectories are provided by the user through the points and after defining the parameters of the control strategy being used, the system triggers a special procedure for the control of trajectories responsible for updating references from pose to be provided to controllers.

Knowing the restrictions on control efforts, implement activities to follow the trajectory of low velocities v_{nav}. These speeds had their order of magnitude as defined previously, for example, it appears that operations with v_{nav} over $1m/s$ require high values for ω. If the trajectories has abrupt changes in the orientation of the robot, this high values of ω can compromise the performance of the controllers. Thus the experiment consists of applying a ellipsis trajectory with major axis of $2m$ and minor axis of $1m$, and, $v_{nav} = 0.5m/s$. Figure 5 illustrates the results obtained.

It was observed a good performance since the positions were achieved satisfactorily and orientations were maintained with tolerable maximum error. For the controller, high-level control efforts are the reference speeds for the center of mass of the robot, these are also presented in this figure.

The Figure 5(c) presents the temporal evolution of poses and 5(d) presents the control efforts for low-level controller, thus ratified the good performance of the strategy implemented. If there is a need to follow trajectories with higher navigation speeds due to supply them in the form of ramp, in order to avoid current peaks thus preserving the actuators.

6 Conclusions

This paper presents the design of a wireless system applied to real-time control of mobile robots. It was developed a system of low computational/finance cost to be used as a platform for both experimental validation of control strategies already established and the development of new applications.

The designed system has provided practical data of great relevance for mobile robotics that allow the resolution of complex problems, often not yet experimentally validated. The decision/supervision system was developed using the *Lazarus IDE*, which is licensed in *GPL*, ie, the softwares developed on it can be distributed under any license, and thus increased the commercial attractives.

Analysis of wireless system are useful for many applications where real-time critical tasks must be perfectly executed, since was estimated the time limitations of the *ZigBee* standard enables increasing the complexity consciously always that necessary.

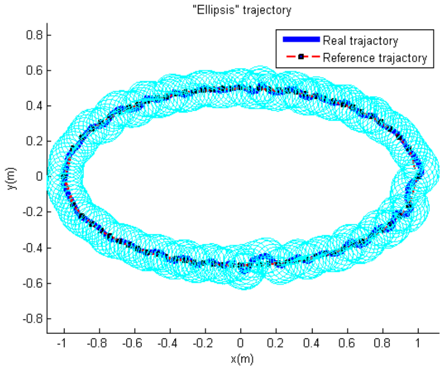
The paper presented a part of a larger study, and future works includes the implementation of a network via the *ZigBee* standard for controlling formation of a multi-robot system.

Acknowledgments

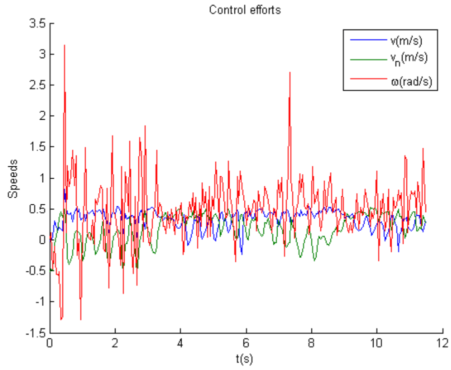
The authors gratefully acknowledge the financial support of FAPESB (Fundação de Amparo à Pesquisa do Estado da Bahia).

References

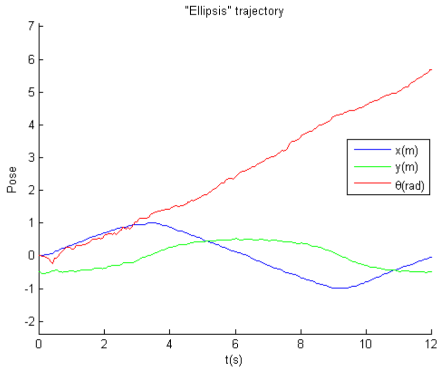
Borenstein, J. and Koren, Y. (1990). Real-time obstacle avoidance for fast mobile robots



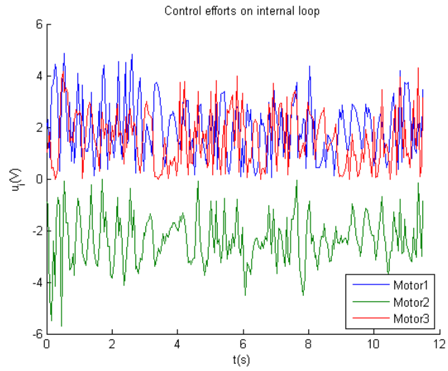
(a) Pose evolution on plane xy.



(b) References Speeds.



(c) Pose evolution on time.



(d) Motors voltages.

Figure 5: "Ellipsis" trajectory, $v_{nav} = 0,5m/s$.

in cluttered environments, *Ieee Transactions On Systems, Man, And Cybernetics* **19**: 1179–1187.

Cheng, A. M. K. (2002). *Real-time systems*, Vol. 1, Wiley-Interscience.

Conceicao, A. G. S., Ribeiro, T., da Costa, A. L. and Santos, J. (2011). Sistema microprocessado para controle em tempo real de robôs móveis omnidirecionais, *Anais do X Simpósio Brasileiro de Automação Inteligente*, pp. 1340–1345.

Conceicao, A., Oliveira, H., Sousa e Silva, A., Oliveira, D. and Moreira, A. (2007). A nonlinear model predictive control of an omnidirectional mobile robot, pp. 2161–2166.

Johansson, M., Haataja, K., Mielikainen, J. and Toivanen, P. (2008). Designing and implementing an intelligent zigbee and wlan enabled robot car, *Internet, 2008. ICI 2008. 4th IEEE/IFIP International Conference on*, pp. 1–6.

Jung, J. H., Park, S. and Kim, S.-L. (2010). Multi-robot path finding with wireless multihop communications, *IEEE Communications Magazine* **48**(7): 126–132.

Lima, T. C. B., Umezu, C. K., Cappelli, N. L. and Nunes, E. F. (2006). Equipamento microprocessado para geração de sinal de correção diferencial, em tempo real, para GPS, *Engenharia Agrícola* **26**: 537 – 545.

Oprea, M., Buruiana, V. and Matei, A. (2010). A microcontroller-based intelligent system for real-time flood alerting.

Ru, D., Tong, Z., Li, Z. and Gang, L. (2008). Design and realization of omni-directional mobile robot body based on zigbee technology, *Embedded Computing, 2008. SEC '08. Fifth IEEE International Symposium on*, pp. 207–211.

Ruha, A., Sallinen, S. and Nissilä, S. (1997). A real-time microprocessor qrs detector system with a 1-ms timing accuracy for the measurement of ambulatory hrv., *IEEE Trans Biomed Eng* **44**(3): 159–67.

Shaffer, C. A. and Herb, G. M. (1990). A real-time robot collision avoidance system (abstract), pp. 441–.

Wan, J., Wang, Y., Qin, Q. and Li, Y. (2009). Multi-robots' communication system based on zigbee network, *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, pp. 3–14–3–19.