We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

5,800
Open access books available

142,000
International authors and editors

180M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Conceptual Bases of Robot Navigation Modeling, Control and Applications

Silas F. R. Alves[1], João M. Rosário[1], Humberto Ferasoli Filho[2],
Liz K. A. Rincón[1] and Rosana A. T. Yamasaki[1]
[1]State University of Campinas
[2]University of São Paulo State
Brazil

1. Introduction

The advancements of the research on Mobile Robots with high degree of autonomy is possible, on one hand, due to its broad perspective of applications and, on other hand, due to the development and reduction of costs on computer, electronic and mechanic systems. Together with the research in Artificial Intelligence and Cognitive Science, this scenario currently enables the proposition of ambitious and complex robotic projects. Most of the applications were developed outside the structured environment of industry assembly lines and have complex goals, such as planets exploration, transportation of parts in factories, manufacturing, cleaning and monitoring of households, handling of radioactive materials in nuclear power plants, inspection of volcanoes, and many other activities.

This chapter presents and discusses the main topics involved on the design or adoption of a mobile robot system and focus on the control and navigation systems for autonomous mobile robots. Thus, this chapter is organized as follows:

- The *Section 2* introduces the main aspects of the Robot design, such as: the conceptualization of the mobile robot physical structure and its relation to the world; the state of art of navigation methods and systems; and the control architectures which enables high degree of autonomy.
- The *Section 3* presents the dynamic and control analysis for navigation robots with kinematic and dynamic model of the differential and omnidirectional robots.
- And finally, *Section 4* presents applications for a robotic platform of Automation, Simulation, Control and Supervision of Navegation Robots, with studies of dynamic and kinematic modeling, control algorithms, mechanisms for mapping and localization, trajectory planning and the platform simulator.

2. Robot design and application

The robot body and its sensors and actuators are heavily influenced by both the application and environment. Together, they determine the project and impose restrictions. The process of developing the robot body is very creative and defies the designer to skip steps of a natural evolutionary process and achieve the best solution. As such, the success of a robot

project depends on the development team, on the clear vision of the environment and its restrictions, and on the existence purpose of the robot. Many are the aspects which determine the robot structures. The body, the embedded electronics and the software modules are the result of a creativity intensive process of a team composed by specialists from different fields. On the majority of industrial applications, a mobile robot can be reused several times before its disposal. However, there are applications where the achievement of the objectives coincides with the robot end of life. Such applications may be the exploration of planets or military missions such as bomb disposal.

The project of a robot body is initially submitted to a critical analysis of the environment and its existence purposes. The environment must be studied and treated according to its complexity and also to the previous knowledge about it. Thus, the environment provides information that establishes the drive system in face of the obstacles it will find. Whether it is an aerial, aquatic or terrestrial, it implies the study of the most efficient structure for the robot locomotion trough the environment. It is important to note that the robot body project may require the development of its aesthetics. This is particularly important to robots that will subsist with humans.

The most common drive systems for terrestrial mobile robots are composed by wheels, legs or continuous track. The aerial robots are robotic devices that can fly in different environment; generally this robots use propellers to move. The aquatic robots can move under or over water. Some examples for these applications are: the AirRobot UK® (Figure 1a), an aerial quad rotor robot (AirRobot, 2011); the Protector Robot, (Figure 1b), built by Republic of Singapore with BAE Systems, Lockheed Martin and Rafael Enterprises (Protector, 2010); and the *BigDog* robot (Figure 1c), created by Boston Dynamics (Raibert et al., 2011), a robot that walks, runs and climbs in different environment.
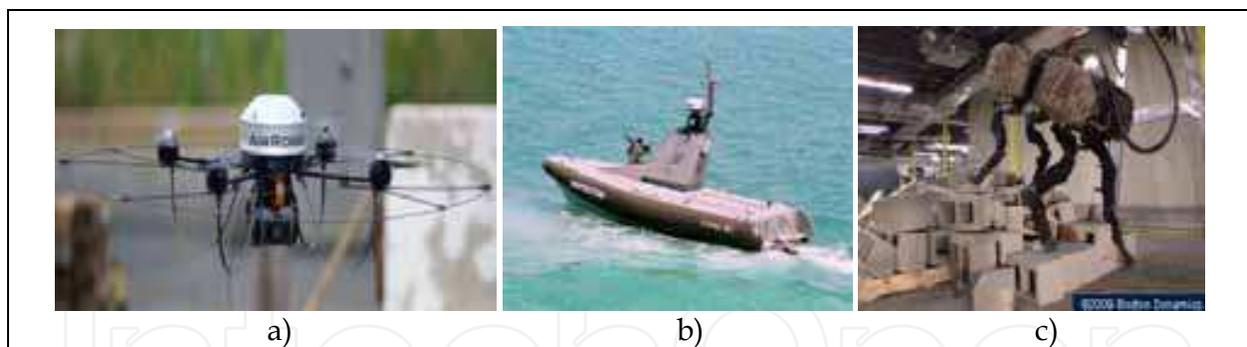

Fig. 1. Applications of Robot Navigation: a) Aerial Robot, b) Aquatic Robot, c)Terrestrial Robot

There are two development trends: one declares that the project of any autonomous system must begin with an accurate definition of its task (Harmon, 1987), and the other proclaims that a robot must be able to perform any task in different environments and situations (Noreils & Chatila, 1995). The current trend on industry is the specialization of robot systems, which is due by two factors: the production cost of general purpose robot is high, as it requires complex mechanical, electronic and computational systems; and a robot is generally created to execute a single task – or a task "class" – during its life cycle, as seem in Automated Guidance Vehicles (AVG). For complex tasks that require different sensors and actuators, the current trend is the creation of a robot group where each member is specialist on a given sub-task, and their combined action will complete the task.

2.1 Robot Navigation systems and methods

Navigation is the science or art of guiding of a mobile robot in the sense of how travel through the environment (McKerrow, 1991). The problems related to the navigation can be briefly defined in three questions: "Where am I", "Where am I going" and "How do I get there?" (Leonard & Durrant-White, 1991). The first two questions may be answered by an adequate sensorial system, while the third question needs an effective planning system. The navigation systems are directly related to the sensors available on the robot and the environment structure. The definition of a navigation system, just like any aspect of the robot we have seem so far, is influenced by the restrictions imposed by both the environment and the robot very purpose. The navigation may be obtained by three systems: a coordinates based system, a behavior based system and a hybrid system.

The coordinates based system, like the naval navigation, uses the knowledge of one's position inside a global coordinate system of the environment. It is based on models (or maps) of the environment to generate paths to guide the robot. Some techniques are Mapping (Latombe, 1991), Occupancy Grid Navigation (Elfes, 1987), and Potential Fields (Arkin et al., 1987). The behavior based system requires the robot to recognize environment features through its sensors and use the gathered information to search for its goals. For example, the robot must be able to recognize doors and corridors, and know the rules that will lead it to the desired location. In this case, the coordinate system is local (Graefe & Wershofen, 1991).

Information about the statistical features of the environment is important to both cited systems. The modeling of the environment refers to the representation of objects and the data structure used to store the information (the maps). Two approaches for map building are the geometric and phenomenological representation. Geometric representation has the advantage of having a clear and intuitive relation to the real world. However, the geometric representation has no satisfactory representation of uncertain geometries, as well as is not clear if knowing the world shape is really useful (Borenstein et al., 1995). The phenomenological representation is an attempt to overcome this problem. It uses a topological representation of the map with relative positioning which is based on local reference frames to avoid the accumulation of relative errors. Whenever the uncertainty grows too high, the robot sets a new reference frame; on the other hand, if the uncertainty decreases, the robot may merge frames. This policy keeps the uncertainty bound locally (Borenstein et al., 1995, as cited in Engelson & McDermott, 1992).

Mobile robots can navigate using relative or absolute position measures (Everett, 1995). Relative positioning uses odometry or inertial navigation. Odometry is a simple and inexpensive navigation system; however it suffers from cumulative errors. The inertial navigation (Barshan & Durrant-White, 1995) uses rotation and acceleration measures for extracting positioning information. Barshan and Durrant-White (1995) presented an inertial navigation system and discusses the challenges related to mobile robot movement based on non-absolute sensors. The most concerning issue is the accumulation of error found in relative sensors. The absolute positioning system can use different kinds of sensors which are divided in four groups of techniques: magnetic compass, active beacons, landmark recognition and model matching. Magnetic compasses are a common kind of sensor which uses Earth's natural electromagnetic field and does not require any change on the environment to be able to navigate through the world. Nevertheless, magnetic compasses readings are affected by power lines, metal structures, and even the robot movement, which introduces error to the system (Ojeda & Borenstein, 2000).

Active beacons are devices which emits a signal that is recognized by the robot. Since the active beacons are placed in known locations, the robot is able to estimate its position using triangulation or trilateration methods. In a similar fashion, the landmark system uses features of the environment to estimate its position. These landmarks may be naturally available on the environment, such as doors, corridors and trees; or they can be artificially developed and place on the environment, such as road signs. On one hand, natural landmarks do not require any modification in the world, but may not be easily recognized by the robot. On the other hand, artificial landmarks modifies the environment, but offer best contrast and are generally easier to be recognized. Nonetheless, the main problem with landmarks is detecting them accurately through sensorial data. Finally, the model matching technique uses features of the environment for map building or to recognize an environment within a previously known map. The main issues are related to finding the correspondence between a local map, discovered with the robot sensors, and a known global map (Borenstein et al., 1995).

Inside model matching techniques, we can point out the Simultaneous Localization and Mapping (SLAM). The SLAM addresses to the problem of acquiring the map of the environment where the robot is placed while simultaneously locating the robot in relation to this map. For this purpose, it involves both relative and absolute positioning techniques. Still, SLAM is a broad field and leaves many questions unanswered – mainly on SLAM in non-structured and dynamic environments (Siciliano & Khatib, 2008).

Other approach for mobile robot navigation is the biomimetic navigation. Some argue that the classic navigation methods developed on the last decades have not achieved the same performance flexibility of the navigation mechanisms of ants or bees. This has led researchers to study and implement navigation behaviors observed on biological agents, mainly insects. Franz and Mallot (Franz & Mallot, 2000) surveyed the recent literature on the phenomena of mobile robot navigation. The authors divide the techniques of biomimetic navigation into two groups: local navigation and path discovery. The local navigation deals with the basic problems of navigation, as the obstacle avoidance and track following, to move a robot from a start point (previously know or not) to a known destination inside the robot vision field. Most of recent researches objectives are to test the implementation of biological mechanisms, not to discover an optimal solution for a determined problem.

The navigation of mobile robots is a broad area which is currently the focus of many researchers. The navigation system tries to find an optimal path based on the data acquired by the robot sensors, which represents a local map that can be part, or not, of a global map (Feng & Krogh, 1990). To date, there is still no ideal navigation system and is difficulty to compare the results of researches, since there is a huge gap between the robots and the environment of each research (Borenstein et al., 1995). When developing the navigation system of a mobile robot, the designer must choose the best navigation methods for the robot application. As said by Blaasvaer et al. (1994): "each navigation context imposes different requirements about the navigation strategy in respect to precision, speed and reactivity".

2.2 Sensors

The mobile robots need information about the world so they can relate themselves to the environment, just like animals. For this purpose, they rely on sensor devices which

transform the world stimuli into electrical signal. These signals are electrical data which represents state of about the world and must be interpreted by the robot to achieve its goals. There is wide range of sensors used to this end.

Sensors can be classified by some features as the application, type of information and signal. As for their usage, sensors can be treated as proprioceptive or exteroceptive. Proprioceptive sensors are related to the internal elements of the robot, so they monitor the state of its inner mechanisms and devices, including joints positions. In a different manner, exteroceptive sensors gather information from the environment where the robot is placed and generally are related to the robot navigation and application. From the viewpoint of the measuring method, sensors are classified into active and passive sensors. Active sensors apply an known interfering signal into the environment and verify the effect of this signal into the world. Contrastingly, passive sensors does not provoke any interfering signal to measure the world, as they are able to acquire "signals" naturally emitted by the world. Sensors can also be classified according to the electrical output signal, thus are divided into digital and analog sensors. In general, sensorial data is usually inaccurate, which raises the difficulty of using the information provided by them.

The sensor choice is determined by different aspects that may overlap themselves or are conflicting. The main aspects are: the robot goals, the accuracy of the robot and environment models, the uncertainty of sensor data, the overall device cost, the quantity of gathered information, the time available for data processing, the processing capabilities of the embedded computer (on-board), the cost of data transmission for external data processing (off-board), the sensors physical dimension in contrast to the required robot dimension, and the energy consumption.

In respect to the combined use of sensor data, there is not a clear division of data integration and fusion processes. Searching for this answer, Luo (Luo & Kay, 1989) presented the following definition: "The fusion process refers to the combination of different sensors readings into an uniform data structure, while the integration process refers to the usage of information from several sensor sources to obtain a specific application".

The arrangement of different sensors defines a sensor network. This network of multiple sensors when combined (through data fusion or integration) functions as a single, simple sensor, which provides information about the environment. An interesting taxonomy for multiple sensor networks is presented by Barshan and Durrant-White (1995) and complemented by Brooks and Iyengar (1997): *Complementary*: There is no dependency between the sensors, however they can be combined to provide information about a phenomena, *Competitive*: The sensors provides independent measures of the same phenomena, which reduces the inconsistency and uncertainties about of the information, *Cooperative*: different sensors which works together to measure a phenomena that a single sensor is not capable of measuring, *Independent*: independent sensors are those of which measures does not affect or complement other sensor data.

2.2.1 Robot Navigation sensors

When dealing with Robot Navigation, sensors are usually used for positioning and obstacle avoidance. In the sense of positioning, sensors can be classified as relative or absolute (Borenstein et al., 1995). Relative positioning sensors includes odometry and inertial navigation, which are methods that measures the robot position in relation to the robot initial point and its movements. Distinctively, absolute positioning sensors recognize structures on the environment which position is known, allowing the robot to estimate its own position.

Odometry uses encoders to measure the rotation of the wheels, which allows the robot to estimate its position and heading according to its model. It's the most available navigation system due to its simplicity, the natural availability of encoders, and low cost. Notwithstanding, this is often a poor method for localization. The rotation measuring may be jeopardized by the inaccuracy of the mechanical structure or by the dynamics of the interaction between the tire and the floor, such as wheel slippage. On the other hand, the position estimation takes into account all past estimations - it integrates the positioning. This means that the errors are also integrated and will grow over time, resulting in a high inaccurate system. Just like odometry, inertial navigation, which uses gyroscopes and accelerometers to measure rotation and acceleration rates, is highly inaccurate due to its integrative nature. Other drawback is the usual high costs of gyroscopes and accelerometers. The heading measure provided by compasses represents one of the most meaningful parameters for navigation. Magnetic compasses provides an absolute measure of heading and can function on virtually all of Earth surface, as the natural magnetic field of Earth is available on all of its surface. Nevertheless, magnetic compasses are influenced by metallic structures and power lines, becoming highly inaccurate near them.

Other group of sensors for navigation are the active beacons. These sensors provide absolute positioning for mobile robots through the information sent by three or more beacons. A beacon is a source of known signal, as structured light, sound or radio frequencies. The position is estimated by triangulation or trilateration. It is a common positioning system used by ships and airplanes due to its accuracy and high speed measuring. The Global Positioning System (GPS) is an example of active beacon navigation system.

The map based positioning consists in a method where robots use its sensors to create a local map of the environment. This local map is compared to a known global map stored on the robot memory. If the local map matches a part of the global map, then the robot will be able to estimate its position. The sensors used in this kind of system are called Time-of-Flight Range Sensors, which are active sensors that measures the distance of nearby objects. Some widely used sensors are sonars and LIDAR scanners (Kelly, 1995).

As sensor industry advances in high speed, this chapter does not cover all sensors available in the market. There are other sensors which may be used for navigation, as odor sensors (Russel, 1995, Deveza et al., 1994) for active beacon navigation, whereas the beacon emits an odor.

2.3 Control architectures for navigation

A mobile robot with a high degree of autonomy is a device which can move smoothly while avoiding static and mobile obstacles through the environment in the pursuit of its goal without the need for human intervention. Autonomy is desirable in tasks where human intervention is difficult (Anderson, 1990), and can be accessed through the robot efficiency and robustness to perform tasks in different and unknown environments (Alami et al., 1998), or the ability to survive in any environment (Bisset, 1997), responding to expected and unexpected events both in time and space, with the presence of independent agents or not (Ferguson, 1994).

To achieve autonomy, a mobile robot must use a control architecture. The architecture is closely linked to how the sensor data are handled to extract information from the world and how this information is used for planning and navigating in pursuit of the objectives, besides involving technological issues (Rich & Knight, 1994). It is defined by the principle of operation of control modules, which defines the functional performance of the architecture, the information and control structures (Rembold & Levi, 1987).

For mobile robots, the architectures are defined by the control system operating principle. They are constrained at one end by fully reactive systems (Kaelbling, 1986) and, in the other end, by fully deliberative systems (Fikes & Nilsson, 1971). Within the fully reactive and deliberative systems, lies the hybrid systems, which combines both architectures, with greater or lesser portion of one or another, in order to generate an architecture that can perform a task. It is important to note that both the purely reactive and deliberative systems are not found in practical applications of real mobile robots, since a purely deliberative systems may not respond fast enough to cope with the environment changes and a purely reactive system may not be able to reach a complex goal, as will be discussed hereafter.

2.3.1 Deliberative architectures

The deliberative architectures use a reasoning structure based on the description of the world. The information flow occurs in a serial format throughout its modules. The handling of a large amount of information, together with the information flow format, results in a slow architecture that may not respond fast enough for dynamic environments. However, as the performance of computer rises, this limitation decreases, leading to architectures with sophisticated planners responding in real time to environmental changes.

The CODGER (Communication Database with Geometric Reasoning) was developed by Steve Shafer et al. (1986) and implemented by the project NavLab (Thorpe et al., 1988). The Codger is a distributed control architecture involving modules that revolves a database. It distinguishes itself by integrating information about the world obtained from a vision system and from a laser scanning system to detect obstacles and to keep the vehicle on the track. Each module consists on a concurrent program. The Blackboard implements an AI (Artificial Intelligence) system that consists on the central Database and knows all other modules capabilities, and is responsible for the task planning and controlling the other modules. Conflicts can occur due to competition for accessing the database during the performance of tasks by the various sub-modules. Figure 2 shows the CODGER architecture.
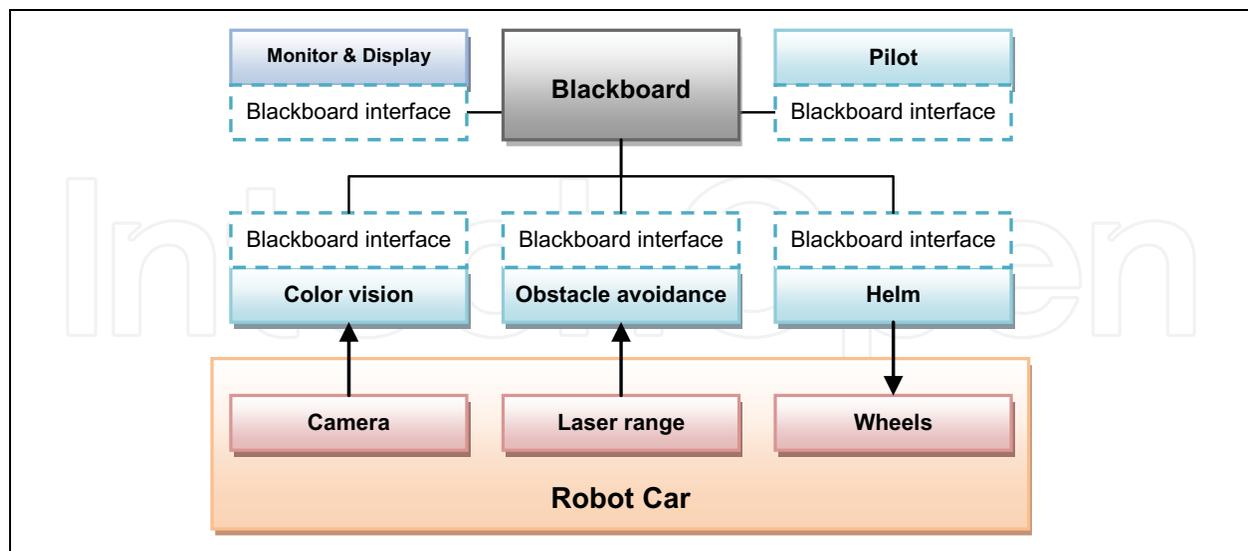

Fig. 2. CODGER Architecture on NavLab project (Thorpe et al., 1988)

The NASREM (NASA/NBS Standard Reference Model for Telerobot Control System Architecture) (Albus et al., 1987; Lumia, 1990) developed by the NASA/NBS consortium, presents systematic, hierarchical levels of processing creating multiple, overlaid control

loops with different response time (time abstraction). The lower layers respond more quickly to stimuli of input sensors, while the higher layers answer more slowly. Each level consists of modules for task planning and execution, world modeling and sensory processing (functional abstraction). The data flow is horizontal in each layer, while the control flow through the layers is vertical. Figure 3. represents the NASREM architecture.
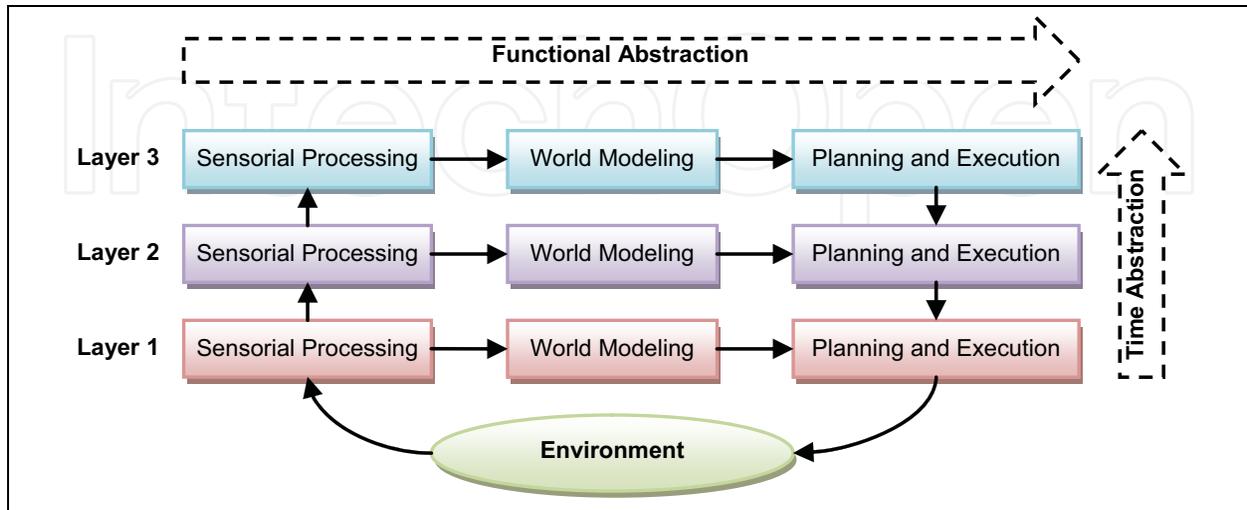

Fig. 3. NASREM architecture

2.3.2 Behavioral architectures

The behavioral architectures have as their reference the architecture developed by Brooks and thus follow that line of architecture (Gat, 1992; Kaelbling, 1986). The Subsumption Architecture (Brooks, 1986) was based on the constructive simplicity to achieve high speed of response to environmental changes. This architecture had totally different characteristics from those previously used for robot control. Unlike the AI planning techniques exploited by the scientific community of that time, which searched for task planners or problem solvers, Brooks (Brooks, 1986) introduced a layered control architecture which allowed the robot to operate with incremental levels of competence. These layers are basically asynchronous modules that exchange information by communication channels. Each module is an example of a finite state machine. The result is a flexible and robust robot control architecture, which is shown in Figure 4.
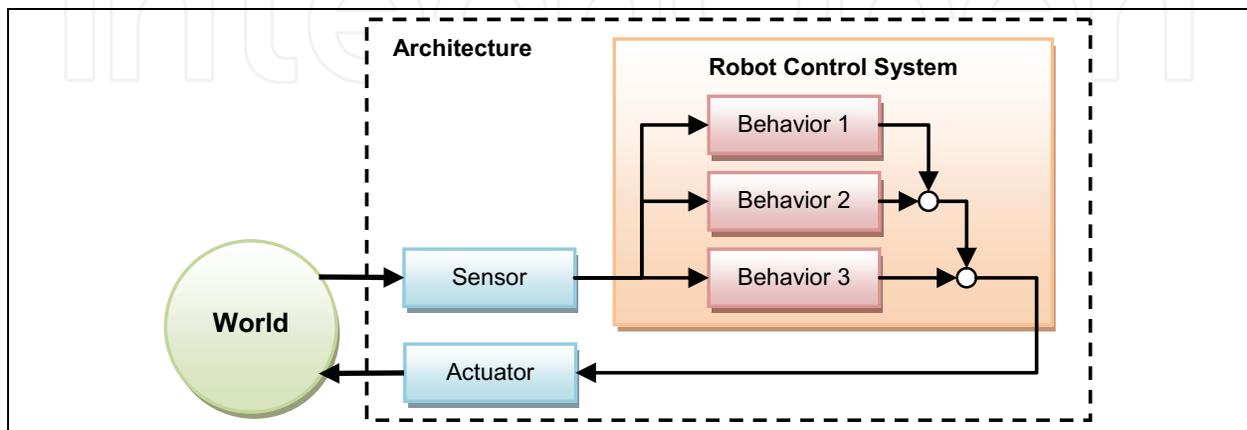

Fig. 4. Functional diagram of an behavioral architecture

Although the architecture is interesting from the point of view of several behaviors concurrently acting in pursuit of a goal (Brooks, 1991), it is unclear how the robot could perform a task with conflicting behaviors. For example, in a objects stacking task, the Avoiding Obstacles layer would repel the robot from the stack and therefore hinder the task execution, but on the other hand, if this layer is removed from the control architecture, then the robot would be vulnerable to moving or unexpected objects. This approach successfully deals with uncertainty and unpredictable environmental changes. Nonetheless, it is not clear how it works when the number of tasks increases, or when the diversity of the environment is increased, or even how it addresses the difficulty of determining the behavior arbitration (Tuijman et al., 1987; Simmons, 1994).

A robot driven only by environmental stimuli may never find its goal due to possible conflicts between behaviors or systemic responses that may not be compatible with the goal. Thus, the reaction should be programmable and controllable (Noreils & Chatila, 1995). Nonetheless, this architecture is interesting for applications that have restrictions on the dimensions and power consumption of the robot, or the impossibility of remote processing.

2.3.3 Hybrid architectures

As discussed previously, hybrid architectures combine features of both deliberative and reactive architectures. There are several ways to organize the reactive and deliberative subsystems in hybrid architectures, as saw in various architectures presented in recent years (Ferguson, 1994; Gat, 1992; Kaelbling, 1992). Still, there is a small community that research on the approach of control architectures in three hierarchical layers, as shown on Figure 5.
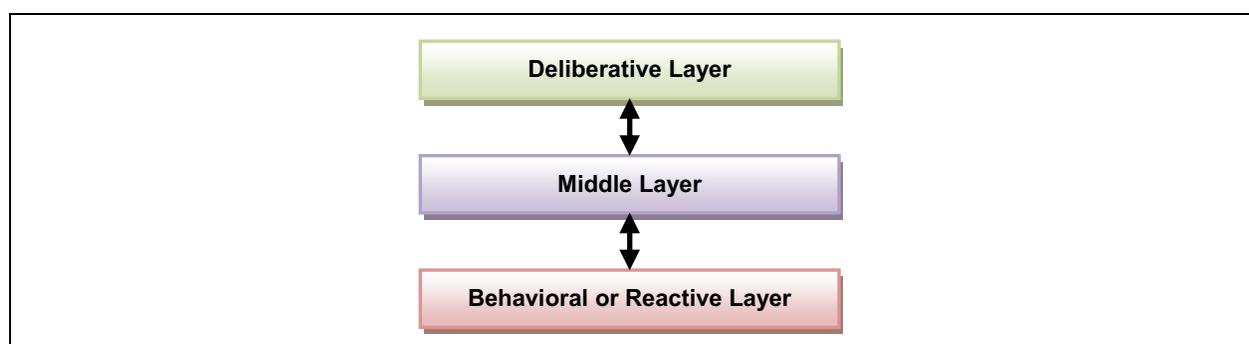

Fig. 5. Hybrid architecture in three layers

The lowest layer operates according to the behavioral approach of Brooks (Brooks, 1986) or are even purely reactive. The higher layer uses the planning systems and the world modeling of the deliberative approach. The intermediate layer is not well defined since it is a bridge between the two other layers (Zelek, 1996).

The RAPs (Reactive Action Packages) architecture (Firby, 1987) is designed in three layers combining modules for planning and reacting. The lowest layer corresponds to the skills or behaviors chosen to accomplish certain tasks. The middle layer performs the coordination of behaviors that are chosen according to the plan being executed. The highest layer accommodates the planning level based on the library of plans (RAP). The basic concept is centered on the RAP library, which determines the behaviors and sensorial routines needed to execute the plan. A reactive planner employ information from a scenario descriptor and the RAP library to activate the required behaviors. This planner also monitors these behaviors and changes them according to the plan. Figure 6 illustrates this architecture.
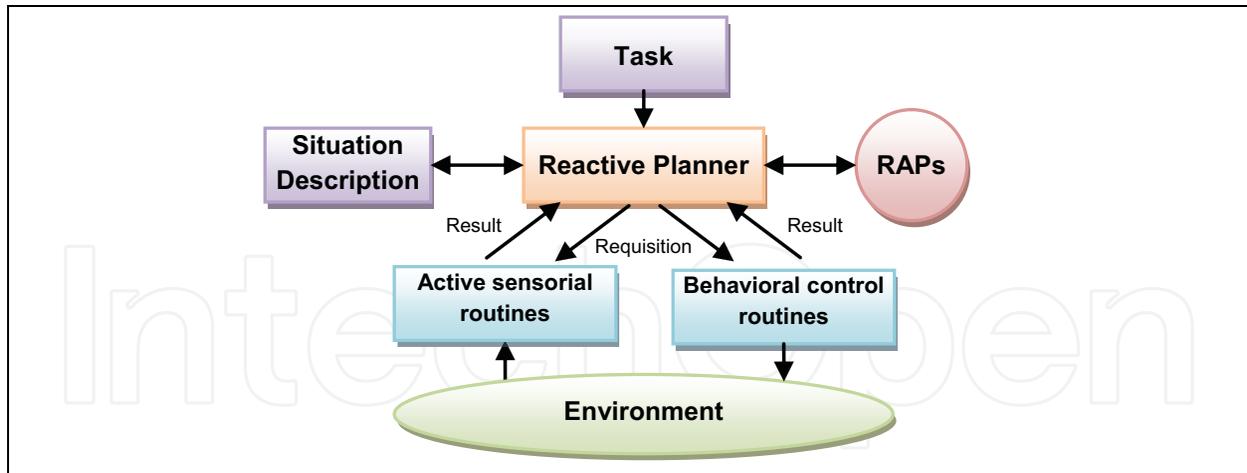
Fig. 6. RAPs architecture

The TCA (Task Control Architecture) architecture (Simmons, 1994) was implemented in the AMBLER robot, a robot with legs for uneven terrain (Krotkov, 1994). Simmons introduces deliberative components performing with layered reactive behavior for complex robots. In this control architecture, the deliberative components respond to normal situations while the reactive components respond to exceptional situations. Figure 7 shows the architecture. Summarizing, according to Simmons (1994): "The TCA architecture provides a comprehensive set of features to coordinate tasks of a robot while ensuring quality and ease of development".
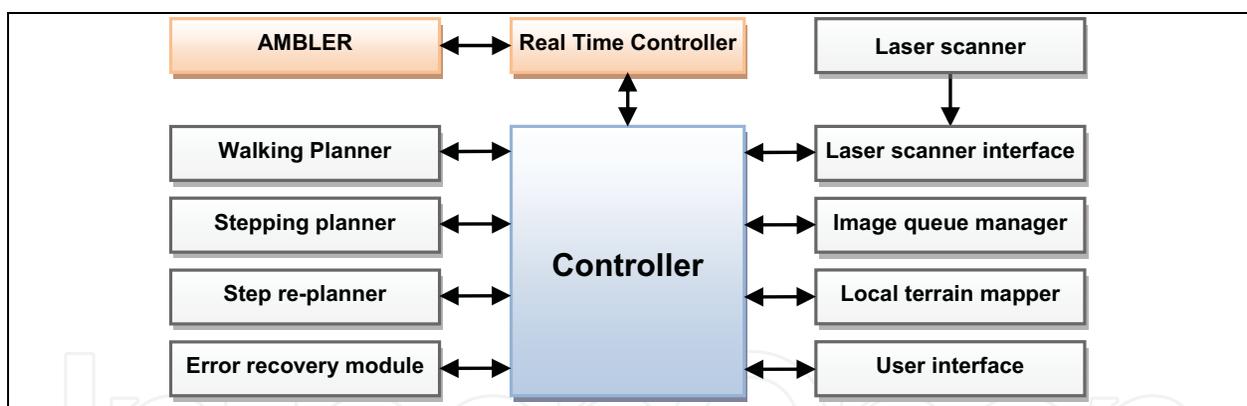
Fig. 7. TCA architecture

2.3.4 The choice of achitecture

The discussion on choosing an appropriate architecture is within the context of deliberative and behavioral approaches, since the same task can be accomplished by different control architectures. A comparative analysis of results obtained by two different architectures performing the same task must consider the restrictions imposed by the application (Ferasoli Filho, 1999). If the environment is known or when the process will be repeated from time to time, the architecture may include the use of maps, or get it on the first mission to use on the following missions. As such, the architecture can rely on deliberative approaches. On the other hand, if the environment is unknown on every mission, the use or creation of maps is not interesting – unless the map building is the mission goal. In this context, approaches based on behaviors may perform better than the deliberative approaches.

3. Dynamics and control

3.1 Kinematics model

The kinematics study is used for the design, simulation and control of robotic systems. This modeling is defined as the movement of bodies in a mechanism or robot system, without regard to the forces and torques that cause the movement (Waldron & Schmiedeler, 2008). The kinematics provides a mathematical analysis of robot motion without considering the forces that affect it. This analysis uses the relationship between the geometry of the robot, the control parameters and the system behavior in an environment. There are different representations of position and orientation to solve kinematics problems. One of the main objectives of the kinematics study is to find the robot velocity as a function of wheel speed, rotation angle, steering angles, steering speeds and geometric parameters of the robot configuration (Siegwart & Nourbakhsh, 2004). The study of kinematics is performed with the analysis of robot physical structure to generate a mathematical model which represents its behavior in the environment. The mobile robot can be distinguished by different platforms and an essential characteristic is the configuration and geometry of the structure body and wheels. The mobile robots can be divided according to their mobility. The maneuverability of a mobile robot is the combination of the mobility available, which is based on the sliding constraints and the features by the steering (Siegwart & Nourbakhsh, 2004). The robot stability can be expressed by the center of gravity, the number of contact points and the environment features. The kinematic analysis for navigation represents the robot location in the plane, with local reference frame $\{X_L, Y_L\}$ and global reference frame $\{X_G, Y_G\}$. The position of the robot is given by X_L and Y_L and orientation by the angle θ. The complete location of the robot in the global frame is defined by

$$\xi^T = \begin{bmatrix} x & y & \theta \end{bmatrix} \tag{1}$$

The kinematics for mobile robot requires a mathematical representation to describe the translation and rotation effects in order to map the robot's motion in tracking trajectories from the robot's local reference in relation to the global reference. The translation of the robot is defined as a P^G vector that is composed of two vectors which are represented by coordinates of local (P^L) and global (Q_0^G) reference system expressed as

$$P^G = Q_0^G + P^L, \quad Q_0^G = \begin{bmatrix} x_0 \\ y_0 \\ \theta \end{bmatrix} \quad P^L = \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \quad P^G = \begin{bmatrix} x_0 + x_1 \\ y_0 + y_1 \\ \theta + 0 \end{bmatrix} \tag{2}$$

The rotational motion of the robot can be expressed from global coordinates to local coordinates using the orthogonal rotation matrix (Eq.3)

$$R_G^L(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The mapping between the two frames is represented by:

$$\dot{\xi}_L = R_G^L(\theta)\dot{\xi}_G = R_G^L(\theta).\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\dot{x} + \sin(\theta)\dot{y} \\ -\sin(\theta)\dot{x} + \cos(\theta)\dot{y} \\ \dot{\theta} \end{bmatrix} \tag{4}$$

The kinematics is analyzed through two types of study: the forward kinematics and the inverse kinematics. The *forward kinematics* describes the position and orientation of the robot, this method uses the geometric parameters β_i, the speed of each wheel α_i, and the steering, expressed by

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\alpha}_1 ... \dot{\alpha}_n, \beta_1 ... \beta_m, \dot{\beta}_1 ... \dot{\beta}_m), \tag{5}$$

The *inverse kinematics* predicts the robot caracteristics as wheels velocities, angles and other geometrical parameters through the calculation of the final speed and its orientation angle:

$$\left[\dot{\alpha}_1 ... \dot{\alpha}_n, \beta_1 ... \beta_m, \dot{\beta}_1 ... \dot{\beta}_m \right] = f(\dot{x}, \dot{y}, \dot{\theta}) \tag{6}$$

In the kinematic analysis, the robot characteristics such as the type of wheels, the points of contact, the surface and effects of sliding or friction should be considered.

3.1.1 Kinematics for two-wheel differential robot

In the case of a two-wheeled differential robot, as presented in Figure 8, each wheel is controlled by an independent motor X_G and Y_G represents the global frame, while X_L and Y_L represents the local frame. The robot velocity is determined by the linear velocity $V_{robot}(t)$ and angular velocity $\omega_{robot}(t)$, which are functions of the linear and angular velocity of each wheel $\omega_i(t)$ and the distance L between the two wheels, $V_r(t)$, $\omega_r(t)$ are the linear and angular velocity of right wheel, $V_l(t)$, $\omega_l(t)$ are the linear and angular velocity of left wheel, θ is the orientation of the robot and the (r_l, r_r) are left and right wheels radius.
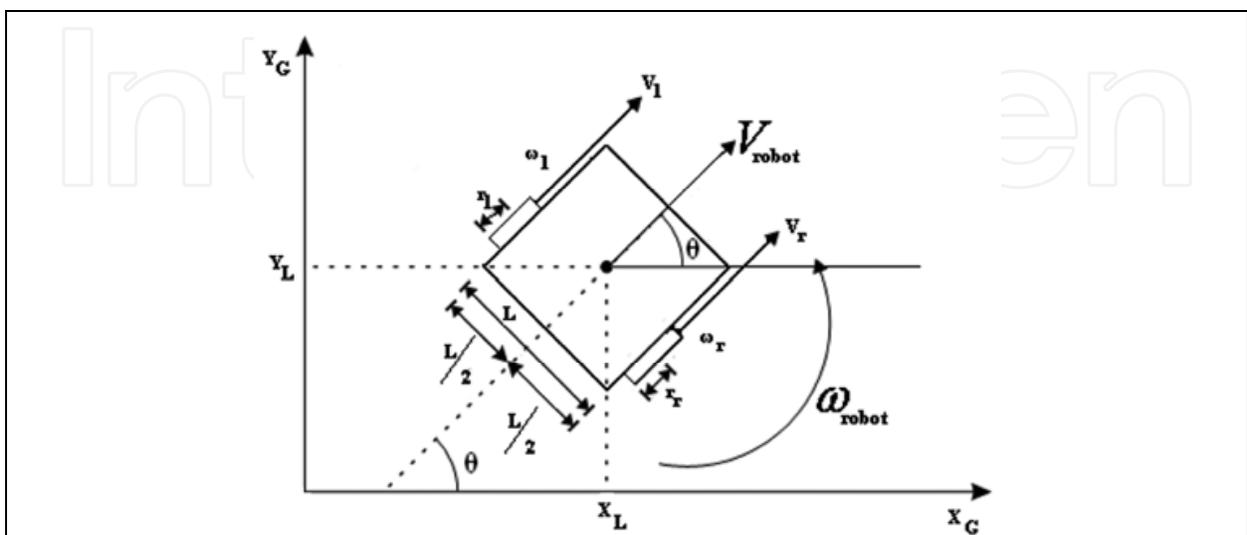

Fig. 8. Two-wheeled differential robot

The linear speed of each wheel is determined by the relationship between angular speed and radius of the wheel as

$$V_r(t) = \omega_r(t)r_r \, , \quad V_l(t) = \omega_l(t)r_l \tag{7}$$

The robot velocities are composed of the center of mass's linear velocity and angular velocity generated by the difference between the two wheels.

$$V_l(t) = V_{robot}(t) - \left(\frac{L}{2}\right)\omega_{robot}(t) \, , \quad V_R(t) = V_{robot}(t) + \left(\frac{L}{2}\right)\omega_{robot}(t) \tag{8}$$

The robot velocities equations are expressed by

$$V_{robot} = \frac{V_r + V_l}{2} \, , \quad \omega_{robot} = \frac{V_r - V_l}{L} \tag{9}$$

The kinematics equations of the robot are expressed on the initial frame (Eq. 10a) and in local coordinates (Eq, 10b) by

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{robot} \\ \omega_{robot} \end{bmatrix} \, , \quad \begin{bmatrix} \dot{x}_L(t) \\ \dot{y}_L(t) \\ \dot{\theta}_L(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_l(t) \\ \omega_r(t) \end{bmatrix} \tag{10}$$

$$\text{a)} \qquad\qquad\qquad\qquad\qquad \text{b)}$$

Therefore, with the matrix of the differential model shown in Eq. 10, it is possible to find the displacement of the robot. The speed in Y axis is always zero, demonstrating the holonomic constraint μ on the geometry of differential configuration. The holonomic constraint is explained by Eq.11, with $N(\theta)$ being the unit orthogonal vector to the plane of the wheels and p the robot velocity vector, it demonstrates the impossibility of movement on the Y axis, so the robot has to perform various displacements in X in order to achieve a lateral position.

$$\mu = 0 \rightarrow N(\theta).\dot{p} = \begin{bmatrix} \sin(\theta) & -\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0 \tag{11}$$

Finally, with the direct kinematics it is possible to obtain the equations that allow any device to be programmed to recognize at every moment its own speed, position and orientation based on information from wheel speed and steering angle.

3.1.2 Kinematics for three-wheeled omnidirectional robot

The omnidirectional robot is a platform made up of three wheels in triangular configuration where the distance between these wheels is symmetric. Each wheel has an independent motor and can rotate around its own axis with respect to the point of contact with the surface. The Figure 9 shows the three-wheeled omnidirectional robot configuration.

As seen of Figure 9, X_G and Y_G are the fixed inertial axes and represent the global frame. X_L and Y_L are the fixed axis on the local frame in the robot coordinates; d_0 describes the current position of the local axis in relation to the global axis, d_i describes the location of the center

of a wheel from the local axis. H_i are positive unit velocity vector of each wheel, θ describes the rotation axis of the robot X_{LR} and Y_{LR} compared to the global axis, α_i describes the rotation of the wheel in the local frame, β describes the angle between d_i and H_i. In order to obtain the kinematic model of the robot, the analysis of the speed of each wheel must be determined in terms of the local speed and its make the transformation to the global frame. The speed of each wheel has components in X and Y directions.
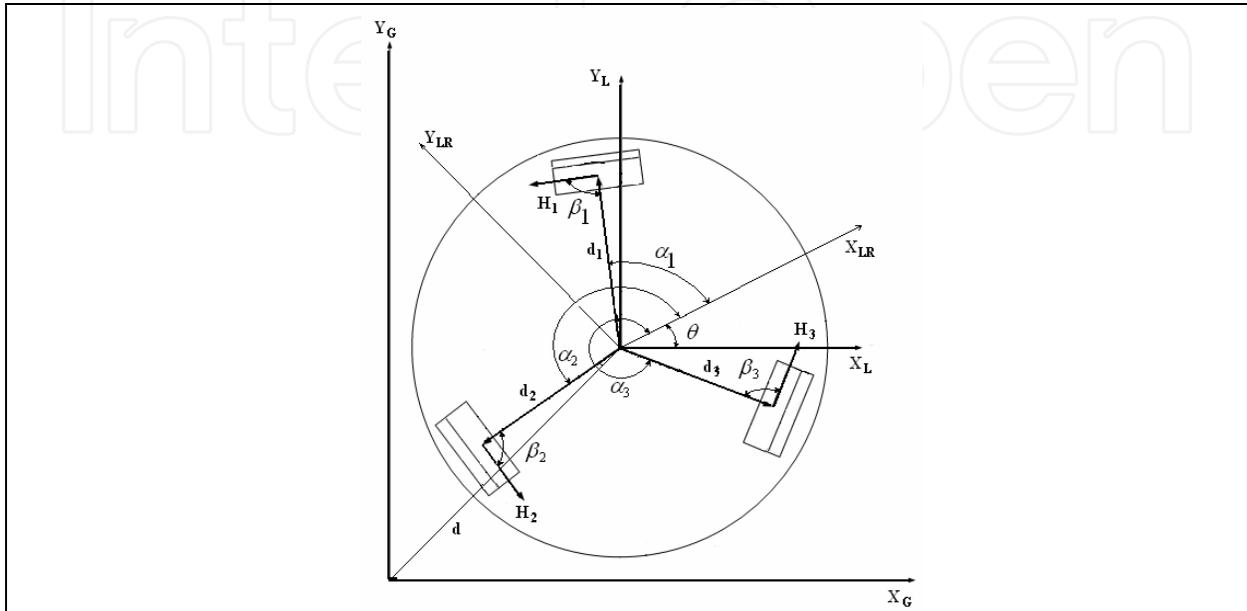
Fig. 9. Three-wheeled omnidirectional robot

The speed of each wheel is represented by the translation and rotation vectors in the robot frame. The position from the global frame P_0^G is added to the position transformation and orientation of the wheel. The rotation $R_L^G (\theta)$ is calculated from local frame to global frame. The transformation matrix is obtained and provides the angular velocity of each wheel in relation to the global frame speeds represented in Eq.12, (Batlle & Barjau, 2009).

$$P_i^G = P_0^G + R_L^G(\theta)\,{}^L P_i \,, \quad \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} v_1/r \\ v_2/r \\ v_3/r \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta+\alpha_1) & \cos(\theta+\alpha_1) & R \\ -\sin(\theta+\alpha_2) & \cos(\theta+\alpha_2) & R \\ -\sin(\theta+\alpha_3) & \cos(\theta+\alpha_3) & R \end{bmatrix} \begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\theta} \end{bmatrix} \quad (12)$$

3.2 Dynamic model

The study of the movement dynamics analyzes the relationships between the forces of contact and the forces acting on the robot mechanisms, in addition to the study of the acceleration and resulting motion trajectories. This study is essential for the design, control and simulation of robots (Siciliano & Khatib, 2008). The kinematic model relates to the displacement, velocity, acceleration and time regardless of the cause of their movement, whereas the dynamic analysis relates to the generalized forces from the actuators, with the energy applied in the system (Dudek, 2000). There are different proposals for the dynamic

model of robot navigation, but the general shape of the dynamic study is the analysis of the forces and torques produced inside and outside of the system. General equations of system motion, and the analysis of the system torques and energy allows developing the dynamic model of the robotic system. For this analysis, it is important to consider the physical and geometrical characteristics of the system such as masses, sizes, diameters, among others which are represented in the moments of inertia and static and dynamic torques of the system.

3.2.1 Dynamic model for robot joint

Each joint of a robot consists of a an actuator (DC motor, AC motor, step motor) associated with a speed reducer and transducers to measure position and velocity. These transducers can be absolute or incremental encoders at each joint. The motion control of robots is a complex issue, since the movement of the mechanical structure is accomplished through rotation and translation of their joints that are controlled simultaneously, which hinders the dynamic coupling. Moreover, the behavior of the structure is strongly nonlinear and dependent on operating conditions. These conditions must be taken into account in the chosen control strategy. The desired trajectory is defined by position, speed, acceleration and orientation of, therefore it is necessary to make coordinate transformations with set times and with great complexity of calculations. Normally the robot control on only considers the kinematic model, so joints are not coupled, and the control of each joint is independent. Each robotic joint commonly includes a DC motor, the gear, reducer, transmission, bearing and encoder. The dynamic model of DC motor is expressed by the electrical coupling and mechanical equation as

$$V(t) = Li(t) + Ri(t) + e(t)$$
$$T(t) = K_m i(t)$$
$$T(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) + T_r(t)$$

(13)

Where $i(t)$ is the current, R is the resistance, L is the inductance, $V(t)$ is the voltage applied to the armature circuit, $e(t)=k_e*\theta$ is the electromotive force, J and B are the moment of inertia and viscous friction coefficient, k_e and k_m are the electromotive torque coefficient and constant torque, T_r and T are the resistant torque due to system losses and mechanical torque. The joint model is shown in Figure 10.
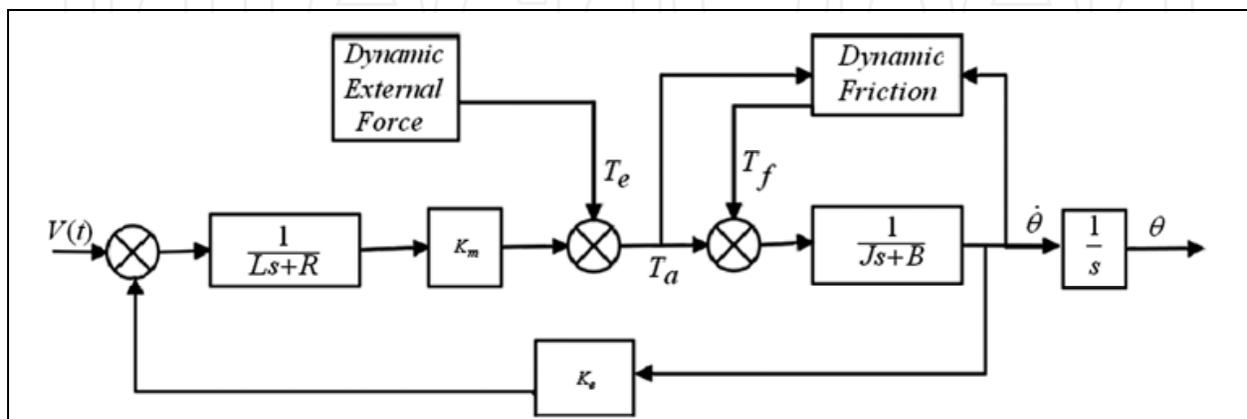

Fig. 10. Joint Model

The reduction model, with *η* as the rate of transmission, *p* as the teeth number of gear and *r* as the gear ratio, where the tangential velocity is the same between the gears. The system without slip can be expressed by

$$\eta = \frac{p_2}{p_1} \text{ and } \theta_2 = \frac{r_1}{r_2}\theta_1 \text{ , then } v = \dot{\theta}_1 r_1 = \dot{\theta}_2 r_2 \text{ for } \frac{\dot{\theta}_1}{\dot{\theta}_2} = \frac{r_2}{r_1} = \eta \tag{14}$$

The model presented above will be increased by the dynamic effect of reducing the loads of coupled system through the motor model and load-reducer as

$$(T(s) - T_r(s))G_2(s) = \Omega_{motor}(s) \text{ , } (T_{load}(s) - T_{per}(s))G_3(s) = \Omega_{load}(s)$$

$$\Omega_{load}(s) = \frac{1}{\eta}\Omega_{motor}(s) \qquad T_{load}(s) = \eta T_{motor}(s) \tag{15}$$

3.2.2 Two-Wheeled differential dynamic model

The dynamic analysis is performed for the Two-Wheeled differential robot (Fierro & Lewis, 1997). The movement and orientation is due to each of the actuators, where the robot position in a inertial Cartesian frame (O, X, Y) is the vector q = {xc, yc, θ}, X_c and Y_c are the coordinates of center of mass of the robot. The robot dynamics can be analyzed from the Lagrange equation, expressed in terms as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = \tau + \mathbf{J}^T(\mathbf{q})\lambda \text{ , } \qquad T(\mathbf{q},\dot{\mathbf{q}}) = T\frac{1}{2}\dot{\mathbf{q}}^T M(q)\dot{\mathbf{q}} \tag{16}$$

The kinematic constraints are independent of time, and the matrix **D** represents the full range for a group of linearly independent vectors and the H(q) is the matrix associated with constraints of the system. The equation of motion is expressed with V1 and V2 as the linear velocities of the system in Eq.17.

$$\dot{\mathbf{q}} = \mathbf{D}(\mathbf{q})\mathbf{v(t)} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -L\sin(\theta) \\ \sin(\theta) & L\cos(\theta) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \ \mathbf{D(q)} = \begin{bmatrix} \cos(\theta) & -L\cdot\sin(\theta) \\ \sin(\theta) & L\cdot\cos(\theta) \\ 0 & 1 \end{bmatrix}, \ \mathbf{v(t)} = \begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{17}$$

The relationship between the parameters of inertia, information of the centripetal force and Coriolis effect, friction on surfaces, disturbances and unmodeled dynamics is expressed as

$$\mathbf{M(q)\ddot{q}} + \mathbf{V_m(q,\dot{q})\dot{q}} + \mathbf{F(\dot{q})} + \mathbf{G(q)} + \boldsymbol{\tau_d} = \mathbf{B(q)\tau} - \mathbf{H^T(q)}\lambda \tag{18}$$

where M (q) is the inertia matrix, Vm (q, q) is the matrix of Coriolis effects, F (q) represents the friction of the surface, G (q) is the gravitational vector, Td represents the unknown disturbance including unmodelled dynamics. The dynamic analysis of the differential robot is a practical and basic model to develop the dynamic model of the omnidirectional robot. For the analysis of the dynamic model it is necessary to know the physical constraints of the system to get the array of restrictions, the matrix of inertia is expressed by the masses and

dimensions of the robot with the geometrical characteristics of three-wheeled omnidirectional system.

3.3 Control structure

Control for robots navigation has been developed primarily with the trajectory tracking, with the aim to follow follow certain paths by adjusting the speed and acceleration parameters of the system, which are generally described by position and velocity profiles. Control systems are developed in open loop or closed loop. Some problems of control systems in open loop are the limitations to regulate the speed and acceleration in different paths, this control does not correct the system to disturbances or dynamic changes, resulting in paths that are not smooth (Siegwart & Nourbakhsh, 2004). Systems in closed loop control can regulate and compare its parameters with references to minimize errors. The feedback control is used to solve the navigation system when the robot has to follow a path described by velocity and position profiles as a function of time from an initial to a final position.

3.3.1 Control structure for robot joint

The controller is an important element of a complete control system. The goal of a controller in the control loop is to compare the output value with a desired value, determining a deviation and producing a control signal to reduce the error to regulate dynamic parameters. This error is generated by the comparison of the reference trajectory and the robot's current path, which is represented in terms of position and orientation (Figure 11).
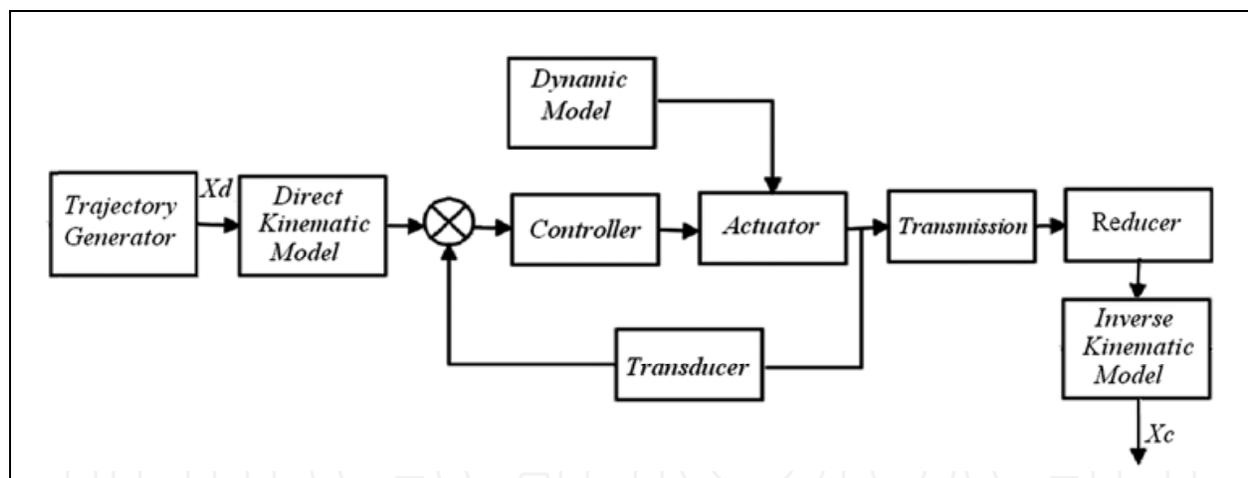

Fig. 11. Control Structure for Robot Joint

The control most used in robotic system is the PID Control that combines the Proportional (*Kp*), Integral (*Ki*) and Derivative (*Kd*) actions shown in the Eq. 19. This type of controller has good performance if the dynamic system is known and the controller parameter has been adjusted. The main limitation of a PID controller is the need to refine procedures of parameters adjustment, in addition it is very sensitive to the dynamic changes of the system.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \tag{19}$$

For setting parameters, different strategies in continuous time or discrete time can be applied as: Ziegler Nichols, Chien Hrones Reswick, and using control and stability analysis as Routh Hurwitz, Root Locus, Nyquist Criteria, Frequency response Analysis, and others.

4. Applications

Industry usually have a structured environment, which allows the integration of different mobile robots platforms. This chapter analyses the implementation of environments with robots in order to integrate multiple areas of knowledge. These environments applies the dynamic and kinematic model, control algorithms, trajectory planning, mechanisms for mapping and localization, and automation structures with the purpose of organizing the production chain, optimizing processes and reducing execution times.

4.1 Navigation robots platforms

The robots navigation platform uses one ASURO robot with hybrid control architecture AuRA (Mainardi, 2010), where the reactive layer uses motor-schemas based on topological maps for navigation. The environment perception is obtained through of signals from sensors. The ASURO robot has three types of sensors: contact, odometry and photosensors. Another robot of the platform is Robotino developed by FESTO. This robot has odometry, collision sensors and nine infrared distance sensors. Robotino is equipped with a Vision System, which consists of a camera to view images in real time. The modular structure allows the addition of new sensors and actuators. Both robots are shown in Figure 12. The odometry is performed with optical encoders. The phototransistors are used to detect the floor color while moving in a certain way. The robot navigates through line following and odometry.
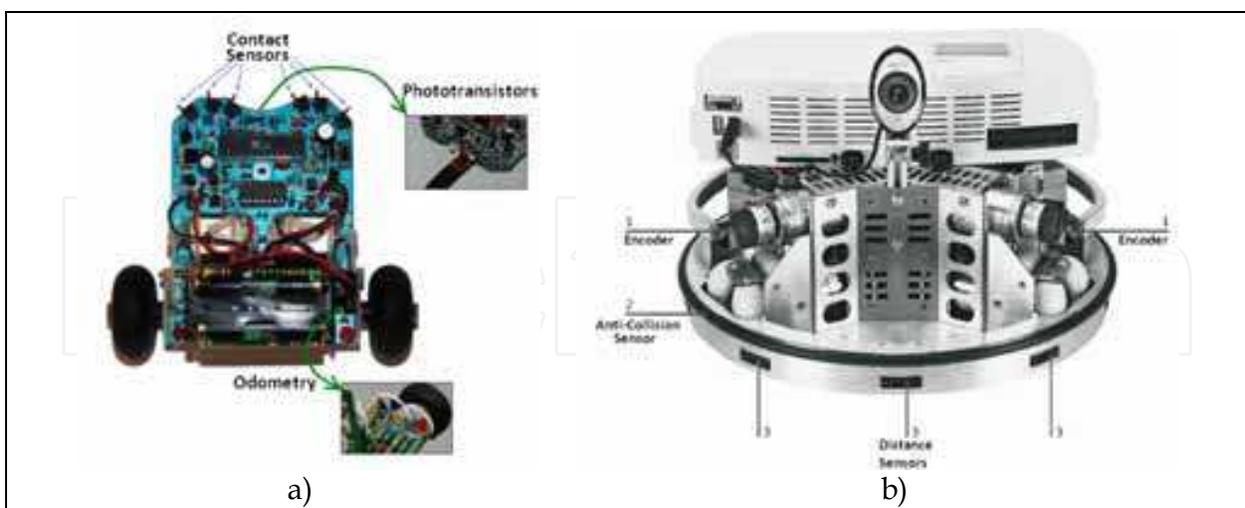

Fig. 12. Platform Robots (Mainardi, 2010): a)ASURO, b) Robotino

4.2 Mapping and location

The localization task uses an internal representation of the world as an map of environment to find the position through the environment perception around them. The topological maps divide the search space in nodes and paths (Figure 13). Mapping and location can guide the

robot in different environments, these methods give information about of objects in the space.
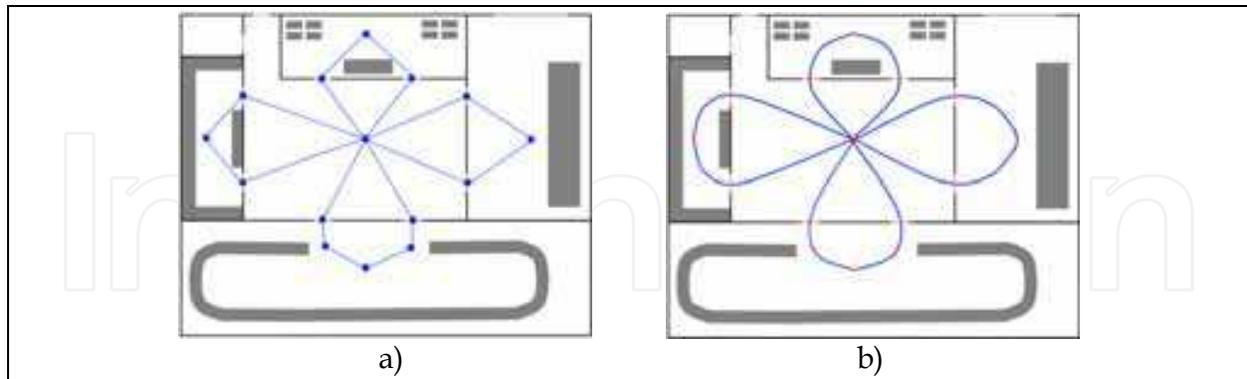

Fig. 13. a) Topological Map, b) Map with frame path (Mainardi,2010)

4.3 Path and trajectory planning

The path planning provides the points where the robot must pass. For this, the planning uses a search algorithm to analyze internal model of the world and find the best path, resulting in a sequence of coordinates to follow without colliding with known objects. For the purpose of determining the best path, the Dijkstra's algorithm is used to find the shortest path between all the nodes on the map. The discovered paths are then archived into a reminiscent table. Therefore, with the topological map of the environment shown in Figure 14, knowing the position and the goal, the path planning module access the reminiscent table to get the best path to the goal or, if necessary, it runs the Djikstra's algorithm to determine the best path. Finally, the path planning module applies two different techniques to generate the robot trajectories that compose the frame path. These techniques are the: Dubins path (geometric and algebraic calculations) and β-splines (polynomials parameterized).
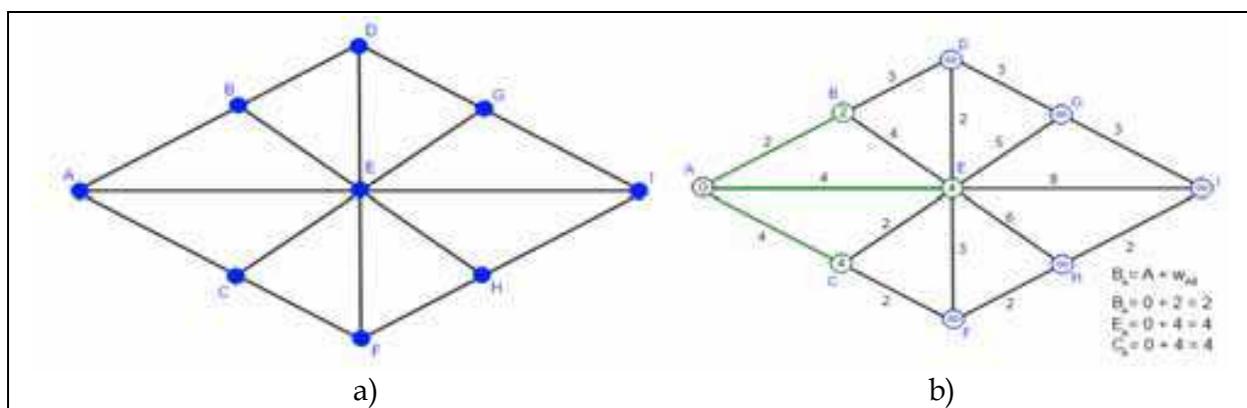

Fig. 14. a)Topological Map , b) Topological map with the weights

4.4 Trajectory execution

During the trajectory execution, the actuators are controlled based on the frame path and the environment perception. In this application, the execution of the trajectory will be conducted in a hybrid structure. The parameters are calculated in the path planning as the

basis to adjust the robot parameter through perception of the environment. The stage of path execution is performed by motor-schemas, which are divided into three distinct components: perceptual schemas, motor-schemas and vector sum. The perceptual schemas are mechanisms of sensorial data processing. The motor-schemas are used to process behaviors, where each schema (behavior) is based on the perception of the environment provided by the perceptual schemas. These schemas apply a result vector indicating the direction that the robot should follow. The sum vector adds the vectors of all schemas, considering the weight of each schema to find the final resultant vector. In this case, the weights of each schema change according to the aim of the schema´s controller. The control signal changes due to the different objectives or due to the environment perception. The wheels speeds V_R and V_L are determined by Eq. 20, where vr_i and vl_i are speeds in each behavior, and p_i is weight behavior of the current state.

$$V_R = \sum_{i=1}^{n} vr_i * p_i \ , \ V_L = \sum_{i=1}^{n} vl_i * p_i \tag{20}$$

The robot behavior to follow a black line on the floor will be informed of the distance between the robot and the line, and calculates the required speeds for each wheel to correct the deviation , applying the Eq.21, where l_W is line width, V_M is maximum desired speed, K_R is reactive gain. The speed on both wheels must be less than or equal to the V_M.

$$V_R = V_M + K_R * \Delta S * V_M * \frac{2}{l_W} \ , \quad V_L = V_M - K_R * \Delta S * V_M * \frac{2}{l_W} \tag{21}$$

The odometric perception is responsible of the calculatation of the robot displacement. The execution of the paths made by motor schemas is represented by Figure 15.
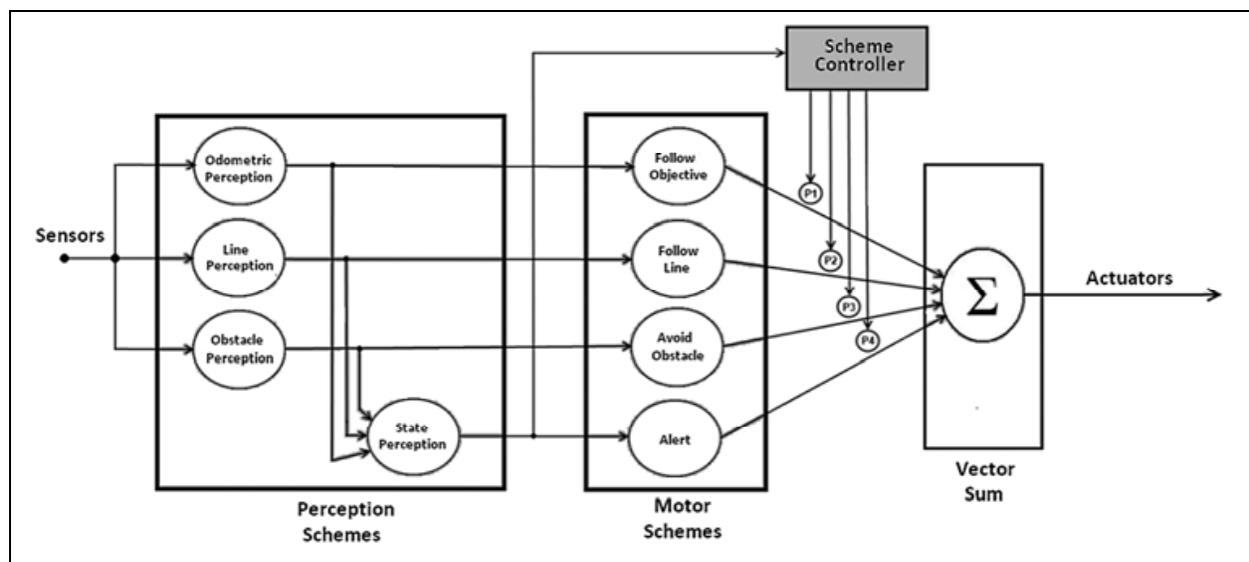

Fig. 15. Motor-schemas Structure (Mainardi,2010)

4.5 Trajectory and control simulator

The library DD&GP (Differential Drive and Global Positioning Blockset) of MATLAB®-Simulink is a simulation environment for dynamic modeling and control of mobile robotic

systems, this library is developed from the GUI (Graphical User Interface) in MATLAB®, that allows the construction and simulation of mobile robot positioning within an environment. This Blockset consists of seven functional blocks. The integration of these blocks allows the simulation of mobile differential robot based on their technological and functional specifications. The kinematic, dynamic and control system can be simulated with the toolbox, where the simulator input is the trajectory generation. The velocities of deliberative behavior are easily found via the path planning, but the reactive velocities behavior is necessary to include two blocks in the simulator, one to determine the distance between robot and desired trajectory (CDRT) and another to determine the velocities (reactive speeds), this simulator is presented in Figure 16. Blocks DD & GP can be used for the simulation of two-wheeled diferential robot. However, to simulate the Robotino robot of three-wheeled omnidirectional, it is necessary to modify some of the blocks considering the differences in the dynamics and kinematics of robots for two and three wheels. In this case, a PID control block was added to control the motor speed of each wheel, and blocks were added according to the equations of kinematics model of three-wheeled omnidirectional robot (Figure 17).
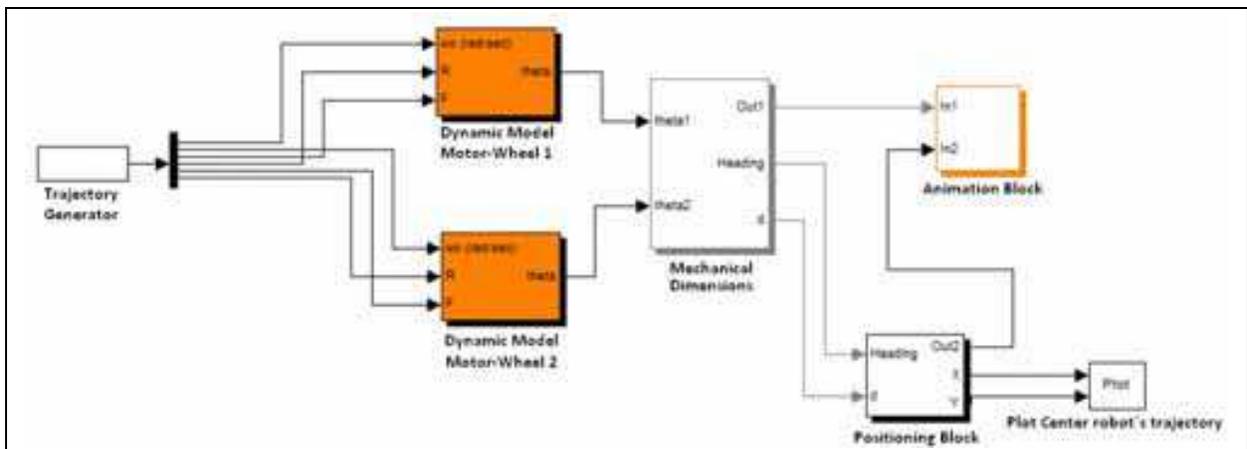

Fig. 16. Simulator with toolbox configuration in MATLAB® (Mainardi, 2010)
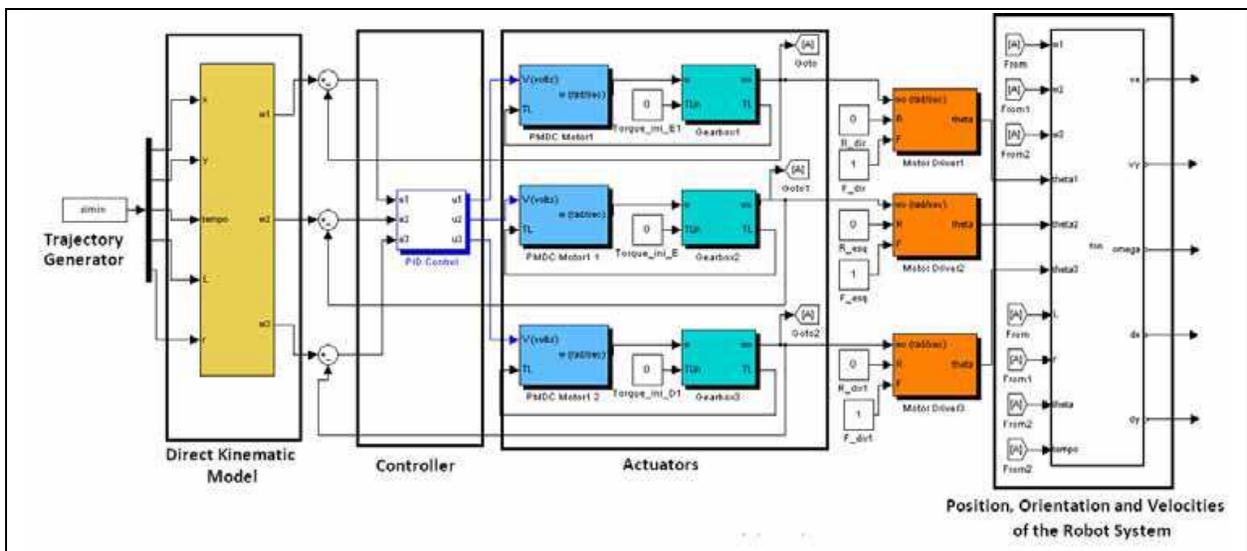

Fig. 17. Simulator for Three-Wheeled Omnidirectional

4.6 Results

First analysis simulated the path represented in Figure 13, with different number of control points to verify which control points respond better at proposed path. The simulations were realized with 17 points (Fig. 18a), 41 control points and 124 control points(Fig. 18b). To compare the results and set the values of desired weight p_d, reference weight p_r and reactive gain K_R, will use the square error in the simulation of paths, where the equation is the quadratic error shown in Eq. 22, where p_i is the current position and p_d is the desired position.

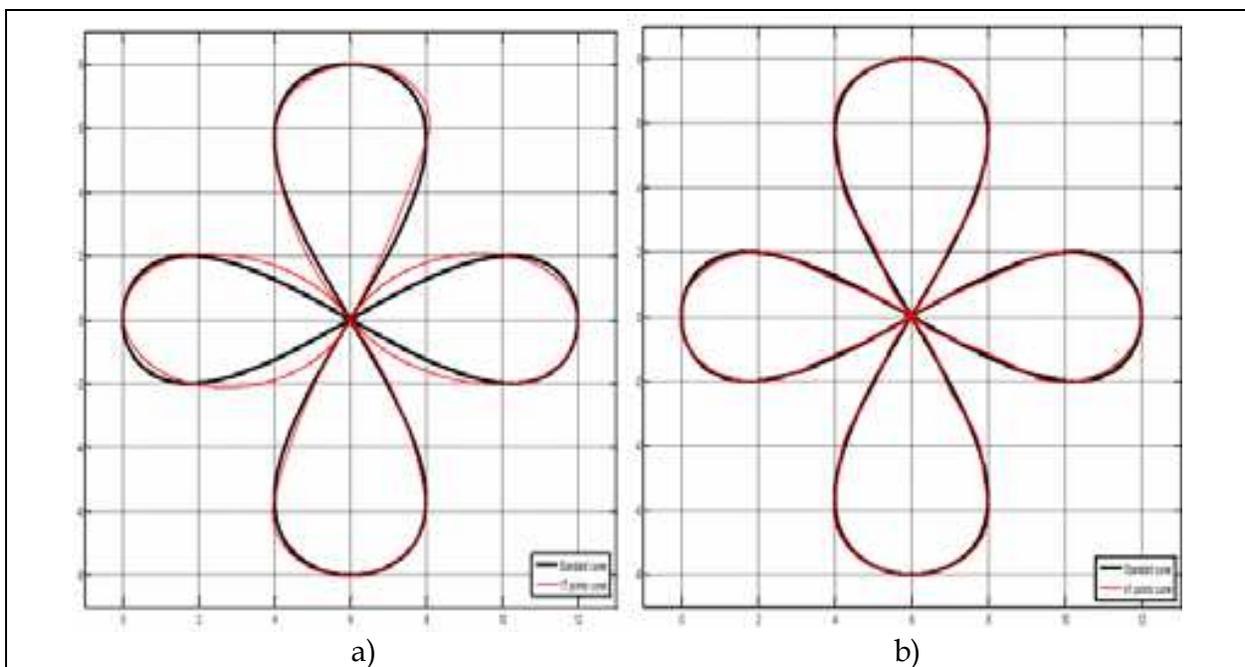$$\Delta x \sqrt{\frac{\sum_{i=1}^{n}(p_i - pd_i)^2}{n*(n-1)}} \tag{22}$$

Fig. 18. a)First Path trajectory with 17 points, b) with 124 points (Maniardi, 2010)

Initially, the K_R values for each simulation were defined by simulation of trajectories using the quadratic error for different values of K_R. The optimal weights p_r and p_d were performed simulations varying the weights from 0 to 1 so that the sum of the weights should be equals 1, indicating that the overall speed should not exceed the desired speeds. The results of simulations of paths considering failures can be observed as : with the straight path, the best result was obtained with a purely reactive pair (0, 1), while on the curve path, the par purely deliberative (1, 0) had a better result. The pair (p_d, p_r)=(0.8, 0.2) had an excellent result on the straight and a good result in the curve, being the second best result in all simulations. For the analysis and selection of the best reactive gain K_R, the average was calculated with different simulations which resulted in the error graphs. The graph obtained is shown in Figure 19. The color lines represent different averages of the error in each gain K_R, where the lines is the result of the sum of simulations. The value K_R =10 was selected because the average error in this value is lower, with a better response in the control system.
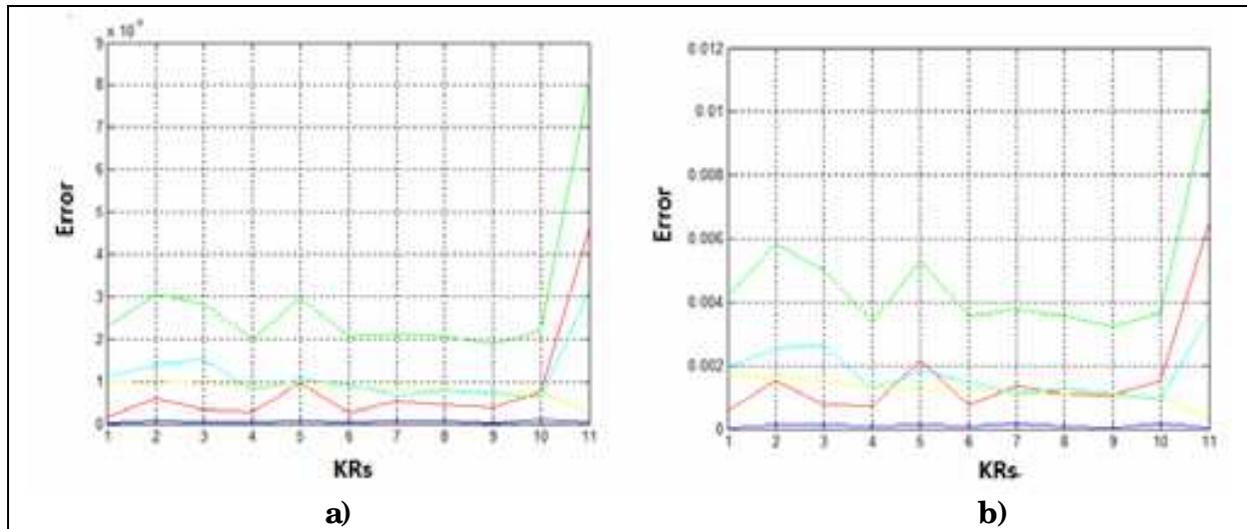
Fig. 19. Errors obtained in the simulations of the K_R's: a)Quadratic error, b) Maximum error

5. Conclusion

This chapter has presented the overall process of robot design, covering the conceptualization of the mobile robot, the modeling of its locomotion system, the navigation system and its sensors, and the control architectures. As well, this chapter provides an example of application. As discussed on this chapter, the development of an autonomous mobile robot is a transdisciplinary process, where people from different fields must interact to combine and insert their knowledge into the robot system, ultimately resulting on a robust, well modeled and controlled robot device.

6. Acknowledgment

7. References

Alami, R. ; Chatila, R.; Fleury, S.; Ghallab, M. & Ingrand, F. (1998). An Architecture for Autonomy. *International Journal of Robotics Research*, 1998.

Albus, J; McCain, H. & Lumia R. (1987). NASA/NBS Standart Reference Model for Telerobot Control System Architecture (NASREM). NBS Technical Note 1235, Robot Systems Division, National Bureau of Standart, 1987.

AirRobot UK. (2010). AirRobotUK, In: *AirRobot UK- links*, March 2011, Available from: < http://www.airrobot-uk.com/index.htm>

Anderson, T. & Donath, M. Animal Behavior as a Paradigm for Developing Robot Autonomy, Elsevier Science Publishers B.V. North-Holland, pp. 145-168, 1990.

Arkin, R. (1990). Integrating Behavioural, Perceptual and World Knowledge in Reactive Navigation. *Robotics and Autonomous Systems*, n6, pp.105-122, 1990.

Arkin, R.; Riseman, E. & Hanson, A. (1987). AuRA: An Architecture for Vision-based Robot Navigation, *Proceedings of the 1987 DARPA Image Understanding Workshop*, Los Angeles, CA, pp. 417-431.

Barshan, B. & Durrant-White, H. (1995) Inertial Systems for Mobile Robots. *IEEE Transactions on Robotics and Automation*, Vol. 11, No 3, pp. 328-351, June 1995.

Batlle, J.A & Barjau, A. (2009). Holonomy in mobile Robots. *Robotics and Autonomous Systems* Vol. 57, pp. 433-440, 2009.

Bisset, D. (1997). Real Autonomy. Technical Report UMCS-97-9-1, University of Kent, Manchester, UK, 1997.

Blaasvaer, H.; Pirjanian, P & Christensen, H. (1994) AMOR - An Autonomous Mobile Robot Navigation System. *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Texas, pp. 2266-2277, october 1994.

Borenstein, J.; Everett, H. & Feng, L. (1995). Where am I? Sensors and Techniques for Mobile Robot Positioning. AK Peters, Ltd., Wellesley, MA), 1st ed., Ch. 2, pp. 28-35 and pp. 71-72, 1995.

Brooks, R. & Iyengar, S. (1997). Multi-sensor Fusion - Fundamentals and Applications with Software. Printice-Hall, New Jersey 1997.

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, Vol. RA-2, No 1, pp. 14-23, March 1986.

Brooks, R.A. (1991). Intelligence Without Representation. Intelligence Artificial Elsevier Publishers B.V, No 47, pages 139-159, 1991.

Deveza, R.; Russel, A. & Mackay-Sim, A. (1994). Odor Sensing for Robot Guidance. *The International Journal of Robotics Research*, Vol. 13, No 3, pp. 232-239, June 1994.

Dudek, G. & Jenkin, M. (2000). Mobile robot Hardware, In: Computational Principles of mobile robotics, (Ed.), 15-48, Cambridge University Press, ISBN 978-0-521-56021-4, New York,USA.

Elfes, A. (1987). Sonar-Based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*, Vol. RA-3, No 3, pages 249-265, June 1987.

Everett, H. (1995). Sensor for Mobile Robot - Theory and Application. A.K. Peters, Ltd, Wellesley, MA, 1995.

Feng, D. & Krogh, B. (1990). Satisficing Feedback Strategies for Local Navigation of Autonomous Mobile Robots. *IEEE Transactions on Systems, Man. and Cybernetics*, Vol. 20, No 6, pp. 476-488, November/December 1990.

Ferasoli Filho, H. (1999). Um Robô Móvel com Alto Grau de Autonomia Para Inspeção de Tubulação. Thesis, Escola Politécnica da Universidade de São Paulo, Brasil.

Ferguson, I. (1994). Models and Behaviours: a Way Forward for Robotics. *AISB Workshop Series*, Leeds, UK, April, 1994.

Fierro, R; Lewis, F.(1997). Control of a Nonholonomic Mobile Robot: Bac Backstepping Kinematics. *Journal of Robotic System*, Vol. 14, No. 3, pp. 149-163, 1997.

Fikes, R & Nilsson, N. (1971). STRIPS: A New Approach to the application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2, pp 189-208, 1971.

Firby, R. (1987). An investigation into reactive planning in complex domains. In *Sixth National Conference on Artificial Intelligence*, Seattle, WA, July 1987. AAAI.

Franz, M. & Mallot, H. (2000). Biomimetic robot navigation. *Robotics and autonomous Systems*, v. 30, n. 1-2, p. 133–154, 2000.

Gat, E. (1992) Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots. *Proceedings of the AAAI92.*

Graefe, V. & Wershofen, K. (1991). Robot Navigation and Environmental Modelling. *Environmental Modelling and Sensor Fusion*, Oxford, September 1991.

Harmon, S.Y. (1987). The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain. *IEEE Journal of Robotics and Automation*, Vol. RA-3, No.3, pp. 266-279, June 1987.

Kaelbling, L. (1992). An Adaptable Mobile Robot. *Proceedings of the 1st European Conference on Artificial Life.* 1992

Kaelbling, L. (1986). An Architecture for Intelligente Reactive Systems. Technical Note 400, Artficial Intelligence Center SRI International, Stanford University, October 1986.

Kelly, A. (1995) Concept Design of a Scanning Laser Rangerfinder for Autonomous Vehicles. Technical Report CMU-RI-TR-94-21, The Robotics Institute Carnegie Mellon University, Pittsburgh, January 1995.

Krotkov, E. (1994). Terrain Mapping for a Walking Planetary Rover. *IEEE Transaction on Robotics and Automation*, Vol. 10, No 6, pp. 728-739, December 1994.

Latombe, J. (1991). Robot Motion Planning, Kluwer, Boston 1991.

Leonard, J. & Durrant-White, H. (1991) Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions on Robotics and Automation*, Vol. 7, No 3, pp. 376-382, July 1991.

Lumia, R.; Fiala, J. & Wavering, A. (1990). The NASREM Robot Control System and Testbed. *International Journal of Robotics and Automation*, no.5, pp. 20-26, 1990.

Luo, R. & Kay, M. (1989). Multisensor Integration and Fusion in Intelligent Systems. *IEEE Transactions on Systems, Man. And Cybernetics*, Vol. 19, No 5, pages 900-931, September/October 1989.

Mainardi, A; Uribe, A. & Rosario, J. (2010). Trajectory Planning using a topological map for differential mobile robots, *Proceedings of Workshop in Robots Application*, ISSN 1981-8602, Bauru-Brazil, 2010.

Mckerrow, P.J. (1991). Introduction to Robotics. Addison-Wesley, New York, 1991.

Medeiros, A. A. (1998). A survey of control architectures for autonomous mobile robots. *Journal of the Brazilian Computer Society*, v. 4, 1998.

Murphy, R. (1998). Dempster-Shafer Theory for Sensor Fusion In Autonomous Mobile Robot. *IEEE Transaction on Robotics and Automation*, Vol. 14, No 2, pp. 197-206, April 1998.

Noreils, F. & Chatila, R.G. (1995). Plan Execution Monitoring and Control Arquiteture for Mobile Robots. *IEEE Transactions on Robotics and Automation*, Vol. 11, No 2, pp. 255-266, April 1995.

Ojeda, L. & Borenstein, J. (2000). Experimental results with the KVH C-100 fluxgate compass in mobile robots. Ann Arbor, v. 1001, p. 48109–2110.

Protector. (2010). Protector USV, In: *The Protector USV: Delivering anti-terror and force protection capabilities*, April 2010, Available from: < http://www.ws-wr.com/epk/BAE_Protector/>

Raibert, M., Blankespoor, K., Playter, R. (2011). BigDog, the Rough-Terrain Quadruped Robot, In: *Boston Dynamics, March 2011*, Available from:
    <http://www.bostondynamics.com/>

Rembold, U. & Levi, P. (1987) Sensors and Control for Autonomous Robots. *Encyclopedia of Intelligence Artificial*. pp. 79-95, John Wiley and Sons, 1987.

Rich, E. & Knight, K. (1994). Inteligência Artificial, Makron Books do Brasil, 1994, São Paulo.

Russell, R. (1995). Laying and Sensing Odor Markings as a Strategy for Assistent Mobile Robot Navigation Tasks. *IEEE Robotics & Automation Magazine*, pp. 3-9, September 1995.

Shafer, S.; Stentz, A. & Thorpe, C. (1986). An Architecture for Sensor Fusion in a Mobile Robot. *Proc. IEEE International Conference on Robotics and Automation, San Francisco, CA*, pp. 2002-2011, April 1986.

Siciliano, B.; Khatib, O. (ORGS.). Springer Handbook of Robotics. Heidelberg: Springer, 2008.

Siegwart, R. & Nourbakhsh, I. (2004). Mobile Robot Kinematics, In: Introduction to Autonomous Mobile Robots, MIT Press, 47-82., Massachussetts Institute of Technology, ISBN 0-262-19502, London, England.

Simmons, R. (1994). Structured Control for Autonomous Robots. *IEEE Transactions on Robotics and Automation*, Vol. 10, No 1, pp. 34-43, February 1994.

Thorpe, C; Hebert, M.; Kanade, T. & Shafer, S. (1988). Vision and Navigation for the Carnegie-Mellon Navlab. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 10, No. 3, pp. 401-412, May 1988.

Tuijnman, F.; Beemster, M.; Duinker, W.; Hertzberger, L.; Kuijpers, E. & Muller, H. (1987). A Model for Control Software and Sensor Algorithms for an Autonomous Mobile Robot. *Encyclopedia of Intelligence Artificial*, pp. 610-615, John Wiley and Sons, 1987.

Waldron & Schmiedeler. (ORGS.). Springer Handbook of Robotics. Heidelberg: Springer, 2008.

Zelek, J. (1996). SPOTT: A Real-Time, Distributed and Scalable Architecture for Autonomous Mobile Robot Control. Thesis, Centre for Intelligent Machines Departament of Eletrical Engineering McGill University, 1996.

Advances in Robot Navigation

Edited by Prof. Alejandra Barrera

Robot navigation includes different interrelated activities such as perception - obtaining and interpreting sensory information; exploration - the strategy that guides the robot to select the next direction to go; mapping - the construction of a spatial representation by using the sensory information perceived; localization - the strategy to estimate the robot position within the spatial map; path planning - the strategy to find a path towards a goal location being optimal or not; and path execution, where motor actions are determined and adapted to environmental changes. This book integrates results from the research work of authors all over the world, addressing the abovementioned activities and analyzing the critical implications of dealing with dynamic environments. Different solutions providing adaptive navigation are taken from nature inspiration, and diverse applications are described in the context of an important field of study: social robotics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Silas F. R. Alves, Joao M. Rosario, Humberto Ferasoli Filho, Liz K. A. Rincon and Rosana A. T. Yamasaki (2011). Conceptual Bases of Robot Navigation Modeling, Control and Applications, Advances in Robot Navigation, Prof. Alejandra Barrera (Ed.), ISBN: 978-953-307-346-0, InTech, Available from: http://www.intechopen.com/books/advances-in-robot-navigation/conceptual-bases-of-robot-navigation-modeling-control-and-applications

INTECH
open science | open minds