Assistive Navigation Control Architecture

URBANO NUNES, GABRIEL PIRES and PAULO COELHO
ISR - Institute for Systems and Robotics
University of Coimbra – Polo II
3030 Coimbra
PORTUGAL

Abstract: - An advanced class of robots and intelligent machines that adapt, extend, and work in a symbiotic way with humans, is required and is under research in University and Industry laboratories. These robots do not only work for humans, but also they work with or assist humans, and share with the users the same environment. This paper presents an hybrid control architecture with reactive, deliberative, and cognitive components appropriate to human-oriented mobile robots. The architecture is under development and has been tested in a motorized wheechair equipped with several sensors and a multimodal HMI (Human-Machine Interface). A joystick and a voice unit interface (VUI) are used to steer the wheelchair, giving accessibility to handicaped persons.

Key-Words: - Assistive navigation, control architectures, human-oriented robotics, reactive behaviours, kinematics

1 Introduction

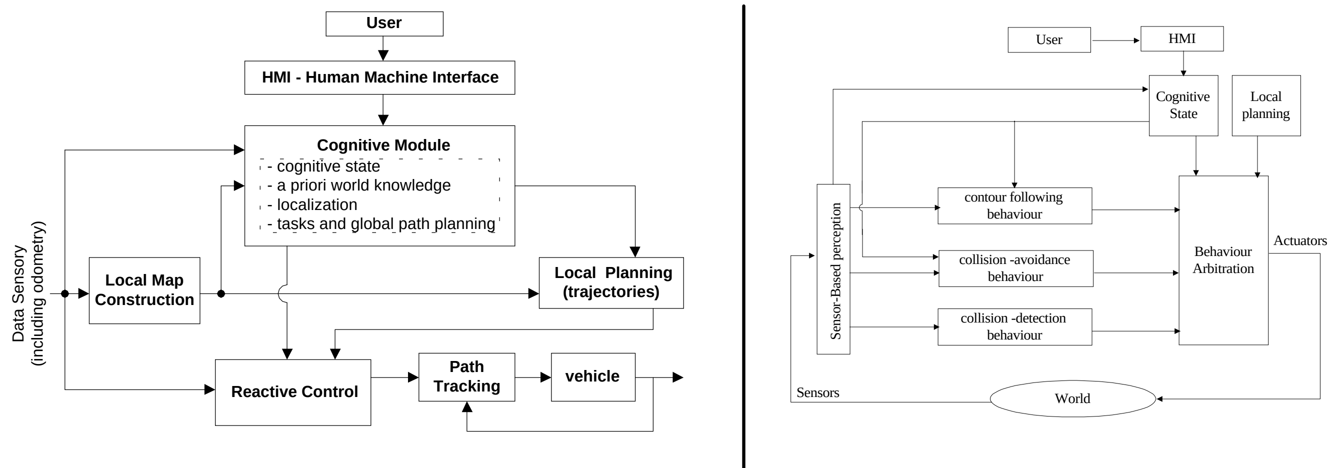
Usually the human is forced to adapt him or herself to the machines in present day manufacturing and process industries. Human-oriented robots and intelligent machines are required and are under research. Its development poses many difficult problems, namely concerning HMI, safety and man-machine shared control.

In the Intelligent Control and Robotics laboratory (IC&R) at the Institute for Systems and Robotics (ISR), research is being done towards the development of an intelligent control system for human-oriented mobile robots (HOMR). We are pursuing an hybrid paradigm composed by a reactive control level and a decision making level supported by a knowledge-based perceptual system, as depicted in Fig.1a. The paper presents globally our control architecture and describes some of its modules in detail. The purpose is to achieve an architecture appropriate to integrate different components necessary to a HOMR. As example we can enumerate some research problems faced in this development:

1)- For a behaviour-based reactive level, what type of behaviours will be necessary? What methods to use for integration/fusion and arbitrage of different behaviours? What learning methods could be considered for performing behaviour integration? How to establish the human interaction in this architecture?
2)- One of the challenges of human-oriented robotics is that robots have to work closely with humans. Can we be inspired by biological behaviours to make robots more friendly and more familiar to humans?
3)- Besides the interaction capabilities with humans, the robot should integrate and acquire knowledge concerning its own state and environment state. This means it should have other capabilities, namely for self-localisation, map building and planning movements and tasks.

2 Control architecture

Fig.1a presents the major components of the control architecture under development. Parts of it have been tested in a motorized semi-autonomous wheelchair prototype being developed under the RobChair project running in ISR [1].



a) b)

Fig.1: **a)** Hybrid control architecture; **b)** RobChair reactive layer

This is achieved sharing the control between the user and the actions provided by the cognitive and sensory systems. A joystick and a VUI (voice unit interface) [2] are used to steer the wheelchair. The VUI is based on a set of voice recognition libraries included in a package that performs the recognition of spoken words (Dragon Voice Tools).

2.1 Conceptual Architecture

RobChair architecture is organized as an hybrid architecture combining deliberative reasoning with low-level reactive behaviours. Deliberative reasoning provides the system with the ability to reach proposed goals. Reactive behaviours are indispensable to ensure a safe navigation, enabling the vehicle to react in real time to environment emergent situations. In most robotics applications, a purposeful navigation depends on the integration and interaction of these two control levels. However, there are others where a unique deliberative or reactive architecture ensures a purposeful navigation. The first, deliberative reasoning, can be used in fully deterministic and static environments. However, this doesn't meet the requirements of most real environments. The second, reactive reasoning, usually lacks purposeful goals. The lack of *a priori* information to plan strategies and trajectories can lead to navigation failure. This can be compensated if the goals are always visible. For example, if the goal is a light always detectable, or a cue in the floor, or a surface to contour, it will be possible to reach purposeful goals.

RobChair is a specific system integrating closely the human and the machine. The human is a cognitive entity that substitutes parts of the deliberative layer. Presently, without having global environment information, RobChair system is unable of a purposeful navigation without user intervention, so the reason we call it a *semi-autonomous* system.

The proposed architecture is a four layer distributed architecture: a *reactive layer* embodying reactive behaviours; a *localised action layer* for execution of specific tasks dependent of local environment; a *deliberative reasoning layer* responsible for high-level planning; and finally a *mission layer* where goals are defined. The wheelchair user is part of this layer and he intervenes in the cognitive state. By this way, he can define goals for the deliberative reasoning layer, as well as, depending of system state, goals for reactive control layer guidance. The main modules of the conceptual control architecture and their interconnections are illustrated in Fig.1a:

Mission Layer - In this layer a set of static or dynamic goals are defined by the user or by other human operator. Examples of deliberative goals might be *go to room B*.

Deliberative Reasoning Layer - It is usually based on *a priori* knowledge of the world. This knowledge takes the form of topological and geometric maps giving, for instance, information of how to go from A to B, or giving the position of objects and landmarks. Based on this information, global trajectory and task planning and execution can be undertaken. This layer relies on long-term memory information, and basically, performs global path-planning providing bottom layers with a set of points defining the path to accomplish the tasks. To perform path-planning it may be required other cognitive capabilities relying on global map updating, integrating over time local maps into the global map, self localisation, etc.

Localised Action Layer - This is an intermediate, short-term memory layer. The action plan relies essentially on short-term memory, which integrates sensory information, in terms of a local map, and guidance information from the upper control layers. Two localized action tasks are being implemented: a door-passage and a table/writing desk approaching.

Reactive Layer - This layer is fully implemented. It embodies three behaviours: collision detection, obstacle avoidance, and contour following. These behaviours rely upon actual sensory information without resorting to environment models. The behaviours are simple, directly coupling perception to action. This layer receives guidance from upper layers. It consists basically on system state information and commands of velocity and direction. An integration/fusion of the guidance variables and data from each behaviour is carried out in this layer.

3 Local Mapping

A local environment model is acquired iteratively and on-line. The local map, an occupancy grid of $n \times n$ cells, is a local view that moves with the robot. The mapping process, based on Thrun's proposal [3] consists of two modules: a feedforward neural network to interpret sensor readings and a Bayesian update rule to integrate over time the range sensing data [4].

4 Reactive Control

Purposeful navigation based upon reactive behaviours can be achieved if a continuous and well-defined goal exist. RobChair goals come from user steering commands. These commands are given through a joystick or voice and represent respectively, continuous and discrete-fuzzy goals that the vehicle has to follow. Real environments, complex and dynamic, force the user to some dexterity to control the wheelchair (this is especially difficult to people with motor impairments). The reactive layer guides the wheelchair, observing the upper levels commands, but changing them in accordance to the actual stimulus sensed by the reactive behaviours.

4.1 Behaviour-Based Control

Behaviours implement specific tasks encapsulating the perception, plan and act necessary to the execution of the task, independently of other behaviours. The vehicle overall behaviour results from the co-ordination between individual behaviours. What is the best way to coordinate behaviours in order to achieve a coherent behaviour? The answer has not yet been reached, but several authors have given their contributions. All of them state that the co-ordination module is the key of behaviour-based architectures. Arkin [5] classifies the co-ordination methods in two main categories: competitive methods and cooperative methods - command fusion methods. In the competitive methods, the co-ordinator acts as "winner-take-all" mechanism. There is just one behaviour that is active (overriding all the others) and controls the robot. The well known competitive method is the *Subsumption Architecture arbitration* developed by Brooks [6]. This architecture obeys to a priority layered structure. High level behaviours with higher priorities subsume low level behaviours, via inhibitor and suppressor mechanisms. Other way of arbitration is based on context rules [7]: *IF context THEN behaviour n*. The context relies on actual sensory information. Competitive methods only allow one behaviour to control the vehicle. This is clearly a disadvantage when the goal is to satisfy simultaneously two or more behaviours (representing distinct objectives). The information included in inactive behaviours is lost, not contributing for the global answer. To overcome this limitation, cooperative methods emerged. The global behaviour results from the fusion of several behaviours. Each one gives a contribution. The well known methods of command fusion are based on vector summation [8-9]. Following this approach, each behaviour is represented by a vector with an

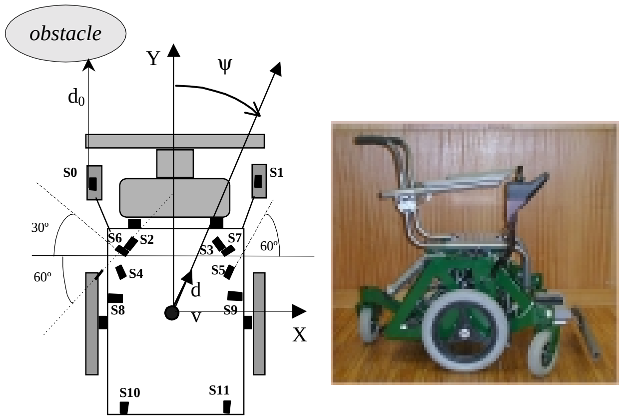


Fig. 2: Left)- IR sensor arrangement (top view). S_i IR sensor *i*. d_0 denotes the distance measured by IR sensor S_0. Ψ denotes the angle between wheelchair heading direction and input direction; Right)- Wheelchair prototype used in RobChair project.

associated gain. Vectors are multiplied by their gains and then summed, resulting an output directly related with each associated gain. The vector can be an attractor to goal or a repulsor from obstacles. The gain can, for instance, be inversely proportional to the square of the distance between the goal and the vehicle. Another method of command fusion is based on fuzzy logic. Each behaviour is modelled by an intention function, represented by fuzzy sets. Behaviours actions are combined via fuzzification and the resulting action is selected via defuzzification.

4.2 RobChair Reactive Behaviours

The reactive layer of RobChair is composed by three behaviours, in the configuration depicted in Fig.1b. Collision detection behaviour ensures the safety of the user preventing or detecting potential collisions. The other two behaviours have inputs from the cognitive module (goals which may represent the input command from the user through a joystick or voice), and actual sensory information from environment.

Obstacle avoidance behaviour - is composed by fuzzification, rule base, decision-making, and defuzzification modules:

A. Fuzzification of the input-output variables

A fuzzy operator converts crisp input data into linguistic values. The input linguistic values d_i (i=0,..,11), Ψ and dv (Fig.2) are expressed by linguistic values (VN, NR, FR, VF), (NB, NM, NS, ZZ, PS, PM, PB), (NS, ZZ, PS, PM) respectively. The output linguistic variables v and $\Delta\theta$ are expressed by the linguistic values (NS, ZZ, PS, PM) and (NB, NM, NS, ZZ, PS, PM, PB) using triangular shaped membership functions. The linguistic terms have the meanings shown in Table 1.

Table 1: Linguistic values.

ZZ: zero	NB: negative big
VN: very near	NM: negative medium
NR:near	NS: negative small
FR: far	PS: positive small
VF: very far	PM: positive medium
	PB: positive big

B. Rule base construction and defuzzification

The rule base for implementing behaviours is constructed based on human experience. The rule base of the obstacle avoidance behaviour is composed by rules taking the form of IF-THEN statements, such as:

IF (dv=PM AND Ψ=ZZ AND IR(0)=VF AND IR(1)=VF) THEN (v=PM, $\Delta\theta$=ZZ)

where dv and Ψ are the input variables and, v and $\Delta\theta$ are the output variables. The defuzzification process uses the centre of gravity method.

Contour-following behaviour - There are situations for which a contour-following behaviour can enhance the movements of the wheelchair. Using this behaviour, smoother trajectories may emerge than under obstacle avoidance control. For example, the navigation in a curved hallway or, in a long corridor, are clearly situations in which it is advantageous the use of the contour following. This behaviour is implemented using fuzzy logic similarly as described for the obstacle avoidance behaviour. Despite being a reactive behaviour it requires cognitive information to decide how to contour the wall (e.g. on the left or at the centre in case of a corridor).

Arbitration - This module integrates the command outputs from behaviours and from upper level control modules. The cognitive

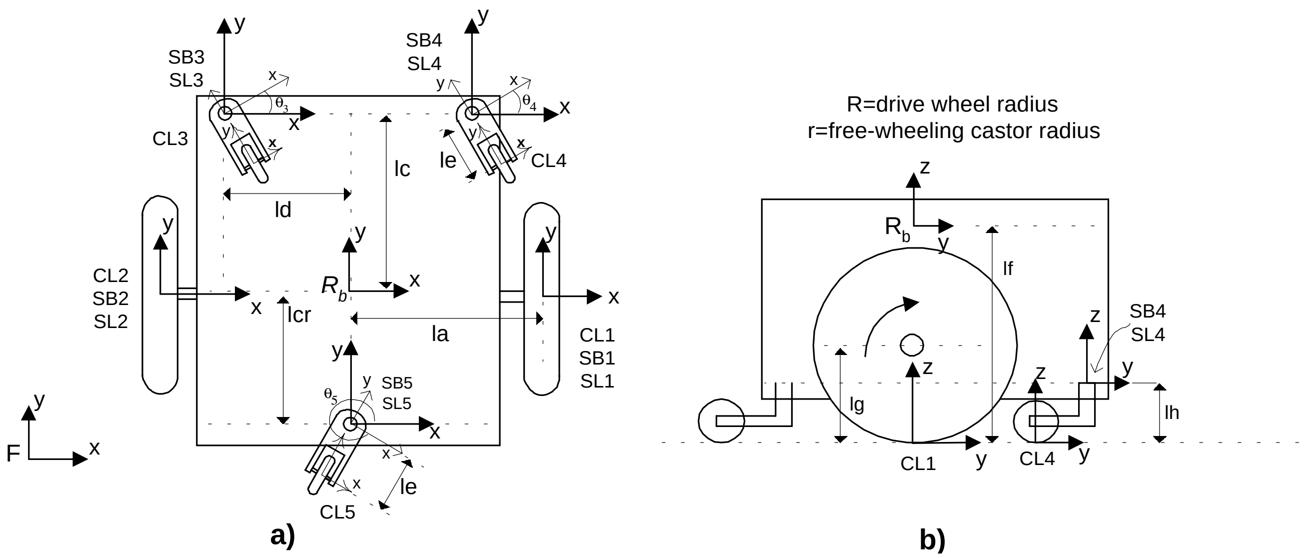


Figure 3 - Geometric model of the wheelchair: **a)** top view; **b)** side view

module, a key element of the system, assigns the system state using information from the user and from sensory data, integrated over time in local and global maps. The mechanism of arbitration is context dependent following rules of the form:

IF *context n* THEN *behaviour n*

For example, if a collision or potential collision is detected the other behaviours are inhibited. If the vehicle suffers a front collision, the vehicle stops and only backward commands are accepted. Collision behaviour has the higher priority and overrides outputs from concurrent behaviours. In a collision free state (no report of possible collisions) the other behaviours can control the wheelchair, if the system is in a state demanding reactivity.

5 Path-tracking control

5.1 Kinematic Model

Fig.3 presents a geometrical model of the wheelchair defining necessary variables to obtain the kinematic model. Muir's methodology [10] was followed. Two coordinates systems are defined: the world coordinates system {F} and the robot coordinates system {Rb}. Fig.3 shows a WMR (Wheeled Mobile Robot) having two diametrically opposed drive wheels (radius R) and a free-wheeling castor (radius r). Both drive wheels are actuated and sensed, while the castor is neither actuated nor sensed. The kinematic model with respect to the robot body frame {Rb} [11] is given by

$$\dot{q}_R = J_x \cdot \dot{w} \qquad (1)$$

with

$$\dot{q}_R = \begin{bmatrix} v_{rx} \\ v_{ry} \\ w_r \end{bmatrix}_{Rb} ; \; J_x = \frac{R}{2l_a}\begin{bmatrix} 0 & 0 \\ l_a & l_a \\ 1 & -1 \end{bmatrix}; \; \dot{w} = \begin{bmatrix} w_{w1x} \\ w_{w2x} \end{bmatrix}$$

where J_x is the Jacobian of actuation (the wheels are actuated around X-axis), v_{rx} and v_{ry} are the linear velocities of the robot along X-axis and Y-axis, w_r is the angular velocity of the robot, w_{w1x} and w_{w2x} are the angular velocities of the wheels 1 and 2 around the X-axis. In equation (1) the velocities are given in respect to the robot frame $\{R_b\}$. In order to get the cartesian velocities in the world frame {F} it is necessary to make the coordinates transformation, as illustrated in Fig. 4:

$$\dot{q}_F = R(\theta) \cdot \dot{q}_R \qquad (2)$$

where

$$\dot{q}_F = \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}_F \; ; \quad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In Fig.4, (X_r,Y_r) defines the origin of the robot coordinate system and θ is the angle between the X-axis of {F} and the X-axis of {Rb} (counter-clockwise by convention).

5.2 Dead-Reckoning

The angular velocities of the wheels (1 and 2) are estimated from encoders readings. At each sampling period, *h*, the signed integration of the encoder pulses provides an estimate of angular displacements, (respectively $\Delta\alpha_1$ and $\Delta\alpha_2$). From equations (1) and (2) we can obtain the incremental linear displacement *(*ΔD*)*, along Y-axis, and incremental rotation *(*$\Delta\theta$*)* of the mid-point of the wheel base as follows: $\Delta D = R(\Delta\alpha_1 + \Delta\alpha_2)/2; \quad \Delta\theta = R(\Delta\alpha_1 - \Delta\alpha_2)/2l_a$, where *R* is the right and left wheel radius, and $2l_a$ is the wheel base, as shown in Fig.3. ΔD and $\Delta\theta$ can be used to estimate the robot position incrementally (making the arc curvature trajectory assumption):

$$\begin{bmatrix} x_r(N) \\ y_r(N) \\ \theta(N) \end{bmatrix} \approx \begin{bmatrix} x_r(N-1) \\ y_r(N-1) \\ \theta(N-1) \end{bmatrix} + \begin{bmatrix} -\Delta D \, sin\,(\theta(N-1) + \frac{\Delta\theta}{2}) \\ +\Delta D \cos(\theta(N-1) + \frac{\Delta\theta}{2}) \\ +\Delta\theta \end{bmatrix} \quad (3)$$

The equation approximates trajectory as a sequence of constant curvature segments of length ΔD. Such an approximation requires that the sampling period *h* be sufficiently small with respect to the vehicle linear and angular accelerations.

5.3 Path-tracking

The design of the low-level controller of the wheels is based on pole placement using discrete state space theory. An augmented state space is formulated considering the process and disturbance states. A stochastic active observer (new concept developed by Cortesão [12]) is embedded in the controller to handle the disturbances in a robust and optimal way.

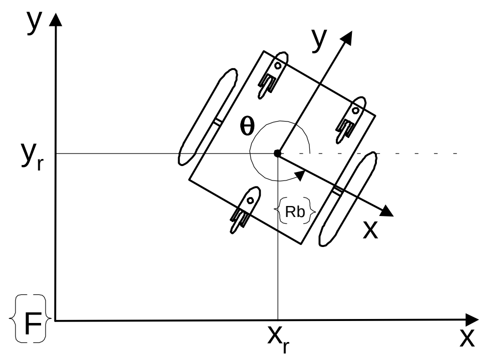


Fig. 4: Coordinates transformation (θ is the angle between the X-axis of {F} and the X-axis of {Rb}).

6 Acknowledgments

This work was partially supported by the project MENTOR PRAXIS/P/EEI/13141/1998.

References:

[1] G. Pires, R. Araújo, U. Nunes, and A.T. Almeida, ROBCHAIR – A Powered Wheelchair Using a Behaviour-Based Navigation, *IEEE 5th Int Workshop on Advanced Motion Control (AMC'98)*, Coimbra 1998, pp. 536-541.

[2] C. Lopes, N. Honório, F. Perdigão, and U.Nunes, Voice Interface for an Intelligent Wheelchair, *Third Portuguese Conference on Automatic Control (CONTROLO98)*, Coimbra, 1998, pp. 841-843.

[3] S. Thrun, Learning Metric-Topological Maps for Indoor Mobile Robots Navigation, *Artificial Intelligence*, 99 (1): pp. 21-71, 1998.

[4] J. Cruz and U. Nunes, Generating Local Maps for Mobile Robots, *Submitted to the 4th Portuguese Conference on Automatic Control (CONTROLO2000).*

[5] R. Arkin, *Behaviour-Based Robotics*, MIT Press, 1998.

[6] A. Brooks, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol. RA-2, nº1, 1986 pp.14-23.

[7] A. Saffiotti, Fuzzy Logic in Autonomous Robotics: Behaviour Coordination, *6th IEEE Int. Conf. on Fuzzy Systems*, 1997, pp. 573-578

[8] R. Arkin, Motor Schema Navigation for a Mobile Robot: An Approach to Programming by Behavior, *IEEE Conf. on Robotics and Automation*, 1987, pp. 264-270.

[9] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *The International Journal of Robotics Research*, vol.5, 1986, pp. 90-98.

[10] P. Muir, *Kinematic and Dynamic Modeling and Control of Wheeled Mobile Robots*, Ph.D Dissertation, Carnegie Mellon University, Pittsburgh, 1988.

[11] P. Coelho, Cinemática de Robôs Móveis, Technical Report, ISR – University of Coimbra, 2000 (in Portuguese).

[12] R. Cortesão, R. Koeppe, U. Nunes, and G.Hirzinger, "*Explicit Force Control for Manipulators with Active Observers*", Submitted to 2000 IEEE Int. Conference on Intelligent Robots and Systems (IROS'2000).