PRESENTATION ROBOT ADVEE

Jiří Krejsa*, Stanislav Věchet*, Jan Hrbáček†, Tomáš Ripel†,
Vít Ondroušek†, Radek Hrbáček‡, Petr Schreiber§

The paper gives detailed overview of the prototype of autonomous mobile robot Advee, developed for presentation purposes. Mechanical, electrical, sensor and software subsystems are described in the paper together with key algorithms such as localization and motion planning. Advee already performed over 600 hours of operation among common people, the experiences gained from the operation are also summarized in the paper.

Keywords: *mobile robot, human-robot interface, localization, motion planning*

1. Introduction

Mobile robotics is currently rapidly expanding from research labs to common routine operation in natural environment. Apart from most visible simple machines such as robotic vacuum cleaners, the more advanced systems are being utilized as in museum tour guides [1], city hall robots [2] or hospital service robots [3]. This paper gives detailed description of such advanced system – presentation robot Advee, developed by Bender Robotics s.r.o. during 2009–2011 period. The idea was to build a large scale autonomous mobile robot that could serve as an information source in common public places, being capable of interaction with common people. The purpose of the robot defines the requirements it is obliged to meet: autonomous motion, robust selflocalization, long duration, highly reliable safety measures, human-robot interface friendly to computer illiterate users. To meet all the requirements while keeping the price of the system in economically reasonable level required the development of robust control algorithms that can cope with cheaper sensors, mainly with the respect to robot localization problem. On the other hand the reliability requirements do not allow any compromise in robot actuation equipment.

Presentation robot works basically in two stages – the first one being the autonomous motion, second one being the interaction with the user. During such interaction the human-robot interface (HRI) plays essential role to keep the users attention. Therefore the HRI must be combining all robot means to get the redundancy in communication channels, so the less computer literate people can still get the message while more literate user is not repelled.

The paper is organized as follows. Chapter 2 presents the concept of the robot and its hardware components. Chapter 3 describes its software architecture, from the lowest layer

* Ing. J. Krejsa, Ph.D., Ing. S. Věchet, Ph.D., Institute of Thermomechanics ASCR v.v.i., Brno branch, Technická 2, 616 69, Brno, Czech Republic

† Ing. J. Hrbáček, Ing. T. Ripel, Ing. V. Ondroušek, Ph.D., Brno University of Technology, Faculty of Mechanical Engineering

‡ R. Hrbáček, Brno University of Technology, Faculty of Electrical Engineering and Communication

§ Ing. P. Schreiber, Bender Robotics s.r.o.

through localization and motion planning to human-robot interface. Chapter 4 summarizes the experiences gained during the operation in public.

2. Hardware

2.1. Chassis

The mechanical design of the robot starts with the chassis type decision. Basically there are only two options ů differential chassis or Ackerman steering type chassis, as all other types [4] while possibly offering better maneuverability/energy efficiency rates are too complex to maintain required high reliability at the reasonable cost. The differential chassis advantage of ability to turn on site makes the motion planning easier and operational area of the robot is restricted only by robot's physical dimensions. However, the energy efficiency is lower and as the sufficiently long duration of the robot operation is one of the major requirements the chassis type selection for Advee ended up with Ackerman steering.

The ability to overcome certain low profile obstacles is necessary when operating in real world environment. In order to keep track of driven wheels there are three possibilities how to design the chassis:

- wheel suspension: + good driving properties, + considerate towards sensitive equipment, − lower stability, − more complex (more expensive, less reliable),
- independently driven wheels: + good driving properties, + redundant in case of drive failure, + best stability, − more complex (more expensive, less reliable),
- swinging axle: + simple, + cheap, + reliable, + medium stability, − inconsiderate towards sensitive equipment.

The authors have experiences with the combination of the independently driven wheels and swinging axle in mobile robot Bender II intended for outdoor environment [5]. While such combination gives the best driving abilities in rough terrain, for indoor use the price and reliability variables are more important, therefore the swinging axle was selected for Advee. The dimensions of the chassis are set based on the spatial and mass position of the equipment the robot has to carry (see chapters 2.3 and 2.4) , keeping the best possible balance between the stability and maneuverability. Resulting chassis drawing is shown on Fig. 1.

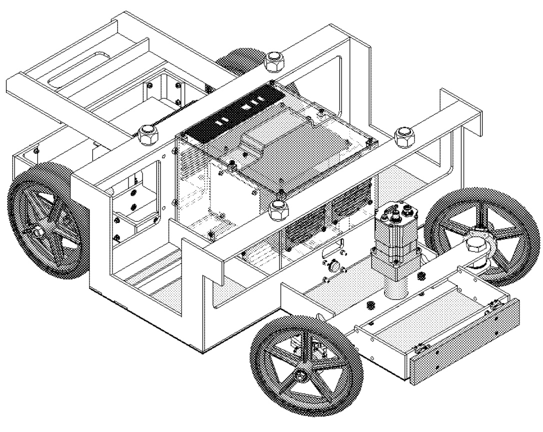

Fig.1: The chassis (accumulators not shown for clarity)

2.2. Motion actuators

Two major actuators handle the steering and the motion of the robot itself. To estimate the requirements for motion drive, the simplified motion model was created taking into account estimated total weight of the robot and required driving properties, such as maximum speed, maximum inclination, etc. Based on the calculation comprising all major physical influences the actuator Maxon RE 50 with 24 V/200 W DC motor, planetary gear 26:1 and incremental sensor has been chosen as the main drive. Common differential is used to distribute the torque to back wheels.

The determination of sufficient power of the steering mechanism is difficult task, as the calculation would depend on very precise acquaintance with mechanical model simulation and used friction model, that can vary for different types of surfaces the robot is intended to move on. Therefore the required power was determined on the base of the measurement of forces actions in the testing platform Bender II steering mechanism, adjusted to the wheel load of Advee, resulting in Berger Lahr actuator IclA N065/2 with planetary gear 40:1.

Both Maxon and Berger Lahr controllers use CAN bus for communication with the low level software layer.

The power supply of the robot consists of 8 LiFePo4 cells with 40 Ah capacity. The cells are placed on the sides of the robot $(3 + 3)$ and at the rear part of the main frame to distribute the load on front and back wheels. Motion actuators represent about 40 % of the total drain during the robot operation.

2.3. Motion sensors

Motion sensors provide robot with data necessary for successful obstacle detection, motion planning and localization. Location of key components is shown on Fig. 2. Obstacle detection is provided by the ring of 16 SRF08 ultrasonic sensors by Devantech, mounted directly onto the outer shell of the robot at about 150 mm above the ground. Beams of the sensors are partially overlapping covering the whole 360 degrees around the robot. SRF08 sensors are connected with the low level software layer through EIA-485 bus. Data from the sensors are obtained in a rate of approximately 4 Hz.

Odometry is measured on front wheels using HEDS-9000 IRC sensor with resolution of 1024 ppr at 4 Hz rate. IRC sensors also use EIA-485 bus for communication with the low level. Odometry data are used in position estimator, see chapter 3.2.

Another key device for position estimation is located in the top part of the robot. It is beacons scanner, a ring of 16 IrDA receivers, placed on a circle, covering whole 360 degrees range. The scanner detects the IrDA based beacons within the range of about 10 meters, that are not hidden behind an obstacle, providing the position estimator with the relative angle of each beacon. The beacons are placed in known locations, see chapter 3.2. The beacon scanner and the emitters communicate with each other using combined radio and infrared communication protocols. The low power consumption radio modules with free 433 MHz modulation are used for one way data transmission, in the direction from the scanner to the beacons. This way the proper beacon is selected for transmission and thus beacon identification problem is solved. For further power consumption reduction only the beacons within the operating range from estimated position are requested to respond. Beacons are powered by single LiPol cell 1100 mAh providing enough power for 55 hours of continuous emitting that represents about 14 days of operation in normal conditions. Beacons scanner also uses EIA-485 bus for communication with the low level and provides the measured angles at about 1 Hz rate.

The last sensors that are connected directly with the motion are the bumpers. Bumpers serve as the last instance safety feature in cases of other mean failure. Bumpers are located in both front and rear bottom part of the robot and when pressed, it directly cuts the power to the motion actuators to stop the robot immediately.

2.4. Human robot interface hardware component

Advee is equipped with several components for communication with the user. Its usage is further described in chapter 3.3, the overview of devices locations is given in Fig. 2. The key components are:

- 19″ LCD monitor with capacitance touch screen, the monitor is mounted directly on the upper frame of the robot;
- color CCD camera by Microsoft, capable of about 15FPS at 640×480 resolution, the camera is mounted in robot's eye directly on the head part of robot outer shell;
- thermal printer Zebra TTP 7030/112 with 112 mm width paper output, both the printer and paper drum are mounted on the upper frame of the robot;
- sound input/output, consisting of the microphone and speakers mounted in robots head part, amplifier and external sound card mounted on the upper frame of the robot.

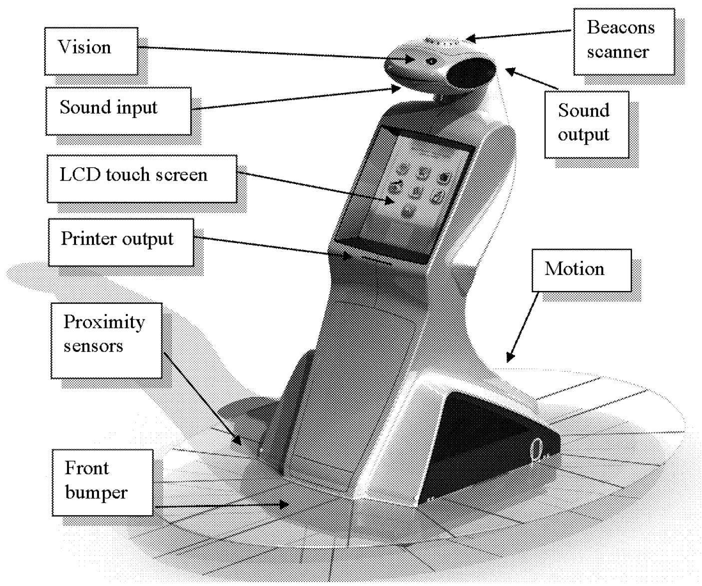

Fig.2: Motion sensors and key human robot interface devices

2.5. Outer shell

The outer shell of the robot has to meet following key requirements:

- attractive look as the robot is intended as presentation tool and must attract people,
- rigidity and reliability to protect the internal equipment,
- low weight to keep the overall weight of the robot in reasonable level.

To meet the requirements the outer look was designed by professional design studio. Based on the extensive study the shape vaguely reminding the human was chosen to make the robot look neat. Surprisingly, the fully humanoid shape was found repelling people, therefore e.g. the head part was designed to only partially resemble an actual person.

To keep the weight at the reasonable levels and with respect to complicated shape of the shell it is made of composite fiberglass sandwich. Covers are both at the front and back part of the shell to allow access to the key components without dismounting the shell itself (paper drum replacement, etc.). The shell is connected to the frame on the most curvatory parts of the shell, as those are the most rigid. The final look of the shell is illustrated on Fig. 2.

2.6. Computers

In order to keep the overall structure modular, the computational units were divided into two separate groups. The first one is responsible for robot motion, the second one handles the human-robot interface, further denoted as lower and upper computer. The key aspects in selecting the lower computer are:
- industrial-grade *ruggedness* needed to lay solid foundations of overall control system reliability and robustness in harsh conditions,
- *communication busses* support necessary to interface peripheral devices (CAN-bus, EIA-485 etc.),
- sufficient *computational power* to run simple low-level data manipulation modules as well as more complex middle-level data processing.

All these features are implemented by the Technologic Systems' TS-7800 single-board computer running the Linux operating system, which has been chosen to host low- and middle-level software equipment.

The upper computer key requirements are different:
- *low power consumption* to keep the operational range of the robot as high as possible,
- *sufficient computational power* for expensive methods such as image processing,
- *powerful graphics card* that can handle attractive yet expensive OpenGL based features of human-robot interface.

At least double core processor is required as HRI has to handle both the main GUI application and extensive image processing such as face detection simultaneously. Upper computer selected for the prototype is based on Zotac IONITX-F-E motherboard, equipped with Intel Atom 330 processor and integrated NVidia GeForce 9400 graphics. This combination meets all the requirements.

3. Software

3.1. Overall scheme and intermodule communication

Overall software equipment of the robot is divided into three layers:
- the *low level* provides interaction with hardware devices – basically it translates device-specific data into standard inter-modular messages and vice versa,
- the *middle level* implements robot state estimation and path planning,
- the *high level* cares handles user interaction.

Two lower layers are hosted on the Linux single-board computer although thanks to used inter-module communication mechanism can be run on separate computers and even under different operating systems without any change of communication-related code. The Linux-based operating system was selected due to its high modularity and low latency. The TS-7800 SBC is primarily designed to operate in conjunction with Linux and the source code of almost all custom hardware drivers is published by the manufacturer which leads to simple adaptability to concrete needs of the application and feasibility of fixing potential errors in implementation.

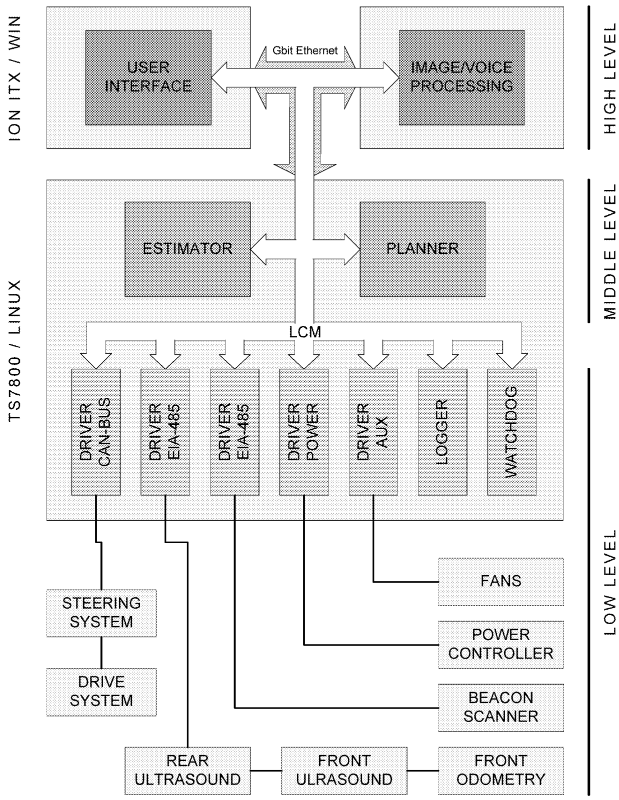

Fig.3: Scheme of hardware and software modules

Low and middle level software functionalities are divided into modules – an independent processes communicating with other modules. Inter-module communication mechanism (or generally inter-process communication, IPC) belongs to the most important parts of the system, as it defines unambiguous interface between modules, provides high-throughput and minimal-latency communication channel and should support multiple platforms – at least Linux and Windows, C and C#.NET. A number of systems that solve these requirements exist, a good comparison of several popular IPC libraries targeted for use in robotics is provided in [6].

As the software equipment of the robot is spread over minimally two computers, many traditional IPC methods are disqualified (shared memory, files, pipes, etc.). The computers are interconnected through Ethernet and solution based on the standard TCP/IP protocol suite seems to fit well. With respect to the paradigm, considering that most of the data flowing through the robot is of periodic nature, the publisher-subscriber model suits better. It can be implemented as 'push' or 'pull' – the 'push' mechanism delivers data instantly while the 'pull' method uses clients asking periodically for any new data. To ensure lowest possible latency, 'push' method has to be used. Unambiguity of the communication between different interface implementations is best achievable using dedicated message definition files and programming language specific generators of message-related code (marshalling, un-marshalling, publishing etc.).

Taking described requirements into account, the Lightweight Communications and Marshalling (LCM) [6, 7] has been chosen as an optimal mechanism for message passing in Advee. LCM is a set of libraries developed originally by the MIT DARPA Urban Challenge Team for their autonomous vehicle Talos. It covered all requirements except for the .NET framework support – this was solved by creating the lcm-dotnet port which has been added to the project. There is now a binding for C/C++, Java, Python, MATLAB and C# what makes it extremely flexible. LCM is composed of three main functional blocks:

- *message definition* using a C-like definition language that can be compiled into language-native message support files using thelcm-gen tool,
- *message marshalling/un-marshalling* – conversion between native and byte representation with defined endianness and runtime type safety checking,
- *communication itself* based on UDP multicast.

Described inter-module communication is used both for low-middle level and middle-high level, as shown on Fig. 3.

3.2. Robot navigation

Middle level layer handles two most critical modules for robot autonomous motion – position estimation and motion planning. The overall scheme used in navigation is shown in Fig. 4. Motion planner inputs are the proximity sensors readings, goal position and position of the robot provided by the estimator.

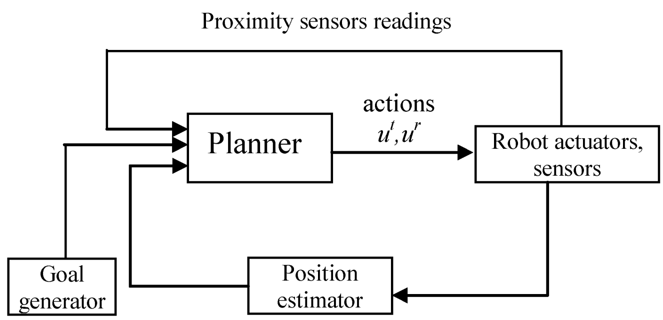

Fig.4: Advee navigation scheme

3.2.1. Position estimator

The goal of position estimator is to localize the robot in environment, i.e. to determine robot state defined by its coordinates and heading angle $\mathbf{x}_k = [x_k^{\mathrm{R}}, y_k^{\mathrm{R}}, \varphi_k^{\mathrm{R}}]^{\mathrm{T}}$. There is a number of methods available for localization, all of them use the fusion of sensor information. As the cost of hi-tech sensors such as laser rangefinders was not acceptable, the approach based on beacons was chosen. This brings the necessity to install and service the beacons, however the beacons based localization is robust and reliable. Beacons used for Advee localization are based on infrared principle and the signal from the beacons is processed in beacon scanner – the ring of infrared receivers, providing the bearing information – relative angle between the beacon and the robot. This information is fused with the actions $\mathbf{u}_k = [u_k^{\mathrm{t}}, u_k^r]$ – translational and rotational velocities. Actions are taken either from the planner or from robot odometry readings.

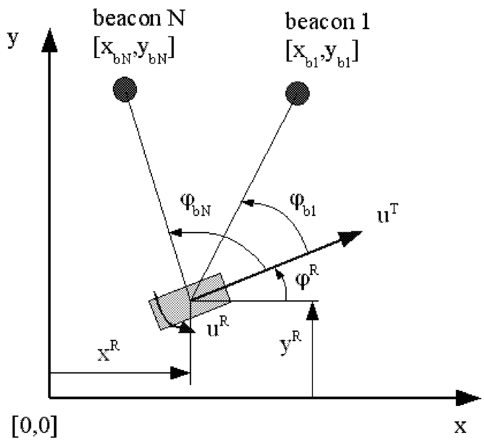

Fig.5: Robot state and beacons measurement scheme

The fusion is performed by Extended Kalman filter (EKF), that combines the motion model with the sensor readings (1).

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{v}_k \ , \\ \mathbf{y}_k &= h(\mathbf{x}_k, k) + \mathbf{w}_k \ . \end{aligned} \tag{1}$$

The motion model is given by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k = \begin{bmatrix} \cos\varphi_k^{\mathrm{R}}\, u_k^{\mathrm{t}}\, \Delta t + x_k^{\mathrm{R}} \\ \sin\varphi_k^{\mathrm{R}}\, u_k^{\mathrm{t}}\, \Delta t + y_k^{\mathrm{R}} \\ u_k^{\mathrm{r}}\, \Delta t + \varphi_k^{\mathrm{R}} \end{bmatrix} + \mathbf{v}_k \tag{2}$$

and measurement for a single beacon of known location at $\mathbf{x}_{\mathrm{b}}$ by

$$\mathbf{y}_{1k} = [h_1(\mathbf{x}_k, \mathbf{x}_{\mathrm{b}1})] + [\mathbf{w}_{1k}] \tag{3}$$

where

$$h_1(\mathbf{x}_k, \mathbf{x}_{\mathrm{b}1}) = [\mathrm{atan2}(y_k^{\mathrm{R}} - y_{\mathrm{b}1}, x_k^{\mathrm{R}} - x_{\mathrm{b}1}) - \varphi_k^{\mathrm{R}}] \ . \tag{4}$$

EKF prediction and correction is of common form $(5, 6)$

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) \ , \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_k\, \mathbf{P}_{k|k}\, \mathbf{F}_k^{\mathrm{T}} + \mathbf{V}_k \ , \end{aligned} \tag{5}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\, \tilde{\mathbf{y}}_{k+1} \ , \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\, \mathbf{H}_{k+1}\, \mathbf{P}_{k+1|k} \end{aligned} \tag{6}$$

where

$$\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \ , \qquad \tilde{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k}, k+1) \ , \qquad \mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\, \mathbf{H}_{k+1}^{\mathrm{T}}\, \mathbf{S}_{k+1}^{-1} \ ,$$

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\, \mathbf{P}_{k+1|k}\, \mathbf{H}_{k+1}^{\mathrm{T}} + \mathbf{W}_{k+1} \ , \qquad \mathbf{H}_{k+1} = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \ .$$

The partial derivatives of nonlinear functions of motion and measurement models that are required are as follows:

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} = \begin{bmatrix} 1 & 0 & -\sin\hat{\varphi}^{\mathrm{R}}_{k|k}\, u^{\mathrm{t}}_k\, \Delta t \\ 0 & 1 & \cos\hat{\varphi}^{\mathrm{R}}_{k|k}\, u^{\mathrm{t}}_k\, \Delta t \\ 0 & 0 & 1 \end{bmatrix} , \tag{7}$$

$$\mathbf{H}_{k+1,1} = \begin{bmatrix} \dfrac{1}{1+\left(\dfrac{y_{\mathrm{b}1}-\hat{y}^{\mathrm{R}}_{k+1|k}}{x_{\mathrm{b}1}-\hat{x}^{\mathrm{R}}_{k+1|k}}\right)^2} \dfrac{y_{\mathrm{b}1}-\hat{y}^{\mathrm{R}}_{k+1|k}}{(x_{\mathrm{b}1}-\hat{x}^{\mathrm{R}}_{k+1|k})^2} \\ \dfrac{1}{1+\left(\dfrac{y_{\mathrm{b}1}-\hat{y}^{\mathrm{R}}_{k+1|k}}{x_{\mathrm{b}1}-\hat{x}^{\mathrm{R}}_{k+1|k}}\right)^2} \dfrac{1}{x_{\mathrm{b}1}-\hat{x}^{\mathrm{R}}_{k+1|k}} \\ -1 \end{bmatrix}^{\mathrm{T}} . \tag{8}$$

The details regarding the beacons based localization can be found in [8]. Navigation scheme using the EKF estimator is shown on Fig. 6.

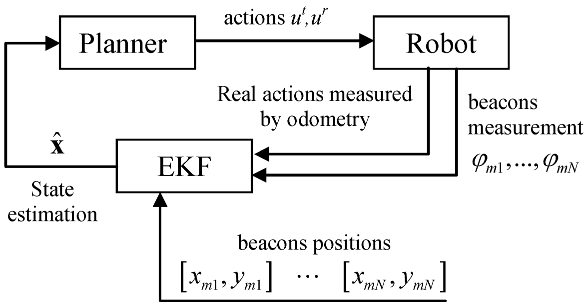

Fig.6: EKF based position estimation in navigation scheme

3.2.2. Motion planner

Motion planner in Advee's navigation scheme directly produces generalized motion commands $\mathbf{u}_k = [u^{\mathrm{t}}_k, u^{\mathrm{r}}_k]$ based on available information. Due to the purpose of the robot the navigation handles different problems depending on the current task. While commonly the motion planner tries to find the best way from point A to B (under various criteria such as shortest path, fastest path, etc.) taking into account various constraints, in presentation robot application reaching the goal is not as important as uniform covering of available space that attracts as many potential users as possible.

Several planners has been tested and so far the best results in real operation where the space is densely populated with people – dynamic obstacles with hard to predict motion – were obtained through finite state machine based planner. The planner works in reactive way, it directly produces generalized motion commands (rotational and translational velocities) based on array of proximity sensors measurement (see Fig. 7), known location of the robot produced by EKF based estimator and goal position. In order to cover the area of available space, the goals are generated in random positions, thus covering the space uniformly. The number of steps in each attempt to reach the goal position is logged and once it exceeds the threshold, the new goal is generated to avoid trapping the robot in unsolvable situation. The goal is also regenerated once the current goal is reached (the estimated position is close enough to the goal).

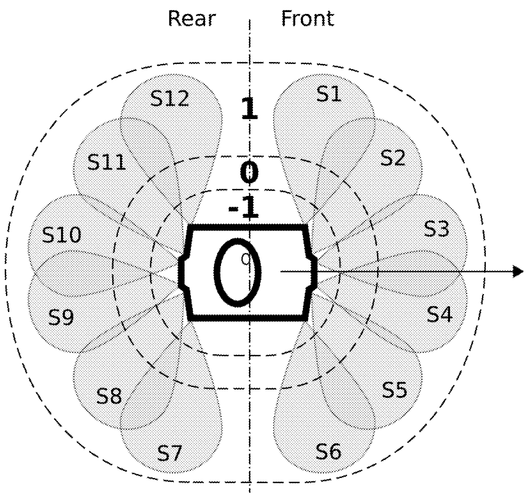

Fig.7: Proximity sensors locations and detection areas

The planner itself is based on Mealy finite state machine (FSM), where outputs are based on both inputs and current state. It is defined as a set of six parameters $(S, S_0, \Sigma, \Lambda, T, G)$, where S it a set of possible states, S_0 is initial state, Σ is the input alphabet $(x_0, \ldots, x_n)$; x_i are possible input values, Λ is the output alphabet $(z_0, \ldots, z_n)$; z_i are possible output values, T is the transition function; $T : S \times \Sigma \to S$ and G is transition function; $G : S \times \Sigma \to \Lambda$. These two transition functions are usually joined to only one transition function : $T : S \times \Sigma \to S \times \Lambda$ which reduces the number of parameters to form $(S, S_0, \Sigma, \Lambda, T)$. The FSM is used in a following way : The sensor measurements are transformed to the input alphabet Σ and output alphabet Λ defines the robot motion. In particular :

– $S = \{s_0, s_1, s_2, s_3\}$; where s_0 = Forward, s_1 = Backward, s_3 = Pause, s_4 = Stop,
– $S_0 = s_2$ = Pause,
– $\Sigma = \{-1, 0, 1\}$ is defined with respect to the robots detection areas (see figure 7)
– $\Delta = \{z_0, \ldots, z_8\}$; where z_0 = Follow_a_goal, z_1 = Turn_right_sharply, z_2 = Turn_left_sharply, z_3 = Turn_right, z_4 = Turn_left, z_5 = Follow_a_corridor, z_6 = Backward, z_7 = Back_right, z_8 = Back_left,
– T and G are joined into one transition function $T : S \times \Sigma \to S \times \Lambda$.

Figure 7 shows the sensors location and its orientation on the robot (top view). Each sensor has its detection range divided into the three discrete areas, which are used as inputs. The discretization is accomplished via the definition (9).

$$F(d) = \begin{cases} 1 & d > 0.6\ , \\ 0 & d = \langle 0.2, 0.6 \rangle\ , \\ -1 & d < 0.2 \end{cases} \tag{9}$$

where d is the measured distance.

The FSM defines the action, when chosen the real action (real velocities of the robot) are calculated also in dependence on real sensors measurements. The transition function is defined as a transition table, shown partially on table 1 (as full transition table has 59049 rows). The transition table can be generated automatically using rules systems that describes the behavior of the planner. For details see [9].

Inputs						State	New State	Output
S1	S2	S3	S4	S5	S6			
1	1	1	1	1	1	s_0	s_0	Follow a goal
1	0	1	1	1	1	s_0	s_0	Turn right: sharply
0	0	1	1	1	1	s_0	s_0	Turn right: sharply
1	1	1	1	0	1	s_0	s_0	Turn left: sharply
1	1	1	1	0	0	s_0	s_0	Turn left: sharply

Tab.1: Part of the Mealy state-machine

3.3. Human robot interface

Human robot interface (HRI) is the interaction module that plays essential role during the interaction with the user. The key demands for the module are:

– redundancy – information should be transferred to user in all possible ways in parallel (e.g. both visual and sound),
– robustness – HRI module must be capable of operation even when minor hardware failure occur (e.g. camera fails, printer goes out of paper, etc.),
– flexibility – simple addition of new feature, sensor, etc.

There are other requirements that should be met as well mainly with respect to operation in changing environment, such se parametrization (overall behavior can be easily changed, certain features can be disabled/enabled upon request) and adaptability (HRI should adapt to the user abilities, e.g. longer timeouts for slower users). The demands require to combine all the communication means if possible. The means were briefly discussed above (see Fig. 2), with respect to HRI the division to input and output means is useful, the overview is shown on Tab. 2.

Inputs			Outputs		
Type	Device	Details	Type	Device	Details
Screen	capacitive touch screen	robust touchscreen returning touch coordinates	Screen	LCD screen	hi resolution screen to present visual information
Voice	microphone, soundcard	incoming sound can be recorded and further processed	Voice	soundcard, amp, speakers	modulated voice, currently prerecorded
Vision	CCD camera	instant flow of images further processed (face detection, etc)	Print	printer	thermal prints up to 112 mm wide
Lower level	bumpers, proximity sensors, odometry	data from lower level of robot control (distances from obstacles, current location, etc.)	Motion	motion actuators	safe motion in given area, towards given goal

Tab.2: Advee means

To combine the means and meet all the requirements while keeping the HRI module maintainable the following system was proposed. HRI module is designed to separate the interaction with the user into independent blocks, that are sequentially activated by main HRI engine and only single block is active at the time. This way the overall behavior of the robot can be defined in a single routine controlling blocks activation. An example of the HRI structure is shown in Fig. 8.

Each block can use different means of the robot, therefore to enable access to the means, the particular means are encapsulated into so called sources. In implementation the blocks and sources are represented by classes inherited from BaseBlock and BaseSource classes. To allow access to all the sources the BaseBlock contains links to all sources instances and as the sources can serve to single block only, the mechanism that assigns and releases sources is

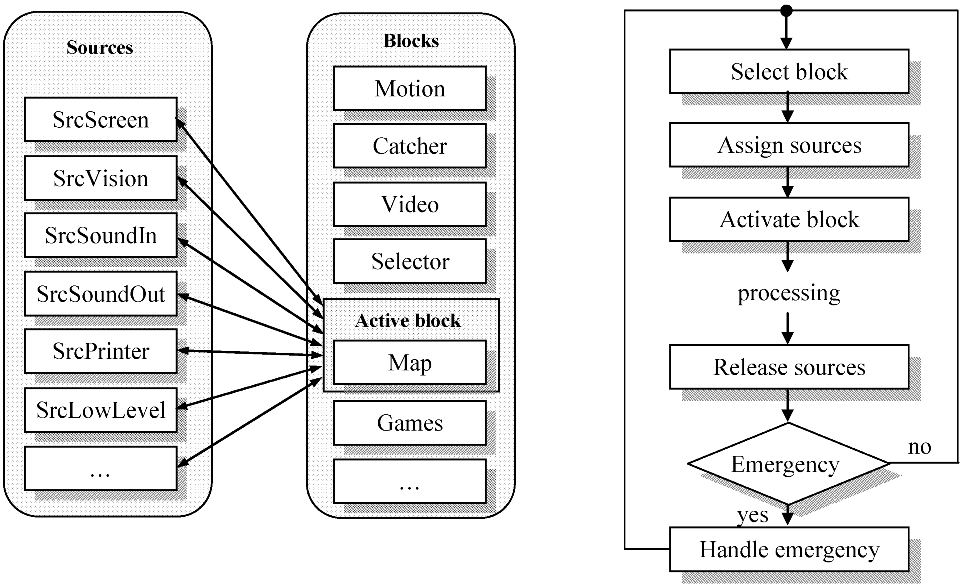

Fig.8: HRI resources management and main HRI loop structure

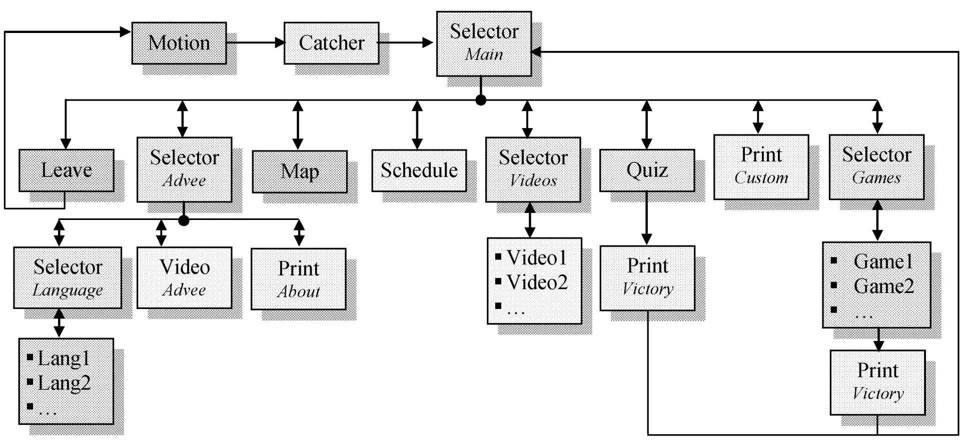

Fig.9: Example of HRI behavior scheme

implemented. Inherited blocks handle particular portion of the interaction, e.g. Motion block is active when robot is moving, Catcher block is responsible for attracting the user to start the interaction, Selector block serves as the menu, etc. Sources are assigned prior to block activation and released once the block is finished. Sources inherited from the BaseSource correspond to the means of the robot shown in Tab. 2. The overall structure of the HRI loop sequence is shown in Fig. 8.

While detailed description of particular sources and blocks is beyond the range of this publication (for details see [10]), the main advantages of given structure should be clear from the schemes. Whenever new source (e.g. sensor) is added, the corresponding source class is implemented and all the blocks can utilize it immediately without changing its code.

Sources can also be mixed together [2], an example is the fusion of proximity sensors data (SrcLowLevel) with the face detector output (SrcVision). Face detector used in Advee is based on well known Viola-Jones algorithm [11]. The distances of obstacles measured by the ultrasound sensors can be used to substantially reduce the false positives of the face detector [12].

4. Operational experiences

Robot Advee has over 600 hours of operation in real environment up to date (May 2011), covering whole range of possible locations, from balls (Fig. 10a) and conferences to shopping malls (Fig. 10b). Also the range of users is wide, from VIP users such as Czech government members to common users in Tesco stores. During the operation all the sensor data, user inputs and robot states were logged for further processing. The key fields to observe are:

- hardware reliability (both mechanical and electrical components),
- middle level algorithms reliability (path planning and localization),
- user interface efficiency (how successful is HRI while working with different types of users).

Regarding the hardware reliability, no essential flaws were observed during the operation in mechanical domain. In the electrical domain the balancing of the individual cells during the operation seems to be the main issue. Balancing the cells only during the charging is

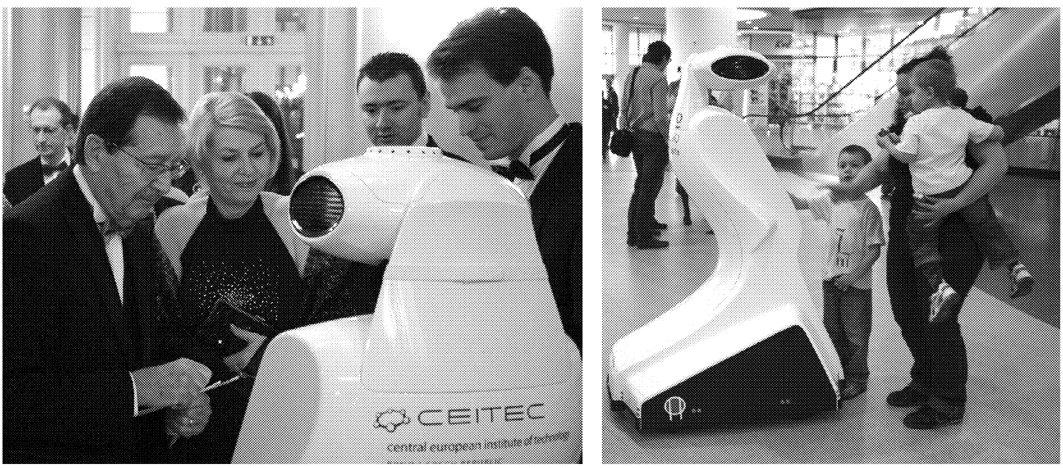

Fig.10: Advee among people – VIPs on Czech ball in Brussels (a), common users in shopping mall (b)

not sufficient and individual cells charging must be applied after every 30 cycles to keep the pack balanced.

Regarding the middle level algorithms, EKF based localization proved to be capable of reliable estimation of robot position. During the operation the loss of position was not observed. The precision of the estimate varied in spaces only partially covered by the beacons, but odometry information is reliable enough to keep the errors low for sufficiently long time periods. The position estimate is quickly restored to precise values once the robot returns to the fully beacons covered areas.

Fig.11: Example of trapped robot, surrounded by people with nowhere to go (Amper trade-fair)

The path planner works reliably, however there is some space for further improvement, mainly in elegance of the motion and also in 'trapped' situations when robot is surrounded by people for extended period of time and should restore the motion, as illustrated on figure 11. Solution of this problem is in direct opposition with the safety demands and will require further investigation.

Human-robot interface evaluation was based on following coefficients:

- ATI – average time of interaction with single user [sec]: interval between the start of the interaction and its end (user left the robot). Longer ATI means single user spent more time with the robot, it is only indirect indicator of how well the interaction went.
- CSR – user catching success rate [–]: rate between successful catch of user (interaction starts) and failure (user walks away). Lower CSR means that people are afraid of the robot, do not know how to start an interaction, etc.
- TIP – interaction timeout percentage [%]: percentage of interactions ended with timeout in arbitrary block other than main menu. Higher percentage means that user left the robot during interaction.
- PPE – person per event [–]: total number of interactions divided by total number of users. If the PPE is higher than 1 then some users used the robot repeatedly.
- BPU – blocks per user [–]: number of blocks activated per user in single interaction. Higher BPU means that user exploited more options during interaction.

The values of the indicators obtained during the operation in real environment with people coming from all the backgrounds are shown in Tab. 3. High variations in ATI and BPU values are caused by different offer of the HRI on different events. CSR value of about 70 % is considered successful, the PPE value is higher than initially expected. The PPE value is essentially influenced by the rewards the robot can provide for given event. With no reward the value of PPE quickly goes to about 1.05 .

Indicator	ATI	CSR	TIP	PPE	BPU
Value	75 ± 22	0.69	14.2	1.20	6.1 ± 4.5

Tab.3: Values of HRI quality indicators

5. Conclusions

The development of autonomous presentation robot Advee took two years from the initial sketches to the prototype operating in real environment. It is to our knowledge the first autonomous mobile robot developed in Czech Republic that is operated commercially among common people. The experiences gained so far indicate that the concept is viable.

There are robots developed for similar purposes by other universities/companies. The closest competition is represented by Scitos based ShopBot [13]. In comparison with Advee the ShopBot offers better maneuverability due to differential drive. It uses vision and laser scanner based navigation. Based on our personal experience the localization of ShopBot fails frequently when the robot is surrounded by the crowd of people (and that should be its natural environment). Advee also surpasses its competition in the HRI, however this is only subjective judgment as there are no hard data available for the competition robot.

The future work on Advee will be focused on extending operational range (currently about 8 hours), improvement of the motion planner to be capable of handling overcrowded areas while keeping the surrounding people safe and further enhancement of human-robot interface (e.g. incorporating the gender identification that is already in progress).

Acknowledgement

The research was partially funded by the AS CR under the research plan AV0Z20760514 and by the project FSI-S-11-15.

References

[1] Burgard W., Cremers A.B., Fox D., et al.: Experiences with an interactive museum tour-guide robot, Artificial Intelligence, 114(1-2), 1999

[2] Bellotto N., Hu H.: Multisensor Integration for Human-Robot Interaction, IEEE Journal of Intelligent Cybernetic Systems, 2005

[3] King S., Weiman C.: Helpmate autonomous mobile robot navigation system, Proceedings of the SPIE Conference on Mobile Robots, 1990

[4] Grepl R. et al.: Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics, IEEE International Conference on Mechatronics (ICM 2011-13-15 April, 2011, Istanbul), 2011

[5] Hrbáček J., Ripel T., Krejsa J.: Ackermann mobile robot chassis with independent rear wheel drives, In Proceedings of EPE-PEMC 2010, Skopje, pp. T5/46–51, 2010

[6] Huang A.S., Olson E., Moore D.C.: LCM: Lightweight Communications and Marshalling, Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010

[7] Huang A.S., Olson E., Moore D.C.: Lightweight Communications and Marshalling for Low-Latency Interprocess Communication, Technical Report MIT-CSAIL-TR-2009-041, Cambridge, USA, 2009

[8] Krejsa J., Vechet S.: Odometry-free mobile robot localization using bearing only beacons, Proceedings of EPE-PEMC 2010, pp. T5/40–T5/45, 2010

[9] Vechet S.: The rule based path planner for autonomous mobile robot, Proc. of Mendel 2011, 17^{th} International Conference on Soft Computing, pp. 546–551, 2011

[10] Krejsa J., Ondrousek V.: The design of human-machine interface module of presentation robot Advee, Recent Advances in Mechatronics, Springer-Verlag, 2011, pp. 423–428

[11] Viola P.A., Jones M.J.: Robust real-time face detection, International Journal of Computer Vision, 57(2):137–154, 2004

[12] Krejsa J.: Reduction of face detection false positives in mobile robot interaction using proximity sensors, Proc. of Mendel 2011, 17th International Conference on Soft Computing, pp. 540–545, 2011

[13] Gross H. et.al.: ShopBot: Progress in Developing an Interactive Mobile Shopping Assistant for Everyday Use, in: Proc. 2008 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC 2008), Singapore, pp. 3471–78, 2008

Received in editor's office: May 27, 2011
Approved for publishing: June 22, 2011