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

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

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

Dynamic decision making for humanoid robots based on a modular task structure

Giulio Milighetti
Fraunhofer Institute IITB
Germany

1. Introduction

Within the last years a new challenging field of robotics has arisen in which assisting robots cooperate and interact with humans, both imitating their behaviour patterns and performing with them and/or in their environment diverse complex tasks.

Due to the ambitious tasks and the complexity of a time varying environment, these "humanoid" robots have to be provided with much more intelligence in comparison with the current industrial robots available on the market.

Thus, the robot has to dispose on the one hand of a large variety of dedicated sensors which cover the full range of human perception modalities and which collect information about the state of the robot and of the environment. On the other hand the control concept has to provide the capability of extracting, combining and analyzing this information in order to cope autonomously with these extreme conditions in the most appropriate way. Moreover, in order to be able to face a manifold of different possible situations, the robot has to manage specific human-like skills. That means it has to perceive, act, deliberate and communicate in a robust manner.

In order to develop advanced cognitive systems, various architectures have been proposed and validated in different application fields (Vernon et al., 2007). However, the implementation of such architectures in the robotics results often in a three-layer structure with a planning, an executive and a functional (control) level (Gat, 1997).

In spite of the many proposed approaches the communication problem between executive and planning is still an unsolved one because of the two different abstraction levels (Joyeux et al., 2007). In order to fill in this gap, some solutions have been developed merging executive and planning into a hybrid level. Following this kind of approach architectures like IDEA (Finzi et al., 2004) and CLARAty (Estlin et al., 2001) have been designed.

Within the Collaborative Research Center 588 "Humanoid Robots – Learning and Cooperating Multimodal Robots" (SFB588) a three-layer cognitive architecture has been presented for coping with the complexity of the challenging tasks that a humanoid robot is supposed to achieve (Burghart et al., 2005). Based on this reference a two-layer discrete-continuous control located in the two lowest levels of this architecture has been developed

at Fraunhofer IITB in order to enable the robot to achieve successfully its goal also in a complex dynamical environment. Only by reconfiguring its plan (discrete control) and by adapting its control strategy (continuous control) the robot can react autonomously to external disturbances and to unexpected events occurred during the execution of its plan.

For choosing the optimal strategy to solve a task, different decision making algorithms driven by the actual situation have been proposed in the past, mainly in the form of reactive planning in mobile robotics (Finzi et al., 2004; Estlin et al., 2001). Most of them rely on the use of Neural Nets, Fuzzy methods or Bayes approaches (Gao & Zhou, 2005; Lundell et al., 2005; Schrempf & Hanebeck, 2005; Serrano, 2006).

The description of discrete-continuous hybrid systems makes often use of discrete models based on Petri nets or automata (Nenninger & Krebs, 1997; Antsaklis et al., 1998). Although the widening of these tools can be led back to applications in the production scheduling, Petri-nets are also used more and more in the field of robotics for modeling and planning complex tasks (Cao & Sanderson, 1994; Chang et al., 2004) as well as for their coordination and supervision (Asfour et al., 2004; Lehmann et al., 2006). Many applications can be found for example both in the area of unmanned vehicles (Palomeras et al.,2006) as well as of humanoid robots (Kobayashi et al., 2002).

The discrete-continuous control concept developed at Fraunhofer Institute IITB (Milighetti & Kuntze, 2006) utilizes a Fuzzy decision making algorithm supported by a modular task sequence based on Primitive Skills (PS) and modelled by Petri nets which considerably improves the autonomy and the flexibility of the robot. The efficiency of the proposed concept is demonstrated in a first experiment on a grasping task in which different vision and acoustic sensors can be deployed.

2. Control Architecture

The quality of a plan depends strongly on the accuracy of the available information about the environment. In a complex time-varying environment where a large variety of events can not be foreseen (e. g. dynamic obstacles) an off-lined scheduled sequence of actions leads often to a failure in the execution of the task. Therefore a discrete-continuous control has to assure that even in the presence of unpredicted conditions the plan is dynamically adapted in order to enable the robot to achieve the goal.

The basic structure of the discrete-continuous control concept developed for the supervision of the robot throughout its task is shown in Fig. 1. The state of the robot and of the environment in which it acts are captured with the help of proprioceptive (e.g. encoder) and exteroceptive sensors (e.g. camera, microphone, force-torque). A hierarchical two-level control is then responsible for the interpretation of such an information.

In the upper level, a discrete control processes the measurements coming from the sensors and uses them in order to generate diagnosis signals that contain quantitative information about the continuous state of the system (e.g. position of objects, sounds, forces). As a second step, this information serves for the identification of the discrete state or event

providing the qualitative information about the present situation (e.g. a contact has been established). By interpreting the acquired knowledge about both the continuous and discrete state of the system, a decision unit analyses the actual task sequence and if necessary adapts it or introduces new appropriate actions for overcoming unexpected situations. A fast online task adaptation and a transparent and efficient structure on Which supports an easy implementation of an online decision making algorithm, have been obtained by adopting a modular architecture based on the concept of Primitive Skills.

Every robot task can be divided into a discrete sequence of elementary actions (called Primitive Skills – PS), each with its own control strategy. A simple assembling process can for example be described by the following five PS:

PS1. Localisation of the part by means of an optical sensor (e.g. a stereo camera);
PS2. Approach to the object with a time optimal and collision-free trajectory;
PS3. Grasping the part by means of tactile and slip sensors;
PS4. Motion to the assembling point and establishment of the contact;
PS5. Assembling using force-torque sensors.

Once the decision unit has determined the PS sequence that has to be performed and the most appropriate controller to execute the currently active PS, in the lower hierarchy level the continuous control ensures an optimal system response and the attainment of the desired values by adapting the parameters of the employed control law.
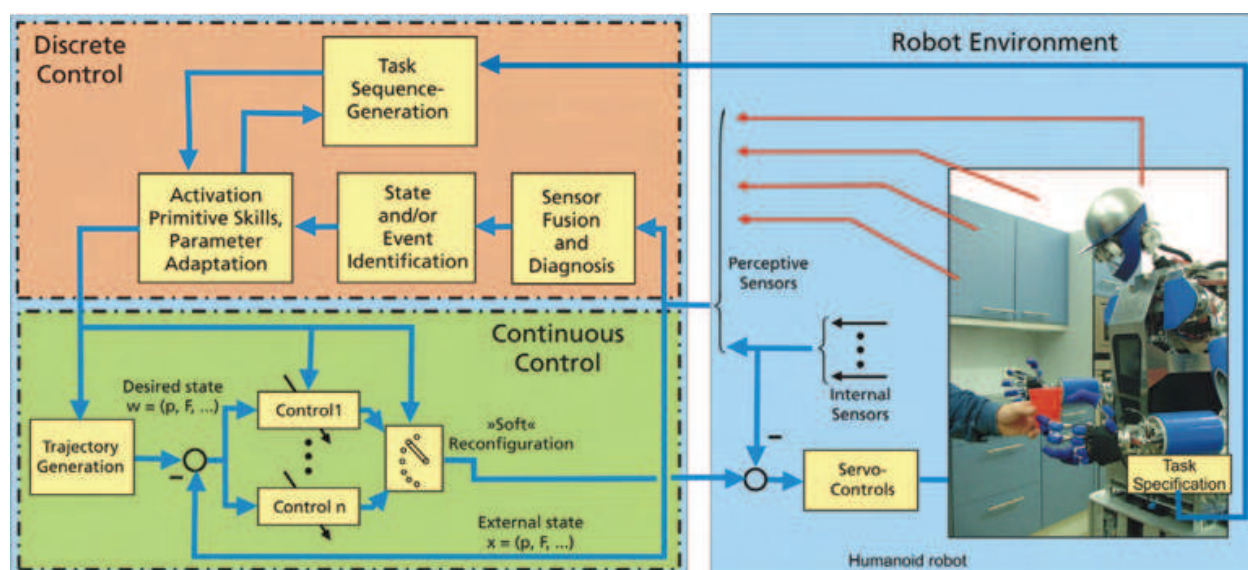

Fig. 1. Discrete-continuous control architecture

3. Task structure based on Primitive Skills and Petri Nets

3.1 Primitive Skills basic concept
A modular structure of a robot task offers numerous advantages. First of all a transparent action sequence is easy to understand and to be handled by the user. Moreover it offers a flexible plan which can be adapted online simply introducing new actions in the sequence and removing or adapting the existing ones. In addition, the presence of software-modules reduces the effort of the programmer since it maximizes the reusability of the code. Finally

such a structure emphasises the discrete nature of the system, offering an optimal basis for the communication with a symbolic planner and supporting also the integration of learned processes, since many learning approaches are based on the decomposition of human actions into sequences of elementary actions (Pardowitz et al., 2007; Peters et al., 2007).

Following this modular approach different concepts have been proposed especially in the area of the manipulation (Hasegawa et al., 1992; McCarragher, 1996; Morrow & Khosla, 1997). In the present contribution an elementary action (Primitive Skill) has been defined starting from the tuple (1) introduced in (Thomas et al., 2003).

$$PS = \{HM, \tau, \lambda\} \tag{1}$$

The variable HM (Hybrid Motion) contains the information needed for the trajectory generation and for the control of the motion (e.g. desired values, active control-law and used parameters, control frame,…) in order to reach the PS-goal. The resources involved in the execution of the PS (e.g. sensors, tools, objects,…) and the associated commands (e.g. activate, open,…) are stored in the variable τ. The third variable λ represents the leave condition of the PS, which has to be fulfilled in order to consider the PS completed and to execute the next one (position reached, malfunctioning,…). More theoretical details about these three variables can be found in (Thomas et al., 2003).

3.2 Extension of the Primitive Skills concept
This concept has been extended in (Milighetti & Kuntze, 2006) by adding two more variables in order to include in the PS information about its actual capabilities and about the influence of the environment on its performance. This extension leads to a PS defined as

$$PS = \{HM, \tau, \lambda, E, a\} \tag{2}$$

The affinity a evaluates the a-priori suitability of the PS for achieving a given goal. Comparing two different PS which are able to reach the same final state, the affinity value tells how optimal is a PS with respect to the other one (e.g. faster, more accurate). For example a localization by means of a visual sensor is faster than a localization with tactile inspection. The affinity is time independent because it is based only on a-priori knowledge about the performance of each PS without taking into account the actual situation.
If PS_{opt} is the a-priori optimal choice in order to achieve a goal g (that means an affinity $a_{opt} = 1$), the affinity a_i of another PS_i compared with it can be expressed for example by:

$$a_i = \begin{cases} 0 & \text{if } g \notin \{goalset_i\} \\ 1 - k \cdot \dfrac{|p_i - p_{opt}|}{\max(p_i, p_{opt})} & \text{if } g \in \{goalset_i\} \end{cases} \tag{3}$$

with p_{opt} the optimal value of the comparison parameter (time, accuracy,…) and

$$0 < k < \max\left(p_i, p_{opt}\right) \tag{4}$$

The efficiency E on the contrary measures the current suitability of the PS by analyzing how good every resource needed for its execution is functioning with respect to the actual state of the system.

The efficiency $E_k(t)$ of a resource k can be decomposed into the following two components:

$$E_k\left(t\right) = av_k\left(t\right) \cdot q_k\left(t\right), \quad av \in \{0,1\}, \quad q \in [0,1] \tag{5}$$

where av_k is the availability of the k-th resource and q_k its quality. While the availability can assume only the values 0 (the resource is available) and 1 (the resource is not available), the quality can be estimated using the whole interval [0,1] with 1 indicating an optimal functional capability and 0 a not functioning resource.

In order to obtain the total efficiency of the PS, the efficiencies of the n involved resources are combined as follows:

$$E\left(t\right) = m \cdot \prod_{i=1}^{n_{comp}} \max_{j=1\ldots n_{coop}} \left(E_{ji}\left(t\right)\right) \tag{6}$$

The single efficiencies calculated by (6) can be adjusted in a n_{coop} x n_{comp} matrix, where n_{comp} is the maximum number of complementary resources and n_{coop} the number of the cooperative ones. With such a structure the efficiencies of the resources, which are working in cooperation (they provide the same kind of measurement) can be listed in each column (see equation (7)). Thus, according to equation (6), the total efficiency of the PS can be evaluated by multiplying the maxima of each column.

$$\uparrow_{coop} \begin{pmatrix} E_{11} & \cdots & E_{1j} & \cdots & E_{1n_{comp}} \\ \vdots & & \vdots & & 0 \\ E_{i1} & & \vdots & & \vdots \\ 0 & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & E_{n_{comp}j} & \cdots & 0 \end{pmatrix}_{\rightarrow comp} \tag{7}$$

In the PS4 of the example introduced in paragraph 2 ("Motion to the assembling point") the assembling point can be identified by means of several fixed cameras while the establishment of the contact with the surface can be supervised with the help of a force-

torque-sensor. In this case the efficiency of the PS is evaluated taking the maximum of the camera efficiencies and multiplying it with the efficiency of the force-torque-sensor.

A memory factor m has been introduced in (6) in order to consider a learning capability of the robot with respect to previous execution of the same PS. A break of the action before the goal has been achieved results in a decrement of m, a successful termination in its increment.

Since the efficiency measures the present performance capability of a PS, its value is actualized online during the whole robot task. In this way the discrete control has access at any times to the information that describes in a compact form the influence of the state of the system on the actions of the robot.

3.3 Task modeling by means of Petri Nets

In order to achieve the desired goal, the robot can plan the execution of different complex actions. The division of each action into a sequence of PS can result either from the segmentation used for learning the action (Pardowitz et al., 2007) or from automatic approaches (Bagchi et al., 2000). Thus, the complete robot task results in a chain of PS, which can be intuitively modelled by means of a Petri net associating every PS to a place. Following this approach, the exemplary assembly task presented in paragraph 2 can be described by the net of Fig. 2.

The marked place represents the actual discrete state of the system, that is the currently performed PS. A transition is activated once the leave condition λ of the executed PS has been fulfilled. By firing the transition the next PS in the net will be activated.
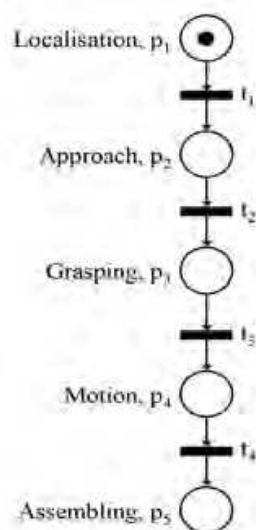

Fig. 2. Petri net modelling an assembling task

Of course the resulting net is not as simple as the one in Fig. 2 for every planned task. In the majority of the cases the robot has to face nets with different alternatives, which could result for example from the following situations:
-    parallel tasks, that the robot has to perform;
-    processes, that can be activated at any time during the execution of the main task;

- different available strategies for solving the given task or some of its sub-actions.

In order to resolve this kind of conflicts in the net (more than one transition firable at the same time) a decision making algorithm at higher level has to be implemented.

4. Decision Making Algorithm

4.1 Decision making algorithm

The decision making algorithm can be seen as the discrete control law of the robotic system. In fact, every time step the actual discrete state is compared with the desired optimal one and corrected in the case that it is not optimal with respect to the actual situation.

The optimal discrete state is the one which ensures the best performances, that is the PS with the highest efficiency (the PS with the currently best functioning resources) and the highest affinity (the a-priori optimal choice). The decision solving the conflict in the net can thus be made by taking the PS with the highest value of the utility function given by the product $E \cdot a$.

$$\max(E) \wedge \max(a) \Rightarrow \max(E \cdot a) \tag{8}$$

At every time step the efficiency of every PS is updated depending on the actual information about its resources and then used to make the optimal decision.

With this approach a decentralized decision making structure is obtained which relies on the measurements attached to every PS independently from its position in the net and thus unrelated to a particular net configuration or conflict (see Fig. 3). In this way the increase in complexity of the decision algorithm is negligible when the number of possible choices rises. Moreover, having most of the intelligence needed for the decision stored locally in every single PS results in an algorithm which works automatically also in case of self-generated task-nets.
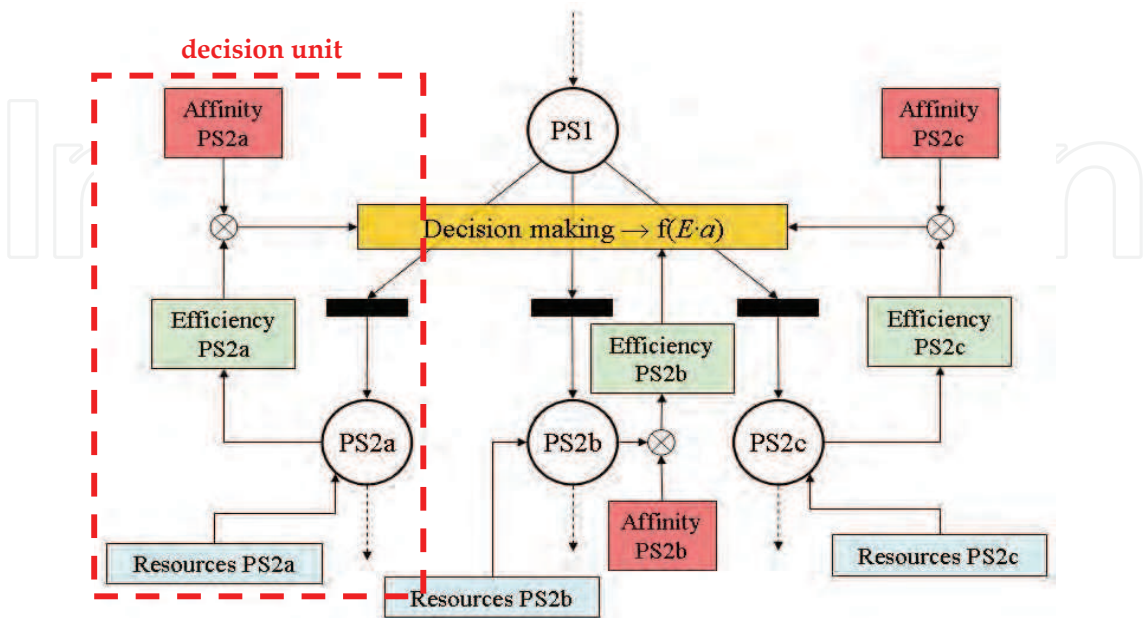

Fig. 3. Decentralized PS-based decision making structure

Comparing by (5) the different PS available in the next execution step, a local optimization problem can be solved finding the optimal action which has to be performed in the discrete task sequence. However, the time horizon of the decision can be extended considering the global net that describes the entire task and finding the optimal path from the currently active PS to the goal.

In order to do this, the arcs entering the k-th PS can be dynamically weighted with $1-E_k \cdot a_k$ obtaining a net where an arc with minimal weight corresponds to a PS with maximum utility ($E \cdot a = 1$).

By using for example a slightly modified version of the Dijkstra algorithm a global optimal path can be evaluated every time step and used to extract the next PS avoiding in this way a deadlock in the task execution that could result by taking an optimal but local decision (see Fig. 4).
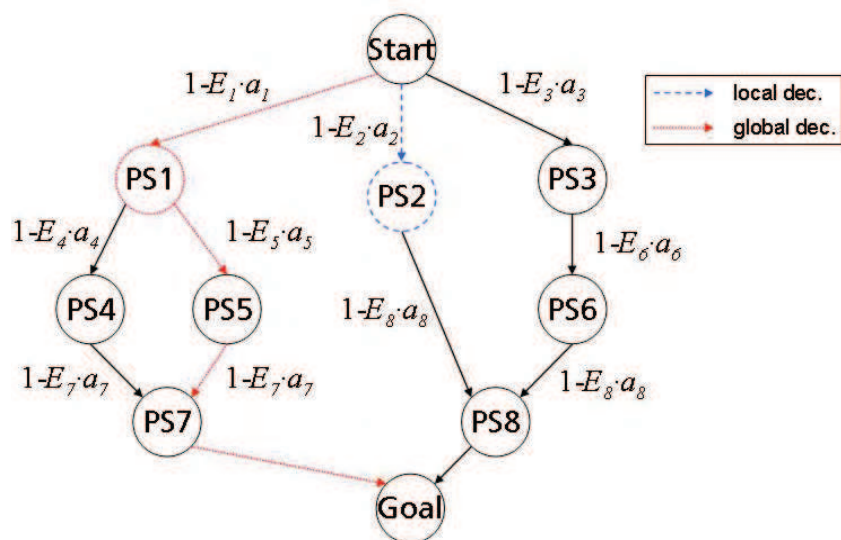
Fig. 4. Global vs. local decision making

4.2 Fuzzy-based efficiency evaluation

Equation (5) has shown that the value of each single efficiency is given by two different parameters:

- the availability av of the resource;
- the quality q of the resource.

Even if the estimation of the quality can be performed using any arbitrary method that returns a value in the interval [0,1], a fuzzy approach has been chosen. Thanks to this kind of approach, it is easier to transfer the human experience into the system, obtaining a more transparent and more comprehensible decision unit.

The fuzzy-based method for the quality evaluation can be clarified by taking as an example the PS1 introduced in paragraph 2, that is the localisation of an object by means of a stereo camera. In this case the two main resources involved are the camera and the object. In order to simplify the example it is supposed that the efficiency of the object is always constant and equal to one. Thus, the efficiency of the PS can be reduced to the efficiency of the camera only.

The evaluation of the efficiency can be carried out on the basis of three main factors:
- availability of the communication between sensor and robot (1 = available, 0 = not available);
- availability of an actual measurement (1 = received, 0 = no new measurement in the last n time steps);
- quality of the last measurement (1 = good and reliable, 0 = bad and/or unreliable).

The quality of a measurement is evaluated by taking into account three more factors that mostly influence a camera:
- luminosity of the environment;
- noise of the measurement;
- working range of the sensor.

The membership functions associated with each of these three factors are shown in Fig. 5.
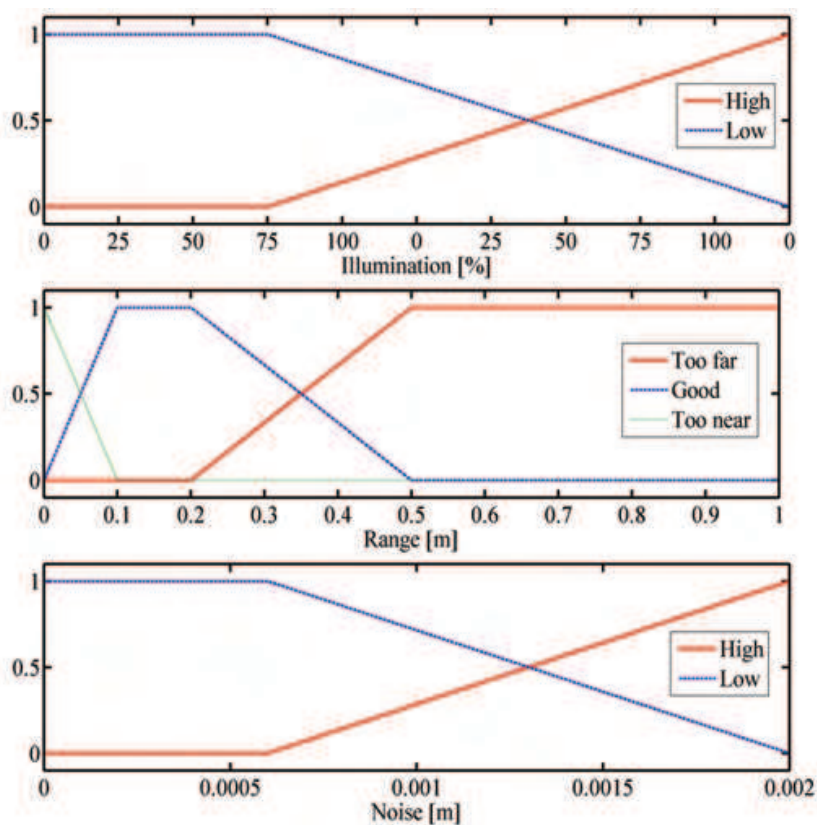
Fig. 5. Membership functions for the variables Illumination, Range and Noise

Once the values have been fuzzified they are evaluated with very intuitive rules like for example

$$\text{If } (\textit{Noise is High}) \text{ or } (\textit{Range is Too far}) \text{ or}$$
$$(\textit{Illumination is Low}) \text{ then } (\textit{Quality is Low})$$

$$\text{If } (\textit{Noise is Low}) \text{ and } (\textit{Range is Good}) \text{ and}$$
$$(\textit{Illumination is High}) \text{ then } (\textit{Quality is High})$$

(9)

After the defuzzification process a value of the quality between 0 and 1 is obtained and is weighted with the corresponding availabilities in order to estimate the value of the efficiency function needed in order to solve the conflict.

5. Experimental Results

5.1 Multi-sensor Robot Platform

The experimental test platform available at Fraunhofer IITB used for the development and investigation of the proposed control concept is shown in Fig. 6. It consists of two robot arms (A) each with 7 degrees of freedom (DoF), a 2DoF pan-tilt sensor head (B) and a five finger fluid hand (C).

For coping with a variety of interactive basic skills the robot is equipped with several redundant (cooperative) and complementary sensors. The head is equipped with a stereo camera able to track predefined objects and with an acoustic sensor (microphone array) able to determine the position of sound sources. Moreover, a miniaturized camera for accurately localizing objects at close range is integrated in the palm of the hand (see Fig. 7).

For the tactile inspection two force-torque sensors are mounted on the wrists (D) and the fingers of the gripper are equipped with tactile arrays and with a slip sensor able to detect the relative motions between end-effector and surfaces in contact with it.

Both cameras as well as the acoustic and slip sensor are connected to a dedicated computer where a first processing of the data takes place. The results are then sent via UDP/IP communication to the main computer where the robot control is implemented.

The different control programs have been developed in C++ under Windows. The control algorithms which have been successfully implemented and optimized on the presented test platform can be transferred and integrated in the common SFB demonstrator ARMAR with the help of the Modular Control Architecture (MCA2).
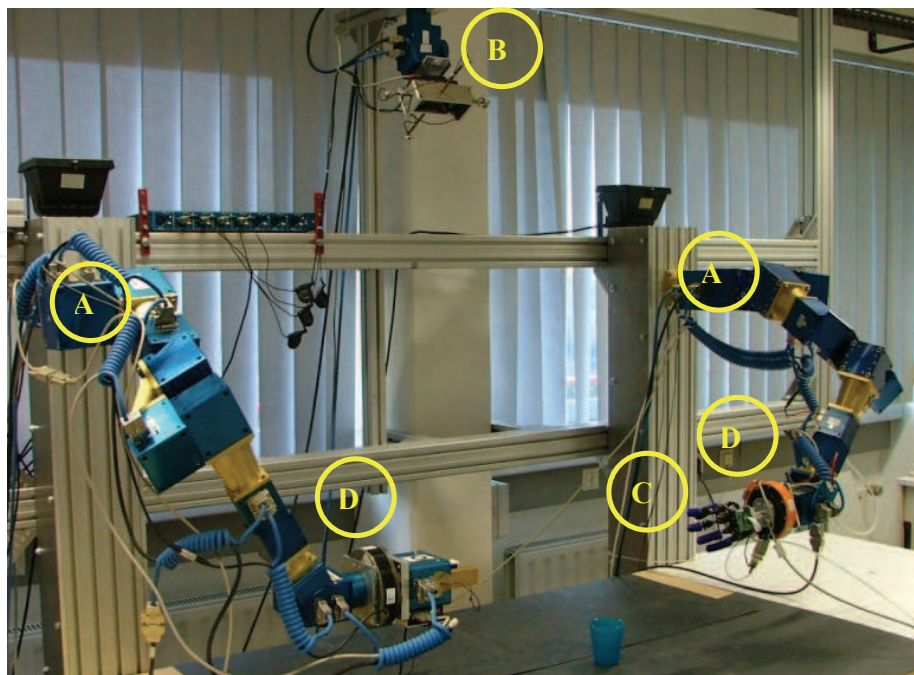

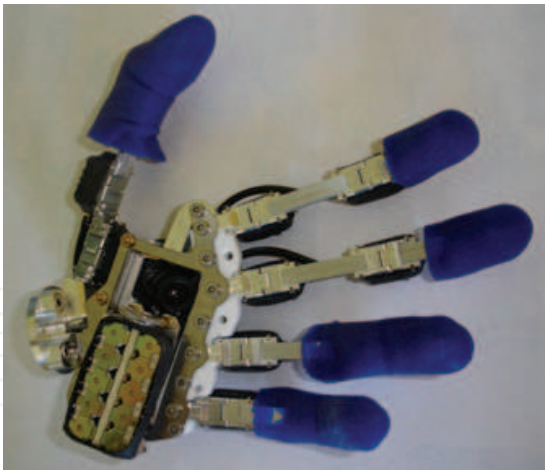Fig. 6. Multi-sensor test and development platform

Fig. 7. Five finger hand with integrated miniaturised camera

5.2. Case study

In order to validate the control concept a case study typical for a kitchen environment has been considered. While the robot is performing a "pick and place" task to transport an object between two points A and B (e.g. taking different ingredients and putting them into a pan), the audio array hears a foreign sound. Thus, one of the three transitions associated with a sound event is triggered:

T_A: the carried object or a similar one has fallen down;

T_B: unknown (or uninteresting) sound;

T_C: an alarm (e.g. microwave) is ringing.

The robot has to cope with this unexpected situation without forgetting the initial task. In Fig. 8 the PS-based task structure in form of a pseudo Petri net describing this example is shown.
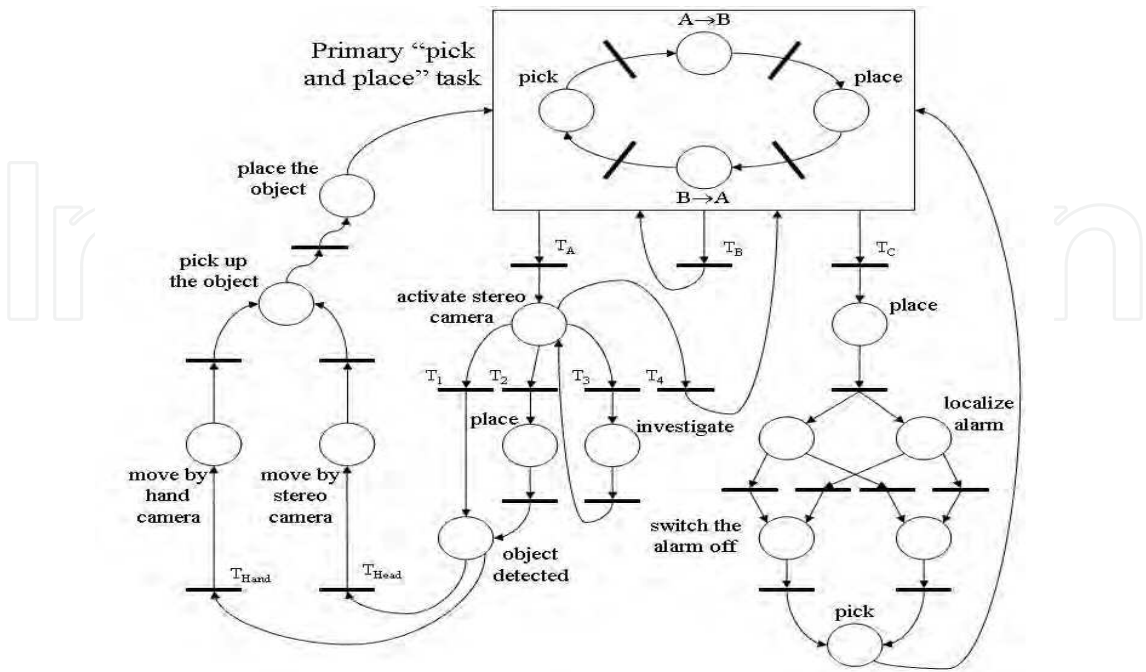
Fig. 8. Pseudo Petri net of a case study

In order to reduce the complexity of the implemented problem, only the first described situation will be discussed (i.e. an object falls down in the robot workspace). An example of the two angles representing the identified impact direction are shown in Fig. 9 (see (Milighetti et al., 2006) for more details).
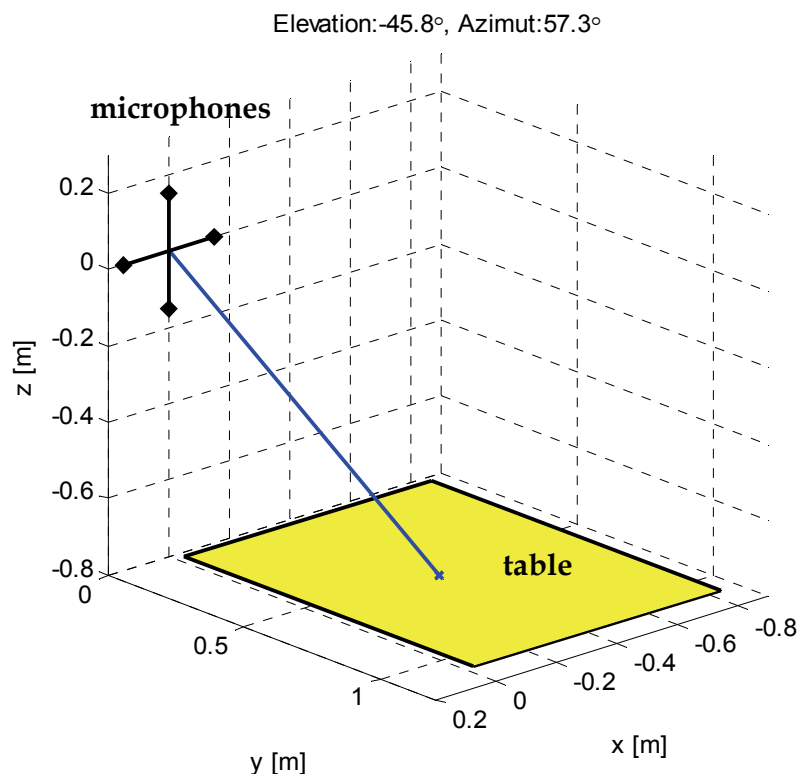

Fig. 9. Example of an audio localization

The robot stops immediately the primary "pick-and-place" task in order to activate a PS-sequence able to cope with the new situation.
First the stereo camera in the head is aligned with the sound direction in order to search for the fallen object. As shown in Fig. 8, four different events are possible at this point and can be distinguished by merging both audio and vision measurements and comparing them with the robot position.
Depending on the identified event, the following four transitions can be fired:

T_1: the carried object has fallen down;
T_2: another similar object has fallen down in the robot workspace;
T_3: no object has been found in the field of view of the camera;
T_4: the fallen object cannot be reached.

Two consistent measurements located in a region far from the actual working area of the robot are shown in Fig. 10. In this case it can be supposed that the impact was caused by a second object that can be picked up only after placing the carried one (T_2). In Fig. 11 instead the two measurements are inconsistent and a more accurate investigation (maybe enlarging the searching area) is required before a decision is made (T_3).
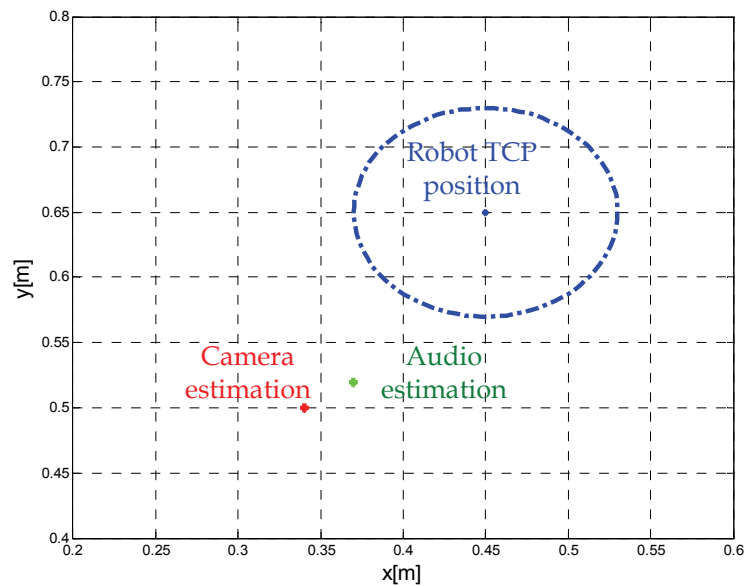
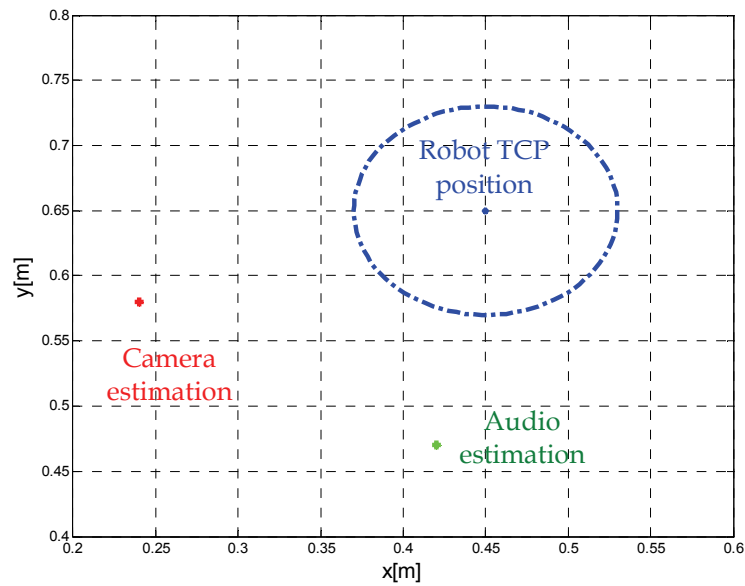Fig. 10. Consistent audio and visual estimations

Fig. 11. Inconsistent audio and visual estimations

Once it has been determined which object has to be picked up, two different vision-based approach strategies can be adopted:

T_{Head}: the robot approaches the object using the measurements of the stereo camera in the head;

T_{Hand}: the robot approaches the object using the measurements of the camera integrated in the hand (see Fig. 7).

Finally, the object can be picked up and the primary "pick and place" task can be resumed.

During the execution of the approach phase it can be shown in detail, how the presented fuzzy-based decision making algorithm based on the evaluation of the PS-efficiency works. The affinities of the two considered PS are defined as follows:

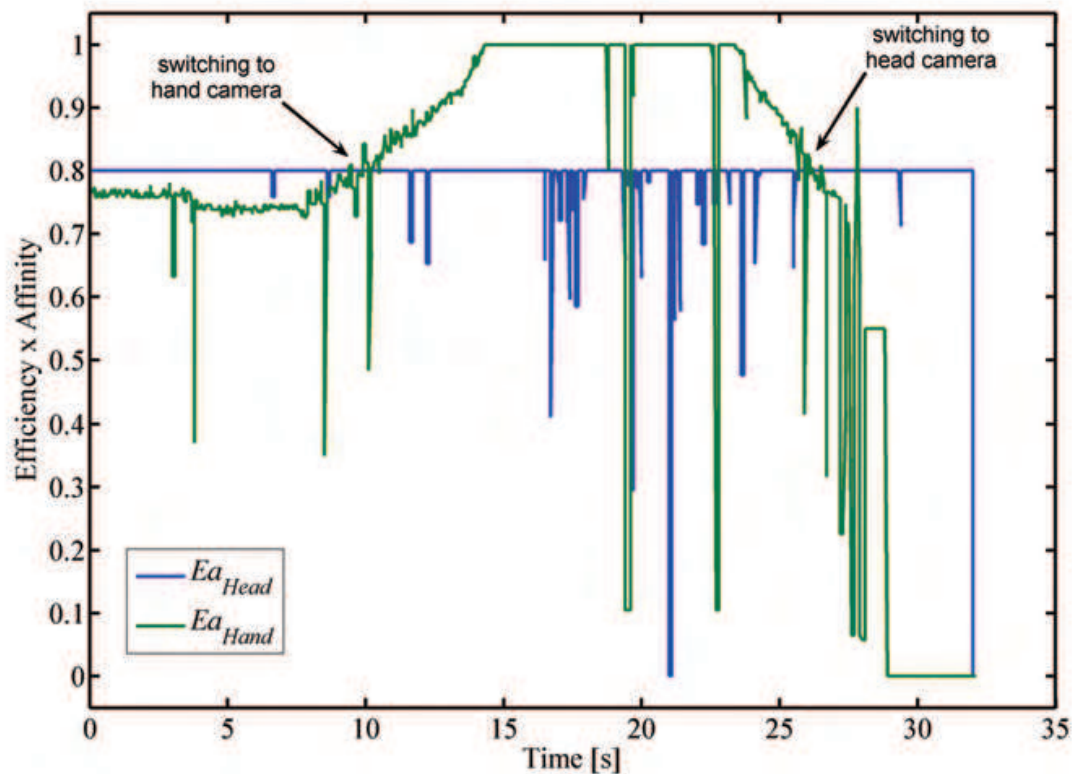$a_{HandCamera}$ = 1 (the most accurate)

$a_{StereoCamera}$ = 0.8

Fig. 12. Utility of the two PS "move by head camera" and "move by stereo camera" during an approach phase

Fig. 12 shows the efficiencies of the two used sensors during a normal approach towards the goal already weighted with the corresponding affinities. Considering for the other employed resources (robot arm and object) a maximal efficiency, the visualised results represent the total utilities of the two PS.

Both efficiencies are influenced by the noise in the measurements and by some false or completely missed measurements (i.e. at ca. 20 seconds for the hand camera or at 21 seconds for the head camera).

Except for these variations, the efficiency of the head camera remains constant during the analyzed time interval because the object is not moving and therefore the working range of the camera is not changing. Only at the end of the task, while the hand is grasping the object, the head camera is no longer able to localize it and its efficiency sinks to zero (at ca. 32 seconds).

On the contrary, the closer the hand camera comes to the object, the better its working range becomes and its efficiency grows accordingly till the optimal value of 1 has been reached.

At a certain instant (at ca. 23 seconds), the hand camera is too close to the object and its efficiency begins to sink once again. In the last grasping phase, shortly before the object has been reached, the localization is no longer reliable as shown also by the extreme fluctuations in the efficiency.

On the basis of the calculated efficiencies, the robot can switch between the two different choices depending on the actual situation, always activating the control strategy based on the currently optimal sensor.

In the presented example the approach is started closing in the control loop the stereo camera. In the central phase of the task the hand camera provides measurements with a higher quality and therefore the PS using it is the best choice. In order to avoid a blind grasping phase at the end of the approach (where the hand camera is no more able to localise correctly the object) the robot has to switch back again to the head camera.
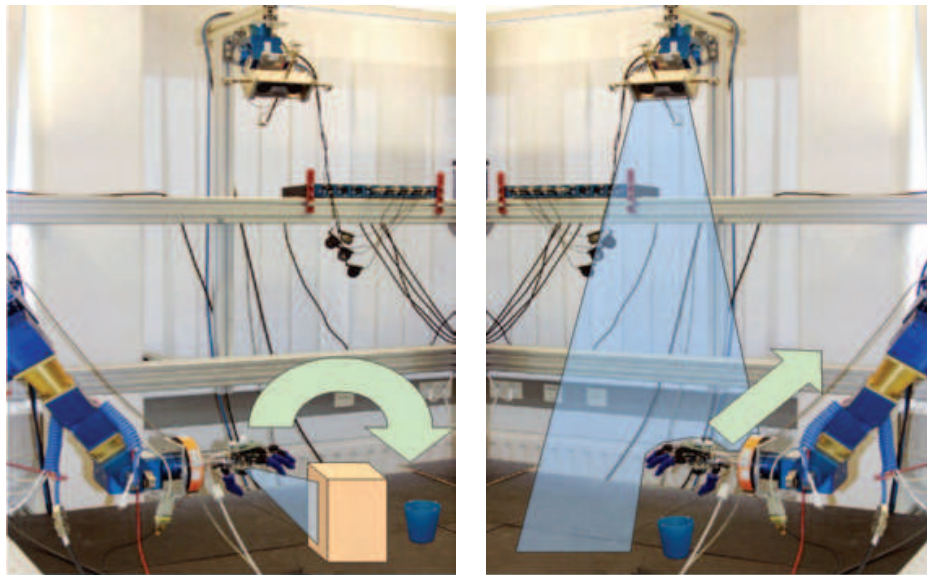

Fig. 13. Unexpected situations during the task execution: occlusion of
a) hand camera b) stereo camera

Also in case of unexpected events like for example the presence of an obstacle (Fig. 13a) or the occlusion of the view due to the motion of the arm (Fig. 13b), the values of the efficiencies of the two sensors and of their associated PS can be used in order to activate some correction in the plan. In the two presented situations, a vertical and a lateral motion can be respectively executed for overcoming the obstructions.

In Fig. 14 and Fig. 15 the efficiencies of the two PS and the cartesian trajectory of the robot TCP during a scenario with several occlusions are respectively shown.

Firstly, four occlusions of the hand camera have been simulated. The correspondence between the lowering of the availability (in this time interval no measurements are received) and the value of the efficiency is clearly observable. The robot reacts with vertical motions until a new measurement is available. Switching off the hand camera (at ca. 45 seconds) leads the robot to choose an execution by means of the stereo camera, although its optimal range was not yet reached.

Finally, the stereo camera was shortly occluded three times and the robot moves laterally

until it has again free field of view. Some isolated peaks in the two efficiencies are visible as in the previous experiment.
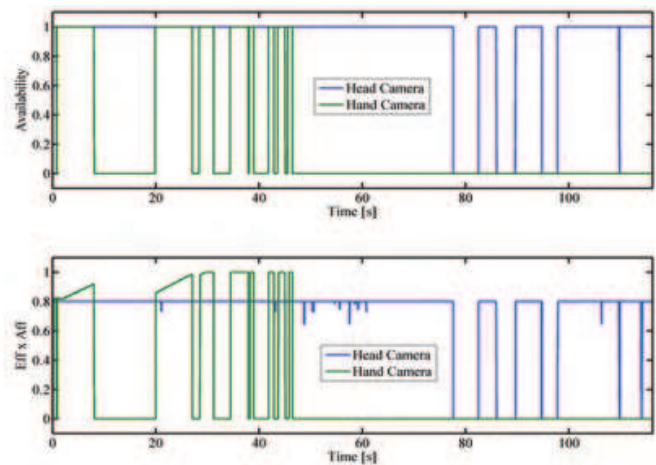

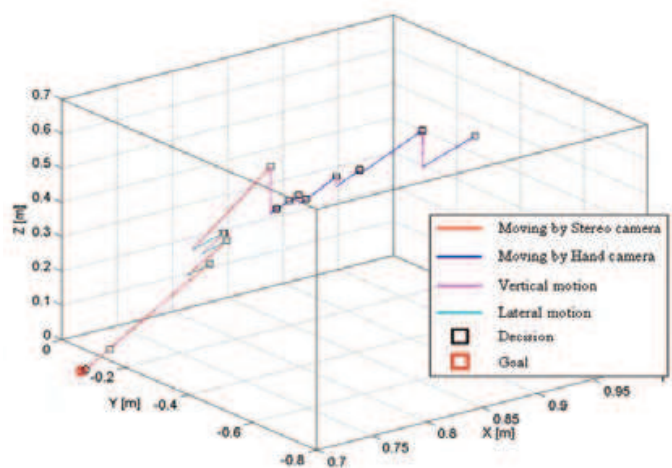Fig. 14. Estimated efficiencies during an approach with several occlusions of the cameras

Fig. 15. Cartesian trajectory during an approach with several occlusions of the cameras

6. Conclusions

A multi-sensor based discrete-continuous control concept able to supervise complex robot tasks in a time varying environment has been presented. A flexible and transparent task architecture has been developed using the concept of Primitive Skills (PS). Every PS can be associated with a place of a Petri net which models the discrete structure of the task.

Through its multi-sensor perception the robot is able to identify failures, unexpected events or circumstances during the execution of the task. A fuzzy approach which models the human knowledge has been investigated in order to give the robot the intelligence to choose always the optimal configuration and control strategy according to the actual situation.

The efficiency of the proposed concept has been demonstrated by first experiments involving a grasping process by means of different visual and acoustic sensors. The achieved results persuasively show that the proposed approach is still valid for more complex tasks.

7. References

Antsaklis, P.J.; Koutsoukos, X.D.; Zaytoon, J. (1998). On Hybrid Control of Complex Systems: A Survey, *Proceedings of the 3rd International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, March 1998.

Asfour, T.; Ly, D.N.; Regenstein, K.; Dillmann, R. (2004). Coordinated Task Execution for Humanoid Robots, *Proceedings of the 9th International Symposium on Experimental Robotics*, Marina Mandarin, Singapore, June 2004, pp. 1-10.

Bagchi, S. ; Biswas, G. ; Kawamura, K. (2000). Task Planning under Uncertainty using a Spreading Activation Network, *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 30, no. 6, pages 639-650, November 2000.

Burghart, C.; Mikut, R.; Stiefelhagen, R.; Asfour, T.; Holzapfel, H.; Steinhaus, P.; Dillmann, R. (2005). A cognitive architecture for a humanoid robot: a first approach, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, 5-7 December, 2005.

Cao, T.; Sanderson, A.C. (1994). Modeling of Sensor-Based Robotic Task Plans Using Fuzzy Petri Nets, *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, 10-12 October 1994.

Chang, Z.-H.; Bian, X.-Q.; Shi, X.-C. (2004). Autonomous Underwater Vehicle: Petri Net Based Hybrid Control of Mission and Motion, *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, 26-29 August 2004.

Estlin, T.; Volpe, R.; Nesnas, I.; Mutz, D.; Fisher, F.; Engelhardt, B.; Chien, S. (2001). Decision-making in a robotic architecture for autonomy, *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space* (iSAIRAS), Montreal, Canada, June 18-22, 2001.

Finzi, A.; Ingrand, F.; Muscettola, N. (2004). Model-based executive control through reactive planning for autonomous rovers, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (IROS), Sendai, Japan, September 28 - October 2, 2004.

Gat, E. (1997). On three-layer architectures, *Artificial Intelligence and mobile robots*, D. Kortenkamp, R. P. Bonnasso, and R. Murphy, MIT/AAAI press, 1997.

Gao, M.; Zhou, M. (2005). Control Strategy Selection for Autonomous Vehicles in a Dynamic Environment, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 10-12 October 2005.

Hasegawa, T. ; Suehiro, T. ; Takase, K. (1992). A Model-Based Manipulation System with Skill-based Execution, *IEEE Transaction on Robotics and Automation*, vol.8, no. 5, pages 535-544, October 1992.

Kobayashi, K.; Nakatani, A.; Takahashi, H.; Ushio, T. (2002). Motion planning for humanoid robots using timed Petri net and modular state net, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 6-9 October 2002.

Joyeux, S.; Alami, R.; Lacroix, S. (2007). A Software component for simultaneous plan execution and adaptation, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (IROS), San Diego, CA, USA, October 29 – November 2, 2007.

Lehmann, A.; Mikut, R.; Asfour, T. (2006). Petri Nets for Task Supervision in Humanoid Robots, *Proceedings of the 37th International Symposium on Robotics* (ISR 2006), München, May; 2006.

Lundell, M.; Jinpeng Tang; Nygard, K. (2005). Fuzzy Petri Net for UAV Decision Making, *Proceedings on the 2005 International Symposium on Collaborative Technologies and Systems*, 15-20 May 2005.

Milighetti, G.; Kuntze, H.-B. (2006). On the Discrete-Continuous Control of Basic Skills for Humanoid Robots, *Proceedings of the International Conference on Intelligent Robots and Systems IROS 2006*, Beijing, China, 9-15 October 2006.

Milighetti, G. ; Emter, T. ; Kuntze, H.-B. ; Bechler, D. ; Kroschel, K. (2006). Primitive-Skill Based Supervisory Control of a Humanoid Robot Applied to a Visual-Acoustic Localization Task, *Proceedings of the Robotik 2006*, Munich, Germany, 15th -17th May 2006.

McCarragher, B.J. (1996). Task primitives for the discrete event modeling and control of 6-DOF assembly tasks. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 2, pages 280–289, April 1996.

Morrow, J.D. ; Khosla, P.K. (1997). Manipulation Task Primitives for Composing Robot Skills. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3354–3359, Albuquerque, Mexico, April 1997.

Nenninger, G.; Krebs, V. (1997). Modeling and Analysis of Hybrid Systems: A New Approach Integrating Petri Nets and Differential Equations, *Proceedings of the Joint Workshop on Parallel and Distributed Real-Time Systems*, 1997.

Palomeras, N.; Carreras, M.; Ridao, P.; Emili, H. (2006). Mission control system for dam inspection with an AUV, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 9-15 October 2006.

Pardowitz, M.; Zöllner, R.; Knoop, S.; Dillmann R. (2007). Incremental Learning of Tasks From User Demonstrations, Past Experience, and Vocal Comments, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol. 37, No. 2, pages 322-332, 2007.

Peters, J.; Schaal, S.; Schoelkopf, B. (2007). Towards Machine Learning of Motor Skills, *20.Fachgespräch Autonome Mobile Systeme (AMS)*, Informatik aktuell, Springer, Kaiserslautern, October, 2007.

Schrempf, O.C.; Hanebeck, U.D. (2005). A generic model for estimating user intentions in human-robot cooperation, *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 14-17 September 2005.

Serrano, N. (2006). A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 9-15 October 2006.

Thomas, U.; Finkemeyer, B.; Kröger, T.; Wahl, F.M. (2003). Error-Tolerant Execution of Complex Robot Tasks Based on Skill Primitives, *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, May 2003.

Vernon, D.; Metta, G.; Sandini, G. (2007). A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents, *IEEE Transaction on Evolutionary Computation*, Vol. 11, No. 2, April 2007.
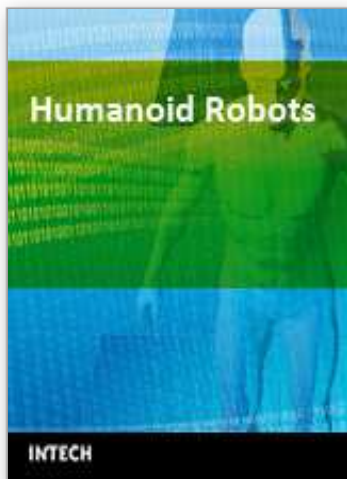
Humanoid Robots

Edited by Ben Choi

Humanoid robots are developed to use the infrastructures designed for humans, to ease the interactions with humans, and to help the integrations into human societies. The developments of humanoid robots proceed from building individual robots to establishing societies of robots working alongside with humans. This book addresses the problems of constructing a humanoid body and mind from generating walk patterns and balance maintenance to encoding and specifying humanoid motions and the control of eye and head movements for focusing attention on moving objects. It provides methods for learning motor skills and for language acquisition and describes how to generate facial movements for expressing various emotions and provides methods for decision making and planning. This book discusses the leading researches and challenges in building humanoid robots in order to prepare for the near future when human societies will be advanced by using humanoid robots.

How to reference

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

Giulio Milighetti (2009). Dynamic Decision Making for Humanoid Robots Based on a Modular Task Structure, Humanoid Robots, Ben Choi (Ed.), ISBN: 978-953-7619-44-2, InTech, Available from:
http://www.intechopen.com/books/humanoid_robots/dynamic_decision_making_for_humanoid_robots_based_on_a_modular_task_structure

INTECH
open science | open minds