Proceedings of the International Conference on
Information Technologies (InfoTech-2017)
20-21 September 2017, Bulgaria

DYNAMIC HIERARCHY IN DECISION-MAKING IN BEHAVIOUR SELECTION OF MOBILE ROBOTS

Roumen Trifonov, Galya Pavlova

Technical University of Sofia,
e-mail(s): r_trifonov@tu-sofia.bg, raicheva@tu-sofia.bg,
Bulgaria

Abstract: The control of mobile robots with autonomous and intelligent behavior makes a significant development. The present paper proposes a dynamic hierarchical control approach that takes into account the current state of the robot subsystems, the specific environment in which it is operated and, on the basis of typified behaviors, the robot finds out an adequate solution for movement and action.
Key words: mobile robot, intelligent behavior, decision making

1. INTRODUCTION

The implementation of intelligent mobile robots requires them to be equipped with a number of sensors that assess the current state of their subsystems, the parameters of the environment, in which the movement is realized, and to use intelligent behavior algorithms for behavour control. The sensor system helps to determine the current state of the robot and the specific dynamically changing environment of movement; intelligent algorithms allow the robot to plan and operate in accordance with its physical capabilities and perform the assigned task [1].

Planning the movement of the body is realized in the same way for all mobile robots - walking, wheeled, tracked, etc. The special feature of legged robots is the presence of a large number of controllable degrees of freedom. Each foot should have at least three drives to be able to place it at any point of the three-dimensional space. Intelligent behavior is the ability of a robot to achieve a particular purpose or to maintain its movement within acceptable tolerances of the normal state and uncertainty of the environments in which it moves and perform the tasks. Where several characteristics are changed simultaneously and unexpectedly, and it is not possible to determine in advance how the robot has to respond to any combination of events (for example, in the control device do not exist programmed behavior) [2].

Intelligent behaviour is characterized by the following features: adaptability, Self-Maintenance, communication (with a human and robots), autonomy, learning and self-learning, forecasting and decision-making [3]. For realization of such behavior the intelligent robot must have the appropriate systems, which have changed over the time corresponding to the stage of development of the technique and technology. A dynamic hierarchical approach for decision making in behavior selection of mobile robot, operating in an environment and state, determined by sensors, is proposed in this paper.

2. BLOCK DIAGRAM FOR CONCEPTUAL PRESENTATION OF THE APPROACH

Decision making about the movement and operation of a smart robot is a complex task that requires taking into account the robot's current state as well as the specific state of the environment and according to this information to select the adequate behaviour. In Fig. 1 is shown a block diagram for conceptually representation of dynamic hierarchy approach in decision making about robot's behavour based on its particular state and the environment in which it is located.

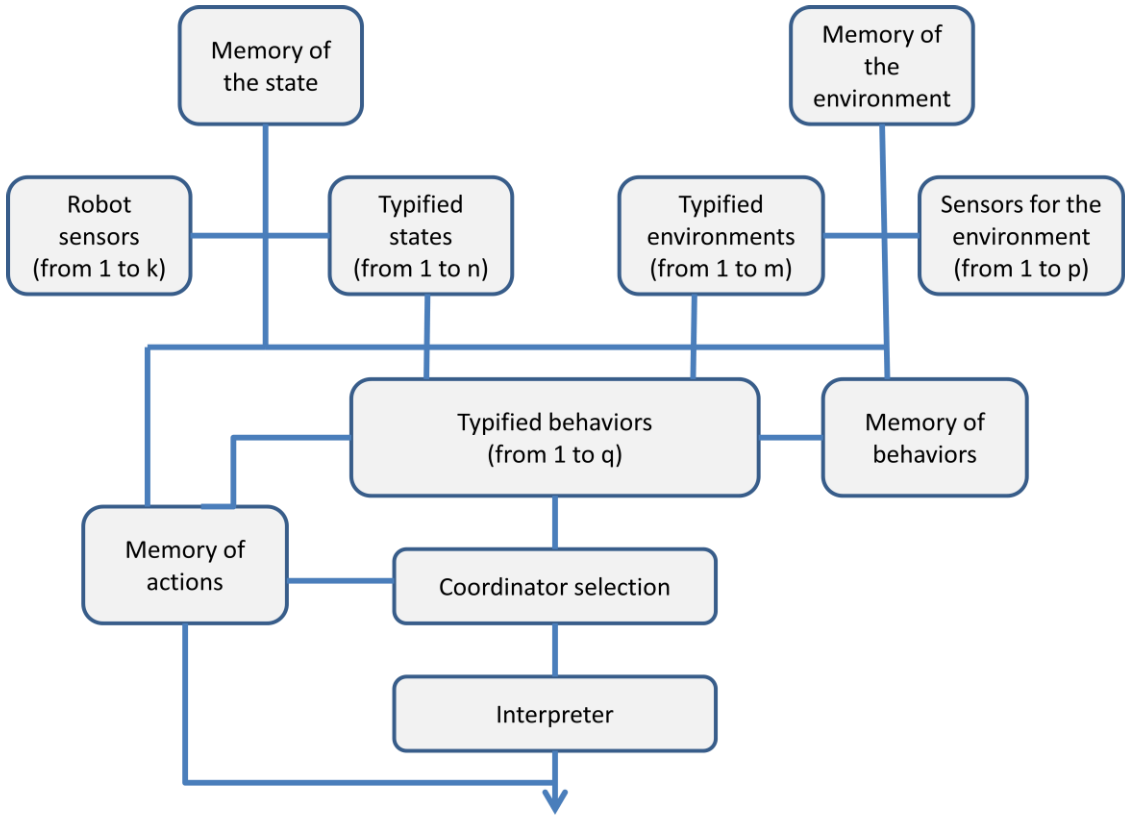

Fig. 1. Block diagram for conceptually representation of the approach

The "dynamic hierarchy" means that the individual behaviors are selected on the basis of a specific situation rather than an authoritatively imposed sequence. In view of the fact that the modern technical means are not able to make a full creative interpretation of the dynamically changing both the state of the robot and of the environment in which it is located, typing the cases is offered.

3. TYPIFYING OF ROBOT STATES

The assessment of the particular state goes through the detection and control of malfunctions. The detection of errors, malfunctions and failures in mobile robots is especially important, especially when they are remote and the human operator can not correct or compensate the failure or problems [4]. Malfunctions could be occurred in: the sensor system, the mechanical system, the power supply and the control. It is difficult to list the whole set, but it is necessary to mark cases of partial and complete loss of mobility. On the base of obtained information from robot sensors (1 to k) the typifying of the states is made and the results are saved in memory of states. Such example for possible state of legs of six legged robot will be considered below taking into account possible combinations of partial loss of mobility of the leg with three controllable degrees of freedom [5].

If the motors are numbered 1, 2, 3 starting from the body of the robot, then normal mobility of a leg is 123 and complete immobility — 000. When there is failure in one of the leg motors, the possible combinations are: 023; 103; 120 and respectively 003; 020 and 100 for failures in two of the leg motors.

If the same approach is applied to the six-legged robot and legs are numbered 1, 2, 3, 4, 5, and 6 and if the legs, denoted with even numbers, are situated on one side and odd-numbered legs on the other, the possible combinations are:

- When one leg is disabled: 023456; 103456; 120456; 123056; 123406 and 123450.
- When two legs are out of action:
  - on the same side: 020456; 120406; 103056 and 123050;
  - adjacent opposite: 003456; 120056; 123400.

Stable movement is difficult to be realized if three legs are out of action, but the combinations are: 003056; 003450; 023050; 100056; 100450; 120050; 103006; 103400 и 123000.

When the robot partially losses its mobility for continuation of its movement it is important to:

- configure the active joints so that the robot is resistant to turnover;
- recalculate movement of active joints so that the robot is stable during the motion on planned trajectory or to perform another acceptable functional task [5].

The robot could also lose stability while having full designed capabilities. It could happen when the inclination of the terrain, on which the robot moves, exceed exposure limits. When some of these values exceed allowed limits, arisen tumble overturns the robot.

These examples typify the condition of the legs. The same approach could also be applied to the rest subsystems of the robot and the states could be typified in sequence from 1 to n, determining for each of them what actions a robot can perform.

4. DEFINED BEHAVIORS ACCORDING TO THE INTERNAL STATE AND THE ENVIRONMENT

The movement algorithm of six legged robot could be summarised with elementary behaviors as follows [6]:

1. the six legged robot moves forward with normal 3+3 gait (legs number 1,3,5 and 2,4,6) until detects an obstacle.
2. if obstacles are detected then:
   a) timer on for t seconds
   b) consider the terrain as uneven
   c) go backward (behavior 4), then uses behavior 2 or 3 to rotate (90 angle) if the obstacle is in the left or right side.
   d) use behavior 1 until detect obstacle
3. else tripod gait with behavior 1.

The obstacle avoidance simple behaviours are: Behavior 1 - no obstacle; Behavior 2 - left obstacle; Behavior 3 - right obstacle; Behavior 4- front obstacle

Similarly, other behaviors can be typified and saved in beheavioural memory M*p*, for example changing the gait.

5. FORMATION OF A CONTROL VECTOR

To improve the proposed approach, it is proposed to introduce three parameters for all elementary behaviors (the approach proposes each more complex behavior to be decomposed to the simplest behaviors). These are memory modules in which information about previous robot and environment states should be saved [7].

If in classical behavioral control architectures the output signal is control information for electric motors, in the considered architecture with dynamical hierarchy, the output B_n is a three-dimensional magnitude. One element is the control vector of the electric motors D, the second vector reflects the state of the robot S_i and the third one (S_r) is the specific characteristic of the environment, i.e.:

$$Bn\ [D, Si, Sr] \qquad (1)$$

The control vector of engines D contains information on how the motors schould move (direction, speed) according to the behavior algorithm. The movements that schoud be accomplished are discreet. This allows the use of fixed and finite number of d_j values for the control vector:

$$D = \begin{bmatrix} d_1 \\ d_2 \\ .. \\ d_j \end{bmatrix}, \text{where } d_j = \{\text{direction, velocity, angle}\} \qquad (2)$$

To simplify the conversion process d_j is coded with numbers. Parameters for determining behavior activity are also defined. Behavior activity means a state that results in a specific reaction at the output based on sensory information or internal processes. When behavior is not active, the output does not receive control information.

The state S_i is determined by information from the robot's internal state sensors. This involves describing and storing specific statesfor a particular robot. The state is typical for a given time moment and that is why a timer or counter that is individual for each subsystem is required.

The specific characteristic of the environment S_r is determined by information from the sensors intended for measurement of its parameters (those that are not measurable are defined by other characteristics). Typifying and saving of some main features makes decision-making easier.

Memory of state, environment, behaviors and actions

At the end of the program cycle the information from the output vectors is saved in the respective blocks (Fig.1). The size of the memory depends on the requirements of the assigned tasks. The output of the corresponding module is an array of stored values. For example, for the memory of stat Ms is:

$$M = \begin{bmatrix} (B_1, B_2 ......B_n)_{t-1} \\ (B_1, B_2 ......B_n)_{t-2} \\ .. \\ (B_1, B_2 ......B_n)_{t-m} \end{bmatrix} \qquad (3)$$

From a software point of view, the memory module is an elementary function containing a m x n array for storing the output vectors and a counter to n.

This memory scheme allows having better information about the whole system. The stored data, together with the information from a particular sensor, are the input parameters that are taken into account by the behavior algorithm. They determine its output activity and its current state. The ability of each behavior to follow the processes in the other ones is provided.

Memory output actions

As the approach is competitive, the arbitrator always chooses only one elementary behavior. In the memory module described above are stored typified behaviours and it is not known at this stage which one will be selected. In order to avoid situations of local traps and to optimize the trajectory, one more module is defined. Its purpose is to store a certain number of cycles. This memory register is also available as an input parameter for any elementary behavior.

6. DYNAMIC COORDINATION MODULE AND INTERPRETER

The implementation of the complex selector begins with the determination of the number and type of particular coordinators, organization, as well as the conditions for their selection. All coordinators are merged into one library module. At a certain point in time only one of the possible coordinators is allowed to work. The organization of each coordinator is connected with the type of the specific task or stage of implementation. The selection of the specific coordinator is done in the coordinator block.

In the interpreter module the codded values of d_j are transformed in appropriate form for direct control of the robot.

7. CONCLUSION

The presented dynamic hierarchy approach for decision-making about mobile robot behaviour, taking into account the current state of the robot and the specific environment in which it is located, has been conceptually developed. Typified robot states and behaviors are considered. More work is needed to personalize the problems to a certain robot and to the environments of movement and action. An essential point is the refinement of the parameters of the output B_n, which consists of components for engine control, the state of the robot and the external environment.

Acknowledgements

Research, the results of which are presented in this publication, is funded by the Internal Competition of Technical University-Sofia 2017, Project № 172PD0001-09.

REFERENCES

[1] Cheng-Hsiung C., Liang-Hsuan C., (**2015**), A Self-Adaptive Intelligent Control System with Hierarchical Architecture, *Journal of Information Science and Engineering*, vol. 31, pp. 1027-1049 ,.

[2] Rzevski G. (**1993**) On Behaviour and Architectures of Autonomous Intelligent Agents: An Engineering perspective, *1-st Int. Round-Table on Abstract Intelligent Agents*, Rome, Italy, 25-27 January,.

[3] Arora, D. (**2016**). Intelligent Mobile Robotics, *A Bimonthly S&T Magazine of DRDO*, vol. 24 No. 6, November-December

[4] Bongard J., Lipson H. (**2004**)Automated Damage Diagnosis and Recovery for Remote Robotics," in *IEEE International Conference on Robotics and Automation (ICRA04)*,.

[5] I. Chavdarov, T. Tanev, V.Pavlov(**2012**)"Statical stabibity of a walking robot," *Mechanika na Machinite*, 20, 4, No 99, pp. 115-120

[6] R. Trifonov, G. Pavlova, I. Pavlov. (**2016**), " Architecture of six-legged robot for simulation of movement under partial loss of leg mobility," in *Proc.of the Int.Conf. on Information Technologies (InfoTech-2016)*, 20-21 September, Bulgaria.

[7] Mohiuddin, A.,M. Khan, Md. Billah, S. Farhana, (**2008**) Walking Hexapod Robot in Disaster Recovery: Developing Algorithm for Terrain Negotiation and Navigation, *Int. Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* , 2, No:6, pp. 795-800, .

[8] V. Nikolov, *Mobile robot control*, Sofia: Abstract of PhD thesis, 2010.