ISSN 0146-4116, Automatic Control and Computer Sciences, 2011, Vol. 45, No. 1, pp. 39–46. © Allerton Press, Inc., 2011.
Original Russian Text © A. Baums, A. Gordyushin, 2011, published in Avtomatika i Vychislitel'naya Tekhnika, 2011, No. 1, pp. 55–63.

Physical Model for Solving Problems of Cost-Effective Mobile Robot Development

A. Baums and A. Gordyushin

Institute of Electronics and Computer Engineering, ul. Dzerbenes 14, Riga, LV-1006 Latvia

E-mail: baum@edi.lv

Received November 3, 2010; in final form, November 17, 2010

Abstract—A physical model including hardware and software is designed for the purpose of the development of a cost-effective mobile robot capable of moving in a heavy-going environment. The analysis of the realizable control system processes is performed and the trial of a physical model is carried out.

DOI: 10.3103/S0146411611010020

Keywords: mobile robot, physical model, detection of objects, distance measurement, routing, energy consumption

1. INTRODUCTION

Presently, the economic circumstances are such that the robots participating in hostility [1–5] being the most complicated devices are among the most expensive machinery. The next level both in structural complexity and in cost is occupied by the robots for serving patients domiciliary and for tidying up [6–8]. The robots for games and teaching of school children and students [9–11] are the most popular, cheapest, and simplest.

Cost-effective robots for the national economy are those capable of working under difficult field conditions and are little known to authors. When creating unmanned mobile robots for information gathering work in a difficult environment, realizable problems concerning the physical model [12] would be appropriate for investigation. By the term physical model is meant a natural prototype that allows modeling the problems and processes of the created field robot operation.

2. PLANNED TASKS OF UNMANNED FIELD ROBOTS

A field robot is meant for solving different types of formidable tasks. Such distinctive tasks are the following:

1—the robot's motion along a path with obstacles that need to be negotiated to reach a given aim;

2—locality tracing in a circular path that is not known in advance and the researching of the objects;

3—tracing of a predetermined area of ground with information gathering from all the objects encountered; and

4—the making of a field map and the use of it for further navigation.

The use of unmanned mobile robots is indispensable for the observation of areas dangerous for human health and life.

The particular tasks requiring realization for the robot's relocation are as follows:

—the orientation of a robot in space;

—the detection of objects that are carriers of useful data;

—the detection of obstacles;

—decision making about the realization of obstacle avoidance;

—planning the sequence of operations;

—stability protection;

—travel and the collected data storage; and

—the realization of a flexible engine management system.

The particular tasks performed according to the robot's function are as follows:

—sensor-based information gathering;

—the estimation of dimensions, color, and other parameters;

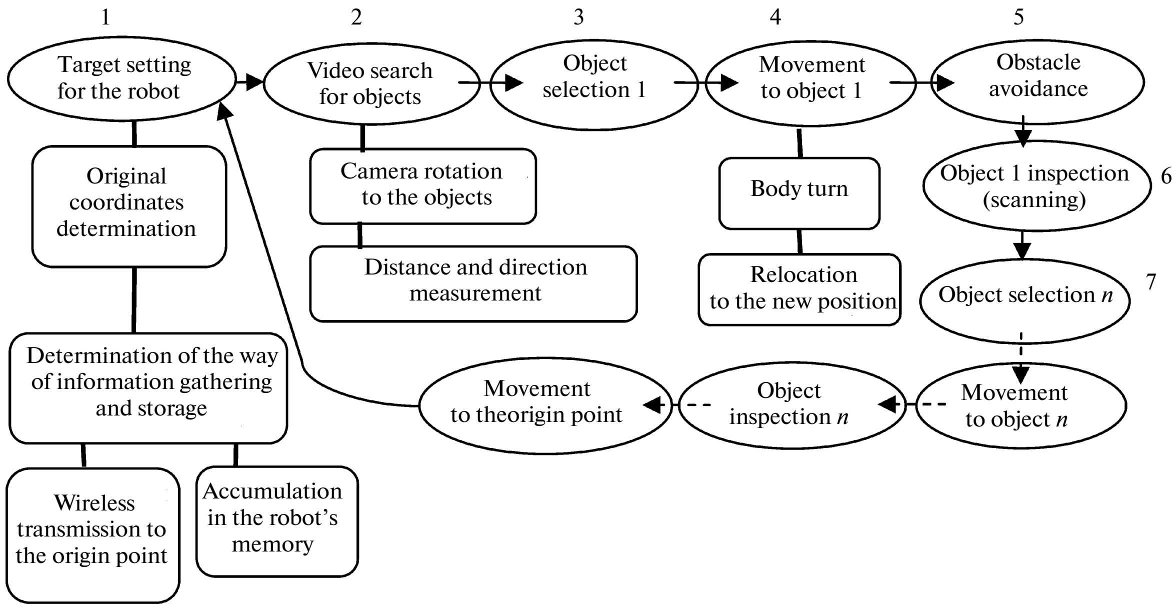

Fig. 1. Fundamental robot states during the locality tracing.

—in-line processing;
—collected data storage;
—collected data transmission;
—the back-out path estimation; and
—the implementation of certain actions with useful objects.

Figure 1 shows 7 fundamental robot states during locality tracing.

3. HARDWARE AND SOFTWARE SUPPORT OF THE MODEL

The cost of a robot created for operation in a difficult environment is due to the cost of the platform on which the robot moves. Thus, the platforms such as the six-legged Mars rover [14] and the BigDog military robot [1] moving with the use of manipulators in the form of legs are both very expensive and complicated to manufacture. A platform on the caterpillar tracks of a remotely controlled toy tank [15] was used in the created model within the framework of the estimation of inexpensive mobile robot building opportunities for use in difficult environments. This platform provides high passability and maneuverability of the mobile robot and allows bearing a load weighing as much as 4 kg. The robot's movement is provided by two electric motors linked with the caterpillars via a reducer. The velocity and direction of the robot's movement are controlled by the motor driver transmitting a PDM signal to the motors. The power source is a lithium polymeric battery with a capacity of 4300 mAh and a working voltage of 9–12.6 V capable of providing the robot with a feed current of 60 A. The directions of the model's extension were defined in the course of the research.

A sensor system including a video system, an electronic compass, ranging sensors, a calculation system of the travel, and other measures was created for the orientation in space, the ambient conditions' perception, and the characterization of the objects considered.

The objects' identification and the determination of their color and dimensions are performed by the image analysis system [16], which consists of a CmuCam3 module with an embedded processor (Philips LP2106 ARM7 and CMOS) and a sensor module (Omnivision OV7630) with a resolving capacity of 352×288 RGB. A controller with the opportunity for the management of four servomotors for the camera's guidance over the objects investigated or for the management of the actuators is installed in the module. A Custom C code was used for the program's development. The program's compiling and debugging was performed in the Cygwin environment.

The following modules were created for the video camera's controlling and the image processing:

—a screenshot module;

—a module for searching for given objects in the screenshot;

—a module for the camera's guidance to the object detected; and

—a module of the link between the camera and the basic controller.

The program for the camera's *X*-rotation processing is given as an example:

```
if ((mas[1]< 950)&&(mas[1]>=x_mid)){
    if (mas[2]< 950){
        do{
            t_step=0;
            if (mas[1]>x_mid+servo_settings.pan_range_far){
                t_step=servo_settings.pan_step;
                }
            else if (mas[1]>x_mid+servo_settings.pan_range_near){
                t_step=1;//(servo_settings.pan_step/2);
                }
            if (mas[1]<x_mid–servo_settings.pan_range_far){
                t_step=–servo_settings.pan_step;
                }
            else if (mas[1]<x_mid–servo_settings.pan_range_near){
                t_step=–1;//(servo_settings.pan_step/2);
                }
            servo_settings.x=cc3_gpio_get_servo_position(3)+t_step;
            if (servo_settings.x>SERVO_HMAX){
                mas[0]=3;
                servo_settings.x=SERVO_HMAX;
                cc3_gpio_set_servo_position(3, servo_settings.x);
                return;
                }
            if (servo_settings.x<SERVO_HMIN){
                servo_settings.x=SERVO_HMIN;
                cc3_gpio_set_servo_position(4, servo_settings.x);
                return;
                }
        cc3_gpio_set_servo_position(3, servo_settings.x);
    }.
```

For the compensation for the measurement errors, the investigation, and the determination of the optimal principles of ranging to the observable objects and obstacles, the sensor system includes two sensor types—infrared and ultrasonic. The advantage of the infrared range finders is the high accuracy of the measurement, but bright sunlight affects the accuracy as well as the objects' color. The infrared range finders are prevented from these drawbacks, but the objects' stippling in turn affects the measurement accuracy. Furthermore, because of the fact that the ultrasonic front extends with the increasing of the distance to the object, it is possible that the signal's front is reflected from a larger object located at a shorter range than the desired object, e.g., from the floor or ground.

The range measurer sensor SRF08 [18] operating at the ultrasonic frequency of 40 Khz is capable of ranging from 3 centimeters to 6 meters.

The following is an example of the distance value reader for the ultrasonic range finder:

```
unsigned int getDistance(int a)
{
unsigned int distance;
```

```
        if (a==0){
    delay_ms(50);
            angle=0;
            angle=i2cread(0xA0,2);
            angle<<=8;
            angle+=i2cread(0xA0,3);
            return(angle);
        };
```

In addition, to measure the distance, two optical sensors are used: an R316-GP2Y0A710YK for ranging from 100 to 550 cm, and an R144-GP2Y0A02YK for ranging from 20 to 150 cm [19].

An example of a processing program for the infrared range finders is as follows:

```
long int getDistance(void){
        float res,l1,l2;
        PORTB.0=1;
        PORTB.1=0;
        delay_ms(40);
        for(i=0;i==19;i++){
            temp_array[i]=read_adc(0);
            delay_ms(10);
        }
        res=0.00489*get_mean();
        l1=9999;
        if ((0.480<=res)&&(res<=2.1)){
            l1=57.3/(res–0.107);
        };
        PORTB.0=0;
        PORTB.1=1;
        delay_ms(40);
        for(i=0;i==19;i++){
            temp_array[i]= read_adc(1);
            delay_ms(10);
        }
        res=0.00489*get_mean();
        l2=9999;
        if ((0.85<=res)&&(res<=2.92)){
            l2=191.111/(res–0.526);
        };
        PORTB.0=0;
        PORTB.1=0;
        if (l1<=l2){
            i=(int)l1;
            }
        else{
            i=(int)l2;
            };
        return(i);
}.
```

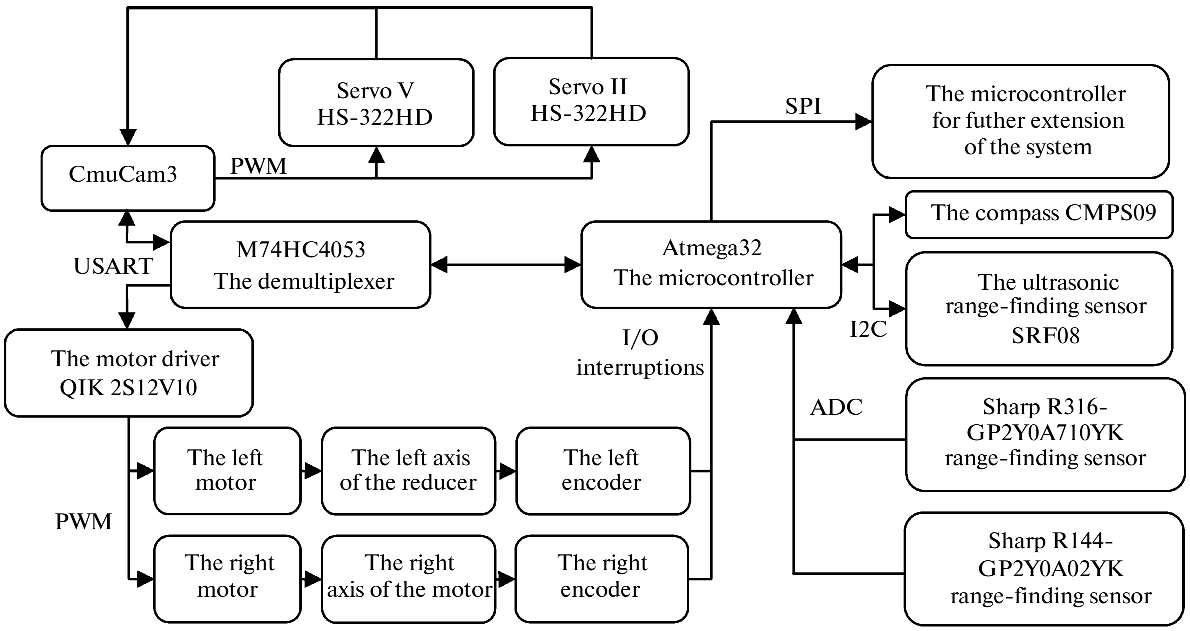

Fig. 2. The electronic structural scheme of the mobile model robot.

To organize the navigation and the calculation of the position relative to the surrounding objects, a CMPS09 electronic compass [17] with a three-axis sensor of the terrestrial magnetic field and a three-axis accelerometer was used.

The following is an example of a reader from the compass module:

```
unsigned int getAngle(int a){
unsigned int angle;
    if (a==0){
  delay_ms(50);
      angle=0;
      angle=i2cread(0xC0,2);
      angle<<=8;
      angle+=i2cread(0xC0,3);
      return(angle);
    };
```

To extend the physical model, the project of the robot's orientation in space was designed on the basis of a GPS receiver [19].

To decrease the power consumption, the possibility of the disconnection of the unused sensors when they are not needed has been organized.

To extend the research possibilities, analog and digital ports were allocated for the connection of sound, temperature, and moisture sensors; a gas analyzer; and an actuator.

The sensor data's processing, the engine management by the driver, and the route calculation are performed by the central RISC processor (by the microcontroller) (Atmega 32) [21] operating at a frequency of 14 MHz. The processor has 2 kB of RAM, 2 types of ROM (1 kB), and flash memory of 32 kB. To accumulate a lot of information, the possibility of the connection of external memory with a volume from 2 to 128 MB is provided. The microcontroller has I2C, SPI, and USART interfaces and 8 ADC channels. The link between the processor and the motor driver is realized by USART via an M74HC4053 demultiplexer [22].

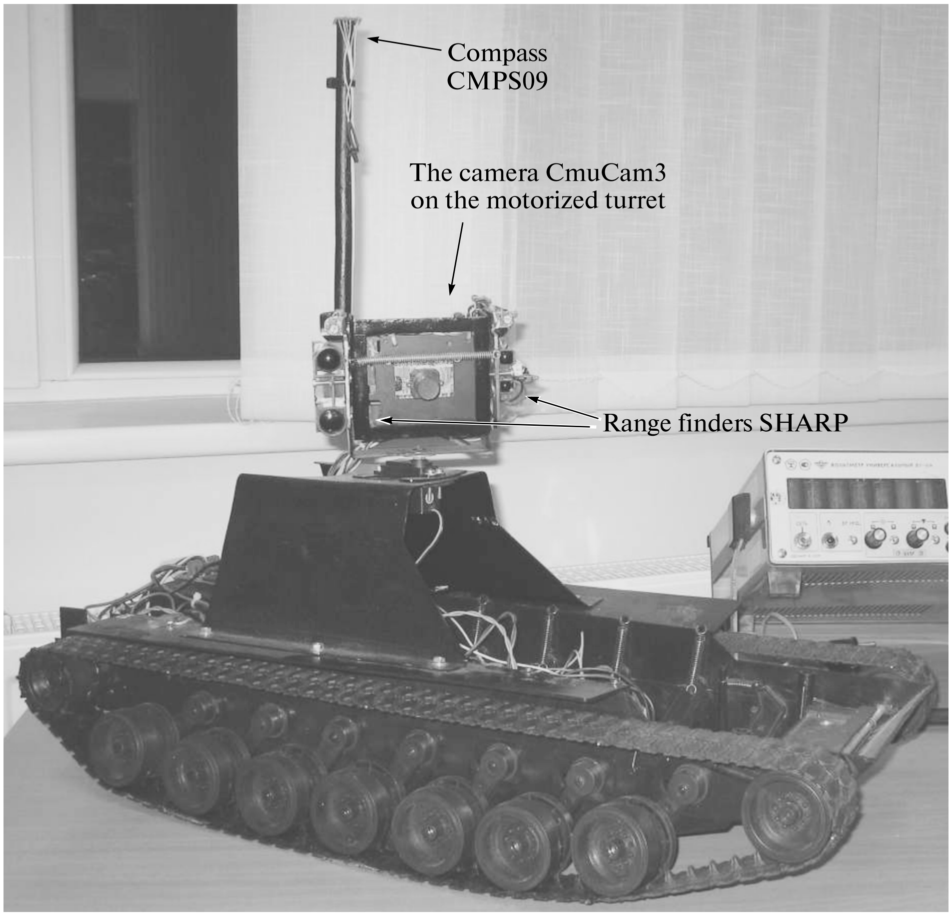

Fig. 3. An external view of the mobile robot prototype.

4. ROUTING AND SEARCHING FOR THE SHORTEST CIRCULAR ROUTE

During the mobile robot operation, the problem of routing and moving is one of the basic and power-consuming problems. As an illustration of the first test task, the following problem has been solved.

N red markers denoting information gathering points and *M* yellow markers denoting filling or battery charging points are located without obstacles on an infinite field. In addition to passing the distance *L*, the robot has an energy storage that can be filled at the yellow markers. The problem is to calculate the shortest circular route among the red markers and to optimize it in terms of the yellow markers in the absence of complete information about the location of all the markers.

To solve this problem, the following sequence of operations was realized.

Step 1. Scanning of the territory

A camera turret assembly is taken to the leftmost position. An azimuth $\alpha 1$ as the initial position is read from the compass. The camera takes a frame and carries out its analysis. If red and yellow markers are in the frame, the closest marker to the vertical axis from the right is chosen. The turret points the camera at the chosen object, and then the azimuth and object distance readout occurs. The data are compared with the table of stored object data (which is empty at the first iteration). In the case of an object's absence in the memory, the object is added to the table of the objects detected. The next frame is taken. If there are no markers in the frame, the turret rotates 30 degrees and the process iterates. When the turret reaches the rightmost position, the process stops. Then, azimuth $\alpha 2$ is read from the compass, the calculation of the azimuth difference $\Delta\alpha = \alpha 1 - \alpha 2$ occurs, and the robot's body turns through this angle. The process repeats. As soon as the robot performs a revolution through 360 degrees, the process stops.

Step 2. Finding the optimized shortest circular route

In the case when the azimuth and object matrix changes, i.e., a new object is found, the calculation of the distance matrix between the previous marker and the next one occurs. The distance matrix obtained among the red objects is sent to the algorithm for finding the shortest circular route based on the ant colony algorithm [23]. The shortest circular route obtained is optimized by the ultimate fuel margin condition and by the necessity to get to the yellow markers for refueling. The obtained sequence is recorded in the one-dimensional matrix of the transitions between the markers. By obtaining the transition matrix, the azimuth sequence matrix is calculated to know from what point and to what angle the turn should be performed for entrance to the next point.

Step 3. Movement among the markers

The group of two vertices consisting of the current marker and the next one are selected from the transition matrix, and the distance between the markers is read. The appropriate azimuth is chosen from the azimuth matrix for what the robot's body is required to be pointed to for the transition to the next marker. Further, the robot points its body at the necessary azimuth and the movement starts. The distance covered and the correction of the course's bend are estimated and corrected by means of an encoder fitted to the reducer motor's axes. As soon as the robot reaches the finite point, the process stops, and the algorithm performs from the beginning.

5. MODEL EXPERIMENT

The model's testing was carried out under ambient conditions. The maximum rate of motion was 1 m/s. The maximum turn rate was 1 revolution per 3 seconds.

The one frame processing rate is 400 ms/frame.

The distance sampling rate is 200 ms/sample. The accuracy of the distance measurement is ±5 cm.

The azimuth sampling rate is 200 ms/sample. The accuracy of the azimuth measurement is 5 degrees.

The maximum current consumption is 7 A/h. The consumption in the direct-current state is 1 A/h.

The capacity of the accumulator is 4500 mA. The average action period is 45 minutes.

6. PROSPECTS FOR THE MODEL's UPDATING

The physical model's platform is appropriate for extending of the video system and the orientation system and linking with external objects. The connection of the transmission media of the research results by wireless lines (Wi-Fi and Wi-Max) as well as the adding of new sensors such as a binocular system of cameras with fine resolution, a laser rangefinder (LIDAR), and an inertial navigation system are planned.

7. CONCLUSIONS

The physical model designed allows the simulation and the processing of algorithms that will be used in real robotic systems in the future. The model is rather flexible and can be easy extended. The model is a good basis for teaching students.

REFERENCES

1. Raibert, M., Blankespoor, K., Nelson, G., Playter, R., and the BigDog Team, BigDog, the Rough-Terrain Quadruped Robot, *Proc. 17th World Congr.*, The International Federation of Automatic Control, Seoul, Korea, 2008, pp. 10823–10825.
2. http://www.foster-miller.com/products-maars.htm
3. http://en.wikipedia.org/wiki/File:Gladiator_240G.jpg
4. http://en.wikipedia.org/wiki/Military_robot
5. http://www.irobot.com/gi/ground/510_PackBot
6. Maaref, H. and Barret, C., Sensor-Based Navigation of a Mobile Robot in an Indoor Environment, *Robot. Autonom. Systems*, 2002, vol. 38, p. 1–18.
7. http://en.wikipedia.org/wiki/Category:Home_automation
8. Srinivasa, S.S., Ferguson, D., Helfrich, C.J., et al., HERB: a Home Exploring Robotic Butler, *Springer Auton Robot*, 2010, vol. 28, p. 5–20.

9. Werry, I. and Dautenhahn, K., Applying Mobile Robot Technology to the Rehabilitation of Autistic Children, *Proc. SIRSX99, Symp. Intell. Robot. Syst.*, 1999, p. 265–272.
10. Iovine, J., *PIC Robotics. A Beginner's Guide to Robotics Projects Using the PICmicro*, McGraw Hill, 2004.
11. http://www.surveyor.com/blackfin/SRV_setup_bf.html
12. http://en.wikipedia.org/wiki/Physical_model
13. Eller-Meshreki, R., Saundurs, T., and Meshreki, S., An Architecture for Planning with External Information Points in a Real-Time Systems, *Proc. 24 ACM Conf. Comp. Sci.*, 1996, p. 58–66.
14. Kermujian, A., *From the Moon Rover to the Mars Rover. The planetary report*, 1990, no. 4, p. 4–11.
15. http://www.tamiyausa.com/product/item.php?product-id=56020
16. Srini, V.P., A Vision for Supporting Autonomous Navigation Urban Environments, *IEEE Computer,* 2006, p. 68–77.
17. http://www.seattlerobotics.com/cmucam3.htm
18. http://www.robot-electronics.co.uk/htm/srf08tech.shtml
19. http://www.acroname.com/robotics/parts/R316-GP2Y0A710YK.pdf
20. http://www.robot-electronics.co.uk/htm/cmps09doc.htm
21. http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf
22. http://www.alldatasheet.com/datasheet M74HC4053.html
23. Bianchi, L., Gambardella, L.M., and Dorigo, M., An Ant Colony Optimization Approach to the Pro-Babilistic Traveling Salesman Problem, *PPSN-VII, 7th Int. Conf. on Parallel Problem Solving from Nature*, Berlin: Springer, 2002.