The 3rd International Conference on ″Computational Mechanics and Virtual Engineering″
COMEC 2009
29 – 30 OCTOBER 2009, Brasov, Romania

ORDERING A MOBILE ROBOT SIMULATION TELEOPERATE

Marius-Constantin Popescu, Niculae Boteanu, Gheorghe Manolea
Faculty of Electromechanical and Environmental Engineering, Craiova, Romania popescu.marius.c@gmail.com

Abstract: *This paper presents a simulated control applications and remote perception, generically called telepresence and teleoperare. In essence, it is transmitted remotely by the IT environment, the signals captured by equipment and controls to the equipment, decided by an operator or an automatic system. In general, sending commands to remote equipment is called teleoperare, who decides whether the control is a human operator or an automatic driving system. Applications where the operator is connected to sensors in remote locations as they are in the environment, in which sensors are installed, are called telepresence applications.*
Keywords: *applications of telepresence, MOBOTSIM program.*

1. INTRODUCTION

According to Webster's encyclopaedia, a robot is "an automatic device that performs functions normally assigned to men, or a machine with human resemblance". Another definition was given by the Robotics Institute of America in 1979. Under this definition, a robot is "a reprogrammable machine, multifunctional designed to move materials, pieces, tools or specialized devices through various programmed motions for achieving a large variety of tasks. A short and widely accepted definition today is: a robot is an intelligent system that interacts with the physical environment through sensors and effects. A summary of similarities between "characteristics" of a human person and a robot is given in Table 1 [9], [11], [12].

Table 1: Similarity of human-robot.

Human	Robot
Brain	Microprocessor, microcontroller.
Eye	Video cameras, light sensors, sonar.
Ears	Sound sensors, ultrasound sensors.
Mouth	Receivers and transmitters for video / data / sound.
Skeleton	Mechanical structure.
Balance	Orientation sensors.
Muscle	Actuators hydraulic / electric / pneumatic.
Food	Power source, batteries.
Legs	Wheels, feet and tracks.

We can distinguish different types of robots [7]: androids, robots built to mimic human behaviour and appearance; static robots, robots used in various factories and laboratories such as robot arms; mobile robots, robots moving in an environment without human intervention and targets achieved; autonomous robots, robots perform their tasks without intervention by a human operator and obtain their energy needed for the functioning of the environment; tele-robots, robots that are guided by remote control devices by a human operator.

What is a mobile robot? A mobile robot could be described as a robot that must move in a certain environment, without a human operator, and to perform certain tasks given by him. So its main feature is mobility.

What is teleoperarea? Sending commands to a remote device is called teleoperare, who decides whether the controls are a human operator or an automatic driving system. Scope:
a) Detection of landmines (Fig.1). One of the serious humanitarian problems facing modern civilization is an antipersonnel mine. Only in 2008 have been destroyed 100,000 mines, but another 2.5 million were placed (particularly in South East Asia). Destruction of these mines is a dangerous and costly operation [5], [10].

Figure 1: Example of robot used in the detection of landmines.

b) Inspection of nuclear contaminated areas (Fig. 2). Following a nuclear accident should be inspected and evaluated the accident damage. Because usually the core status is unknown, the possible consequences of radiation or sending a mobile robot is preferred for assessment of damage [1].

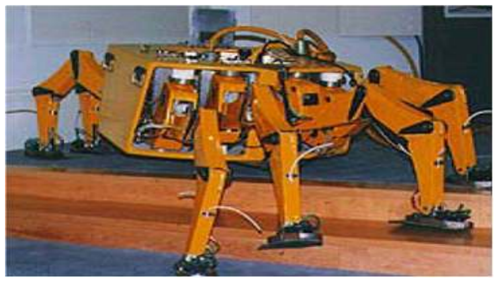


Figure 2: Example of robot used in the inspection of nuclear contaminated areas.

c) Intervention for bomb threat (Fig.3). In civil protection, mobile robots are used to occur in crowded places (airports, railway stations, markets, etc.). To detect and neutralize some explosive devices trap (packages, briefcases, etc.).

Figure 3: Example of robot used in intervention for bomb threats.

d) In space research (Fig.4). Since the '60s space research institutes have investigated the possibility of applying technology teleoperare of mobile robots in space research [6].

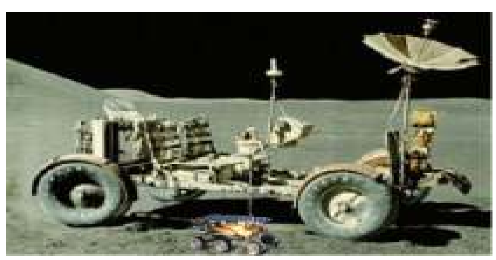


Figure 4: Example of robot used in space research.

e) Pipe inspection (Fig.5). Pipe inspection systems usually consist of several parts: mobile robot platform that provides movement, a typical video camera mounted on a device that allows you to rotate and tilt it and the tools necessary to perform other tests or repairs.

Figure 5: Example of robot used in pipe inspection.

f) Field underwater (Fig.6). They usually operate at relatively large depths of up to 7000-8000 m depth. Some common applications include: mapping, detection of shipwrecks, return to the surface of various objects, inspection Wreck (Magellan 725 was used several years ago in an investigation related to the sinking of a cargo vessel as it is shown that the vessel was deliberately sunk collection of insurance premium).

Figure 6: Example of robot used in the underwater.

g) The military (Fig.7). In military, using the mobile robots brings many benefits. It can carry out reconnaissance, espionage without risking the loss of troops (or to divulge information to their capture), logistic support (transportation of ammunition, medicines, fuel), medical evacuation of wounded soldiers, search and rescue operations, etc.

Figure 7: Example of robot used in military field.

f) Healthcare (Fig.8). They occur in surgical operations [8].

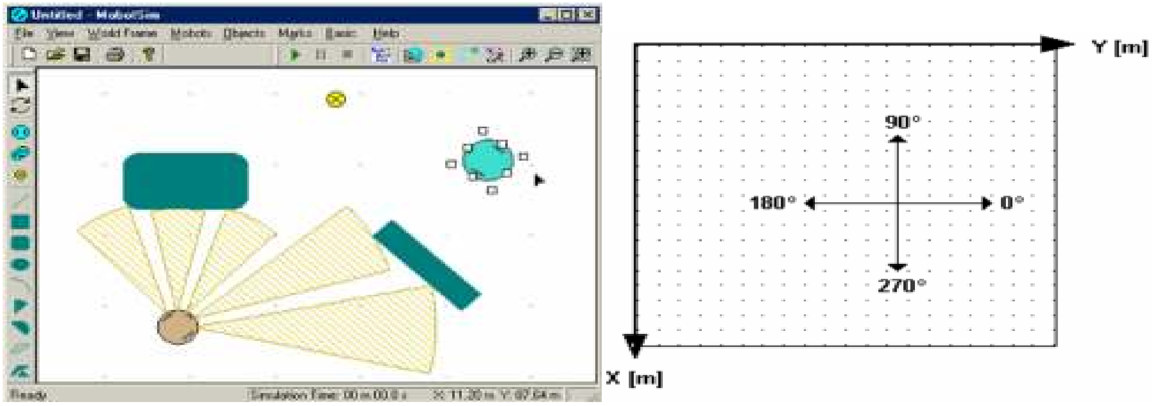
Figure 8: Application of the results of planning ghost.

2. SIMULATION OF A MOBILE ROBOT

Two simulations were done in a special program designed for applications with mobile robots called MOBOTSIM conducted MOBOTSOFT Company. Programming language used is Visual Basic for compatible applications and includes a set of specific functions for programming robots, such as for example [2], [3], [4]: `SetTimeStep(…)`, `SetWheelPosition(…)` etc. Thus, in addition to the basic instructions that you use Visual Basic's (`If. .. Then`

`... Else, For ... Next`, etc..) Was extended "with an adequate language to be dealing with simulated robots `(SetMobotPosition (...), (...) MeasureRange etc..)`

Main Screen. The main screen at MOBOTSIM show the box (the) main (World Frame) that can add elements such as robots, objects, signs, etc. Main box (Fig.9), is a two dimensional region and is working environments for robots (operating space itself). First time to run, start with a Box Main MOBOTSIM default, with size of 20x20 m, but we can resize later with the new desired values for the document. Main box uses a standard Cartesian coordinate system in meters and degrees, and the origin is upper left corner as shown in Fig. 9b.



a) b)

Figure 9: Example of robot used in military field.

Configuring robots. Configuration window allow robots to edit all properties of a robot. These properties are grouped into three categories: General, Geometry and Ranging Sensors as shown in Fig. 10.

Figure 10: Window for configuring robots.

General Properties

Name: Each robot has a unique name, a string of maximum 30 characters. In setting the new robots are called: Mobot 0, Mobot 1, 2 ..., but their names can be changed at any time.

Index: Integer is a number unique to each mobot and is designated by MOBOTSIM dynamically added or deleted when robots and can not be changed by the user.

Position: Robot platform allows choosing colour, determines the colour of the robot trajectory.

X, Y, *Theta Position*: Determines the position and orientation of the robot in Main box, X and Y are given in meters, and Theta in degrees.

Draw Sensor Ranging: If checked, reading sensors milestones is drawn while running. Also they will be drawing as a reference while editing the Main box.

Draw Trajectory: If checked, the robot trajectory is drawn while moving.

Platform Diameter: Allows you to change the circular platform of the robot, which is given in meters.

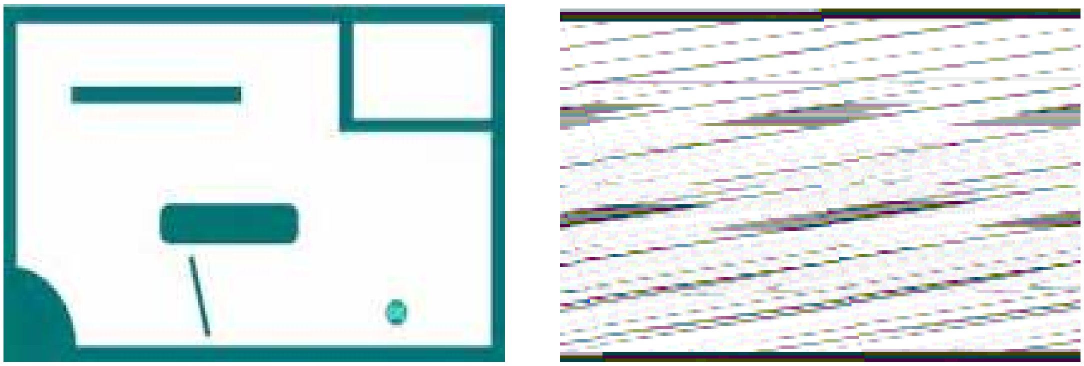
Distance between wheels: Allows you to change the distance between wheels, from centre to centre along the axis.

Wheels diameter: Allows you to change the wheel diameter, which is given in meters.

Radiation Cone: Radiation sensors benchmarking is approximated by a cone. This value is the angle sector is represented by that con-dimensional.

Range from-to: Sets minimum and maximum distance that a sensor can read.

First simulation. In the following we make a simulation in which a robot will follow exactly the route drew new walls will stake through the obstacles, and its final aim is to establish a route. The first step we will do is to build a kind of 'map' in the box we mainly used forms are available in Toolbox (Fig.11a). After we made this map we have no choice but to realize our program to enable our robot to perform "mission". To write our program we have to open the Basic editor. The first thing the robot needs to follow exactly the wall is to take knowledge about the working environment. Thus we will use sensors benchmarking or division (ranging sensors) to measure the distance from the obstacles they encountered. In this simple algorithm we use only one sensor, namely sensor number 4. For this we use the following instructions: `s=MeasureRange(0,4,0)`



a) b)

Figure 11: a) The map was built in the Main box; b) The trail that leds to the robot.

The first parameter,, 0 "is the name of the robot, the second parameter,, 4" is the number of sensor that we used, and the third parameter is 0 "when no one wants to implement due to grid method". Before reading the sensor is convenient to place the robot in a position enabling it to detect a wall. According to the picture we created the environment to place the robot 0 ex: coordinates X=13, Y=12.5, and orientation angle =0 degrees, with instruction `SetMobotPozition`: `SetMobotPosition(0,13,12.5,0)`. The code written in Basic Editor has the following form:

```
Sub Main  SetMobotPosition(0,13,12.5,0); s=MeasureRange(0,4,0) End Sub
```

Now the robot can „see" working environment. A simple algorithm could be the following: if the robot is too close to the wall, then turns to the left; if the robot is too far from the wall, then turns right; otherwise, go forward, translated in Visual Basic this shows thus:

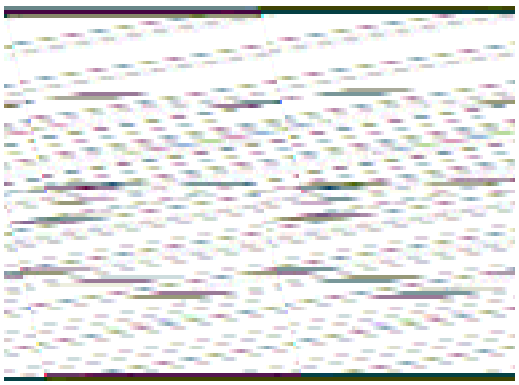
```
If s<0.7 Then SetWheelSpeed(0,0,10)
ElseIf s>0.9 Then SetWheelSpeed(0,10,0)
Else SetWheelSpeed(0,10,10) EndIf
```

Simulation of the robot is now ready, all that remains is we'll test it to see him conduct by pressing „play" for simulation (Fig.11b).

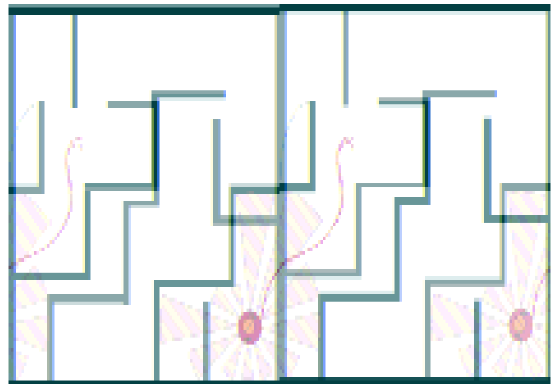
Second simulation. In the following, we will draw a different map for our robot, so we'll write another program to enable besides establish a specific path to reach a specific target (a sign). So his mission is to avoid what he runs into obstacles and reach, the target point. The program in Visual Basic is:

```
Sub Main
 Dim PI,Fcr,Fct,X_target,Y_target,X,Y As Single;Dim X_grid, Y_grid, i,j, C As Integer
 Dim Frx,Fry,d,dist_targ,rot,Fx,Fy As Single; Dim Fcx,Fcy,Rx,Ry As Single
 PI=3.1415927; Fcr=1 `Force constant (repelling)
 Fct=1`Force constant(attraction to the target)
 X_target=GetMarkX(0)`Target coordinates(mark 0); Y_target=GetMarkY(0)
End Sub
SetCellSize(0,0.1) 'Set cell size 10 cm x 10 cm
SetTimeStep(0.1) ,Set simulation time step of 0.1. seconds
Do     'Start main loop
X=GetMobotX(0); Y=GetMobotY(0) 'Present mobot coordinates (in meters)
X_grid=CoordToGrid(0,X);Y_grid=CoordToGrid(0,Y)'indexes of cells where the mobot center is
MeasureRange(0,-1,3) 'Perform a range scan and update the Certainty Grid (max. cell value=3)
Frx=0;Fry=0 'Repulsive Force (x component and y)
'Each occupied cell inside the windows of 33x33 cells applies a repulsive force to the mobot.
For i=X_grid-16 To X_grid+16
        For j=Y_grid-16 To Y_grid+16 C=GetCell(0,i,j)
        If C<>0 Then d=Sqr((X_grid-i)^2+(Y_grid-j)^2
        If d<>0 Then Frx=Frx+Fcr*C/d^2*(X_grid-i)/d
        Fry=Fry+Fcr*C/d^2*(Y_grid-j)/d
        End if
        End if
        Next
        Next
        dist_targ=Sqr((X-X_target)^2+(Y-Y_target)^2)
        'The target generates a constant-magnitud attracting force
        Fcx=Fct*(X_target-X)/dist_targ; Fcy=Fct*(Y_target-Y)/dist_targ
        Rx=Frx+Fcx; Ry=Fry+Fcy 'Resultant Force Vector
        Rot=RotationalDiff(0,X+Rx,Y+Ry) 'shortest rotational difference between
        'current direction of travel and direction of vector R
        SetSteering(0,0.5,3*rot) 'mobot turns into the direction of R at constant speed and
                                 'steering rate proportional to the rotational difference
```

```
        StepForward 'Dynamics simulation progresses time step
        Loop Until dist_targ<0.1 'Loop until mobot reaches the target
End Sub
```



a)



b)

Figure 12: a) The route followed by the robot; b) The robot is reaching the goal pursued.

3. CONCLUSION

The present paper addresses a topic of interest in present: teleoperate respectively telepresence. In general at present trend is towards distributed systems, for globalization applications. This is possible due to the development of computer technology, computer networks, etc. This work can be further developed and applied in different areas where human activity is difficult and sometimes impossible. Mobile robots can be used successfully in dangerous places, where man can not hardly stand or face, for example to get particles of rock from Mars where the air is not breathable, or to check the contents of a car is not any bomb placed, or to carry out various military operations without endangering the life of any soldier. In addition to these uses which require operation in various hazardous environments mobile robots can be used for purposes other less noble such as household cleanliness, already are known around the world famous domestic robots "that give the vacuum by house, remove dust furniture, wash dishes", etc. As possible further developments remember: creating a small robotic system for specific applications; implementing an autonomous robot capable of making decisions; implementing a system of StereoVision, with two rooms that can be attached to the robot; provide platform with various additional sensors such as GPS (Global Position System), radar, sonar, etc.

REFERENCES

[1] Popescu M.C., Petrişor A., *Implementation of moving system in a labyrinth*, International Symposium on System Theory**,** pp.195-199, Vol.2, Editura Universitaria Craiova, 2007,

[2] Popescu M.C., Ilie B., Popescu L., Onisifor O.*, Simulation Hybrid Fuzzy Control of SCARA Robot,* Proceedings of the 3rd WSEAS International, Conference on Applied and Theoretical Mechanics, pp.175-181, Puerto De La Cruz, Tenerife, Spania, 2007.

[3] Popescu M.C., Ilie B., Popescu L., Onisifor O.*, Simulation Hybrid Fuzzy Control of SCARA Robot,* Journal WSEAS Transactions on Systems and Control, pp.105-114, Issue 2,Vol.3, February 2008.

[4] Popescu M.C., Onisifor O., I. Borcosi *Simulation of n-r Robots*, Journal WSEAS Transactions on Systems and Control, pp.149-158, Issue 3, Vol.3, March 2008.

[5] Popescu M.C., Petrisor A., Drighiciu M.A., *Fuzzy Control of the Position for the Piston of an Industrial Robot*. Proceedings of the 12th WSEAS International Conference on Systems, New Aspects of Systems, pp.222-227, Heraklion, Greece, july 2008.

[6] Popescu M.C., Olaru O., *Conducerea optimala a proceselor-Proiectare asistata de calculator in Matlab si Simulink,* Editura Academiei Tehnice Militare, Bucuresti, pp.309-312, 2009.

[7] Popescu M.C., Petrisor A., *Roboţi-Sisteme de control pentru roboti,* Editura Universitaria Craiova, pp.133-139, 2009.

[8] Popescu M.C., *Aplicaţii* in informatica*,* Editura Universitaria Craiova, pp.179-181, 2009.

[9] Popescu M.C., Bulacu T., Radoi A., *Industrial and Mobile Robots Tutorial,* 16th Int. Workshop Robotics in Alpe-Adria-Danube Region, Brasov, may 2009.

[10] Popescu M.C., Petrişor A., Drighiciu M.A., *Fuzzy Control Algorithm Implementation using LabWindows–Robot,* WSEAS Transactions on Systems Journal, Issue 1, Vol.8, pp.117-126, January 2009.

[11] Williams R.L., *Wireless Community Wireless Community Networks*, Texas State Library and Archives Commission Library Development Division, 1999.

[12] Woodward W.S., Peterson J., *Bidirectional H-BridgeDC-MotorMotion Controller*, Electronic design, July 1999.