

## Intelligent OkiKoSenPBX1 Security Patrol Robot via Network and Map-Based Route Planning

Mbaitiga Zacharie

Department of Media Information Engineering,  
Okinawa National College of Technology, 905 Henoko, Nago, 905-2192, Okinawa, Japan

---

**Abstract: Problem statement:** With an increased demand for security and limited numbers of trained security personnel, some security managers have a lot of ground to police and limited staff to cover it. To compensate for shortages of security staff and to reduce the stress of security managers, we have developed an intelligent patrol robot system called "OkiKoSenPBX1". The system integrates a variety of sensors to gather environmental information and to detect abnormal events including intruders. **Approach:** In our approach, the route planning procedure was based on determining a sequence of intermediary goal points or coordinates x and y composing the robot trajectory. **Results:** A qualitative running experimental evaluation had been performed on the 1st floor of the Okinawa national college of technology as a preliminary practical implementation and its real-time performance was excellent, where a student like-guard can take control of the camera pan and tilt functions remotely. **Conclusion:** The real-time performance of the developed system that can leave security personnel hands-free for other important tasks is an irresistible system that can be put into practical use in a public offices facility, manufacturing facilities and various construction sites-everywhere there's a need for advanced frontline security.

**Key words:** Intelligent patrol robot, security robot, wireless network

---

### INTRODUCTION

A robot is a machine designed to execute one or more tasks repeatedly, with speed and precision. There are different types of robots as there are tasks for them to perform. Security robots are protecting property and documenting facilities autonomously around the world. An important aspect of robotic security systems is surveillance of specified area. They provide surveillance on wheels and may be one of the security industry's best kept secret. The Naval Postgraduate School in Monterey, California first attempted the concept of a fully autonomous indoor security robot in the late 1970s. Other agencies including NASA and US Government of Defense were also working on implementation of robotics technology. One of the first robots was ROBARTI, which helped face a full gamut of technical challenges. Subsequent robots grew in capability and with the end of the cold war; the technology was made available for sale to private firms through Reagan-Bush government to industry privatization. After a decade continuous development, security robot keeps as an intensive research issue because of its ever-increasing application to different places and its economical and technological relevance. Interesting application can be seen in robot scanning areas to find explosive devices<sup>[1]</sup>.

Access to program robot units is secured via software. Programming of robot preferences can include instructions such as: If X occurs; do Y. Asset and location protection systems using robots allow hands-free operation via pre-operational programming to response to external stimuli. Over the long haul, it is easy to see that security robots can provide significant cost savings, while they may never replace a human security professional. With an increased demand for security and limited numbers of trained security personnel, some security managers have a lot of ground to police and limited staff to cover it. Others may need to approach an armed barricaded suspect or enemy combatant. Still others need to go into a nuclear reactor to check if all is well. Increasingly, security managers are turning to robots to help get the job done. To compensate for shortages of security staff and to reduce the stress of security managers, we have developed an intelligent patrol robot system called "OkiKoSenPBX1" for University, College Campus and Hotel security control. The system consists of an autonomous mobile robot that can move independently outside as well inside the facility to be patrolled via deployed wireless internet device to that area. Due to its contribution and costs saving, many researchers have done a lot of significant work on patrol robot or security robot whose

some of them can be found in<sup>[2,3]</sup>. These works mainly focus on target perception and identification, robot localization, terrain map updating. Here we are interested specifically in investigating a very critical and still open issue that is paramount for the success of these applications: The route planning. The main goal is to generate a very convenient trajectory to be followed by the robot so that it increases as much as possible the probability of finding the intruders or abnormal events inside the surveillance region.

## MATERIALS AND METHODS

**Mechanics:** The wheels-based platform consists of two 12V DC motors with Integrated 800 Count per Cycle Optical Encoder each supply 550oz.inch (40 kg cm) of the torque to the i90's 18 cm wheels, yielding a top speed in excess of  $0.75 \text{ m sec}^{-1}$ . Weighing only 6 kg (13 lb), the system is light, but it can carry an additional payload of 15 kg (33 lb).

**Sensors:** The sensors consist of i90 that offer full Wi-Fi 802.11G wireless, multimedia, sensing and motion capabilities and comes with a wide range of sensor, camera and audio modules, sufficient to serve in any variety of applications. The OkiKoSenPBX1 offers broad expandability as well for projects that may require additional sensors, even specialized modules. Powered by separate RC servo motors, the integrated camera head can pan and tilt independently.

**Architecture:** The OkiKoSenPBX1 originally developed for Humanoid (HR) Robot. Using this approach, high-level control of the robot is maintained by a remote or local PC/server communicating by a secure wireless link. Low-level functionality is managed by an onboard Digital Signal Processor (DSP) while computationally intensive operations are performed off board. The result is a robot that's lighter, draws less power, runs longer and is dramatically less expensive than a fully bundled or self-contained system. Moreover, since primary processing resides in a server, any hardware upgrades to the central unit are shared by all the robots it controls. With its integrated high bandwidth (54 Mps) wireless fidelity (Wi-Fi 802.11G) module, the system can upload all sensor data to a PC or server at rates in excess of 10Hz. Similarly, streaming audio and video (up to 30 fps) either for direct monitoring or for processing by high-level Artificial Intelligent (AI) schemes is a snap. Commands and instructions sent to the robot via the same wireless link also pass at rates exceeding 10Hz, providing real-time control and access.

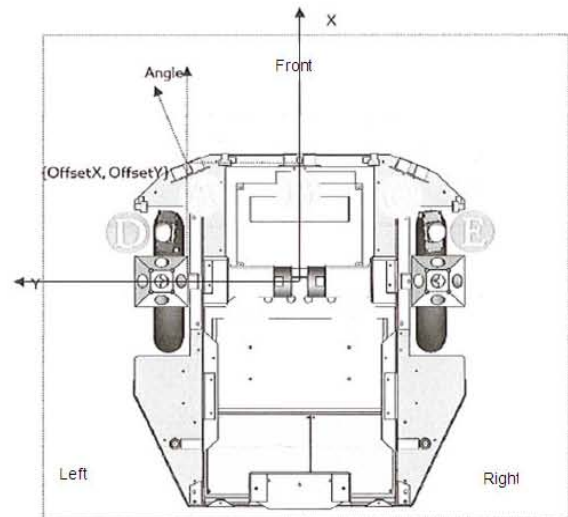


Fig.1: OkiKoSenPBX1 Robot sky view

The OKiKoSenPBX1 includes all WiRobot development software components, enabling easy access to all data and information in a standard Microsoft Windows programming environment. Under the approach of using a separate PC for high-level control, there are no longer mobile system's processing power, memory and storage. For users of non-Windows operating systems OkiKoSenPBX1 can provide the raw communication protocol for direct integration with any other system or device.

**Robot Kinematic:** Robot considered in this study shown in Fig. 1 is a differential motion with two degrees of freedom, composed by two active parallel and independent wheels, a third passive wheel with exclusively equilibrium function and proximity sensors capable of obstacles detection. The active wheels are independently controlled on velocity and sense of turning. The sensors provide short-range distance to obstacles. Additionally the robot is equipped with specific sensors for detection and recognition of search objects or intruders. The robot model presents an interesting compromise between control simplicity and degrees of freedom that allow the robot to accomplish mobility requirements. Its motion is obtained by driving the active wheels. The resultant motion is described in terms of linear velocity  $v(t)$  and direction  $\theta(t)$ , representing on instantaneous linear motion of the medium point of the axis and a rotational motion or rotational velocity  $\dot{\theta}(t)$  of the robot over the same point. The robot motion control is done by providing the wheels velocity  $\omega_{\text{left}}(t)$  and  $\omega_{\text{right}}(t)$ , or equivalently the body linear and angular velocities  $v(t)$  and  $\theta(t)$  called

input or control variables. The mathematical model of this kinematics problem considers these two input variables and also three state variables: The robot position and orientation  $x(t)$ ,  $y(t)$  and  $\theta(t)$ , describe in the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

These equations constitute a nonholonomic dynamical system. The control of this system has been studied extensively by various research groups and diverse solutions are available, for example in<sup>[4-6]</sup>. The motion control strategy adopted in this work involve a state feedback controller proposed in<sup>[7]</sup>, which is an appropriate approach to produce a desired trajectory described by a sequence of coordinates  $x_1(t)$ ,  $y_1(t)$ . This means that the route-planning task is given by a specialized robot module, independent of the motion control module that sets intermediate position lying on the requested path. The adopted control law considers the geometric situation, where the robot is placed at an arbitrary configuration, that is, the position  $x$ ,  $y$  and orientation  $\theta$ , and a desired position is defined by the robot route planner. In the robot reference frame  $X_R Y_R$ , the configuration error vector is defined by  $e = (\rho\phi)^T$ .

Where:

$$\left. \begin{matrix} \rho \\ \phi \end{matrix} \right\} = \text{Targetpositions}$$

The target positions provide the following coordinate change:

$$\begin{aligned} \rho &= \sqrt{\Delta x^2 + \Delta y^2} \\ \phi &= 180 + \theta - \eta \end{aligned} \quad (2)$$

We defined the angle  $\phi$  between the  $X_R$  axis of the body reference frame and the vector connecting the robot center and the desired position. The other configuration variables  $\rho$  and  $\eta$  represent respectively the distance between present desired positions and the angle between the direction to the target and the axis  $X_0$ .

**Map-based route planning:** The planning procedure is based on determining a sequence of intermediary goal points or coordinates  $x$  and  $y$  that will compose the

robot trajectory. The route-planning generator module uses an area-preserving map that is considered as a paradigm for area-preserving chaotic systems: This map, also called Taylor-Chirikov map<sup>[8]</sup>. It is a two-dimensional map which results from a periodic impulsive kicking of a rotor. This map was firstly proposed by Brian Taylor and then independently obtained by Chirikov<sup>[9]</sup> to describe the dynamic of magnetic field lines on the kicked rotor. The dynamic effect of this system is expressed mathematically through the map equations, given by:

$$\begin{aligned} \alpha_{n+1} &= \alpha_n + K \sin \beta_n \\ \beta_{n+1} &= \beta_n + \alpha_{n+1} \end{aligned} \quad (3)$$

Where:

- $\alpha$  = Periodic configuration variable or angular position
- $\beta$  = Momentum variable or angular speed

These map variables are both computed mod  $(2\pi)$ . The map parameter  $K$  represents the strength of the nonlinear kick applied in the rotor mechanism. In its phase space and according to the value associated with the parameter  $K$ , it has stable and unstable periodic orbits, Kolmogorov-Arnold-Moser (KAM) surfaces and chaotic regions. Depending on the nonlinear parameter  $K$ , the regions of regular motion and the regions of chaotic motion are complexly interwoven, but the chaotic regions are confined between KAM. As this parameter is increased, the KAM surfaces start to be destroyed, chaotic regions occupy increasingly large areas until, for a specific value of  $K$ , the last KAM is destroyed and the entire region of the phase space appears to be densely covered by a single chaotic orbit. Our route-planning generator module is implemented based on the map builder that presents this dynamics. Let us now show an example of how the map is used in the context of our route-planning generator module. By numerically simulating the map equations, we analyzed the properties of terrain covering considering the basic mission requirements for fast terrain scanning. We defined a terrain with different dimension in a normalized measurement unit. The map simulation begins with an arbitrary initial position and considers the gain value  $K = 17$ . As the map equations simulation continue over and over, we note in fact that, the necessary condition for the complete region to be patrolled is  $K > 16$ . The terrain covering can be judged through a performance index. This index is defined using a terrain division on square unit cells and computing the visited cells percentage after the robot locations planning.



Fig.2: OkiKoSenPBX1 Robot with its 2CCD Cameras.

**Software:** The software is the robot oxygen as we human being relay on oxygen to respire hence life. This means that the software is the robot life, it is what the robot shown in Fig. 2 needs to accomplish its mission and here, we introduce a new strategy that generates an opportunistic and proper crafted trajectory that works as follows. We associate a route-planning generator module with a closed-loop locomotion control module.

At each step, the first one generates a position goal defined by its coordinates in the phase space. This position goal is provided to the second module, which drives the robot from its actual position to the desired one. When the robot arrives at the desired position, the route (path)-planning generator module is used again to give another position goal, which is subsequently provided to the second module. This sequence of action is repeated over and over again. The route-planning generator module is implemented by exploiting the dynamics of an area-preserving map in a chaotic regime. Different to a dissipative chaotic map, in which the chaotic evolution takes place on attractors, the chaotic region of an area-preserving map for specific parameter values extends practically over all of its phase spaces.

For these parameter values, the entire phase space is covered by a single chaotic orbit. It is therefore possible to make an association between the physical regions that we wish to patrol with the phase space defined for the area-preserving map. Thus, the position goals are generated as transformed iterations of an area-preserving map and so present a chaotic dynamics with its remarkable characteristic that makes long-time prediction based on measurements very difficult between the position goals. The robot trajectory is driven by a closed-loop locomotion control module which allows a short trajectory between the points and introduces an element of regularity to the strategy. By properly combing the parameters of both modules can

exploit a multitude of possibilities between a quick patrolling of a region by using a trajectory with “small” level of unpredictability to a “slow” patrolling with a highly unpredictable trajectory. Actually, by properly combining the parameters, we are able to come up with a very efficient, opportunistic and properly crafted trajectory that fits the desired requirements for patrol missions.

**Control architecture:** The software for running the robot for its patrol mission has been developed using WiRobot Software Development Kit (SDK) through PMS5055 robot sensing-motion controller. The WiRobot software development kit is a part of the robot development system. Being a PC Based-Software framework for robotic, the SDK contains the facilities for memory management, system communication and user interface and utilities for audio, video input-output, sensor data acquisition and motion control.

Teaching robot its patrol routine is driving it around the Building (inside or outside) to be patrolled once as it maps mission plan into its memory. This contains destinations, actions, camera position, schedules and access to forbidden areas. This is achieved by an accurate registration of dada which involved many code modules. The program is written in c# with Visual Studio 2008 Express under .Net 3.5 frame work whose some functions used are express as follows:

**SensorMapBuilder:** This function is used to build an environment map based on infrared range sensors and ultrasonic sensors reading information and output the sensor map to P2P Drive Service for collision avoidance. The configuration data class is defined as follows:

```
public class IrConfigData
{
    public double OffsetX = 0;
    public double OffsetY = 0;
    public double Angle = 0;
    public double Weight = 0;
    public int DisTag = 0;
}

public class UsConfigData
{
    public doubleOffsetX = 0;
    public doubleOffsetY = 0;
    public doubleAngle = 0;
    public doubleWeight = 0;
    public intDisTag = 0;
}
```

These classes define the locations of each sensor and if the weight = 0; means this sensor is disable. This program will read IrSensorConfig.xml to a dataset and UsSensor Config.xml to another dataset, then inform load function, call sensorMapBuilder.ConfigSensor (Sensor mapconfig) to send this configure data to sensor map builder service. After Configuration, this service needs the robot position information and sensor distance position to builder sensor map.

**tmrSensorMapBuilder:** This function is used to update these information to the map builder service as follows:

```
private void tmrSensorMapBuilder_Tick(Object sender,
EventArgs e)
{
    robotMapPosition.robotX = robotPosition.robotX;
    robotMapPosition.robotY = robotPosition.robotY;
    robotMapPosition.robotDir = robotPosition.robotDir;
    sensorMapBuilder.UpdateSensorInfo(sensorData,robotMapPosition);
}
```

**Set P2P DriveParameters:** This function is used to set the robot drive parameters as follows:

```
public class P2PDriveParameter
{
    public double Tolerance Step;
    public double SlowGainTurnRatio;
    public double SlowGain;
    public double MinTurnSpeed;
    public double MinForwardSpeed;
    public double MoveMoment;
    public P2PDriveParameter;
}
```

**Set Target Point:** This function is used to set the destination point information and drive-behavior of the robot using the xml path to read the setting dada. The xml file describes the P2PDrive requests are shown in Table 1.

TargetX to TargetY will drive the robot from its current position to target position, on arrival; it will turn to direction at 90° with 5° tolerance where the robot will stay at this point for 50 sec (StopTime = 50 and Non Stop = false) before pursuing its mission. The maximum forward speed is 0.2 m sec<sup>-1</sup>. The robot must arrive this target within the given tolerance 0.05 m (Forgetable = false, Tolerance = 0.05). The given task time is 3 min (TargetTime = 300).

Table 1: Point configuration

<TargetX >0.8 </TargetX>
<TargetY >0.6</Target Y>
<TargetDir >90.0</TargetDir>
<StopTime> 50</StopTime>
<ForwardSpeed>0.2</ForwardSpeed>
<Forgetable>>false</Forgetable>
<NonStop>>false</NonStop>
<FinalPosture>>true</FinalPosture>
<TargetTime>300</TargetTime>
<TargetTolerance<0.05</TargetTolerance>
<MaxTumSpeed<35</MaxTumSpeed
<CAEnable<false</CAEnable>
<TargetTolerance<0.05</TargetTolerance>
<ReverseDrive>>false</reverseDrive>

The maximum turn speed is 35° sec<sup>-1</sup>. The collision avoidance feature is enabled. The Robot drives toward this target with forward motion with Reverse Drive = false. But depending on the patrol situation or condition the reverse is allowed. The decision to make or not a reverse should be decided by the robot its self (robot self decision making).

## RESULTS AND DISCUSSION

The setup for testing performance verification of the proposed system shown in Fig. 3 consists of:

- Two computers: the remote monitoring computer or client computer where the client program is installed and the central security control computer or server computer where the server program is installed.
- Wireless Fidelity (Wi-Fi 802.11G) router and broadband internet modem.
- A joystick for direct control.

Both monitoring computer and central security control computer operating on a windows XP platform are connected to the wireless Wi-Fi router which in turn is connected to the broadband internet. The central security control computer IP and the port number for connecting to the robot have been assigned as follows: IP Address: 192.168.0.104, Subnet mask: 255.255.255.0, Gateway: 192.168.0.200 and port number: 10001 respectively. The robot has three operating modes: the Manual, Auto and Idle mode.

- In Manual mode, the robot relies on instructions from a remote user to navigate or controlled by mouse by first clicking the show control button in the control panel button on the left side of the program window. This will prompt a navigation control panel. Alternatively the joystick shown in Fig. 3 can be used to control the robot directly

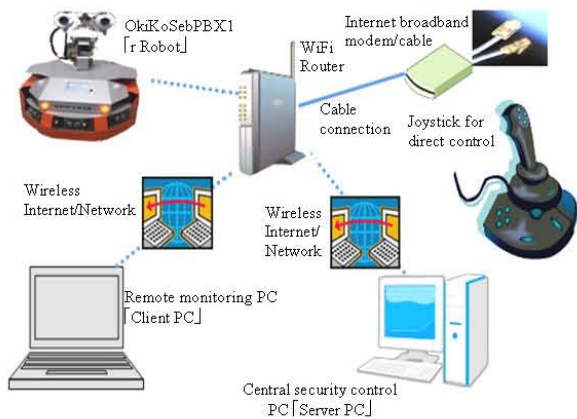


Fig. 3: Robot testing performance setup

- In Auto mode the robot will wander its environment, using its build-in collision avoidance system to aid it from hitting any objects
- In Idle mode the robot remains powered-on in a standby setting while minimizing power consumption

To test our security robot patrolling approach, we have simulated the robot kinematic motion by starting with low speed, then automatically and gradually the robot adjusted its speed to reach the desired speed for the patrol mission. Once the desired speed is reached it becomes constant throughout the patrol mission where it follows successfully the planned route and finished the first tour of the patrolling mission within the allocated patrol time set to 3 min (for the test purpose). The test has been repeated three times of which the robot did accomplish its mission successfully without colliding with any objects.

Next, to test the robot collision avoidance capability, two set of experiments have been done. The first test scenario consists of putting three chairs in the robot path then starts the robot from the remote monitoring computer. The robot has managed to get through the chairs and accomplished its patrol mission successfully again in the 3 min as expected. The second test consists of putting three chairs in the robot path as in the first test scenario and one big piece of wood just where the robot is expected to get out once avoiding these three chairs. The robot managed and gets through the three chairs and once arrived in front of the wood it made a backward and changed direction without leaving its planned mission route and then accomplished successfully its sixth mission with two seconds late. It was expected to finish its patrol mission in 3 min but instead of 3 min the robot arrived at the

destination in 3 min and two seconds which is due probably to the backward. During the patrol mission through its two CCD cameras the robot sent a real-time video to the central security control computer, where a student like-guard took control of the camera pan and tilt-zoom functions remotely.

The system's pan-tilt-zoom video camera allows the security guard to control view, angle, zoom and distance with more flexibility than common fixed security camera allow. For archival purposes, continuous time-lapse video is recorded to a hard drive on board the robot as well as to a hard drive of the central security control computer. The robot is capable of continuously capturing and storing images from one to thirty frames per second on the central security control PC for a period of up to seven days. In this study, unlike battle field or unknown region exploration applications, the operating environment is known a priori, is normally under friendly control and can to some degree be tailored to support robot operation. Some remarkable works have been described in<sup>[10,11]</sup>. In<sup>[10]</sup> for example, the author proposed two methods for controlling an intelligent robot system over a wireless network. The researcher first method deals with the robot control without network while the second method uses Zigbee network.

The results of both methods are somehow good, but compare to our method, the author first method that uses the proportional integrated controller where the parameter  $K_p$  that has been chosen arbitrary in order to get the closed-loop locomotion can have some difficulty to really drive the robot in some region. In<sup>[11]</sup> the modular mechatronic design approach has been used as a control method of the domestic robot, the proposed method as well results obtained are very good, but compare to our method this work used low level control commands to instruct the robot to perform a task. Hence the probability of missing an intruder is high.

## CONCLUSION

In this study, a map-based route planning for security patrol robot is presented. The approach of the proposed system consists of numerically simulate the map equations and then analyzed the properties of the terrain covering considering the basic mission requirements for fast terrain patrolling. The system was verified in a real world experiment inside our College Campus Building. The performance validation test proves that the suggested system is able to accomplish its patrol mission successfully within the time required. The robot capability of avoiding any objects or obstacles

has been also proven as the robot is able of making self-decision once faces an obstacle in front of it.

A part from the description of this system for autonomous patrol mission the particular contribution of this study is the introduction of a novel method that associates a route-planning generator module with a closed-loop locomotion control module. Our future work will include a more thorough evaluation of the capability of the suggested system outside the Campus and during the rain to check if the software is to be modified. Next, we will develop the human detection presence application, where the application should compare the difference of two samples: detect the change from "absence" to "presence" and compare the sample data to a user defined threshold to determine whether to report an alarm or not.

### REFERENCES

1. Brian, R., 2005. Robotics outside the Metals Industries. *Ind. Robot*, 32: 205-208. <http://cat.inist.fr/?aModele=afficheN&cpsidt=16891433>
2. Everett, H.R. and D.W. Cage, 1999. From laboratory to warehouse: Security robots meet the real world. *Int. J. Robot Res.*, 18: 760-768. DOI: 10.1177/02783649922066448
3. Capezio, F., A. Sgorbissa and R. Zaccaria, 2005. GPS-based localization for surveillance UGV in outdoor areas. Proceedings of the 5th IEEE International Workshop on Robot Motion and Control, June 23-25, IEEE Computer Society, Washington DC., USA., pp: 157-162. <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/10414/33076/01554396.pdf?arnumber=1554396>
4. Xu, G. and D. Wang, 2000. Exponential stabilization of extended chained forms. Proceedings of the American Control Conference, June 28-30, IEEE Xplore, Chicago, IL, USA., pp: 3259-3263. DOI: 10.1109/ACC.2000.879167
5. Krotkov, E. and J. Blitch, 1999. The Defense Advanced Research Projects Agency (DARPA) tactical mobile robotics program. *Int. J. Robot. Res.*, 18: 769-776. DOI: 10.1177/02783649922066457
6. Canudas De Witt, C. and O.J. Sørдалen, 1992. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Trans. Automat. Control*, 37: 1791-1797. <http://cat.inist.fr/?aModele=afficheN&cpsidt=4411563>
7. Lee, S.O., Y.J. Cho, M. Hwang-Bo, B.J. You and S.R. Oh, 2000. A stable target-tracking control for unicycle mobile robots. Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems, Oct. 31-Nov. 5, IEEE Xplore, Takamatsu, Japan, pp: 1822-1827. DOI: 10.1109/IROS.2000.895236
8. Meiss, J.D., 1992. Symplectic maps, variational principles and transport. *Rev. Modern Phys.*, 64: 795-848. DOI: 10.1103/RevModPhys.64.795
9. Chirikov, B.V., 1979. A universal instability of many-dimensional oscillator systems. *Phys. Rep.*, 54: 263-379. DOI: 10.1016/0370-1573(79)90023-1
10. Mechraoui, A., J.M. Thiriet and S. Gentil, 2008. Control of an intelligent robot system over a wireless network. Proceedings of the 23rd IAR Workshop on Advanced Control and Diagnosis, Dec. 1-1, Coventry: Royaume, pp: 1-6. [http://hal.archives-ouvertes.fr/docs/00/34/33/84/PDF/Mechraoui\\_AC D08.pdf](http://hal.archives-ouvertes.fr/docs/00/34/33/84/PDF/Mechraoui_AC D08.pdf)
11. Potgieter, J., G. Brighth, O. Diegel and S. Tlale, 2002. Internet control of a domestic robot using a wireless LAN. Proceedings of 2002 Australian Conference on Robotics and Automation, Nov. 27-29, Auckland, pp: 1-4. <http://www.araa.asn.au/acra/acra2002/Papers/Potgieter-Bright-Diegel-Tlale.pdf>