International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
ISSN 2319 - 4847
Special Issue for National Conference On Recent Advances in Technology and Management for Integrated Growth 2013 (RATMIG 2013)

BOMB DETECTION AND DEFUSION IN PLANES BY APPLICATION OF ROBOTICS

Prashant Limje[1] Shailesh N. Khekale[2]

Department of Mechanical Engineering
GNIEM,Nagpur
[1]*prashant.limje02@gmail.com*
[2]*shaileshkhekale@gmail.com*

ABSTRACT

Terrorism is a very actual problem. Although terrorist attacks came in succession from the early seventies and eighties, recent events have raised the technological scale and the dimension of the actions. High level of security is required and bomb-disposal experts are frequently exposed to risks in mission. The effects on the society are blameworthy. Media like television and radio tend to scare people; travels and flights become less attractive. This trend risks to flow into cultural and religious forms of racism. Entertainment and leisure are means that can help to fight against this fear. Earlier we are using human power to defuse the bomb; this paper describes the development of a videogame that simulates the rescue of a potential bomb from an airplane by means of a mobile robot. The user can control real-time the robot, using the keyboard. The device, actuated by two crawlers, carries a 7 DoF arm and moves inside a virtual airplane. The videogame, based on Open Dynamics Engine (ODE), supports dynamics and collision detection. The graphic engine Open Scene Graph (OSG) enhances the reality of the game. The system has been also designed for training of bomb-disposal experts that will maneuver explosive ordnance disposal robots.

1 Introduction

A simulation game is composed of three main elements: scenery, one ore more characters and some rules. The characters can be real or fictional. Users control, for example, airplanes, helicopters or robots and try to accomplish missions. These games allow to analyze the behavior of certain characters in a constrained environment. Videogames can have different levels of abstraction. A simulator may reduce the complexity of a real system or extrapolate only some main aspects, highlighting some details not completely clear by the trend or data analysis. Usually, for detailed/realistic simulations, the user needs a 2D/3D mouse in combination with several keys combinations. Often many players can interact in the same environment; some examples are surgery theatre simulation, racing or role-playing games. Simulators can be used for amusement, training or to predict likely outputs from a real system. A remotely controlled robot is able to enhance the safety of the artificers and, at the same time, to execute successfully the task. Since water cannons and micro charges, typical equipment of robots that operate outdoor, cannot be used inside an airplane, any other technique that allows to identify potential explosive devices is extremely valuable: cameras, X rays, etc. The robots existing on the market can be classified depending on "what" they are able to handle (mass and size of the object) and "where" they should operate (indoor, outdoor, structured environment). The mass and the size of the object are crucial for the correct design of the arm and of the grasping device.
The classification is as follows:-

I. Commercial EOD (explosive ordnance disposal) robots:

Commercial explosive ordnance disposal robots derived from military robots cannot operate in narrow spaces such as planes and busses. Ideally complete equipment for EOD tasks is composed of: - a big size robot, for open space missions, equipped with a water cannon
- A small size robot to work in restricted spaces
- A miniature robot for the inspection of vehicles.

Commercially available EOD robots are derived from devices used for the treatment of dangerous substances or the achievement of military tasks. Robots especially suitable for EOD have been designed only later. These robots can be classified by size. Small size robots, about 10 kg, can easily accomplish inspection tasks. Bigger size robots, up to 300 kg or more, can work outdoor, can lift heavy objects, but due to their size, they have a low dexterity. Middle size robots show intermediate performances. The robots can have wheels or crawlers. Crawlers, despite their limited speed, offer several advantages compared to wheels: high torque, capability to avoid obstacles, zero curvature ray and low center of gravity. Vanguard, from *EOD Performance*, is the only robot commercially available equipped with a really dexterous arm and that

could be employed, in principle, in an airplane; the other EOD robots, suffer lack of arm dexterity, and generally have a cart too cumbersome.

II.The PMAR-EOD Robot

The mission environment, for EOD robots operating in airplanes, includes narrow corridors between the banks of seats and along the plane, and places difficult to reach, like the inside of the baggage rooms above the seats or the service cabinets. The EOD robot designed at the PMARlab of the University of Geneva, named PMAR-EOD (Fig. 1), is a mobile system tailored for this complex environment. It is composed by a crawled base and an arm. The 7R arm is agile and light. The design process of the PMAR-EOD availed of fully implemented digital mock-ups and extended virtual test campaigns.

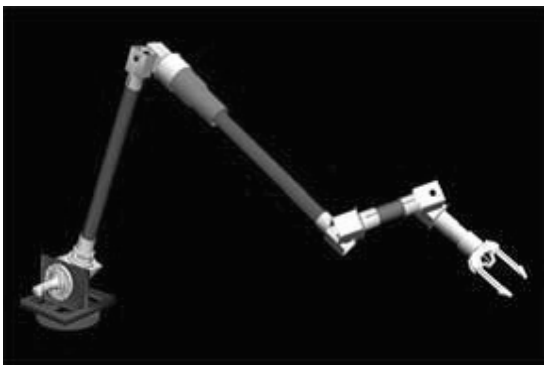

Figure 1. **The PMAR-EOD robot**

A 3D mock-up of an airplane was prepared to simulate the robot in action (Fig. 2).

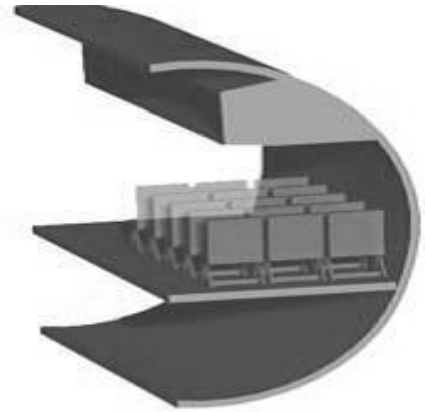

Figure 2. **Cockpit model**

The robot is able to carry a payload up to 3 kg. The arm is sized to reach all "sensible" places inside of an airplane. Its frame is made of aluminum and composite hollow pipes that envelope the cables, the actuators and the gears. The overall weight is low compared to the payload. The clamp has a compact design and its architecture and geometry are selected to grasp both big and small objects, from 4 mm to 100 mm diameter. The fingers use four-bar mechanisms assuring the parallelism of the tips and a linear force/displacement characteristic.

III. The videogame

For designing the videogame the simulation tool is required. The appropriate tool should be selected to get the better performance system. Out of the available tools virtual simulation tool is found to be good for performance.

Choice of the virtual simulation tool

For any system and machine, virtual simulation is much cheaper than any physical, partial or complete, prototype and the assessment of the system performances is faster. Code libraries for the definition of environments for virtual simulation are an effective mean for custom made implementation of specific simulation software. The result is remarkably different from traditional CAD results, since kinematics/dynamics analyses, motion planning and collision check are enabled in the same environment. Several virtual simulation libraries and packages exist, some oriented to robotics, other general-purpose. Most of

these code libraries, packages and environments are freeware and simply downloadable from the web. A rough subdivision is: freeware/open-source packages, university packages (e.g., developed in PhD theses or such and owned by a research team) and commercial software. After a comparison of the candidate simulation packages, it has been chosen to adopt the C library ODE.

ODE – Open Dynamics Engine

ODE is an open-source C/C++ library, implemented by Russel Smith. It can be used for real-time, multi-body, dynamic simulation of rigid bodies, with high level interaction facilities, such as motion commands and monitoring. The library supports a satisfactory set of joints and contact fixtures, also with friction. Hard contact conditions are implement and basic collision detection should be provided for elementary geometries such as cylinders, spheres, capsules, planes, lines and triangular meshes. For performing integration user can refer first order (Euler) algorithm that is not accurate but intrinsically stable. The main features of ODE are: - rigid bodies with arbitrary distributed mass properties; - spherical, revolute, prismatic, universal joints, weld connections and linear and rotational actuators; - Trinkle-Stewart and Anitescu-Potra; - LCP. Dantzig solver for the modeling of contacts with friction and a faster Coulomb friction modeling; - C++ interface on the native C interface; - OpenGL graphic library for real time representation of the simulation results; - run in single and double precision floats.

2. The simulator implementation

Each object is represented in the virtual environment as three separate virtual entities: a body, representing the mass properties, used for the dynamic simulation; a geometry used for the collisions detection; a rendered geometry to provide a virtual-reality representation of the simulation. Each of these three entities has its own location and orientation coordinates, which usually coincide. In this way, the three simulation layers are kept independent each other and the implementation of the model is simplified. Note that the simulation can be performed without being associated to any graphical virtual-reality representation, which is useful just for a faster understanding of the results.

Approaches needed for the simulator:

The virtual simulation of the PMAR-EOD robot is performed in double precision. The simulator needs following approaches for implementation:

1. Kinematics model of the robotic arm
2. Simple dynamic model of the robotic arm
3. Introduction of collision detection of the arm links each other
4. Accurate dynamic model of the arm
5. Introduction of the environment and related collision detection
6. Setup of the control system
7. Graphic engine for the rendering
8. Tuning and tests on the complete simulator. These levels are briefly described in the following sections.

Arm kinematics model

Featherstone's algorithm is fast and can be used for chain-like structures, but problems are encountered when there are contacts or if the chain needs to be broken to add another object. Mirtich type methods, where one constraint at a time is satisfied, are difficult to make stable, and objects suffer from jitter. The arm kinematics model (Fig. 3) is solved with the *RRG Kinematix* library, implemented by the Robotics Research Group of the University of Texas at Austin. The geometry is described by the Denavit-Hartenberg parameters, measured on the Pro/ENGINEER virtual mock-up of the robot. The joint zeros chosen (arm fully deployed frontally and parallel to the ground) differ from the Denavit-Hartenberg representation, so a conversion routine is required.

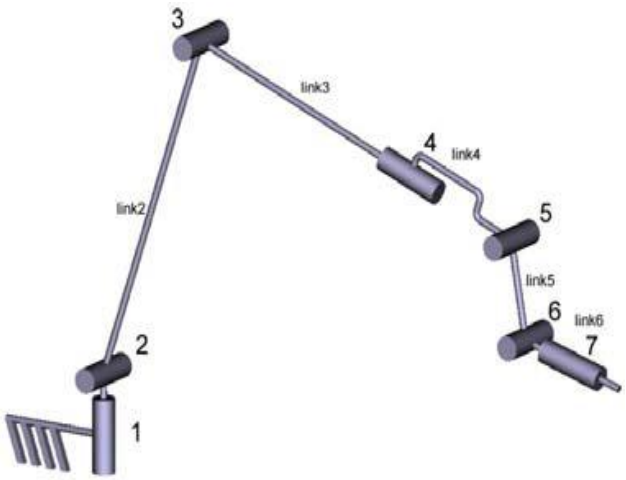

Figure 3. **Kinematics architecture of the Arm**

The simulation steps are:
1. Choice of a feasible set of joint values
2. Creation of a handling to control the robot; the D-H parameters are assigned
3. Run of the forward kinematics; the position of the end-effector is computed
4. Main loop: -
   1.Wait for keyboard commands from the user.
   The current arm configuration is stored and changed according to the keyboard input;
   2. Solution of the inverse kinematics:
      a. If no feasible solution is obtained, the previous saved configuration is kept
      b. If the Measure of Manipulability (MOM) is lower than a specified threshold, then the inverse kinematics is solved by "Singularity- Robust Inverse" (SR-inverse) algorithm. (SR-inverse computed with Matrix TCL Lite library). If some joint overcome the specified angular mobility, the previous saved configuration is kept.
   3. Output of the solution

Simple arm dynamic model

A simple dynamic model of the arm is implemented using ODE functions. The mass properties of the links are approximated. Each link is a constant-density cylinder without reaction force limitation. The inverse kinematics is solved again using RRG Kinematix. Additional joint angle conversions routines are used to link RRG Kinematix to ODE. The pose of the gripper (end-effector) changes accordingly to the keyboard input coming from the user. The inverse kinematics routine provides the input for simple PID loops controlling the joints.

Collisions

Geometry is associated to each link the simulator is provided with a collision detection routine. This routine considers the interaction between non-subsequent links, and between links and the virtual environment.

Accurate dynamic model

The arm and the mobile platform of the robot are modeled with basic solids. The mobile platform moves front and back and steers. The crawlers are replaced by four cylindrical wheels; the platform steers due to different rotation velocities of the left and right wheels. The tangential component of the wheel-ground contact force depends on the orthogonal component (by the friction factor) and on the actuation torque applied to the wheel.

Environment

An accurate modeling of the environment has fundamental importance to get significant results from the simulation. The aircraft interior has been created with Pro/ENGINEER wildfire. The model has been exported to AC3D format and further filtered; each surface has been represented with a triangular mesh. Finally the robot is inserted in the new environment.

Control system

The user can select one of three control schemes developed for the PMAR-EOD robot:
- *Direct dynamics*

- *Inverse dynamic - Global mode*
- *Inverse dynamic - Local mode*

During the simulation, it is possible to switch from one mode to another. The passage from one to any another is always possible during the simulation. PID controllers are used for all joints in all three cases. The control of the mobile platform is independent from the control of the arm, thus the pose of the arm gripper refers to the platform and it is not possible to keep the gripper fixed while the platform moves. The initial control imposed torque thresholds to the motors together with a torque control scheme, but this led to instability. Then the velocity control scheme natively supported by ODE has been adopted. In the case of *Direct dynamics* scheme (Fig. 5), each joint is controlled separately from the others in the joint space. This scheme is used for large arm displacements while reaching the workspace region containing the target object to be disposed, e.g., in the bottom-front of a seat.

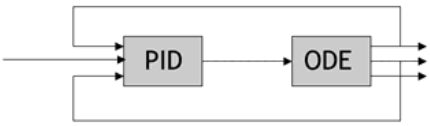

Figure 5. **Direct dynamics**

The Inverse dynamics scheme (Fig. 6) allows the control of the robot in the end-effector space. In *Global mode* the pose of the gripper is controlled with respect to the mobile platform reference frame, while in *Local mode* the gripper is controlled with respect to its own reference frame.

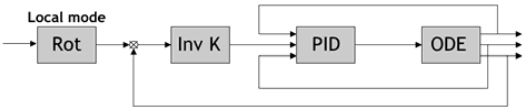

Figure 6. **Inverse dynamics**

The normal robot operation sequence is:
- *Direct dynamics* for fast approach of the workspace region where the target object relies
- *Inverse dynamics – Global mode* to make the
gripper closer to the target
-*Inverse dynamics – Local mode* to reach the target

Need of Graphic engine for rendering

A realistic representation of the robot in the environment, together with an impressing rendering is a useful option to make better advantage of the PMAR-EOD simulator. The graphic engine used is Open Scene Graph (OSG), an open source library for advanced 3D graphics that is normally used for videogames, virtual reality and modeling. It is written in C++ and uses OpenGL. The pose of each body is computed by ODE and transferred to OSG, which place the bodies in the simulation space. The 3D models of every link and component of the system come from Pro/ENGINEER, from which they are converted in AC3D format, directly supported by OSG. During the simulation ODE computes the poses of the bodies and transmits them to OSG. The real-time data visualization is provided by a suitable HUD, switched on/off by the user through the keyboard. The rendering process slows remarkably the simulation, up to 16 frames per second on an Athlon XP 2000+ with 512 MB SDRAM and GeForce 2 Mx 32 MB video card. Although the simulation is not real time, the delay is small and the user can easily perceive and command the robot. With an improved hardware, the simulation could also be performed real time.

User interface

The user can interact with the simulation through the mouse and the keyboard. The former controls the camera, while the latter commands the robot and some parameters of the simulation. A keyboard handler class of OSG manages the keyboard. At each time step, the keyboard handler class checks for a key press.
In *Direct dynamics* each joint has one key associated to it. In *Inverse dynamics* three keys move the gripper along the x, y and z axes of the actual reference frame, while three other keys control its rotation around the same axes. The crawled base is controlled by the arrow keys. They suitably increase/decrease the speeds of the two crawlers. The user can also disable/enable gravity and show/hide the data on the HUD.

3. Conclusions

A dynamic 3D videogame is realized: the paper provides the reference for the mission to find and retrieve a bomb placed inside an airplane. With PMAR EOD as a reference simulator .It helps in finding out how security robot can pick the bomb by

means of a gripper positioned on the tip of its redundant arm. The PMAREOD simulator is suitable either for training of the airport bomb squads either as a core of a videogame. The game is inspired by some recent events: terrorist attack, bombs, sabotages etc. with the aim of making people rationally conscious instead of unconscious or irrationally sensible. Both the left side and the right side of the cockpit are simulated. Crucial data like maximum angular range of each joint and maximum torque were computed with available algorithms .The simulation speed is limited only by the CPU performance. For this reason, more powerful computers will allow to execute real time simulators of complex systems with arbitrary temporal steps and at the same time high stability and accuracy. Moreover, some simplifications would fast the simulation speed in order to make the software compatible with less powerful hardware. The modular and object oriented structure of the simulator allows to upgrade and modify the code; new features can be implemented with relatively low effort. Moreover ODE is continuously improved and upgraded; in the future new program features will allow to enhance the accuracy, stability and speed of the simulations. Similarly OSG is continuously upgraded in order to offer always more realistic simulations.

References

[1] Anitescu M., Cremer J., Potra F.A., 1995, Formulating 3D Contact Dynamics Problems, Mechanics of Structures and Machines,
22/4: 405-437
[2] Anitescu M., Potra F.A., 1997, Formulating dynamic multirigid- body problems with friction as solvable linear complementarity
problems, Reports on Computational Mathematics, 93
[3] Anitescu M., Potra F.A., Stewart D.E., 1998, Time-Stepping for Three-Dimensional Rigid Body Dynamics, Comp. Methods Appl. Mech. Eng., 177/3-4: 183-197 [4] Anitescu M., Potra F.A., 2001, A Time-Stepping Method for Stiff Multibody Dynamics with Contact and Friction, Argonne National Laboratory, Argonne, IL
[5] Baraff D., 1989, Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies, Computer Graphics, 23/3: 223-232
[6] Baraff D., 1990, Curved Surfaces and Coherence for Non- Penetrating Rigid Body Simulation, Computer Graphics, 24/4: 19-28
[7] Baraff D., 1990, Coping with Friction for Non-Penetrating Rigid Body Simulation, Computer Graphics, 24/4
[8] Baraff D., 1992, Dynamic Simulation of Non-Penetrating Rigid Bodies, PhD Thesis
[9] Baraff D., 1993, Issues in Computing Contact Forces for Non- Penetrating Rigid Bodies, Algorithmica, 10: 292-352
[10] Baraff D., 1994, Fast Contact Force Computation for Non- Penetrating Rigid Bodies, Computer Graphics Proceedings, Annual Conference Series, Orlando: 23-34
[11] Bowen I.G., Fletcher E.R., Richmond D.R., 1968, Estimate of Man's Tolerance to Direct Effects of Air Blast, Technical Progress Report, DASA-2113, Defense Atomic Support Agency, Department of Defense, Washington, D.C.
[12] V. Ramya, B. Palaniappan, and Subash Prasad, "Embedded Controller for Radar based Robotic Security Monitoring and Alerting System", International Journal of Computer Applications (IJCA), Volume 47-No. 23, June 2012.

AUTHOR

PRASHANT C. LIMJE received the Bachelor Degree in Mechanical Engineering from Umrer College of Engineering in 2012. During 2012-2013, he worked as a lecturer in Kamptee Polytechnic. He is now in Gurunank Institute of Engineering and Management as an Assistant Professor.

SHAILESH N. KHEKALE is pursuing Ph.D. in Mechanical Engineering, received M. Tech Degree from R.K.N.E.C. in 2010 and working as Asst. Prof. in Guru Nanak Institute of Engineering and Management.