Simulation of Robotic Manipulators

Leon Žlajpah

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
leon.zlajpah@ijs.si

Abstract: *Simulation is a powerful visualization, planning, and strategic tool in different areas of research and development. The simulation has also a very important role in robotics. In the paper we focus on different tools for the dynamic simulation of robot manipulators, especially on those based on MATLAB/Simulink or using general dynamic engines and graphic languages. We present and compare some tools for the simulation of robot manipulators and their visualization.*

Keywords: *Robotic manipulators, Simulation, Visualization, Dynamic engines*

I INTRODUCTION

Simulation has been recognized as an important tool since the beginning of the 20th century and robotics as a modern technological branch is no exception. Actually, in robotics simulation plays a very important role, perhaps more important than in many other fields. We present the simulation of dynamics of robotic manipulators, and compare some tools for the simulation of robot manipulators in MATLAB/Simulink. Finally and we give an overview of simulation and visualization tools suitable for the simulation of robot systems using general dynamic engines and graphic languages.

II SIMULATION TOOLS

Being able to simulate opens a wide range of options for solving many problems creatively. You can investigate, design, visualize, and test an object or even if it does not exists. A large amount of simulation software is available for robot systems, and it is already being used extensively. Simulation allows us to study the structure, characteristics and the function of a robot system at different levels of details each posing different requirements for the simulation tools. As the complexity of the system under investigation increases the role of the simulation becomes more and more important. Advanced robotic systems are quite complex systems. Hence, the simulation tools can certainly enhance the design, development, and even the operation of robotic systems. Augmenting the simulation with visualization tools and interfaces, one can simulate the operation of the robotic systems in a very realistic way. Depending on the particular application different structural attributes and functional parameters have to be modelled. Therefore, a variety of simulation tools has been developed for the robotic systems which are used in mechanical design of robotic manipulators, design of control systems, off-line programming systems, to design and test the robot cells, etc. The majority of the robot simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems they all include kinematic or dynamic models of robot manipulators.

The simulation tools for robotic systems can be divided into two major groups: tools based on general simulation systems and special tools for robot systems.

Tools based on general simulation systems are usually special modules, libraries or user interfaces which simplify the building of robot systems and environments within these general simulation systems. One of the advantages of such integrated toolboxes is that they enable you to use other tools available in the simulation system to perform different tasks. For example, to design control system, to analyse simulation results, to visualize results, etc. The most popular general simulation tools which are used also for simulation of robot systems are MATLAB/Simulink, Dymola/Modelica, 20-sim, Mathematica, etc.

Special simulation tools for robots cover one or more tasks in robotics like off-line programming, design of robot work cells, kinematic and dynamic analysis, mechanical design. They can be specialized for special types of robots like mobile robots, underwater robots, parallel mechanisms, or they are assigned to predefined robot family.

2.1 MATLAB-based Tools

MATLAB is definitely one of the most used platforms for the modelling and simulation of various systems and it is not surprising that it has been used intensively for the simulation of robotics systems. Among others the main reason for that are its capabilities of solving problems with matrix formulations and easy extensibility. As an extension to MATLAB, SIMULINK adds many features for easier simulation of dynamic systems, e.q. graphical model and the possibility to simulate in real-time. Among special toolboxes that have been developed for MATLAB we have selected for comparison the following four:

(a) **Planar Manipulators Toolbox** (PMT) [1] is intended for the simulation of planar manipulators with revolute joints and is based on Lagrangian formulation. Planar Manipulators Toolbox can be used to study kinematics and dynamics, to design control algorithms, for trajecory planning. It enables also real time simulation. Due to its concept it is very good tool for education. To gain the transparency, special blocks have been developed to calculate the kinematic and dynamic models. These blocks are then used to build the desired model. Figure 1 shows the dynamic model where an external force acts on the end-effector. The block dymodall which calculates the system vectors and matrices x, J, $\dot{J}$, H, h and g and then joint accelerations are calculated using Lagrangian equation.

(b) **SD/FAST** [2] can be used to perform analysis and design studies on any mechanical system which can be modelled as a set of rigid bodies inter-

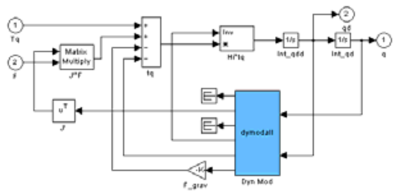

Figure 1
Dynamic model (PMT)

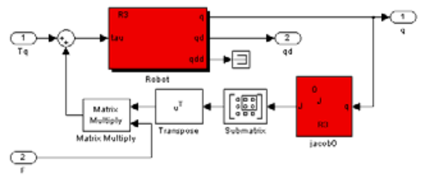

Figure 2
Dynamic model (RT)

connected by joints, influenced by forces, driven by prescribed motions, and restricted by constraints. Here the dynamic model is calculated using SD/FAST library which is based on the advanced Kane's formulation. (Order(n) formulation). Then using the SD/FAST compiler the dynamic model is generated. The simulation of the whole system is based on PMT. The only difference is that a special S-function is used to interface SD/FAST procedures and Simulink.

(c) **The Robotics Toolbox** (RT) [3] provides many functions that are required in robotics and addresses areas such as kinematics, dynamics, and trajectory generation. RT is useful for simulation as well as for analysing the results from experiments with real robots, and can be a powerful tool for education. RT is based on a general method of representing the kinematics and dynamics of serial-link manipulators by description matrices. The inverse dynamics is calculated using the recursive Newton-Euler formulation. Although it was initially meant to be used with MATLAB, it can be used also with Simulink. Figure 2 shows the block scheme of the dynamic model using Robotics Toolbox.

(d) SimMechanics (SM) [4] extends Simulink with the tools for modelling and simulating mechanical systems. With SimMechanics, you can model and simulate mechanical systems with a suite of tools to specify bodies and their mass properties, their possible motions, kinematic constraints, and coordinate systems, and to initiate and measure body motions. To get a dynamic model of a robot manipulator we have first to build the link model, i.e. to connect link masses with joints as it is shown on Figure 3. All link models are then connected together to the complete model.

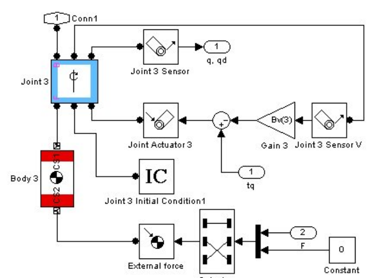

Figure 3
Model of one link (SM)

2.2 Other General Simulation Tools

Similar as in MATLAB the robot system can be simulated in Dymola and Modelica, or 20-sim. Here, the Multi-Body libraries provide 3-dimensional mechanical components to model rigid multibody systems, such as robots. The robot system is built by connecting blocks representing parts of the robot like link bodies, joints, actuators, etc. Figure 4 shows the simulation of KUKA robot in Modelica [5] and Figure 5 shows the simulation of a parallel robot manipulator with 20-sim [6].

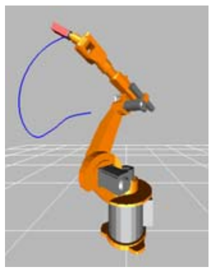

Figure 4
Simulation of a robot with Modelica

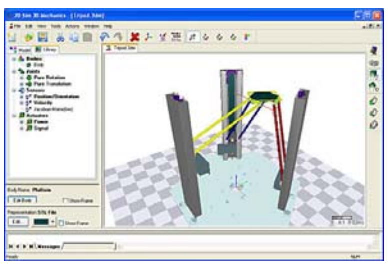

Figure 5
Simulation of a Tripod with 20-sim 3D Mechanics Toolbox

2.3 Multibody Dynamic Enginess

In the last years new simulation tools have been available based on general engines for the simulation of physics environments [8, 9, 2]. These engines provide libraries for simulating the multi-body dynamics, i.e. the physics of motion of an assembly of constrained or restrained bodies. As such they encompass the behaviour of nearly every object and among them are, of course, also robot manipulators. These dynamic engines have beside the dynamics simulation engine also a collision detection engine. The collision engine is given information about the shape of each body and then it figures out which bodies touch each other and passes the resulting contact point information to the user. The user can then take the proper actions. Building the model of a robot is straightforward. First you have to create all bodies and connect them if desired with proper joints. For example, the 3DOF model can be defined using Open Dynamic Engine (ODE) as shown in Figure 6.

```
// create world
contactgroup.create (0);
world.setGravity (9.81,0,0);
dWorldSetCFM (world.id(),1e-5);
dPlane plane(space,0,0,1,0);
// fixed robot base
xbody[0].create(world);
xbody[0].setPosition(0,0,SIDE/2);
box[0].create(space,SIDE,SIDE,SIDE);
box[0].setBody(xbody[0]);
bjoint = dJointCreateFixed (world,0);
dJointAttach (bjoint,xbody[0],0);
dJointSetFixed (bjoint);
// robot links
for (i=1; i<=NUM; i++) {
xbody[i].create(world);
xbody[i].setPosition(0,(i-0.5)*LENG,(i+0.5)*SIDE);
m.setBox(1,SIDE,LENG,SIDE);
m.adjust(MASS);
xbody[i].setMass(&m);
box[i].create(space,SIDE,LENG,SIDE);
box[i].setBody(xbody[i]);
}
// robot joints
for (i=0; i<NUM; i++) {
joint[i].create(world);
joint[i].attach(xbody[i],xbody[i+1]);
joint[i].setAnchor(0,(i)*LENG,(i+1)*SIDE);
joint[i].setAxis(0,0,1);
}
```

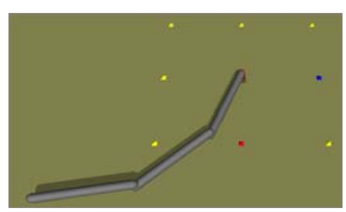

Figure 6
3R planar manipulator simulated with ODE

Unfortunately, most of dynamics engines do not support functionality necessary to include robot models in the control algorithms. Advanced control algorithms including robot models include Jacobian matrices, inertia matrices, gravity forces, etc., and they are not explicitly defined. The user can use some implicit algorithms or other tools to get this parameters. The dynamic simulation of multibody systems becomes very important when introducing robotics into human environments [10, 11, 12], where the success will not depend only on the capabilities of the real robots but also on the simulation of such systems. For example, in applications like virtual prototyping, teleoperation, training, collaborative work, and games, physical models are simulated and interacted with both human users and robots.

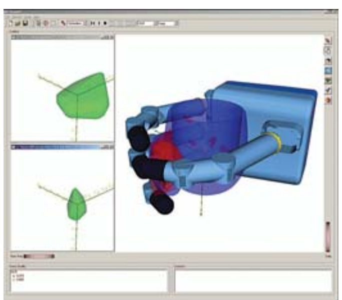

Figure 7
A force-closure grasp of the mug using the DLR hand

For example, the dynamics engine within a robotic grasping simulator known as GraspIt! [13] computes the motions of a group of connected robot elements, such as an arm and a hand, under the influence of controlled motor forces, joint constraint forces, contact forces and external forces. This allows users to dynamically simulate an entire grasping task, as well as test custom robot control algorithms. Fig. 12 shows how a robot hand can grasp a mug. In this example, all contacts between the fingers and the mug and related forces are analysed.

III COMPARISON

For comparison reasons we have simulated only the dynamics, without any task controller. In all four cases it has been very easy to build the robot system. One of the differences is that special toolboxes for robots have predefined more specific functions and blocks as the general toolboxes and the other is the execution time. In Fig. 5 we give the execution time for the dynamic model for all four approaches. First we can see that SD/FAST is significantly faster than other and is increasing more slowly versus the degrees-of freedom than other. Next, Planar Manipulators Toolbox is fast for small number of degrees-of-freedom and the execution time increases fast with the number of degrees-of-freedom. The Robotics Toolbox is relatively fast as long as we use only the inverse dynamics. Otherwise, e.g. for the calculation of the Jacobian matrix, it is significantly slower, because the calculation is based on M-functions.

Conclusion

We have presented that the simulation is widely used in robotics Actually, advanced robot systems require sophisticated simulation tools which can model accurately enough the physical world at sufficient speed and allow user interaction. New challenges in the simulation of robotic systems are multi-body dynamics that computes robot and object motions under the influence of external forces, fast collision detection and contact determination, realistic visualization of the robot and environment, and haptic interaction. Advanced simulation tools are the foundation for the design of sophisticated robot systems, for the application of robots in complex environments and for the development of new control strategies and algorithms. The simulation being once a tool for the analysis of a robot system and task planning has become an open platform for developing new robot systems. In the end, I believe simulation in robotics has reached a very important role and by using different simulation software, the current and future capabilities of complex robotic systems can be significantly improved.

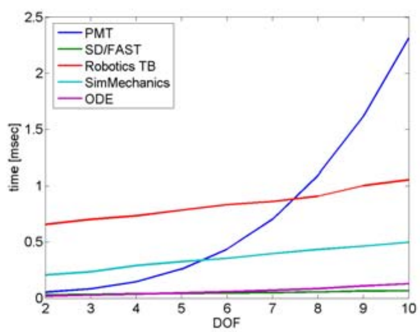

Figure 8
Calculation time of a dynamic model of 3R planar robot manipulator

References

[1] L. Žlajpah: Integrated environment for modelling, simulation and control design for robotic manipulators. *J. of Intelligent. and Robotic Systems*, 32(2):219-234, 2001

[2] Symblic Dynamics, Inc. *SD/FAST User's Manual*, 1994

[3] P. I. Corke: A Robotics Toolbox for MATLAB. *IEEE Robotics & Automation Magazine*, 3(1):24-32, 1996

[4] The Mathworks. *SimMechanics, User's Guide*, 2005

[5] Kazi, G. Merk: Experience with RealSim for Robot Applications. Technical report, KUKA Roboter GmbH, 2002

[6] 20-sim, Controllab Products, http://www.20sim.com/

[7] J. F. Nethery, M. W. Spong: Robotica: a Mathematica package for robot analysis. *IEEE Robotics & Automation Magazine,* 1(1), 1994

[8] Open Dynamics Engine, http://ode.org/ode.html

[9] Newton Game Dynamics, http://physicsengine.com/

[10] O. Khatib, O. Brock, K.-S. Chang, F. Conti, D. Ruspini: Human-Centered Robotics and Interactive Haptic Simulation. *Int. Journal of Robotic Research*, 23(2):167-178, 2002

[11] A. T. Miller, H. I. Christensen: Implementation of multi-rigid-body dynamics within a robotic grasping simulator. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2262-2268, Taipei, Taiwan, 2003

[12] J. Go, B. Browning, M. Veloso: Accurate and flexible simulation for dynamic, vision-centric robots. In *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*, 2004

[13] A. Miller and P. K. Allen: Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110-122, Dec 2004