Development of a Dynamics Model for the Baxter Robot

Tomoya Kashifuku[1], Wenxu Wei[1]

[1] Ostfalia Hochschule für angewandte Wissenschaften-Wolfenbüttel, Germany
{t.kashifuku,w.wei}@ostfalia.de

Abstract. The dynamics model of a robot is important to find the relation between the joint actuator torques and the resulting motion. Commonly there are two relevant methods to do this: The Lagrange formulation, which gives a closed form of the dynamics equations. And the Newton-Euler method, which uses a recursive form. Then we organized two experiments which are about Forward Kinematics and Inverse Kinematics. But in fact, we only use the first method in our experiments. In this paper, we use the simulation which simulate the torques using the Lagrange formulation. After that we analyzed the data and the details of the experiments and sum up simply. At the end of the paper we give an overview of further work, which we may realize later.

Keywords: Dynamics, inverse and forward Kinematics, Lagrange formulation, Baxter Robot

1. Introduction

1.1. Dynamics

The dynamics of a robot manipulator describes how the robot moves in response to these actuator forces. For simplicity, we will assume that the actuators do not have dynamics of their own and then we can command arbitrary torques at the joints of the robot. Once the equations of motion for a manipulator are known, the inverse problem can be treated: the control of a robot manipulator entails finding actuator forces which cause the manipulator to move along a given trajectory. If we have a perfect model of the dynamics of the manipulator, we can find the proper joint torques directly from this model. In experiments, we must design a feedback control law which updates the applied forces in response to deviations from the desired trajectory. Care is required in designing a feedback control law to insure that the overall system converges to the desired trajectory in the presence of initial condition errors, sensor noise, and modeling errors. But at first we primarily concentrate on one of the simplest robot control problems, that of regulating the position of the robot.

1.2. Methods for dynamics

We have already mentioned, that dynamic modeling means deriving equations that explicitly describes the relationship between force and motion in a system. To be able to control a real robot manipulator as required by its operation, it is important to consider the dynamic model in design of the control algorithm and simulation of motion. There are many methods for generating the dynamic equations of a mechanical robotic system. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation or analysis. For our own experiment and analysis, we also need to choose some methods to realize our target. Although in fact there is only the Euler-Lagrange formulation applied in our experiments, but generally there are two important approaches available: the Euler-Lagrange formulation and the Newton-Euler formulation.

1. Langrange-Euler method:

The Euler–Lagrange equation, is a second-order partial differential equation whose solutions are the functions for which a given functional is stationary.

Because a differentiable functional is stationary at its local maxima and minima, the Euler–Lagrange equation is useful for solving optimization problems in which, given some functional, one seeks the function minimizing (or maximizing) it. This is analogous to Fermat's theorem in calculus, stating that at any point where a differentiable function attains a local extremum, its derivative is zero. In classical field theory there is an analogous equation to calculate the dynamics of a field.

The essence of this method are dynamic equations in a symbolic/closed form, and is best for study of dynamic properties and analysis of control schemes.

2. Newton-Euler (RN-E) method:

Traditionally the Newton–Euler equations is the grouping together of Euler's two laws of motion for a rigid body into a single equation with 6 components, using column vectors and matrices. These laws relate the motion of the center of gravity of a rigid body with the sum of forces and torques (or synonymously moments) acting on the rigid body. In classical mechanics, the Newton–Euler equations describe the combined translational and rotational dynamics of a rigid body.

The essence of this method are dynamic equations in a numeric/recursive form, which is also called Recursive Newton-Euler method in some situations, and is best for implementation of control schemes (inverse dynamics in real time).

1.3. Forward/Inverse Kinematics

The definition of these two expressions are a little different, the forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters.

But On the other hand, the inverse kinematics refers to the use of the kinematics equations of a robot to determine the joint parameters that provide a desired position of the end-effector. That means, specification of the movement of a robot so that its end-effector achieves a desired task is known as motion planning. Inverse kinematics transforms the motion plan into joint actuator trajectories for the robot.

1.4. Baxter Robot

Baxter is an industrial robot built by Rethink Robotics, a start-up company founded by Rodney Brooks. Baxter Robot is a 3-foot tall, two-armed robot with an animated face. It is used for simple industrial jobs such as loading, unloading, sorting, and handling of materials.

Baxter Robot has sensors surrounding its head that allow it to sense people nearby. Other sensors around its head give the Baxter Research Robot the ability to adapt to its environment, unlike other industrial robots which will either continue to do their one task repeatedly, or will shut down and stop working at the slightest change in their environment.

Baxter runs on the open-source Robot Operating System on a regular, personal computer which is embedded in its chest. Baxter also has extra sensors in its hands that allow it to pay very close attention to detail. There are also many advantages about the Baxter, any regular worker could program Baxter and it only takes a matter of minutes, unlike usual industrial robots that take extensive programs and coding in order to be used. This means Baxter isn't programmed by traditional software engineers writing code. And the Baxter can be easily taught to perform multiple, more complicated tasks.

1.5. ROS

Robot Operating System (ROS) is a collection of software frameworks for robot software development, providing operating system-like functionality on a heterogeneous computer cluster. ROS provides standard operating system services. The primary goal of ROS, as the statement in the definition, is to provide code reuse support for robot research and development. ROS is a distributed process framework, these processes are packaged in easy to be shared and released in the package and feature packages.

Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages. Despite the importance of reactivity and low latency in robot control, ROS, itself, is not a real-time OS (RTOS), though it is possible to integrate ROS with real-time code.

On the other side, there are also several advantages of ROS, which reflect in these parts:

ROS provides a publish-subscribe communication framework for simple and fast building of distributed computing systems.

ROS not only provides a wide range of tools simulation, but also provides a wide range of libraries to achieve the mobility, operation control, various robot function, which is very easy to operate.

1.6. Gazebo

Gazebo is a multi-robot simulator in a 3-dimensional world. It comes with advanced plugin interfaces that can be used to simulate the sensor feedback and plausible interactions between objects. Robots and other objects can be modeled into gazebo using their SDF. Apart from these, there are also some standard models available within gazebo that can be used with custom models. Besides this, Baxter can also be modeled into gazebo using its URDF. Controller plugins like block laser, point laser, and cameras are used to simulate the sonar, IR and the cameras on Baxter. The simulator can be used in conjunction with the rviz and MoveIt! as with the real robot.

2. Experiment

2.1. Steps for the experiment

We planned our experiments according to the following steps, the whole process shows in the Flow chart 1.

First, we get the position of the end-effector varied with time from a function.Then we can use inverse kinematics, in order to calculate the angle vector q which represents the angles of each joint from the positions. Of course they are varied with time as well.As a result, we can calculate the angular velocity v and acceleration vector a as the flow chart shows, in which the parameter

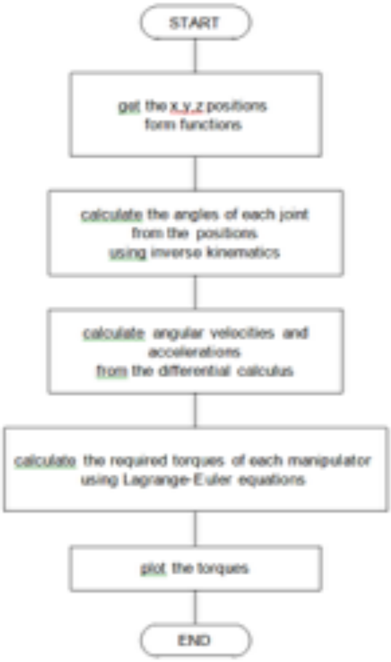
$\Delta t = 1/f$

$dx/dt = x(t+\Delta t)-x(t) / \Delta t$

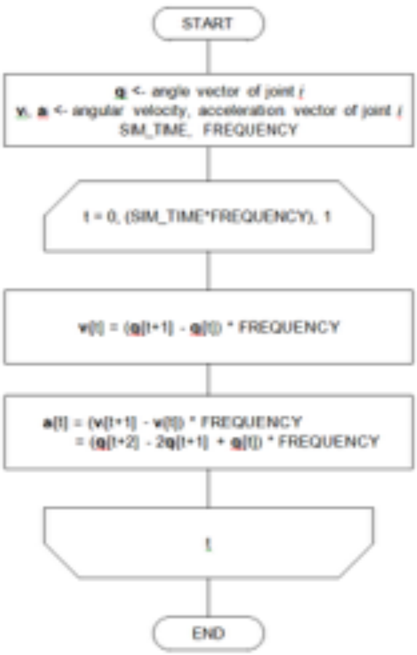
$v = dq/dt$, $a = dv/dt = d^2q/dt^2$

Finally, we calculate the torque vector using the Lagrange-Euler function with q, v, and a vector, about which is explained in detail in the flow chart 2.

2.2. Flow charts



Flow chart 1. The whole process of experiment



Flow chart 2. How to get v, a vectors

3. Results and conclusion of the experiment

-The explanation about graph
The horizontal line shows the simulation time and the vertical one shows the required torques for each joint.

4. Overview of our further work

The field of Baxter Robot is still at its early stage and needed to be developed. As a result of that, our research is still not mature and should be constantly improved, there still exist some deficiencies for us to make up in the future. There are a lot of further work that we plan to do and about our research orientation, which can be divided into the following several aspects in general:

1) About the applied method:

That can be diversified, in our experiments we only concentrated on the the Lagrange formulation. In the future we can add other relevant methods to our experiments and simulation, such as the important method, Newton-Euler method.

2) About the amount of the chosen joints:

In our experiments we only choose two of the seven joints on each of the robot`s arms, while adding all the parameters of seven joints into calculation is very complex and the data is also very hard to process. But on the other hand, this is also a new challenge that we may deal with and overcome in the future.

3) About the connection between virtual and real worlds

With the development of robot technology, people will pay more and more attention to the baxter robot. We have reasons to believe that the robots are where the virtual and the real meet most directly. Therefore, it is also necessary to apply our simulation and the dynamics model of baxter robot into reality in the near future.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)

4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`