한국동력기계공학회지 제18권 제4호 pp. 91-96 2014년 8월 (ISSN 1226-7813)
Journal of the Korean Society for Power System Engineering http://dx.doi.org/10.9726/kspse.2014.18.4.091
Vol. 18, No. 4, pp. 91-96, August 2014

Integrated SolidWorks & Simscape Platform for the Model-Based Control Algorithms of Robot Manipulators

Doo-Sung Ahn*

(Received 31 March 2014, Revised 23 June 2014, Accepted 23 June 2014)

Abstract: The application of the recent model-based control schemes for robot manipulators require the solution of problems concerning various aspects, from the mechanical design to the necessity of determining a robot model suitable for control, and of experimentally testing the control performances. For one solution, integration of SolidWorks with Simscape for designing and controlling robot manipulators is presented in this paper. The integration provides a platform for rapid control prototyping of robot manipulators without the need for building real prototypes. Mechanical drawings of a robot are first created using Solidworks and imported into the Simscape, where a robot is represented by connected block diagrams based on the principle of physical modeling. Simulation examples for 7-DOF SAM ARM made by Berrett Technology Inc. are testified to show effectiveness of the presented platform.

Key Words : Robot Manipulator, Model Based Control, Physical Modeling, SolidWorks, Simscape

— Nomenclature —

$C(q,\dot{q})$: centripetal-Coriolis matrix
$F(\dot{q})$: friction effects vector
$G(q)$: gravity vector
$p(q)$: end-effector position
$J(q)$: Jacobian matrix of the manipulator
K_v, K_p : closed-loop feedback gain matrixes
$k(q)$: forward kinematic equation
$M(q)$: inertia matrix
$q(t)$: joint position vector
r : reference signal
$x(t)$: end-effector position and orientation

Greek Symbols

α : new control variable
ζ : damping ratio
$\phi(q)$: end-effector orientation
τ : input torque vector
ω : undamped natural frequency

Superscripts

$\wedge$: estimations

---

*Doo-Sung Ahn(corresponding author) : Department of Mechanical & Automotive Engineering, Pukyong National University. E-mail : dsahn@pknu.ac.kr, Tel : 051-629-6154

Subscripts

c : computer commanded

f : friction

h : induced by end-effector forces

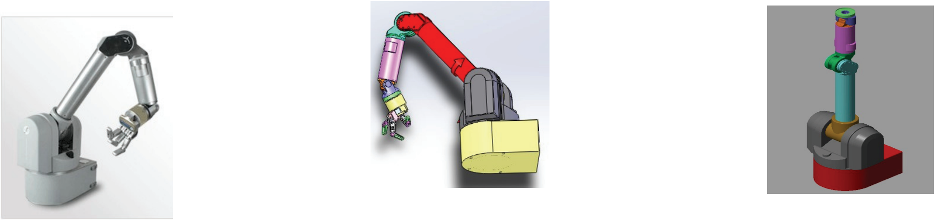
1. Introduction

Since the advent of robot manipulators, they are widely applied in many areas of industry. Many new research theories and technologies are developed with the advancement of various kinds of robots such as redundant manipulators or humanoids. In the robot control design field, its is well known that model-based control strategies for robot manipulators are very effective since they take into account modeling uncertainties thereby, enhancing robustness by making the robot track a time-varying reference trajectory1). However, the drawback of these approaches is to obtain the dynamic motion equations. They are essential to analyze the kinematic and dynamics solution, to plan the trajectory of the robot manipulators and to control during the course from their designs to experiment. However, both the kinematic and the dynamics problems of robot manipulators are very complex with difficult computing due to the multi degree of freedom and multilink space mechanisms of the robot. The effective solution is simulations, that is, the application of virtual modeling technology adopted by many researchers and scholars nowadays 2). It allows the design and development of new control algorithms, mechanical structures of robots and helps to define the optimal parameter specifications of a system (control gains, link lengths, masses, etc.). As the complexity of the system under investigation increases the role of modeling/simulation becomes more important.

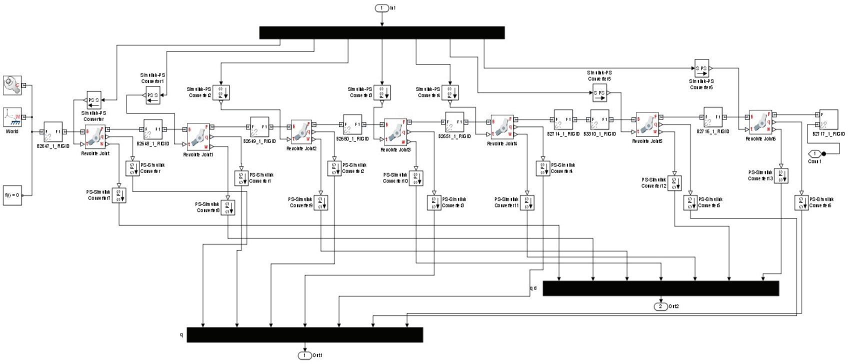
However, with traditional programming languages, even a simple system is difficult to model. First, you need to manually derive the equations of dynamic, and then manipulate them into a form that can be entered into a block diagram representation. Second, it is very difficult to figure out what kind of physical system it is and how the components in this system is practically connected. On the other hand, physical modeling language is also a kind of language but in a totally different way. The essential differences can be identified from the following two aspects: equation-based representation and bidirectional energy-based port. The object of this work is to research a newly-invented physical modeling language, MATLAB/Simscape3), and then explore the ability of Simscape in simulating robot manipulators.

2. MODELING ROBOT MANIPULATORS

Modeling is done in two stages. In the first stage, 3D CAD model of a robot manipulator is made by SolidWorks software4), Then, the CAD model is exported in 3D XML format by using the plug-in, SimMechanics Link, to MATLAB Simscape. Simscape/SimMechanic provides a multibody simulation environment for 3D mechanical systems. The multibody system is modeled using blocks representing bodies, joints, constraints, and force elements, and then SimMechanics formulates and solves the equations of motion for the complete mechanical system. Models from CAD systems, including mass, inertia, joint, constraint, and 3D geometry, can be imported into SimMechanics. As a result of the transfer of the model from CAD environment to SimMechanics software, the model could be used for simulation studies that are conducted to test the controller. Second stage includes the modeling of the control system and development of the necessary



(a) SAM 7 (b) SolidWorks model (c) Simscape model



(d) Physical model

Fig. 1 Whole modeling process of 7 DOF SAM ARM

kinematics and dynamics equations for the robot using Simulink blocks.[5] Imported model can be parameterized using MATLAB variables and expressions, and control systems for multibody system can be easily designed by powerful functions in Simulink. The visualization tools of SimMechanics is also used to animate 3D machine geometries. In this work, 7-DOF SAM Arm made by Berrett Technology[6] is selected as a sample manipulator robot. Fig.1 shows the whole modeling process of 7 DOF SAM ARM.

3. KINEMATICS & DYNAMICS MODEL

3.1 Kinematics model

The end-effector position and orientation in the operation space, denoted by $x(t) \in R^m$, is

$$x = k(q) = \begin{bmatrix} p(q) \\ \phi(q) \end{bmatrix} \tag{1}$$

where $k(q)$ is the forward kinematic equation, $q(t)$ denote the joint position vector of an n-link manipulator and $p(q)$ and $\phi(q)$ are the vectors representing the end-effect or position and orientation respectively. Based on Eq. (1), the differential relationships between the end-effector position and the link position variables are obtained as follows;

$$\dot{x} = \dot{J}(q)\dot{q} \tag{2}$$

$$\ddot{x} = \ddot{J}(q)\dot{q} + J(q)\ddot{q} \tag{3}$$

where $J(q)$ is the Jacobian matrix of the manipulator and $\dot{q}, \ddot{q}$ denote the joint velocity and acceleration vectors, respectively.

3.2 Dynamic model

The dynamic model for an n-link robot manipulator is developed in the following form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \tag{4}$$

where $M(q), C(q,\dot{q}), G(q), F(\dot{q})$ and τ represent the inertia matrix, the centripetal-Coriolis matrix, the gravity vector, the friction effects vector and input torque vector respectively.

4. Model-Based Control Scheme

This section presents a joint space control architecture to explicitly provide model-based control scheme. The proposed control architecture takes into account non-linear feedback linearization based on manipulator dynamics, allowing to control each joint separately.[7)]

4.1 Non-Linear Feedback Linearization

The generalized force contributions in the joint space, τ, are given by

$$\tau = \tau_c + \tau_f + \tau_h \tag{5}$$

where τ_c is the joint actuator torque (i.e. the computer commanded torque sent to the motor), τ_f is the joint friction torque and τ_h is the joint torque induced by end-effector forces in contact with the environment. Frictions are very difficult to model and since the SAM frictions are low, τ_f can be neglected in the dynamic model.τ_h is given by

$$\tau_h = J^T F_h \tag{6}$$

where F_h is end-effector external force/moment. In view of Eq. (4), Eq. (5) and by not taking into consideration the torque contribution due to frictions, the manipulator dynamic model in the joint space can be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau_c + \tau_h \tag{7}$$

Making computer commanded torque τ_c equal to

$$\tau_c = \hat{M}(q)\alpha + \hat{C}(q,\dot{q})\dot{q} + G(q) \tag{8}$$

where α is the new control variable and $\hat{M}(q)$, $\hat{C}(q,\dot{q})$ and $\hat{G}(q)$ are estimations, the following linear and decoupled system is achieved

$$\ddot{q} = \alpha \tag{9}$$

which corresponds to a double integrator plant. Equation (9) assumes no estimation errors and no external forces (τ_h=0). The linearization performance is directly connected with the dynamic model accuracy, being an accurate estimation of the dynamic parameters crucial for the overall system performance. In our scheme, robot dynamic parameters are automatically provided by 3D CAD.

4.2 Secondary Closed-Loop Pole-Placement

The closed-loop poles of the simple system given above in Eq.(9) can be easily specified by using the following statefeedback control law

$$\alpha = r - K_v\dot{q} - K_p q \tag{10}$$

where $K_v, K_p \in R^{n \times n}$ are diagonal matrixes of closed-loop feedback gains, and $r \in R^n$ is the reference signal. If reference signal is defined as

$$r = \ddot{q}_d + K_v\dot{q}_d + K_p q_d \tag{11}$$

where, q_d is the desired joint position, then from Eq. (11), Eq. (10) and Eq. (9)

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{12}$$

where $e \in R^n$ is the tracking error, and $e = q_d - q$. Eq. (12) means that by using the reference signal as in Eq. (11), the tracking error can be made to go to zero asymptotically by specifying appropriate values of K_v and K_p. By taking the Laplace transform of Eq. (12), the individual equation for the i-th link has the following closed-loop characteristic equation in the form of second-order dynamic system

$$s^2 + K_{v,i}\,s + K_{p,i} = 0 \tag{13}$$

Similarly, the following relationship can be obtained as $K_{p,i} = \omega_{n,i}^2$ and $K_{v,i} = 2\zeta_i\omega_{n,i} = 2\zeta_i\sqrt{K_{p,i}}$ That is, by specifying the values of the damping ratio ζ_i and the undamped natural frequency $\omega_{n,i}$ (i.e, the closed-loop poles), we can thus determine the values of closed-loop state feedback gains $K_{v,i}$ and $K_{p,i}$ to achieve a desired dynamic performance for the i-th link of the manipulator. Generally, overshoot is eliminated by choosing a large enough damping factor (i.e., $\zeta_i \geq 1$) so that the system is critically damped or overdamped. Fig.2 shows the overall diagram of control.

5. Simulations & Results

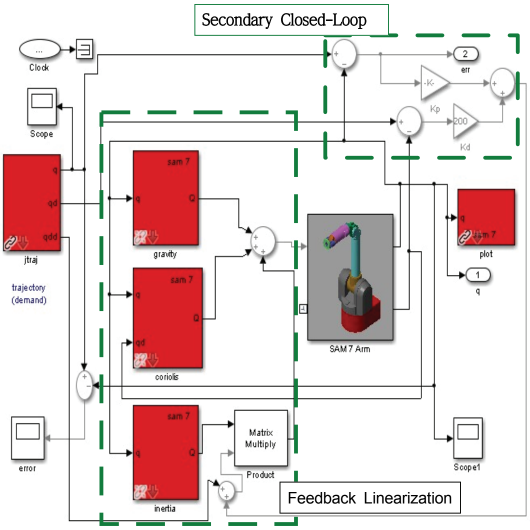


Fig. 2 Simulink/Simsape Control Model for 7 DOF SAM ARM

To illustrate the performance of the proposed integrated platform presented in this work, a set of simulation results are presented in this section. In these simulations, the aim is to utilize the virtual model of 7-DOF SAM Arm made by Berrett Thnology. The simulations are conducted MATLAB/Simulink simulation environment with variable-step sample time. The manipulator moves from at rest at the joint angles q = [0 0 0 0 0 0 0] at rest to q = [pi/2 0 0 －pi/2 0 0 0] at rest in 3 second. Fig. 3 shows the desired joint-space trajectories. In the controller presented Eq.(5) and Eq.(8), the nonlinear terms include centripetal and Coriolis $C(q,\dot{q})\dot{q}$ and the gravity effects, but frictional $F(\dot{q})$ and disturbance are neglected since the robot moves in slow motion. The controller parameters are tuned to the following values to adapt critical damping after some experimental tests. K_p =diag {10000, 10000, 10000, 10000, 10000, 10000, 10000}, K_v =diag {200, 200, 200, 200, 200, 200, 200}.

Fig. 4 shows the successive 3D-view of SAM 7 ARM. Due to 3D view, SAM Arm movement in manipulation space is easily understandable. Fig. 5 shows trajectory tracking errors of seven joints in case of no friction.

6. CONCLUSION

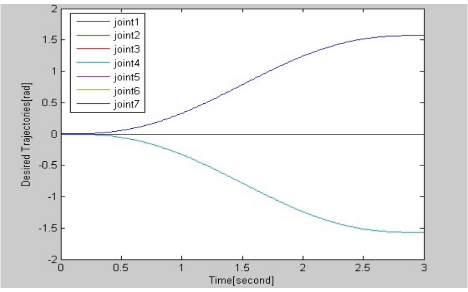


Fig. 3 Desired trajectories in joint space

The presented platform shows rapid control prototyping of robot manipulators without the need for building real prototypes. Simulation examples for 7-DOF SAM ARM made by Berrett Technology Inc. are testified to show effectiveness of the presented platform. In this paper a model based control algorithm of SAM-7 redundant robot is rapidly developed from SolidWorks 3D model seamlessly. In addition, it is possible to integrate the physical model completely in a simulation environment for motion control, verification and optimization. Based on this principles it enables us for further researches in fields like autonomous locomotion, human-robot-interaction, precise positioning or miniaturization.

Fig. 4 Successive view of SAM 7 ARM motion

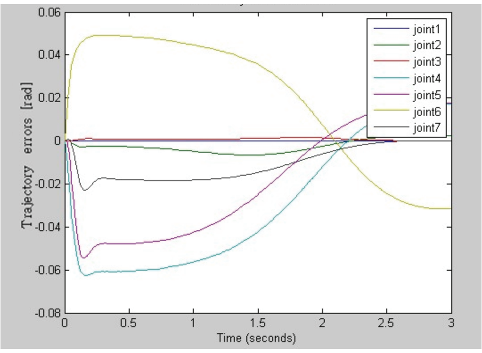


Fig. 5 Trajectory tracking errors

Acknowledgement

“This work was supported by a Research Grant of Pukyong National University (2013)”

References

1. I. Saha and N. Shankar, 2012, “ModelRob: A Simulink Library for Model-Based Development of Robot Manipulators”, IEEE International Conference on Robotics and Automation,USA, pp. 2654-2659.
2. D. S. Ahn and V. P. Nguyen, 2012, “Unified Motion and Force Control of JS-10 Robot Manipulator Based on Operational Space and 3D CAD”, J. of KSPSE, Vol. 16, pp. 57-63.
3. http://www.mathworks.co.kr/products/simscape
4. http://www.solidworks.co.kr
5. http://www.petercorke.com/RVC
6. http://www.barrett.com/robot/products-arm.htm
7. L. Santos and R. Cortesao, 2012, "Joint Space Torque Control with task Space Posture Reference for Robotic-Assisted Tele-Echography", 21st Int. Symposium on Robot and Human Interactive Communication, Paris, pp. 126-131.