DOUBLE ARM ROBOT DESIGN, CONTROL AND OPTIMIZATION USING MODELICA AND OPTIMICA

Moh’d Sami Ashhab\textsuperscript{a, b}, Thilo Breitsprecher\textsuperscript{a} and Sandro Wartzack\textsuperscript{a}

\textsuperscript{a} Lehrstuhl für Konstruktionstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Martens Strasse 9, 91058 Erlangen, Germany
\textsuperscript{b} Department of Mechanical Engineering, The Hashemite University, Zarqa, 13115, Jordan
e-mail: sami@hu.edu.jo, telephone: (962) 5-390-3333, fax: (962) 5-382-6348

ABSTRACT
A double arm robot is analyzed using the Modelica and Optimica software tools. Modelica is a tool for modeling complex physical systems whereas Optimica is provided by Jmodelica.org and is a recent easy-to-use extensible Modelica-based source for dynamic optimization. The double arm robot was not previously optimized with Optimica. The robot system consists of two rigid arms and two motors to actuate the two arms. The optimization algorithm objective is to move the tip of the robot from one position to another in a certain time with the least energy consumption. In order to program the optimization algorithm in Optimica it is necessary to write the differential equations of motion for the system which are derived using the Lagrangian’s formula. The optimization algorithm for the double arm robot is programmed in Optimica and the results are plotted and compared. Furthermore, a closed loop PID controller is implemented for each actuator in Modelica with the objective of moving the tip of the robot from one position to another. The simulation results of the controlled robot are shown and compared with those of Optimica.

KEY WORDS
Modeling and simulation, optimization, double arm robot, Modelica and Optimica

1. Introduction
Dynamic optimization has gained a lot of importance recently and its valuable benefits are continuing to grow with the advent of new engineering processes, products and technologies. The benefits of dynamic optimization include but are not limited to power consumption reduction, cost reduction, space saving, technical requirements and regulations satisfaction. The research work of [1] presents a global optimization method focused on gear vibration reduction by means of profile modifications where a nonlinear dynamic model is used to study the vibration behavior; such model is validated using data available in literature. Dynamic optimization has been used to maximize rice quality synthesis by seeking the corresponding configuration of units and their corresponding operating conditions [2]. A reduced dynamic model for a solar cell characterized by inputs and outputs has been used in [3] to dynamically optimize the operating parameters of the cell behavior by means of model predictive control in order to maximize the mean flow rate of distilled water confronted to a variable solar radiation.

Dynamic optimization deal with systems that vary with time, or equivalently, that can be described with differential equations. In order to perform the dynamic optimization, modeling of the system under consideration should be first performed. For complex systems it is very difficult to obtain the differential equations that describe the system dynamics by mathematical analysis. The use of available modeling software will be a more suitable alternative. A powerful software tool that can model complex systems is Modelica. Block diagram models that mimic real applications can be constructed and simulated with Modelica where there is no need to derive or write the corresponding differential equations. At the same time, Modelica have the option of programming differential equations. Other software tools for modeling are available. An example of those is Matlab/Simulink. Once the dynamic model of a system is generated by software, an important question arises: can optimization tools be applied to such a model? In the Modelica software case, the answer is yes with the recent tool Optimica which is provided by Jmodelica.org and is reported in [4]. Jmodelica.org is a new Modelica-based open source project targeted towards dynamic optimization and its objective is to bridge the gap between the need for high-level description languages and the details of numerical optimization algorithms.

Optimica enables compact and intuitive formulations of optimization problems, static and dynamic, based on Modelica models [4]. However, the differential equations of the system under consideration should be derived and written in the Optimica code. In other words, there is still a gap between Modelica block diagram representations and Optimica. In the literature, research related to dynamic optimization with Optimica is limited to a few applications which have been demonstrated and solved.
Examples of such applications are model predictive control of a continuously stirred tank reactor containing an exothermic reaction [4], distillation column load change [4], industrial robot minimum time problem [5] and a cart and pendulum minimum time [6].

The dynamic optimization problem of a double arm robot is considered in this paper. The double arm robot forms the basis for many industrial processes. A block diagram representation can be easily constructed and simulated in Modelica. Due to the fact that Optimica cannot deal with Modelica block diagram representations, the double arm robot differential equations of motion are derived using the Lagrangian’s formula. The robot system consists of two rigid arms and two motors to actuate the two arms. Two scenarios are considered: 1) the two motors are stationary and 2) one of the motors is stationary and the other motor is carried by the first arm. The optimization algorithm objective is to move the tip of the robot from one position to another with the least energy consumption. The optimization algorithm for the double arm robot is programmed in Optimica and the results are plotted and compared for the two above scenarios. Furthermore, a closed loop PID controller is implemented and compared with those of Optimica. In Modelica the system is modeled with the aid of block diagrams without the need to write the equations of motion.

2. Double Arm Robot Model

The double arm robot forms the basis for many industrial processes. It can be represented by the schematic diagram shown in Fig. 1. The robot system consists of two rigid arms and two motors to actuate the two arms with torques $T_1$ and $T_2$. The first arm (arm 1) with length $L_1$ and mass $m_1$ rotates around the fixed point A whereas the second arm (arm 2) with length $L_2$ and mass $m_2$ rotates around the free end of arm 1, namely, point B. The torque $T_1$ is applied to arm 1 at the point A and torque $T_2$ is applied to arm 2 at point B. The rotational motion takes place in the horizontal $x$-$z$ plane and therefore the gravity has no inertial effect on the system. However, we will show how to include the gravity effect for rotation in the vertical plane. The motor that produces torque $T_2$ can be mounted on arm 1 at point B and its mass is denoted as $M_m$. This is not the best scenario since the motor will be moving which requires extra torque $T_1$ and energy. A better alternative is to mount the motor that produces torque $T_2$ at point A and connect it to point B through pulleys and cables. This will not impose any inertia from the motor and the system will be more efficient. We will analyze the two different scenarios for the motor.

![Fig. 1 Schematic diagram of the double arm robot](image)

The equations of motion for the double arm robot will be now derived to be exploited later by the optimization tool Optimica. The angular positions of arms 1 and 2 are $\theta_1$ and $\theta_2$, respectively (see Fig. 1). Note that $\theta_1$ is measured from the fixed $z$-axis whereas $\theta_2$ is measured from a moving axis parallel to arm 1. The absolute angular position of arm 2 is $\theta_1 + \theta_2$. The position vectors for the center of mass of arms 1 and 2 measured from the origin A are written as

$$r_1 = -\frac{L_1}{2}\sin\theta_1 i + \frac{L_1}{2}\cos\theta_1 k$$

and

$$r_2 = -(L_1\sin\theta_1 + \frac{L_2}{2}\sin(\theta_1 + \theta_2))i + (L_1\cos\theta_1 + \frac{L_2}{2}\cos(\theta_1 + \theta_2))k$$

respectively. Our target is to write the energy equations for the system and thus the velocities of the centers of mass are needed. They are the time derivatives of the of the above position vectors and are written as

$$v_1 = -\frac{L_1}{2}\theta_1 \cos\theta_1 i - \frac{L_1}{2}\theta_1 \sin\theta_1 k$$

and

$$v_2 = -(L_1\theta_1 \cos\theta_1 + \frac{L_2}{2}(\theta_1 + \theta_2)\cos(\theta_1 + \theta_2))i - (L_1\theta_1 \sin\theta_1 + \frac{L_2}{2}(\theta_1 + \theta_2)\sin(\theta_1 + \theta_2))k$$

respectively. The Lagrangian’s formula is given as
\[ L = T - V \]

where, \( T \) is the kinetic energy and \( V \) is the potential energy of the system. The kinetic energy is the sum of the translational and rotational kinetic energies for both links or arms in addition to the kinetic energy of the motor at point \( B \) (if the motor is mounted there). Recall that this motor's mass is \( M_m \). The kinetic energy is obtained as

\[
T = \frac{1}{2} (m_1 v_1^2 + m_2 v_2^2 + \frac{1}{12} m_1 L_1^2 \dot{\theta}_1 + \frac{1}{12} m_2 L_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_B L_1 \dot{\theta}_1 )
\]

where, \( m_B \) is the concentrated mass at point \( B \) and is given

\[
m_B = \begin{cases} 
0 & \text{no motor at point } B \\
M_m & \text{motor at point } B 
\end{cases}
\]

The potential energy of the system stays at a zero value since the robot motion is constrained in the horizontal plane. To have a complete work, it is worthwhile to mention that if the motion is in the vertical plane (the \( z \)-axis is replaced by the \( y \)-axis in Fig. 1) then the potential energy, \( V_{ver} \), with reference at a horizontal line passing through point \( A \) can be written as

\[
V_{ver} = m_1 g \frac{L_1}{2} \cos \theta_1 + m_2 g (L_1 \cos \theta_1 + \frac{L_2}{2} \cos(\theta_1 + \theta_2)) + m_B g L_1 \cos \theta_1
\]

Note that for the motion in the horizontal plane, \( g \) can be set to zero in Equation (8). Thus, in order to generalize the results it is a good practice to include the potential energy of equation (8) in the Lagrangian given in Equation (5), and set \( g \) to zero when the motion is constrained in the horizontal plane. The differential equations of motion for the double arm robot are obtained from the Lagrangian as

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = T_i, i = 1, 2
\]

Applying Equation (9) to the Lagrangian given in Equation (5) we get the following two differential equations in terms of the angular positions and their derivatives

\[
M_{1} \ddot{\theta}_1 + M_{12} \ddot{\theta}_2 = P_1 + T_1
\]

\[
M_{12} \ddot{\theta}_1 + M_{2} \ddot{\theta}_2 = P_2 + T_2
\]

where,

\[
M_{11} = \left( \frac{m_1}{3} + m_2 + m_B \right) L_1^2 + m_2 L_2 \left( L_1 \cos \theta_2 + \frac{L_2}{3} \right)
\]

\[
M_{12} = m_2 L_2 \left( L_1 \cos \theta_2 + \frac{L_2}{3} \right)
\]

\[
M_{22} = \frac{m_2}{3} L_2^2
\]

\[
P_1 = m_2 L_4 L_2 \sin \theta_1 \left( \frac{\theta_1 + \frac{\theta_2}{2}}{2} \right) \theta_2 + \left( \frac{1}{2} m_2 L_2 \sin(\theta_1 + \theta_2) + \left( \frac{m_1}{2} + m_2 + m_B \right) L_1 \sin \theta_1 \right) g
\]

\[
P_2 = -\left( \frac{1}{2} m_2 L_4 L_2 \sin \theta_2 \right) \left( \frac{\theta_1 + \frac{\theta_2}{2}}{2} \right) \theta_1^2 + \left( \frac{1}{2} m_2 L_2 \sin(\theta_1 + \theta_2) \right) g
\]

\[
g = \begin{cases} 
0 & \text{horizontal motion} \\
9.81 \text{m/s}^2 & \text{vertical motion}
\end{cases}
\]

There are four state variables for the double arm robot, namely, the two angular positions and their first time derivatives or angular velocities. Note that the above equations take into account both motions in the horizontal and vertical planes. In addition they can solve the two scenarios of mounting or not mounting the motor at the tip of arm 1 (point \( B \)).

The availability of the differential equations of the system given in Equations (10) and (11) allows the use of the software tool Optimica (from jmodelica.org) to perform dynamic optimization on the system as we will see in the next section. It is worth to mention here that the equations of motion are needed solely for the dynamic optimization procedure provided by Optimica which is an extension of Modelica. The double arm robot can be more easily constructed in Modelica as a block diagram representation without needing the above tedious differential equations but Optimica cannot deal with block diagram representations. However, feedback control of the double...
3. Dynamic Optimization with Optimica

The double arm robot model derived in the previous section will be coded in Optimica for dynamic optimization. Optimica is an extension of the Modelica language which has gained increased attention during the last decade. Modelica is about to establish itself as a de facto standard in the modeling community with strong support both within academia and industry [4]. While systems similar to the double arm robot have been modeled with Modelica, the dynamic optimization of the double arm robot has not been performed with Optimica. In order to perform such a dynamic optimization, the mathematical differential equations of motion for the double arm robot are needed. They were derived in section 2. Note that Optimica does not deal with the easily constructed Modelica block diagram representations.

Optimica offers several dynamic optimization options such as optimal control, minimum energy, minimum time and parameter optimization. In this research, the optimization algorithm objective is to move the tip of the double arm robot (point C in Fig. 1) from an initial position to a final position in a certain time with the least energy consumption. Constraints on angular positions, velocities and accelerations in addition to torques or inputs can be included.

Based on the double arm robot model, the following dynamic optimization problem is formulated

$$\min_{\theta_1(t_f), \theta_2(t_f)} t_f \left( T_1 d\theta_1 + T_2 d\theta_2 \right)$$  \hspace{1cm} (18)

subject to the dynamics given in Equations (10) and (11), where $t_f$ is the final time. Note that the torque times the corresponding angular displacement represents energy. Therefore, the integral given in Equation (18) represents the total energy required by the torques (or motors generating these torques) to move the system during the time interval $[0,t_f]$. The problem will also include constraints for the required final position of the tip of the double arm robot. The initial position is provided as initial conditions for the differential equations (10) and (11). Other constraints can be included on the various signals of the system such as angular positions, velocities and accelerations in addition to torques.

The double arm robot model dynamic optimization problem as defined above will be now solved with Optimica. It will be assumed that the motion is constrained in the horizontal plane, that is, $g = 0$. Let $m_1 = 3$ kg, $m_2 = 2$ kg, $L_1 = 0.9$ m, $L_2 = 0.6$ m and $M_m = 12$ kg. It is desired to move the robot from zero angular positions and velocities to a final position with the values $\theta_1 = \theta_2 = 45^\circ = 0.785$ radian and zero angular velocities. The corresponding final position for tip C can be easily calculated. The final time is taken as 1 second. Torque $T_1$ is constrained to 0.5 N.m whereas Torque $T_2$ is constrained to 0.1 N.m. We will first consider the better scenario of having the two motors stationary at point A. Arm 1 motor provides torque $T_1$ directly to arm 1 (through a revolute component) and arm 2 motor provides torque $T_2$ to arm 2 through a cable and pulley system. In this case the concentrated mass at point B is equal to zero, that is, $m_B = 0$. The Optimica code which performs the double arm robot dynamic optimization problem is listed below.

```
optimization DPend (objective = cost(finalTime),
                     startTime = 0, 
                     finalTime = 1)

Real M11;
Real M12;
Real M22;
Real P1;
Real P2;
Real m1 = 3;
Real m2 = 2;
Real L1 = 0.9;
Real L2 = 0.6;
Real g = 0;
Real mB = 0;

Real th1 (start=0, fixed = true);
Real dth1 (start=0, fixed = true);
Real th2 (start=0, fixed = true);
Real dth2 (start=0, fixed = true);

input Real T1;
input Real T2;

Real cost(start=0,fixed=true);

equation

M11 = ((m1/3+m2+mB)*L1*L1+2*m2*L2*(L1*cos(th2)+L2/3));
M12 = (m2*L2*(L1*cos(th2)+L2/3));
M22 = M2^2/3;
P1 = m2*L1*L2*cos(th2)*(dth1+dth2/2)*dth2+(m2*L2*sin(th1+th2)/2)+(m1/2+m2+mB)*L1*sin(th1))\^g;
P2 = (m2*L1*L2*sin(th2/2)/2)*dth1^2+(m2*L2*sin(th1+th2/2)*g;

der(th1) = dth1;
der(dth1) = (M12/(M12^2-2*M11*M22))*(P2-M22*P1/M12+2*M22*P1/M12)*dth1;
der(th2) = dth2;
der(dth2) = (M11/(M11*M22-2*M12^2))*(P2-M12*P1/M11+2*M12*P1/M11)*dth2;
der(cost) = T1*dth1 + T2*dth2;
```
Optimica enables compact and intuitive formulation of the double arm robot dynamic optimization problem as can be seen from the above code. The programming language python is used to run the Optimica code and generate and plot the results. The angular positions and torques corresponding to the above dynamic optimization problem are shown in Figs. 2 and 3, respectively. The torque values are negative because they act counterclockwise, or equivalently, in the negative \( y \)-direction.

The worse scenario is to mount the motor that drives arm 2 at point \( B \). In this case, the concentrated mass at point \( B \) is equal to the motor mass, namely, \( m_B = M_m \). In the Optimica code this is easily set. The corresponding angular positions and torques are shown in Figs. 4 and 5, respectively. It is clear in this case that larger torques are required which agrees with common sense. Extra energy is required to carry the moving motor at point \( B \).

4. Feedback Control with Modelica

Feedback control of the double arm robot can be performed with Modelica. There are two main advantages of using Modelica: 1) its block diagram representation for the system mimics the real life application and 2) there is no need to derive the complex differential equations of motion for the system. Modelica can actually model very complex systems with the use of block diagrams.

The feedback control problem that we will simulate in Modelica assumes that the two angular positions \( \theta_1 \) and \( \theta_2 \) are available for measurement. Each angular position measurement is fed back to a separate PID (Proportional Integral Derivative) controller. The controller
corresponding to $\theta_1$ provides the torque $T_1$ and the controller corresponding to $\theta_2$ provides the torque $T_2$. PID controllers are widely used in industry due to their tracking and disturbance rejection capabilities. Modelica provides the tools for constructing the PID controlled double arm robot. The corresponding block diagram representation is shown in Fig. 6. Note that the model mimics to some extent the real life application.

The objective of the feedback controller is to move the tip of the double arm robot from an initial position to a final position. Unlike the dynamic optimization problem which can provide better results, the final time and additional constraints are not set. It is unfortunate that the dynamic optimization Modelica extension tool Optimica cannot deal with block diagram representations. It is assumed that the two motors are mounted at point A, that is, $m_B = 0$. The initial and final system positions are selected as in the previous section, that is, the initial position is described by zero angular positions and velocities whereas the final position is described with the values $\theta_1 = \theta_2 = 45^\circ = 0.785$ radian and zero angular velocities. The gains of the PID controllers were selected by experimentation to get a suitable performance. The proportional, integral and derivative gains of the PID controller corresponding to $\theta_1$ were set to 100, 0.1 and 1, respectively, whereas for the PID controller corresponding to $\theta_2$ they were set to 100, 1 and 1, respectively. The simulation results are plotted in Figs. 7 and 8. Fig. 7 shows the angular positions of the arms and Fig. 8 shows the torques. The angular positions go to their desired values in a longer time than that set by the dynamic optimization problem (which is 1 second as given in section 3). In addition, the torques are larger than those of the optimization results which consume minimum energy. Therefore, the PID controllers achieve their objective of tracking the desired angular positions and velocities, but they do not provide the optimum solution which is obtained by Optimica simulation.

5. Conclusion

The dynamic optimization of a double arm robot was performed using the new intuitive tool Optimica which is an extension of the Modelica language. The differential equations of motion of the double arm robot were derived and then used by Optimica to obtain and plot the results of the minimum energy dynamic optimization problem.
Two different scenarios regarding one of the actuating motors position were compared. Optimica proved to be a very useful optimization tool but it requires the availability of mathematical differential equations for the system under consideration. Furthermore, the feedback control of the double arm robot was demonstrated with Modelica where the mathematical differential equations of motion were not required. A block diagram representation which mimics the real life application was constructed instead. An interesting future work will be bridging the gap between Modelica block diagram representations and dynamic optimization through Optimica. This would facilitate the procedure for dynamic optimization and avoid deriving the tedious differential equations of motion.

![Fig. 7 The feedback control system angular positions with $m_B = 0$](image1)

![Fig. 8 The feedback control system torques with $m_B = 0$](image2)

**References**


