N-link inverted pendulum, LQR control, some observations

Martin Grossman, Alexander Gmiterko

Abstract

In this article is the automatized procedure for deriving of n-link inverted pendulum motion equations presented. Example of 2 link inverted pendulum is included. The LQR algorithm using Maple input equations is proposed. Also the comparison between SimMechanics and Simulink is included.

Keywords: pendulum, n-link, maple, simulink, chaos, SimMechanics, Simulink

Introduction

The inverted pendulum has become a standard benchmarking problem. At the department of Applied Mechanics and Mechatronics, we have decided to build single, double and later triple inverted pendulum. In most of the publications [1], [2], [3], [4], we see that authors use already derived equations or derive them for each case of a single, double or triple pendulum system.

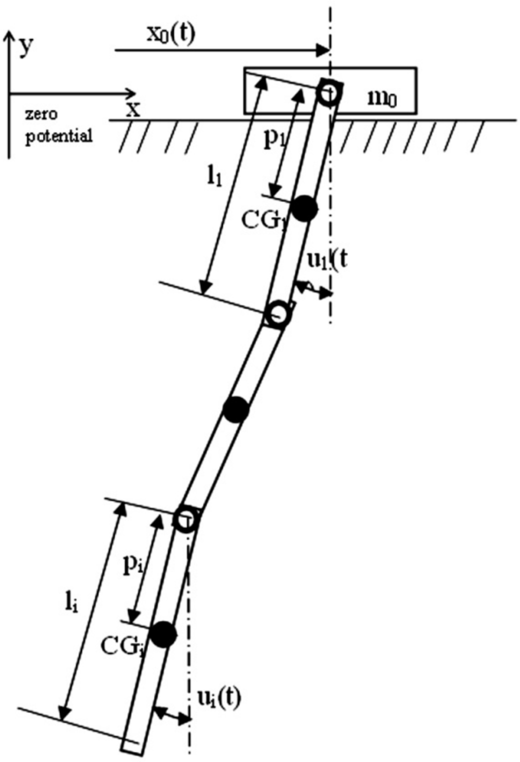

Fig. 1 N-link pendulum

Nevertheless, there is a lack of general equation for the n-link pendulum. For this purpose, I decided to derive and automatize the process of generation of this general equation. It turned out that it was an invaluable help for us, in time of building the inverted pendulum model in Matlab and Simulink. This paper is divided into three parts. In the first part, the geometric model is shown. In the second part will be introduced the derivation of general equation using Maple and an example will be presented. In the last part, I discuss the further possibilities, using Maple and other symbolic mathematical software. By its mean, hopefully one gets better understanding of behavior these systems (non-minimal phase type system).

1. Pendulum's geometry

The basic assumptions are:

- Pendulum movement is in E^2(plane)
- Cart and links are rigid homogeneous bodies (RBD rigid body dynamics)
- Joints with or without friction, but no compliance
- Links are cylinders with variable lengths and masses

In the Table.1 are listed all variables and constants used in the model. From the illustration at the Fig.1, we see that the n-link inverted pendulum consists of n links connected with the revolute joints. The cart motion is allowable only in x direction by translational joint. I decided not to picture the forces, since their implementation will be very straightforward using Lagrange-Euler formula. Instead, the angles between the links (which are actually measured by sensors), the angles between link and its rotation angle (u_i) from vertical position were chosen. This is just for more clear derivation.

Variable	description	dimension
$x_0(t)$	x position of the cart	m
$x_{1\ldots n}(t)$	x position of the i-th link	m
$y_{1\ldots n}(t)$	y position of the i-th link	m
$u_{1\ldots n}(t)$	angle of i-th link with y axis	rad
$Icg_{1\ldots n}$	moment of intertia of i-th link to its center of gravity	$kg.m^2$
n	number of links	-
m_0	mass of the cart	kg
$m_{1\ldots n}$	mass of the i-th link	kg
$l_{1\ldots n}$	length of the i-th link	m
$p_{1\ldots n}$	distance of the gravity center of the i-th link (CG_i) from its beginning	m
CG_i	center of gravity of the i-th link	-

Tab. 1 Table of variables

2. Dynamic behavior and equations

The standard method of deriving dynamical equations of multi rigid body systems is using Euler-Lagrange formula. Its usage is relatively very easy, and for our purpose, it is especially adequate. This method does involve only derivatives of time, speed and position and therefore is very suitable for manual and of course half-manual (by software usage) procedure. We have though remember, that we are losing information about inner forces in joints. Other type of dynamical formulation will later obtain these.

The most important and only little more difficult part of the procedure is obtaining the kinetic and potential energy of the whole n-link pendulum. By inspection, we find that the kinetic energy and potential energy of the system consisting of n links and one cart is *Ek(n)* (1) and *U(n)* (2), respectively.

$$Ek(n) = m_0\left(\frac{d}{dt}x(t)\right)^2 + \sum_{k=1}^{n}\frac{1}{2}m_k\left(\left(\begin{array}{l}\frac{d}{dt}x(t)-\\ -\sum_{i=1}^{k-1}l_i\cos(u_i(t))\frac{d}{dt}u_i(t)-\\ -p_k\cos(u_k(t))\frac{d}{dt}u_k(t)\end{array}\right)^2 + \left(\begin{array}{l}\sum_{i=1}^{k-1}-l_i\sin(u_i(t))\frac{d}{dt}u_i(t)-\\ -p_k\sin(u_k(t))\frac{d}{dt}u_k(t)\end{array}\right)^2\right) + \sum_{k=1}^{n}\frac{1}{2}Icg_k\left(\frac{d}{dt}u_k(t)\right)^2 \tag{1}$$

$$U(n) = -g\sum_{k=1}^{n}\left(m_k\left(\sum_{i=1}^{k-1}l_i\cos(u_i(t)) + p_k\cos(u_k(t))\right)\right) \tag{2}$$

After deriving, the equation (1) and (2) we can put these in suitable form into any software allowing symbolic mathematical computation like Mathematica, Maple, Maxima or even Matlab with symbolic toolbox (Maple 3.0) etc (fig. 2). The next easy step is computing the Lagrangian L (3) and motion equations (4),

$$L = Ek(n) - U(n) \tag{3}$$

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial\dot{\xi}}L\right) - \frac{\partial}{\partial\xi}(L) = F^{\xi} \tag{4}$$

where ξ is generalized coordinate (in our case $x(t)$ or $u(t)$) and $\dot{\xi}$ is its time derivative. F^{ξ} is generalized non-conservative force acting on generalized coordinate.

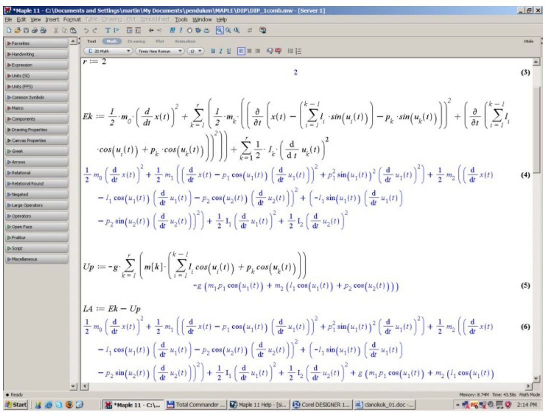

Fig. 2 Maple environment

As an example, let us consider inverted pendulum consisting of 2 links. Then the *L(2)* would be (5) and the equations of motion are : (6)(7)(8)with external forces F^x (this could be input force acting on cart, friction of rail, damping, etc.).

$$L(2) = 1/2m_0\left(\frac{d}{dt}x(t)\right)^2 + 1/2\cdot m_1\left(\begin{array}{l}\left(\frac{d}{dt}x(t)-p_1\cos(u_1(t))\frac{d}{dt}u_1(t)\right)^2+\\ p_1^2(\sin(u_1(t)))^2\left(\frac{d}{dt}u_1(t)\right)^2\end{array}\right) + 1/2\cdot m_2\left(\begin{array}{l}\left(\begin{array}{l}\frac{d}{dt}x(t)-l_1\cos(u_1(t))\frac{d}{dt}u_1(t)\\ -p_2\cos(u_2(t))\frac{d}{dt}u_2(t)\end{array}\right)^2+\\ \left(\begin{array}{l}-l_1\sin(u_1(t))\frac{d}{dt}u_1(t)-\\ p_2\sin(u_2(t))\frac{d}{dt}u_2(t)\end{array}\right)^2\end{array}\right) + 1/2\cdot Icg_1\left(\frac{d}{dt}u_1(t)\right)^2 + 1/2Icg_2\left(\frac{d}{dt}u_2(t)\right)^2 + g\left(m_1p_1\cos(u_1(t)) + m_2\left(\begin{array}{l}l_1\cos(u_1(t))+\\ p_2\cos(u_2(t))\end{array}\right)\right) \tag{5}$$

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial\dot{x}}L(2)\right) - \frac{\partial}{\partial x}(L(2)) = F^x \tag{6a}$$

$$\begin{aligned}&m_0\frac{d^2}{dt^2}x(t) + m_1\frac{d^2}{dt^2}x(t) + m_1p_1\sin(u_1(t))\left(\frac{d}{dt}u_1(t)\right)^2 -\\ &m_1p_1\cos(u_1(t))\frac{d^2}{dt^2}u_1(t) +\\ &m_2\frac{d^2}{dt^2}x(t) + m_2l_1\sin(u_1(t))\left(\frac{d}{dt}u_1(t)\right)^2 -\\ &m_2l_1\cos(u_1(t))\frac{d^2}{dt^2}u_1(t) + m_2p_2\sin(u_2(t))\left(\frac{d}{dt}u_2(t)\right)^2 -\\ &m_2p_2\cos(u_2(t))\frac{d^2}{dt^2}u_2(t) = F(t) - c\frac{d}{dt}x(t)\end{aligned} \tag{6b}$$

$$\begin{aligned}&-m_1p_1\cos(u_1(t))\frac{d^2}{dt^2}x(t) + m_1p_1^2\frac{d^2}{dt^2}u_1(t) -\\ &m_2l_1\cos(u_1(t))\frac{d^2}{dt^2}x(t) - m_2l_1\cos(u_1(t))p_2\sin(u_2(t))\left(\frac{d}{dt}u_2(t)\right)^2 +\\ &m_2l_1\cos(u_1(t))p_2\cos(u_2(t))\frac{d^2}{dt^2}u_2(t) + m_2l_1^2\frac{d^2}{dt^2}u_1(t) +\\ &m_2l_1\sin(u_1(t))p_2\cos(u_2(t))\left(\frac{d}{dt}u_2(t)\right)^2 +\\ &m_2l_1\sin(u_1(t))p_2\sin(u_2(t))\frac{d^2}{dt^2}u_2(t)\\ &+Icg_1\frac{d^2}{dt^2}u_1(t) + gm_1p_1\sin(u_1(t)) + gm_2l_1\sin(u_1(t)) = 0\end{aligned} \tag{7}$$

Therefore, with this procedure we can derive all n equations of motion.

Next step would be solving these equations. These equations are nonlinear differential equations and the Matlab was used to find numerical solution. Already the double inverted pendulum could be considered as the chaotic system, and thus it is important to bear in mind this behavior during the process of finding solution.

There has been extended work done, regarding comparison of numeric integrators used for this chaotic type problem using matlab, maple and MSC.ADAMS. We will compare these results with real physical model.

$$\begin{aligned}
&-m_2 p_2 \cos\left(u_2(t)\right)\frac{d^2}{dt^2}x(t)- \\
&m_2 p_2 \cos\left(u_2(t)\right) l_1 \sin\left(u_1(t)\right)\left(\frac{d}{dt}u_1(t)\right)^2 + \\
&m_2 p_2 \cos\left(u_2(t)\right) l_1 \cos\left(u_1(t)\right)\frac{d^2}{dt^2}u_1(t)+ \\
&m_2 p_2 \sin\left(u_2(t)\right) l_1 \cos\left(u_1(t)\right)\left(\frac{d}{dt}u_1(t)\right)^2 + \\
&m_2 p_2 \sin\left(u_2(t)\right) l_1 \sin\left(u_1(t)\right)\frac{d^2}{dt^2}u_1(t)+m_2 {p_2}^2\frac{d^2}{dt^2}u_2(t)+ \\
&Icg_2\frac{d^2}{dt^2}u_2(t)+g m_2 p_2 \sin\left(u_2(t)\right)=0
\end{aligned} \tag{8}$$

3. Symbolic computation, Maple

In my experience Maple have many advantages but also some disadvantages. It has relatively very simple interface allowing even the beginners quick and impromptu problem solving.

During my work I have found out, that the version 11, have problems with handling more then 3 inner sums with other then latin-2 (not Greek) letters and subscripts. It is also considerably slower then the Mathematica 5.2 (tried the same problem solving). In addition, graphics quality of implicit equation is not satisfactory.

4. LQR controller design, for stabilizing the Double Inverted Pendulum in upright position

The motion equations of DIP were derived in Maple and linearized in upright position (9) in Maple.

$$\begin{aligned}
&m_2 p_2\frac{d^2}{dt^2}x(t)+m_2 l_1 p_2\frac{d^2}{dt^2}d_1(t)-g m_2 p_2 d_2(t) \\
&+\left(m_2 {p_2}^2+Izz_2\right)\frac{d^2}{dt^2}d_2(t)=0 \\
&\left(m_1 p_1+m_2 l_1\right)\frac{d^2}{dt^2}x(t)+\left(-g m_1 p_1-g m_2 l_1\right)d_1(t) \\
&+\left(m_1 {p_1}^2+m_2 {l_1}^2+Izz_1\right)\frac{d^2}{dt^2}d_1(t)+m_2 l_1 p_2\frac{d^2}{dt^2}d_2(t)=0 \\
&c\frac{d}{dt}x(t)+\left(m_0+m_1+m_2\right)\frac{d^2}{dt^2}x(t)+\left(m_1 p_1+m_2 l_1\right)\frac{d^2}{dt^2}d_1(t) \\
&+m_2 p_2\frac{d^2}{dt^2}d_2(t)=0
\end{aligned} \tag{9}$$

From (9) the A,B,C,D matrices were derived using Maple. After deriving state matrices, the LQR K matrix coefficients were calculated with Matlab.
These coefficient are already able to stabilize DIP but with this approach we don't include other objectives and constraint as time, maximal travel in x direction or other. Therefore for the first and rough estimate of K constants we used optimization toolbox in Simulink to decrease travel of cart in x direction. This was achieved by constraining the maximal travel in x direction and optimizing all 6 constants in K matrix.

After this parametric optimization using gradient method, we were able to decrease the travel by 50% and keep the power consumption under acceptable level.

5. Simulink vs. SimMechanics Control Loop Simulation

The control loop was performed by using two different toolboxes in Matlab.

First, was using Simulink (Fig.3). We translated nonlinear equations to Simulink by means of F(u) function blocks. Then the computed K matrix was implemented and optimization of K matrix parameters was performed.

The second approach, we modeled the inverted pendulum using SimMechanics, again use the generated K matrix a perform optimization.

After successful stabilization in both cases, we implemented noise disturbance in joints, wind noise force air friction and cart friction and damping.

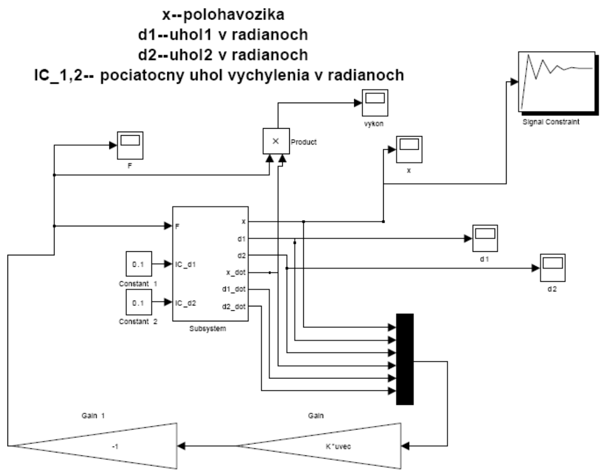

Fig. 3 Simulink LQR implementation

In (Fig.4) and (Fig.5), we can see decrease of cart travel by using optimization and changing coefficients of K.

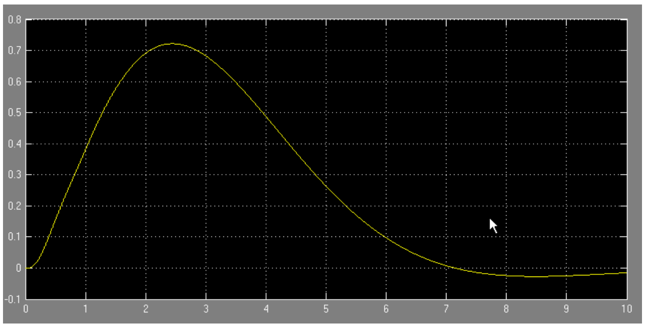

Fig. 4 cart travel; K generated from Ricatti eq.

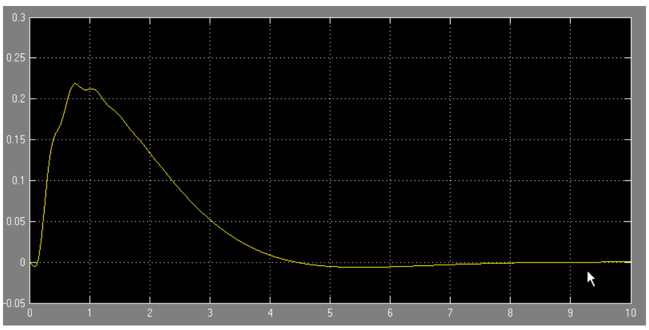

Fig. 5 cart travel; K changed using optimization

Conclusion

Symbolic computation software is already few years becoming standard tool, helping to solve engineering problems. The typical commercial products are Maple and Mathematica.

The student version of Maple was used. During writing this article we have come across the free Maxima, which we will try and possibly move all my computation to its platform.

After implementation of LQR algorithm using Simulink and SimMechanics we can conclude that SimMechanics gave exactly same results as standard block implementation in Simulink. In fact the SimMechanics building procedure was very fast and straightforward. By using SimMechanics we gained option of Animation.

On the other hand, the simulations in SimMechanics were about 10 times slower. Beside that, the equations of DIP are still need for computing the LQR K matrix. After our experience, the SimMechanics is a good choice for very complicated models, were deriving of equations could take long time, in other cases we would suggest using the Simulink or Matlab itself.

ACKNOWLEDGMENT

The authors would like to thank to Slovak Grant Agency - project VEGA 1/3078/06 "Specific aspects of the dust mass concentration measurement in working environment with via optical methods and calibration of optical methods with reference methods" and VEGA 1/0201/08 "Research of structures and behavior of the mechatronics mobile engineering system on the organ level and construction item level in order to properties improvement of the mobile engineering system".

References

[1] SPONG, MARK, W.: The Swing Up Control Problem for the Acrobot, IEEE Control Systems, pp. 49-55, February 1995

[2] STIMAC, A.K.: Standup and Stabilization of the Inverted Pendulum, MIT, master thesis, 1999

[3] NIEMANN, H., POULSEN J. K.: Analysis and Design of Controllers for a Double Inverted Pendulum, Proceedings of the American Control Conference,Denver, Colorado June 4-6, 2003

[4] ZHONG,W., ROCK H.: Energy and Passivity Based Control of the Double Inverted Pendulum on a Cart, Proceedings of the 2001 IEEE International Conference on Control Applications, September 5 - 7, 2001, México City, México.

Ing. Martin Grossman
prof. Ing. Alexander Gmiterko, PhD.

Technical University Kosice
Department of Applied Mechanics and Mechatronics
Institute of special technical science
Faculty of Mechanical Engineering
Letna 9
040 01 Kosice
E-mail: martin.grossman@gmail.com ,
alexander.gmiterko@tuke.sk