Дмитриев М.Г., Макаров Д.А.

«Методы классического и интеллектуального управления динамическими системами»

Лабораторная работа №1

Пример. Перевернутый маятник на тележке[1]

M – масса тележки, 0.5 кг;

m – масса маятника, 0.2 кг;

b – коэффициент трения тележки, 0.1 Н/м/с;

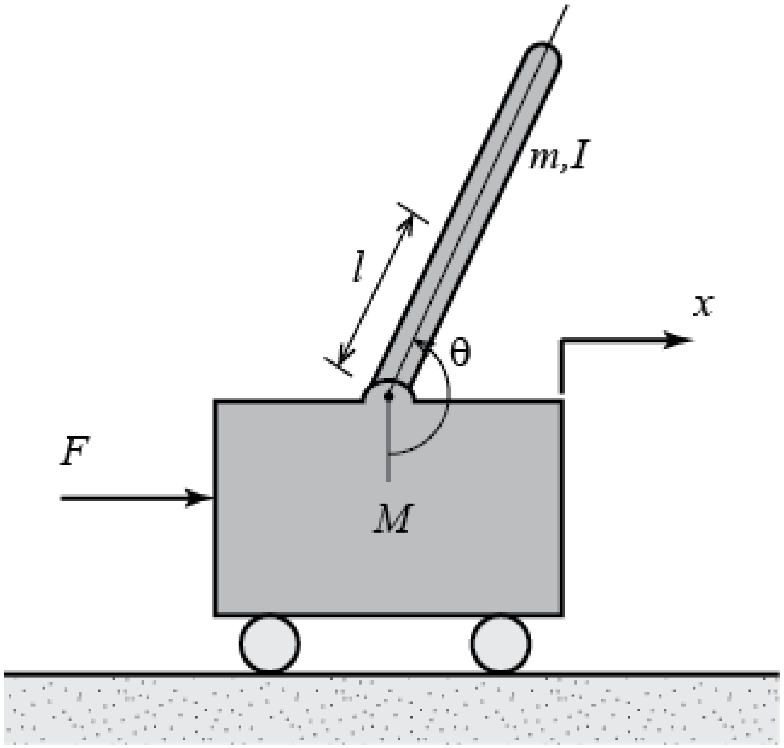
l – расстояние до центра масс маятника, 0.3 м;

I – момент инерции маятника, 0.006 kg*m^2;

F – сила, приложенная к тележке, Н;

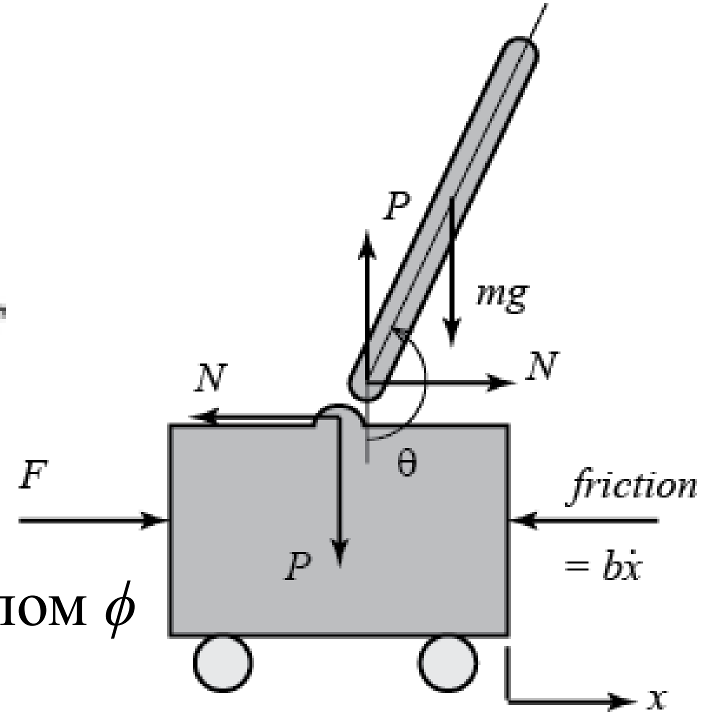
x – координата тележки, м;

θ – угол отклонения маятника от вертикальной полуоси, направленной вниз, рад.



[1] Пример и программа взяты из «Control Tutorials for MATLAB and Simulink» by Prof. Bill Messner at Carnegie Mellon and Prof. Dawn Tilbury at the University of Michigan. http://ctms.engin.umich.edu/CTMS/index.php?aux=Home

Уравнения движения



Нелинейная модель:

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F$$

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta$$

Положим, что $\theta = \pi + \phi$, где ϕ – угол отклонения от вертикали. Тогда при малом ϕ

$$\cos\theta = \cos(\pi+\phi) \approx -1$$

$$\sin\theta = \sin(\pi+\phi) \approx -\phi$$

Итак, при малой скорости $\dot{\theta}$ поучаем линеаризованные уравнения

$$(M+m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u$$

$$(I+ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x}$$

где $u = F$.

Линеаризация и выходные величины

Поскольку полученные выше уравнения являются линейными, то их можно представить в виде системы ОДУ первого порядка, которую можно представить в виде

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

Введем вектор наблюдений *y*

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Упражнения в MATLAB

- Проверка устойчивости разомкнутой системы (без управления) с помощью функции **eig.**
- Проверка управляемости системы с помощью функции **rank.**
- Синтез стабилизирующего в нуле линейно-квадратичного регулятора по состоянию с помощью функций **care** или **lqr.**
- Проверка наблюдаемости системы с помощью **rank.**
- Построение наблюдателя и стабилизирующего в нуле линейно-квадратичного динамического регулятора.

Уравнения замкнутой системы вдоль регулятора с наблюдателем полного порядка

С учетом построенного наблюдателя

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

для системы $\dot{x} = Ax + Bu, \quad y = Cx$ для ошибки наблюдения $\mathrm{e} = x - \hat{x}$ имеем

$\dot{\mathrm{e}} = \dot{x} - \dot{\hat{x}} = (Ax + Bu) - \big(A\hat{x} + Bu + L(y - C\hat{x})\big) =$ $A(x - \hat{x}) - LC(x - \hat{x}) = (A - LC)e$. Тогда уравнения ***расширенной*** замкнутой системы вдоль $u = -K\hat{x} = -K(x - e)$ имеют вид

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}$$

Определение коэффициентов матрицы наблюдателя

Необходимо, чтобы собственные значения матрицы $A - LC$ лежали в левой полуплоскости для устойчивости ошибки наблюдения. Формально, определение матриц коэффициентов усилений регулятора и наблюдателя (K и L) – две ***независимые*** задачи. Однако, *желательно*, чтобы действительные части собственных чисел матрицы $A - LC$ по модулю были значительно (на порядок) больше действительных частей собственных значений $(A - BK)$.

Команда **place(**A, B, *poles***)** находит K, при которой $A - BK$ имеет вектор собственных чисел равный заданному параметру *poles*. Т.к. собственные значения не меняются при транспонировании, то используем L^T= place(A^T, C^T, *poles*).