Demonstration: Oscillating Masses For Linear-MPC Benchmarks

İ. Ç. Yılmaz *May 30, 2021*

1 Mathematical Modeling

This demonstration consists of a sequence of six masses connected by springs to each other, and to walls on either side, as shown in Fig. 1. There are three actuators, u_1, u_2, u_3 , which exert tensions between different masses. The masses, m_1, m_2, m_3, m_4, m_5, m_6, have value 1 kg, the springs all have spring constant 1 kg/s^2, and there is no damping [4].

Continuous time system can be sampled by using a first order hold model, with a period of 0.5 (which is around 3 times faster than the period of the fastest oscillatory mode of the open-loop system). The state vector is $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ \dot{x}_1\ \dot{x}_2\ \dot{x}_3\ \dot{x}_4\ \dot{x}_5\ \dot{x}_6]^T$ where x_i is the position of mass i while $\dot{x}_i$ is the velocity of mass i on the horizontal axis. Dynamic equations for all masses with inputs of the system can be written as

$$m_1\ddot{x}_1 = u_1 - k_1x_1 + k_2(x_2 - x_1) \tag{1a}$$
$$\ddot{x}_1 = \frac{1}{m_1}u_1 - \frac{k_1+k_2}{m_1}x_1 + \frac{k_2}{m_1}x_2$$
$$m_2\ddot{x}_2 = -u_1 - k_2(x_2 - x_1) + k_3(x_3 - x_2) \tag{1b}$$
$$\ddot{x}_2 = -\frac{1}{m_2}u_1 + \frac{k_2}{m_2}x_1 - \frac{k_2+k_3}{m_2}x_2 + \frac{k_3}{m_2}x_3$$
$$m_3\ddot{x}_3 = u_2 - k_3(x_3 - x_2) + k_4(x_4 - x_3) \tag{1c}$$
$$\ddot{x}_3 = \frac{1}{m_3}u_2 + \frac{k_3}{m_3}x_2 - \frac{k_3+k_4}{m_3}x_2 + \frac{k_4}{m_3}x_4$$
$$m_4\ddot{x}_4 = u_3 - k_4(x_4 - x_3) + k_5(x_5 - x_4) \tag{1d}$$
$$\ddot{x}_4 = \frac{1}{m_4}u_3 + \frac{k_4}{m_4}x_3 - \frac{k_4+k_5}{m_4}x_4 + \frac{k_5}{m_4}x_5$$
$$m_5\ddot{x}_5 = -u_2 - k_5(x_5 - x_4) + k_6(x_6 - x_5) \tag{1e}$$
$$\ddot{x}_5 = -\frac{1}{m_5}u_2 + \frac{k_5}{m_5}x_4 - \frac{k_5+k_6}{m_5}x_4 + \frac{k_6}{m_5}x_6$$
$$m_6\ddot{x}_6 = -u_3 - k_6(x_6 - x_5) - k_7x_6 \tag{1f}$$

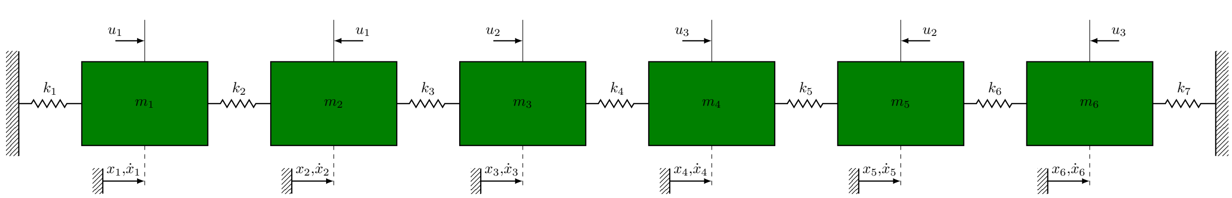

Figure 1: The model of oscillating masses.

$$\ddot{x}_6 = -\frac{1}{m_6}u_3 + \frac{k_6}{m_6}x_5 - \frac{k_6+k_7}{m_6}x_6$$

The state space representation of the oscillation interconnected masses can be written in matrix form as follows:

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_1+k_2}{m_1} & \frac{k_2}{m_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k_2}{m_2} & -\frac{k_2+k_3}{m_2} & \frac{k_3}{m_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{k_3}{m_3} & -\frac{k_3+k_4}{m_3} & \frac{k_4}{m_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_4}{m_4} & -\frac{k_4+k_5}{m_4} & \frac{k_5}{m_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{k_5}{m_5} & -\frac{k_5+k_6}{m_5} & \frac{k_6}{m_5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{k_6}{m_6} & -\frac{k_6+k_7}{m_6} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{A}_c} \boldsymbol{x}(t)$$

$$+ \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & 0 \\ -\frac{1}{m_2} & 0 & 0 \\ 0 & \frac{1}{m_3} & 0 \\ 0 & 0 & \frac{1}{m_4} \\ 0 & -\frac{1}{m_5} & 0 \\ 0 & 0 & \frac{1}{m_6} \end{bmatrix}}_{\boldsymbol{B}_c} \boldsymbol{u}(t) \tag{2}$$

All dynamic equations are in the form of $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{B}_c u(t)$. If the state equations $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{B}_c u(t)$ are rearranged, and all terms pre-multiplied by the square matrix $e^{-\boldsymbol{A}_c t}$:

$$e^{-\boldsymbol{A}_c t}\dot{\boldsymbol{x}}(t) - e^{-\boldsymbol{A}_c t}\boldsymbol{A}\boldsymbol{x}(t) = \frac{d}{dt}\left(e^{-\boldsymbol{A}_c t}\boldsymbol{x}(t)\right) = e^{-\boldsymbol{A}_c t}\boldsymbol{B}_c u(t) \tag{3}$$

Integration of Eq. 3 gives

$$\int_0^t \frac{d}{d\tau}\left(e^{-\boldsymbol{A}_c \tau}\boldsymbol{x}(\tau)\right)d\tau = \int_0^t e^{-\boldsymbol{A}_c t}\boldsymbol{B}_c u(t) = e^{-\boldsymbol{A}_c t}\boldsymbol{x}(t) - e^{-\boldsymbol{A}_c 0}\boldsymbol{x}(0) = \int_0^t e^{-\boldsymbol{A}_c \tau}\boldsymbol{B}_c u(\tau)d\tau \tag{4}$$

where $e^{-\boldsymbol{A}_c 0} = \boldsymbol{I}$ is 12×12 identity matrix and $[e^{-\boldsymbol{A}_c t}]^{-1} = e^{\boldsymbol{A}t}$ the complete state vector response may be written in two similar forms

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}_c t}\boldsymbol{x}(0) + e^{\boldsymbol{A}_c t}\int_0^t e^{-\boldsymbol{A}_c \tau}\boldsymbol{B}_c u(\tau)d\tau \tag{5a}$$

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}_c t}\boldsymbol{x}(0) + \int_0^t e^{\boldsymbol{A}_c (t-\tau)}\boldsymbol{B}_c u(\tau)d\tau \tag{5b}$$

A continuous time signal $\{x(t)\}$ can be obtained from a discrete time (DT) signal $x[k]$, by holding the value of the DT signal constant for one sampling period T or Δt, such that: This is known as the zero-order hold. The discrete-time equivalent of continuous time equation is of the form:

$$\boldsymbol{x}[k+1] = \boldsymbol{A_d}\boldsymbol{x}[k] + \boldsymbol{B_d}u[k] \tag{6}$$

Starting from the solution of the continuous state-space equation 5, that the corresponding discrete matrices are obtained as:

$$\boldsymbol{A_d} = e^{\boldsymbol{A}_c T} \quad \text{and} \quad \boldsymbol{B_d} = \left(\int_0^T e^{\boldsymbol{A}_c \tau} d\tau\right)\boldsymbol{B}_c = \boldsymbol{A}_c^{-1}(e^{\boldsymbol{A}_c T} - \boldsymbol{I})\boldsymbol{B}_c \tag{7}$$

It should be note that that this is the *exact* solution to the differential equation, there are no discretization errors. While it is exact, information is still lost by the discretization: the inter-sample behavior.

```
k = 1;              % spring constant
lam = 0;            % damping constant
a = -2*k;
b = -2*lam;
c = k;
d = lam;

n = 12; % state dimension
m = 3; % input dimension

Ac = [zeros(n/2) eye(n/2);
      [a,c,0,0,0,0,b,d,0,0,0,0;
      c,a,c,0,0,0,d,b,d,0,0,0;
      0,c,a,c,0,0,0,d,b,d,0,0;
      0,0,c,a,c,0,0,0,d,b,d,0;
      0,0,0,c,a,c,0,0,0,d,b,d;
      0,0,0,0,c,a,0,0,0,0,d,b]];

Bc = [zeros(n/2,m);
       [1,  0,  0;
       -1,  0,  0;
        0,  1,  0;
```

```
        0,  0,  1;
        0,-1,  0;
        0,  0,-1]];

% convert to discrete-time system
ts = 0.5;        % sampling time
A = expm(ts*Ac);
B = (Ac\(expm(ts*Ac)-eye(n)))*Bc;
```

2 MPC Formulation and Some Solvers In the Literature

In order to evaluate the performance of the solver for various problem sizes, the following MPC problem is formulated for a chain of masses interconnected with spring-dampers [1]:

$$
\begin{aligned}
\min_{\boldsymbol{U}} V_n(\boldsymbol{x},\boldsymbol{U}) := \sum_{k=k_0}^{k_0+N-1} & \boldsymbol{x}_k^T \boldsymbol{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k + V_f(\boldsymbol{x}_{k_0+N}) \\
\text{s.t.} \quad & \boldsymbol{x}_{k_0} = \boldsymbol{x}(k_0), \\
& \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k && k_0 \le k < k_0+N, \\
& -4 \cdot \mathbf{1}_{n_x} \le \boldsymbol{x}_k \le 4 \cdot \mathbf{1}_{n_x} && k_0 \le k \le k_0+N, \\
& -0.5 \cdot \mathbf{1}_{n_u} \le \boldsymbol{u}_k \le 0.5 \cdot \mathbf{1}_{n_u} && k_0 \le k < k_0+N.
\end{aligned}
\tag{8}
$$

with $V_f(k_0+N) := \boldsymbol{x}_{k_0+N}^T \boldsymbol{Q} \boldsymbol{x}_{k_0+N}$. Note that this formulation does not provide stability guarantees since no terminal constraint is included.

For this optimization problem, dozen of academic works have been done. Wang and Body utilizes the above optimization problem via block-wise Cholesky factorization [4]. In addition to the method that have been employed in [4], rank one modifications in the computation is defined in the [1]. Solution time of the problem, especially in the large horizon size and state variables size is improved.

Frison *et. al.*, [3], selects linear chain of masses springs problem by using Dual-Newton strategy for QP problem. In this study main aim is to show demonstration how the linear algebra provided in BLASFEO (Basic Linear Algebra Subroutines for Emdedded Optimization) can enhance the performance of new or existing embedded optimization tools. The results indicate that using BLASFEO throughout the code can lead to significant performance gains. With BLASFEO Library, a HPIPM (High Performance Interior Point Method) framework have been developed for linear model predictive control. Same problem has also been employed in this study [2]. Three types of QP are taken into consideration and some implementation choices, |**speed_abs**|**speed**|**balance**| **robust**|, are given as mode in the solver.

References

[1] Alexander Domahidi, Aldo U. Zgraggen, Melanie N. Zeilinger, Manfred Morari, and Colin N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pages 668–674, 2012. doi:10.1109/CDC.2012.6426855.

[2] Gianluca Frison and Moritz Diehl. Hpipm: a high-performance quadratic programming framework for model predictive control. IFAC-PapersOnLine, 53(2):6563–6569, 2020. 21th IFAC World Congress. doi:https://doi.org/10.1016/j.ifacol.2020.12.073.

[3] Gianluca Frison, Dimitris Kouzoupis, Tommaso Sartor, Andrea Zanelli, and Moritz Diehl. Blasfeo: Basic linear algebra subroutines for embedded optimization. ACM Trans. Math. Softw., 44(4), July 2018. doi:10.1145/3210754.

[4] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. IEEE Transactions on Control Systems Technology, 18(2):267–278, 2010. doi:10.1109/TCST.2009.2017934.