DINCON 2017
CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES
30 de outubro a 01 de novembro de 2017 – São José do Rio Preto SP

Embedded Model Predictive Control of a 2DOF Helicopter

José Genário O. Júnior[1]
Depto. Engenharia Elétrica e Telecomunicações, Escola Politécnica da USP, São Paulo, SP
Fernando dos Santos Barbosa[2]
Depto. Engenharia Elétrica e Telecomunicações, Escola Politécnica da USP, São Paulo, SP
Bruno Augusto Angélico[3]
Depto. Engenharia Elétrica e Telecomunicações, Escola Politécnica da USP, São Paulo, SP
Claudio Garcia[4]
Depto. Engenharia Elétrica e Telecomunicações, Escola Politécnica da USP, São Paulo, SP

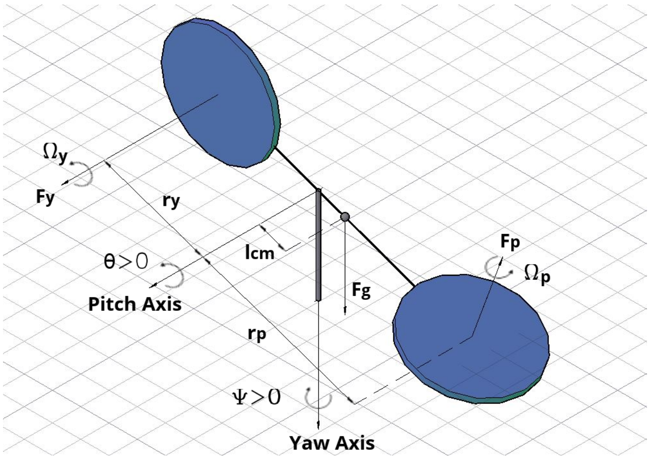
Abstract. It is presented the implementation of a Discrete Finite Horizon Model Predictive Control (MPC) into a 2DOF Helicopter prototype, which makes use of incremental state-space model of the helicopter. The resultant controller is completely embedded in a NXP™ FRDM-K64F development platform. Satisfactory reference tracking capability is presented with experimental results.

Keywords. Embedded Control, MPC, 2-DOF Helicopter.

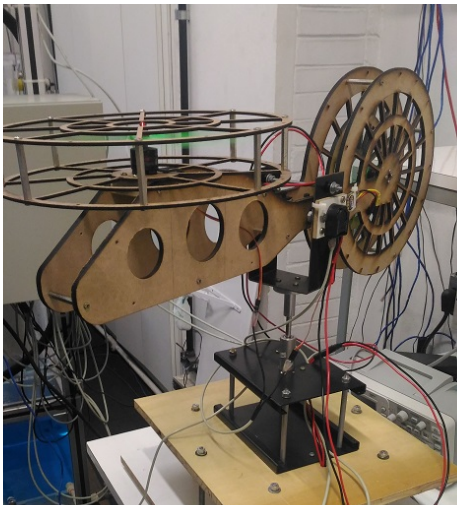
1 Introduction

Currently, Model Predictive Control is the default choice of advanced control in Process Industry. The main reason for this is that it can handle several safety and equipment constraints. In many cases, operation near these constraints is necessary in order to achieve an efficient and profitable operation [1]. However, due to control signal calculation complexity, embedded applications are often scarce and time-consuming. This computational complexity arises from the fact that it solves a quadratic problem every sampling instant and performs high order matrix operations. Thus, since many fast mechanical plants and robotics applications are usually embedded in a microcontroller, it's hard to implement MPC controllers in these systems. This problem is not present in Industrial plants, for example, because of often high sampling times and the wide availability of PCs and PLCs.

[1]genariojr@usp.br
[2]barbosa.fs@usp.br
[3]angelico@lac.usp.br
[4]clgarcia.usp.br



(a) Free-body diagram.



(b) 2DOF Helicopter built.

Figure 1: Free-body diagram and picture of the process

Here is presented the modeling the process used, a 2-DOF Helicopter, then a preliminary on incremental state space models, needed for zero steady-state error, the formulation of the MPC problem and the experimental results. Similar non-embedded work was developed by [2].

2 2DOF Helicopter Mathematical Model

The two-degree-of-freedom helicopter (2DOF) is a nonlinear, MIMO (multiple input, multiple output) system that presents two inputs and two outputs with input-output coupling. A simple free-body diagram is presented in Figure 1a. The conventions adopted in the modeling were such as presented in [3], which results in the following nonlinear equations of motion:

$$\ddot{\theta} = \frac{K_{pp}\, Vm_p + K_{py}\, Vm_y - \left(B_p\dot{\theta} + m_{heli} l_{cm}^2 \dot{\psi}^2 \sin(\theta)\cos(\theta) + m_{heli} g \cos(\theta) l_{cm}\right)}{J_{eq\_p} + m_{heli} l_{cm}^2} \tag{1}$$

$$\ddot{\psi} = \frac{K_{yy}\, Vm_y + K_{yp}\, Vm_p + 2 m_{heli} l_{cm}^2 \sin(\theta)\cos(\theta)\dot{\psi}\dot{\theta} - B_y\dot{\psi}}{J_{eq\_y} + m_{heli}\cos(\theta)^2 l_{cm}^2}, \tag{2}$$

with θ and ψ being the pitch and yaw angles, m_{heli} the helicopter total moving mass, g gravity, l_{cm} the center of mass distance to origin, B_p and B_y the movement resistance acting on respectively pitch and yaw axes, J_{eq_p} and J_{eq_y} are the equivalent moment of inertia related to pitch (J_{eq_p}) and yaw (J_{eq_y}) axes. Vm is the motor signal percentage, K_{pp} is the pitch motor thrust constant, K_{py} is the yaw motor torque constant, K_{yp} is the pitch motor torque constant, K_{yy} is the yaw motor thrust constant.

Table 1: Helicopter Parameters

Parameter [Unit]	Value	Parameter [Unit]	Value
m_{heli} $[kg]$	1.317	l_p $[m]$	0.25
l_{cm} $[m]$	0.038	l_y $[m]$	0.20
K_{pp} $[N/(rad/s)^2]$	$9.013 \cdot 10^{-6}$	B_p $[N/\%]$	0.1
K_{yy} $[N/(rad/s)^2]$	$-5.668 \cdot 10^{-6}$	B_y $[N/\%]$	0.1
K_{py} $[N.m/(rad/s)^2]$	$-1.663 \cdot 10^{-7}$	J_{eq_p} $[kg.m^2]$	0.384
K_{yp} $[N.m/(rad/s)^2]$	$1.489 \cdot 10^{-7}$	J_{eq_q} $[kg.m^2]$	0.0432

The torque produced by each motor is proportional to the motor squared speed (Ω_p^2 and Ω_y^2), and those terms are related to the motors signal percentage by a linear approximation.

Linearizing the system for small angles, around the operating point, the equations of motion are defined as

$$\ddot{\theta} = \frac{K_{pp}Vm_p + K_{py}Vm_y - B_p\dot{\theta}}{J_{eq\_p} + m_{heli}l_{cm}^2} \tag{3}$$

$$\ddot{\psi} = \frac{K_{yy}Vm_y + K_{yp}Vm_p - B_y\dot{\psi}}{J_{eq\_y} + m_{heli}l_{cm}^2}. \tag{4}$$

Based on these equations, the general space-state representation for the case when the system model does not have a direct feedthrough can be written, resulting in:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{5}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6}$$

where $\mathbf{x}(t)$ corresponds to the system states and $\mathbf{u}(t)$ the motors percentage inputs, defined by $\mathbf{x} = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^T$ and $\mathbf{u} = \begin{bmatrix}Vm_p & Vm_y\end{bmatrix}^T$.

The matrices $\mathbf{A}$ and $\mathbf{B}$ of the continuous-time state-space model are written as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-B_p}{J_{eq\_p}+m_{heli}l_{cm}^2} & 0 \\ 0 & 0 & 0 & \frac{-B_y}{J_{eq\_y}+m_{heli}l_{cm}^2} \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{eq\_p}+m_{heli}l_{cm}^2} & \frac{K_{py}}{J_{eq\_p}+m_{heli}l_{cm}^2} \\ \frac{K_{yp}}{J_{eq\_y}+m_{heli}l_{cm}^2} & \frac{K_{yy}}{J_{eq\_y}+m_{heli}l_{cm}^2} \end{bmatrix}. \tag{7}$$

Figure 1b shows the prototype used to implement the control technique proposed in this work. As actuators, it uses two brushless DC motors with 9-inch propellers and, as sensors, two rotary encoders.

The parameters of the system are presented on Table 1, which some were obtained by linearizing thrust and torque equations (obtained with the use of a hall sensor and a scale) [3] and others by simple system identification techniques.

3 Discrete Finite Horizon MPC

In this work, the controller considered is the Discrete Finite Horizon MPC. This means that it will use discrete-time models, as well as having the prediction horizon to be finite.

There is another popular formulation for stable plants called Infinite Horizon MPC, in which the prediction horizon p is treated as infinity.

3.1 Incremental State Space Model

The implementation of the MPC controller on the discrete version of the state space presented in equations 5 and 6 does not eliminate steady-state errors. To solve that, the incremental formulation with new states $\mathbf{x_i}(k) = \begin{bmatrix}\Delta x(k) & y(k)\end{bmatrix}^T$ and control signal $\mathbf{u_i}(k) = \Delta u(k)$ [4] is used and has the following discrete state space matrices:

$$\mathbf{A_i} = \begin{bmatrix} \mathbf{A} & \mathbf{0_{nx,ny}} \\ \mathbf{C} * \mathbf{A} & \mathbf{I_{ny}} \end{bmatrix} \quad \mathbf{B_i} = \begin{bmatrix} \mathbf{B} \\ \mathbf{C} * \mathbf{B} \end{bmatrix} \quad \mathbf{C_i} = \begin{bmatrix} \mathbf{0_{nx,ny}} & \mathbf{I_{ny}} \end{bmatrix},$$

in which **ny,nx** represent, respectively, the number of outputs and the state dimension.

3.2 MPC Formulation

Given a discretized process described by an incremental form:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \tag{8}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \tag{9}$$

find the control sequences $\Delta\mathbf{u}$(k—k),$\Delta\mathbf{u}$(k+1—k),...,$\Delta\mathbf{u}$(k+m-1—k) that minimizes the cost function below, implement the first result, then repeat at the next sampling instant:

$$J_k = \sum_{j=1}^{p}(y(k+j|k - y^{sp})^T Q(y(k+j|k - y^{sp}) + \sum_{j=0}^{m-1} \Delta u(k+j|k)^T R \Delta u(k+j|k), \tag{10}$$

where $\mathbf{p}$ is the prediction horizon, $\mathbf{m}$ is the control horizon, $\mathbf{y^{sp}}$ is the desired output set-point and $\mathbf{Q}$ and $\mathbf{R}$ are output and control signal weighting matrices.

By using the state update equations (8 and 9), one can write the following relationships:

$$y(k+1|k) = CAx(k) + CB\Delta u(k|k) \tag{11}$$

$$y(k+2|k) = CA^2x(k) + CAB\Delta u(k|k) + CB\Delta u(k+1|k) \tag{12}$$

$$\vdots$$

$$y(k+j|k) = CA^jx(k) + CA^{j-1}B\Delta u(k|k) + CA^{j-2}B\Delta u(k+1|k) + \cdots + CB\Delta u(k+j|k). \tag{13}$$

Also, it is considered that the control signal is constant after the control horizon $\mathbf{m}$, i.e.

$$\Delta u(k+m|k) = \Delta u(k+m+1|k) = 0. \tag{14}$$

With that being defined, the output prediction vector can be written as:

$$\bar{y}(k) = \Psi x(k) + \bar{\Theta}\Delta u(k), \tag{15}$$

with

$$\bar{y}(k) = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+m|k) \\ y(k+m+1|k) \\ \vdots \\ y(k+p|k) \end{bmatrix}, \quad \Delta u(k) = \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+m-1|k) \end{bmatrix}, \tag{16}$$

$$\Psi = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^m \\ CA^{m+1} \\ \vdots \\ CA^p \end{bmatrix} \quad \text{and} \quad \bar{\Theta} = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m-1}B & CA^{m-2}B & \dots & CB \\ CA^m B & CA^{m-1}B & \dots & CAB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \dots & CA^{p-m}B \end{bmatrix}. \tag{17}$$

Defining the set-point vector as

$$\bar{y}^{sp} = \begin{bmatrix} y^{sp} & \dots & y^{sp} \end{bmatrix}^T \tag{18}$$

and the weighting matrices as

$$\bar{Q} = diag\begin{bmatrix} Q & \dots & Q \end{bmatrix}^T \quad \text{and} \quad \bar{R} = diag\begin{bmatrix} R & \dots & R \end{bmatrix}^T, \tag{19}$$

then the cost function 10 can be rewritten as

$$J_k = \Delta u(k)^T H \Delta u(k) + 2c_f \Delta u(k) + \bar{c}, \tag{20}$$

where

$$H = \bar{\Theta}^T \bar{Q} \bar{\Theta} + \bar{R} \tag{21}$$

$$c_f^T = (\Psi x(k) - \bar{y}^{sp})^T \bar{Q} \bar{\Theta} \tag{22}$$

$$\bar{c} = (\Psi x(k) - \bar{y}^{sp})^T \bar{Q} (\Psi x(k) - \bar{y}^{sp}). \tag{23}$$

The control law of the Incremental Finite Horizon MPC is then given by the solution of the following Quadratic Programming Problem, where only the first result is implemented:

$$\begin{aligned} \underset{\Delta u_k}{\text{minimize}} \quad & \Delta u_k^T H \Delta u_k + 2c_f^T \Delta u_k \\ \text{subject to} \quad & u_{min} \leq u(k+j|k) \leq u_{max}, \quad j = 0, 1, \dots, m-1, \\ & -\Delta u_{min} \leq \Delta u(k+j|k) \leq \Delta u_{max}, \quad j = 0, 1, \dots, m-1. \end{aligned} \tag{24}$$

4 Embedded Implementation

The MPC controller presented in the previous section was implemented using C++ in a FRDM-K64F board. The code was optimized to a point in which the system is controlled on a 20Hz frequency with the following design parameters: $p = 50$, $m = 5$, $Q = 1.2I_2$ and $R = I_2$.

The choice of the limitations on the control signals were made to cope with the physical limitations of the plant and actuators. Since the inputs are voltage percentages, they simply must be between $0 - 100\%$. The slew rate of the inputs were constrained to a maximum of 5% per sample. The QP problem solution were given by an interior-point solver with a constraint on the maximum number of iterations in order to meet real-time specifications. Also, the states were calculated using a derivative filter.

An outline of the control algorithm can be written as: for every sample, do

1. Read plant outputs;
2. Estimate the states;
3. Calculate the QP Problem Matrices;
4. Solve the QP;
5. Implement first control signal result as u(k) = u(k-1) + Δu(k).

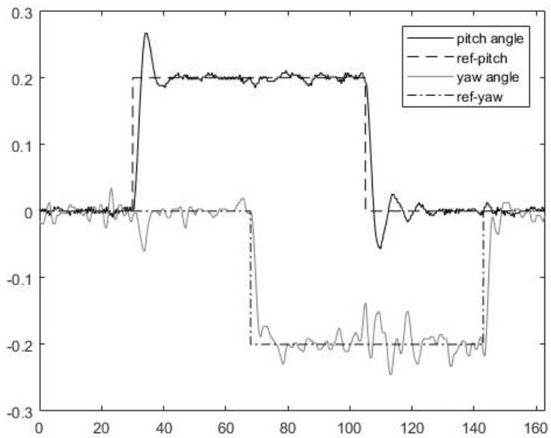
5 Experimental Results

The closed loop response of a two-step test on each output is shown in Figure 2a. The specifications on the system's rise time was about 4 seconds with a maximum overshoot of 20% for both outputs. While the closed loop rise-time was faster, close to 2.8s, the maximum deviation from the reference was higher, about 30%. This can be adjusted by fine-tuning the parameters until the closed loop response meets the specifications. Another approach that was not implemented due to the proof of concept mindset of this work is to design a pre-filter for the current tuning in order to lower the overshoot to meet the specifications. This would increase the rising time, but since the current tuning is faster than the specifications, that would not be a problem.

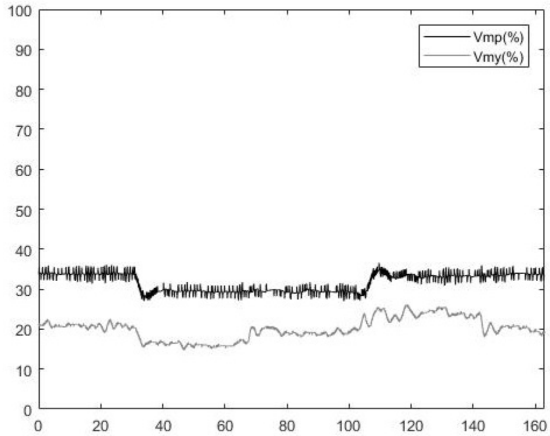
The chosen MPC parameters were found by fine tuning the process starting with the values found on simulations. It is important to note that there is a model mismatch between the theoretical and the real one. This limits the achievable control performance in model-based control schemes. While the control signal was not even close to the control saturations, quantization in the angle sensors causes a chattering in the control signal. This can be solved by filtering the output with a second order filter as indicated in [5] before estimating the output derivatives.

6 Conclusion

Since the 2000's there is an increasing utilization of MPC to control plants outside the Process Industry. This is due to the improvement in the performance of micro-controllers.



(a) Set-Point Tracking



(b) Control Signal

Figure 2: Tracking and control signal plots

This work presents an application of an Embedded MPC to a relatively fast plant. It was found that while improving the solver made it possible to increase the sampling frequency, the control performance did not increase much due to model mismatch. Another interesting approach would be to use an identified model for control. However, as the model order gets higher, the QP problem becomes harder to solve due to more decision variables.

References

[1] J. Maciejowski, *Predictive Control with Constraints.* Prentice Hall, 2002.

[2] R. Raghavan and S. Thomas, "Practically implementable model predictive controller for a twin rotor multi-input multi-output system," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 3, pp. 358–370, June 2017.

[3] G. G. Neto, F. dos Santos Barbosa, and B. A. Angélico, "2-DOF helicopter controlling by pole-placements," in *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, Nov 2016, pp. 1–5.

[4] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB.* Springer Science & Business Media, 2009.

[5] K. J. Astrom and T. Hagglund, *Advanced PID Control.* ISA-The Instrumentation, Systems, and Automation Society, 2006.