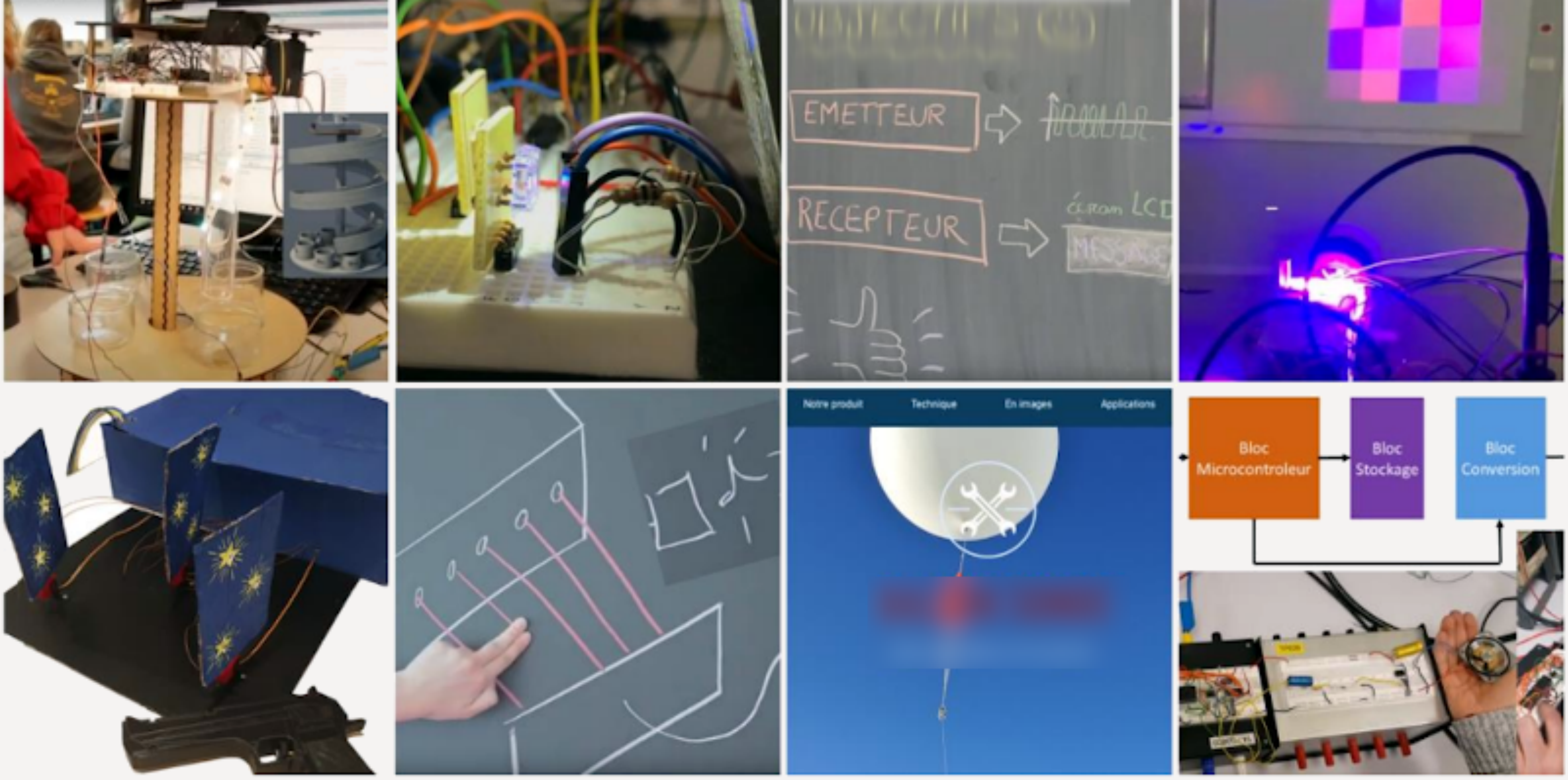

Laser Tracking - Embedded PID Controller

Technical Documentation

Julien VILLEMEJANE
version 2021

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech
LEnsE
Laboratoire d'Enseignement Expérimental

The **embedded PID controller system** presented in this document was designed and realized at the LEnsE lab of INSTITUT D'OPTIQUE GRADUATE SCHOOL / FRANCE by **Cyrille DES COGNETS, Théo MARTIN and Igor RESHETNIKOV** (MSc 1 students) and supervised by **Caroline KULCSAR and Julien VILLEMEJANE**, with the precious help of Thierry AVIGNON and Cédric LEJEUNE.

It was designed for a previous existing didactic demonstrator of a 2D tracking of a laser beam, developped by **Caroline KULCSAR, Fabienne BERNARD and Thierry AVIGNON**.

LENSE.INSTITUTOPTIQUE.FR

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Contents

2D Tracking of a Laser Beam

1. Presentation of the initial system

2. Mathematical model of the loop

2.1 Second order model

Model of the system

Estimation of the parameters

2.2 Simulink® simulation

Some simulations in open loop and with or without PID controller

3. From Analog to Digital PID control

In continous time

Move to digital world : adding ADC and DAC, numerical controller, sampling frequency...

II Embedded PID Controller

4. Introduction and goals

The analog version of the PID controller was sufficient to explore the basics of closed-loop system and all the stability issues it can appear when you want to control a system.

But one of the **limitation** of this controller was the fact that PID coefficients values injected in the controller are not displayed. As a system engineer, it's necessary to compare the prediction of a simulated model and the real experiment.

Additionally, the controller model is **limited to PID**. It's not possible to compare to new kind of controllers based on the state of the system or/and predictive models as Linear Quadratic Gaussian control.

A way to answer to the two problems mentionned here is to use a digital controller where you can adapt "on-the-fly" the running code.

4.1 Test

Bla bla bla

4.2 Materials

A **L476RG Nucleo board** from *STMicroelectronics* is used to perform the embedded calculations and data acquisition on the two channels (X and Y axis). The program is developped with **MBED OS** (version 2) and the **mbed-dsp** library (Digital Signal Processing specific functions are available in this library).

An **electronic board** was designed to adapt the voltages from the detector to the analog-to-digital converter and from the digital-to-analog converter to the motor controller.

An interface to specify the coefficients values was developped with MATLAB® Application Designer[1]. The coefficients are sended from the computer to the embedded board by a RS232 protocol that is virtualized in an USB connexion.

4.3 Voltage adaptation board

[1]This application was tested on Windows, but should also work on Linux and Mac. But a MATLAB® licence is required. The application was developped with the R2020b version of MATLAB®.

5. Differents modes of operation

5.1 Controlling a system step-by-step

Connexion

- Alignment
- Motor control
- Step Response (or frequency response) - Open Loop model
- PID Control

6. Communication protocol

To transmit data from the embedded controller to the computer and in the other direction, a **RS232 (serial) transmission** is used. A virtual communication port is established inside an USB connexion between the board and the computer.

A high-level protocol was established to run the embedded system in the differents working mode specified in the previous chapter. This chapter explains the protocol used between the board - i.e. the embedded controler - and the host system - i.e. the computer.

6.1 Basic Communication Mode

The embedded PID controller uses a **textual data packet** based protocol to communicate with the host system. A communication **session is always initialized by the host system**. No data are sended by the board after powering up.

A data packet sended by the host system is called a **Request**. Once the board receives a request, it will reply the host system with data packets called **Responses** (see the figure 6.1).

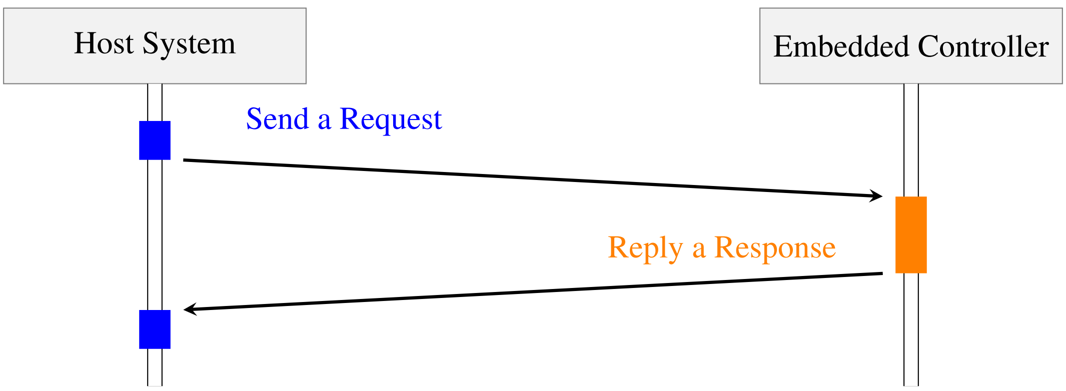

Figure 6.1: Embedded Controller communication : Request/Simple Response Mode

R All the packets must end by a CR/LF terminator ("carriage return" and "linefeed").

6.2 Connexion and Stop Mode

When the serial connexion is setup by the host system, a stop request is sended from the host to the board. The board doesn't send any response in this case.

The stop request packet is :

O_!\r\n

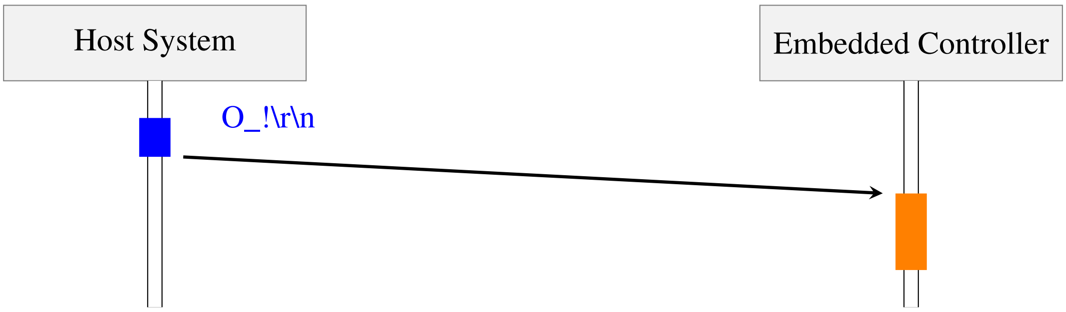

Figure 6.2: Embedded Controller communication : Stop Mode

6.3 Alignment Mode

The alignment mode permits to align the optical banch, including laser system, motors and detectors. In this mode, the two motors (X and Y axis) are positionned in the center of the functionnal range. The board collects **datas from the two detectors** and sends them to the host system **ten times per second**.

The alignment mode request packet is :

A_!\r\n

The alignment mode response packet is :

A_Xvalue_Yvalue_!\r\n

where *Xvalue* is the adapted voltage of the detector on X axis, i.e. the position of the beam on the X axis, and *Yvalue* is the adapted voltage of the detector on Y axis, i.e. the position of the beam on the Y axis.

Xvalue and *Yvalue* are two real numbers (*double* type in C).

Response packets are sending ten times per second by the board, until a stop request is sended.

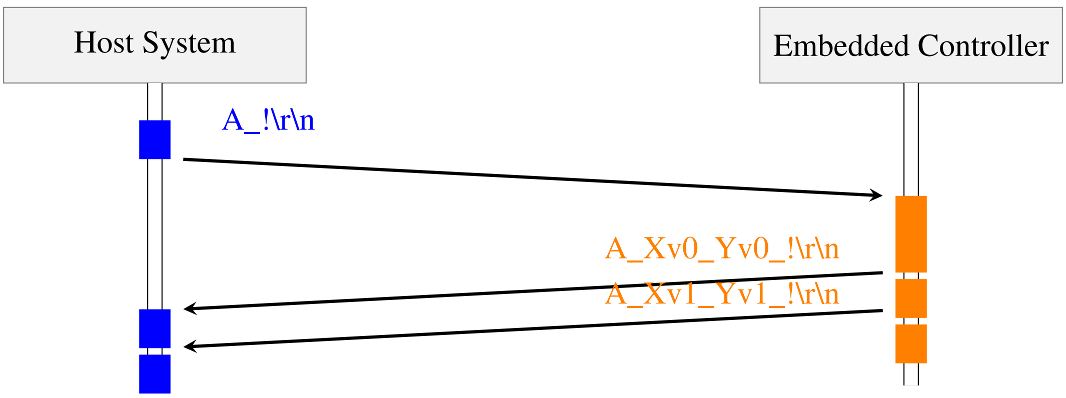

Figure 6.3: Embedded Controller communication : Alignment Mode

6.4 Motor Control Mode

The motor mode permits to **control the two axis motors independently**, in the whole range of operation, to determine the maximum deviation allowed to the beam to stay on the detector. The desired position is sended by the host system by a request packet. The board collects **datas from the two detectors** and sends them to the host system **three times per second**.

The motor mode request packet is :

M_Xpos_Ypos_!\r\n

where *Xpos* and *Ypos* is the desired position given as a pourcentage of the whole possible deviation. *Xpos* and *Ypos* are two real numbers (*double* type in C).

The motor mode response packet is :

M_Xpos_Ypos_Xdet_Ydet_!\r\n

where *Xpos* and *Ypos* are the position of the motors given as a pourcentage of the whole possible deviation, *Xdet* is the adapted voltage of the detector on X axis, i.e. the position of the beam on the X axis, and *Ydet* is the adapted voltage of the detector on Y axis, i.e. the position of the beam on the Y axis.
Xpos, *YposXvalue*, *Yvalue* are real numbers (*double* type in C).
Response packets are sending **three times per second** by the board, until a stop request is sended.

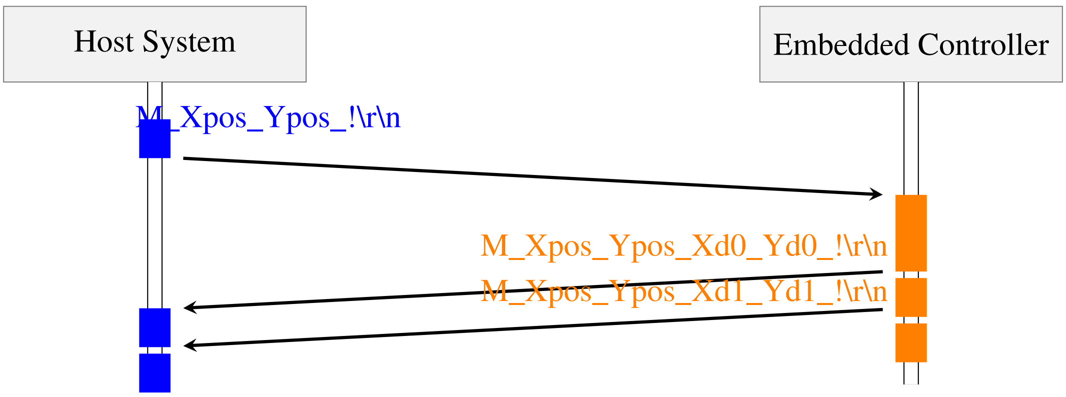

Figure 6.4: Embedded Controller communication : Motor Mode

6.5 Open-Loop Mode

6.6 PID Controller Mode

7. Embedded program in a microcontroller

7.1 Overview

A **L476RG Nucleo Board**, including a *STMicroelectronics* STM32 ARM core, is used to :

- perform the real-time PID control of the system ;
- transfer data from and to the host system (computer).

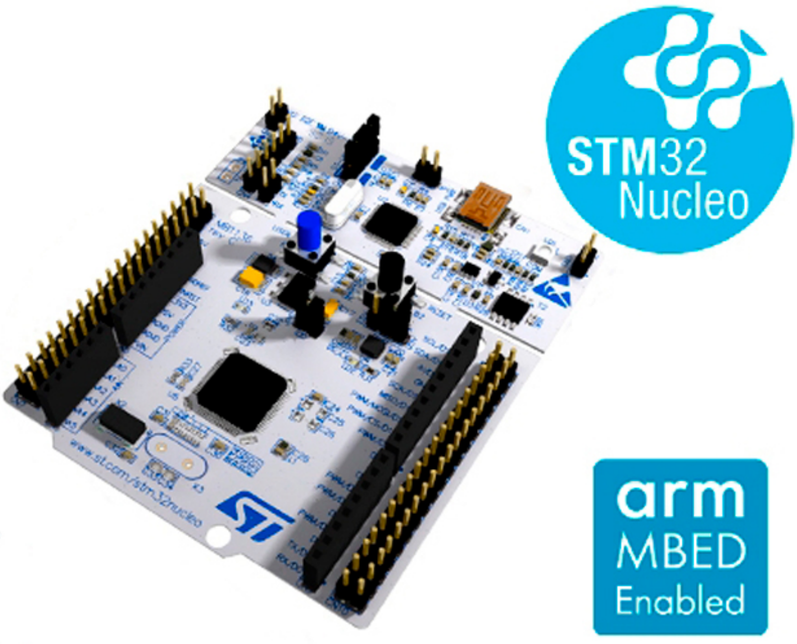

Figure 7.1: Nucleo Board, STMicroelectronics

The program is written for the **MBED OS 2 compiler**.

7.2 RS232 Communication and protocol execution

7.3 Real-time operations

8. MATLAB® Application