Part I

Preliminaries

Chapter I-1

Hardware

I-1.1 General Description

The overall motor control system comprises two main boards, as shown in Fig. I-1.1:

1. the K-CSP (Kylowave Control System Platform) is the board containing the motor components (motor, gear, speed sensor, position sensor, shaft), and most of the other complementary components, such as the flywheel and the supporting electronics.
2. the K-ECS (Kylowave Electronic Control System), is the "brains" of the whole system and supports it with the necessary computational power.

Each board is composed of a number of integrated circuits which can be monitored and controlled through several digital and analog I/O pins, as shown in Fig. I-1.2.

The K-CSP contains the mechanical structure of the motor system along with some basic circuits to run it. It comes with 7 sets of I/O pins:

1. Digital control pins
2. DC motor pins
3. Smart load pins
4. DC generator pins
5. Analog sensors pins
6. Current sensors pins
7. Reserved pins

The pin out of each set is shown in Fig. I-1.2.

The K-ECS is a fully-fledged electronic board with the necessary power electronics to run a large variety of systems, including the K-CSP in open- and closed-loop modes. It comes with 7 sets of I/O pins (most of which are not needed to run the motor, as can be seen from Fig. I-1.2):

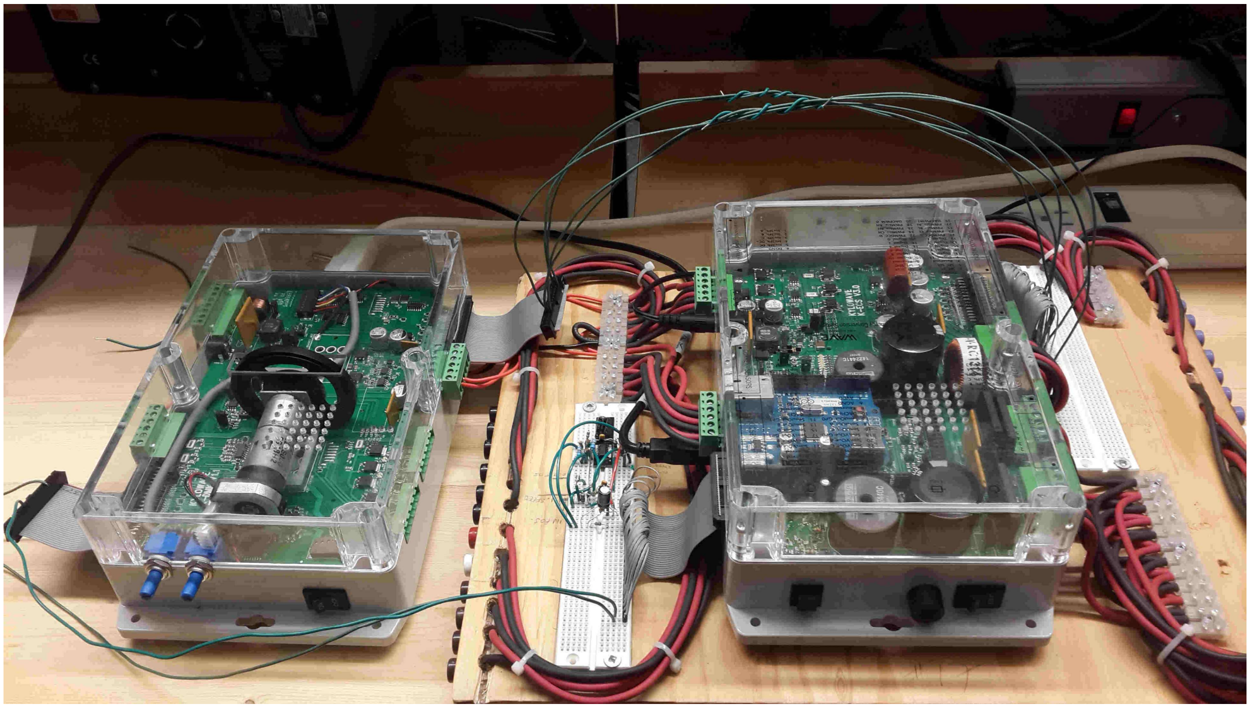


Figure I-1.1: The K-CSP (left), and the K-ECS board (right)

1. Digital control pins
2. DCDC converter pins
3. 3-phase rectifier pins
4. 1-phase rectifier pins
5. Analog sensors pins
6. Current sensors pins
7. 3-phase inverter pins

I-1.2 Hardware Connection

Connecting both kits as shown in Fig. I-1.2 implements the control system depicted in Figs. I-1.3 and I-1.4. To save time, most of these connections are already made for you through a bundle of 3 components:

1. A digital interface board, to connect the digital connectors on the K-ECS and the K-CSP.
2. An analog interface board, to connect the analog connectors on the K-ECS and the K-CSP.
3. 4 pre-wired power connectors, to connect the power/ground signals between the K-ECS and the K-CSP.

I-1.2.1 Connecting Everything Together

To connect all the components together, follow the procedure below:

1. Stack the K-CSP on top of the K-ECS such that their power switches are facing you (Fig. I-1.5(a)).
2. Connect the 4 power connectors, as shown on Fig. Do NOT turn on the power until all the components are connected (Fig. I-1.5(b)).
3. Gently, connect the "Analog Sensors" pins of the K-ECS and the K-CSP using the analog interface board (Fig. I-1.5(b)). Make sure the card is upright. Do NOT force it through. Doing so may damage the pins.
4. Gently, connect the "Digital Control" pins of the K-ECS and the K-CSP using the digital interface board (Fig. I-1.5(c)). Make sure the card is upright. Do NOT force it through.
5. Connect the K-ECS and the computer with the provided USB cable (Fig. I-1.5(b)).
6. Connect the 12 Vdc power adapter to the K-ECS (Fig. I-1.5(b)). Do NOT use the 24 Vdc adapter. Now, you can turn on the K-ECS switch which should automatically turn on the K-CSP as well. This step must be successful before proceeding any further. If the lights on either boards are not on, you have to check your connections.

I-1.2.2 Understanding Figure I-1.2

- Super thick **black** and **red** lines are ground and power lines, respectively. At least 18 AWG wires (thick wires) MUST be used for these connections (NOT the regular breadboard wires).
- Solid continuous connections carry analog signals.
- Dashed connections carry digital signals.
- The circuit in the dashed box at the bottom of the figure is what you will implement on your breadboard as needed.

I-1.2.3 Speed and Position Sensors

The motor system is equipped with a digital encoder to measure the speed and position of the motor (Fig. I-1.3). The position sensor is a relative position sensor as it measures the motor position relative to its starting point. In other words, it takes the starting point as the zero position.

The sensitivities of the speed sensor, K_ω, and of the position sensor, K_θ, are given in Table I-1.1. As can be noted from the table, the speed sensor generates 5.0 V for every 9.225 rad/s. Likewise, the position sensor generates 5.0 V for every 2π rad of rotation.

This means that if the speed sensor's output is 1.5 V, for example, then the motor's speed is $\omega_m = 1.5/K_\omega = 2.77\,\text{rad/s}$.

Likewise, if the position sensor's output is 2 V, then the motor's angular position is $\theta_m = 2/K_\theta = (2/5) \times 2\pi = 0.8\pi\,\text{rad}$ (which is 2/5 of a turn).

Description	Symbol	Value	Unit
Speed sensor's gain	K_ω	5.0/9.225	V·s/rad
Position sensor's gain	K_θ	$5.0/(2\pi)$	V/rad

Table I-1.1: Sensor gains

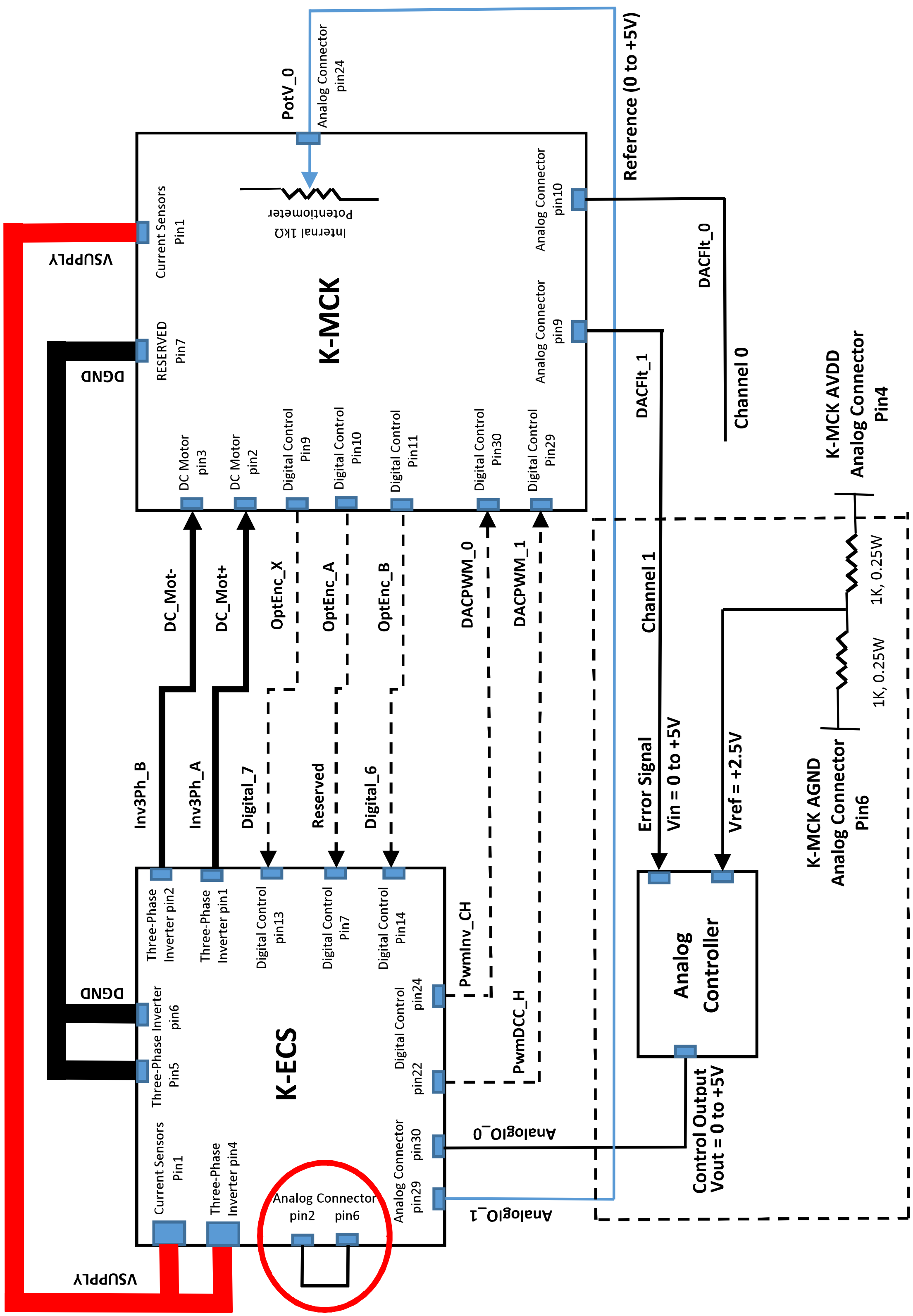


Figure I-1.2: Interfacing the motor kit with the K-ECS board (courtesy of Kylowave Inc.)

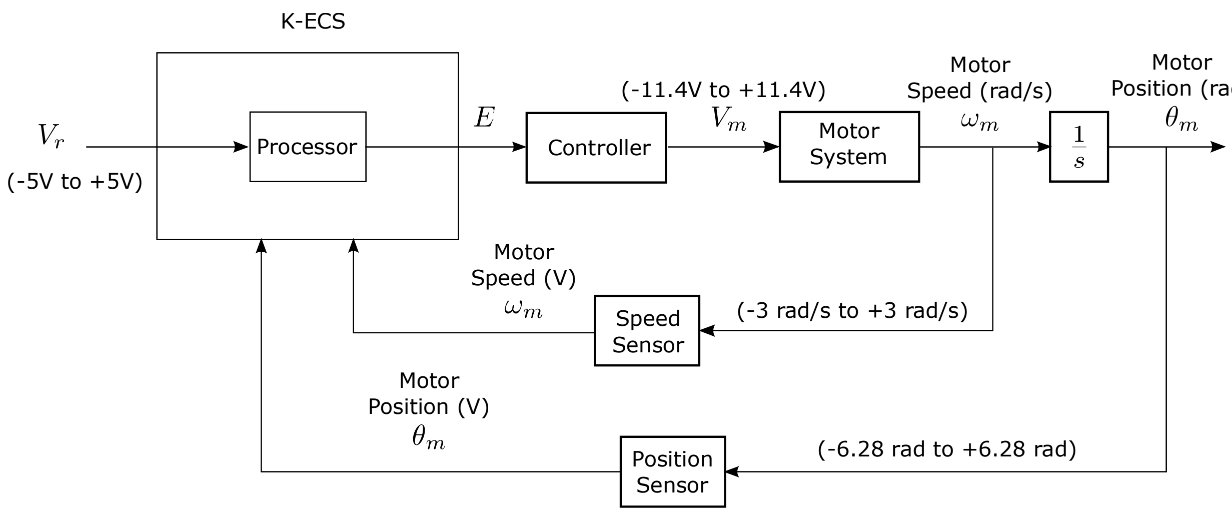
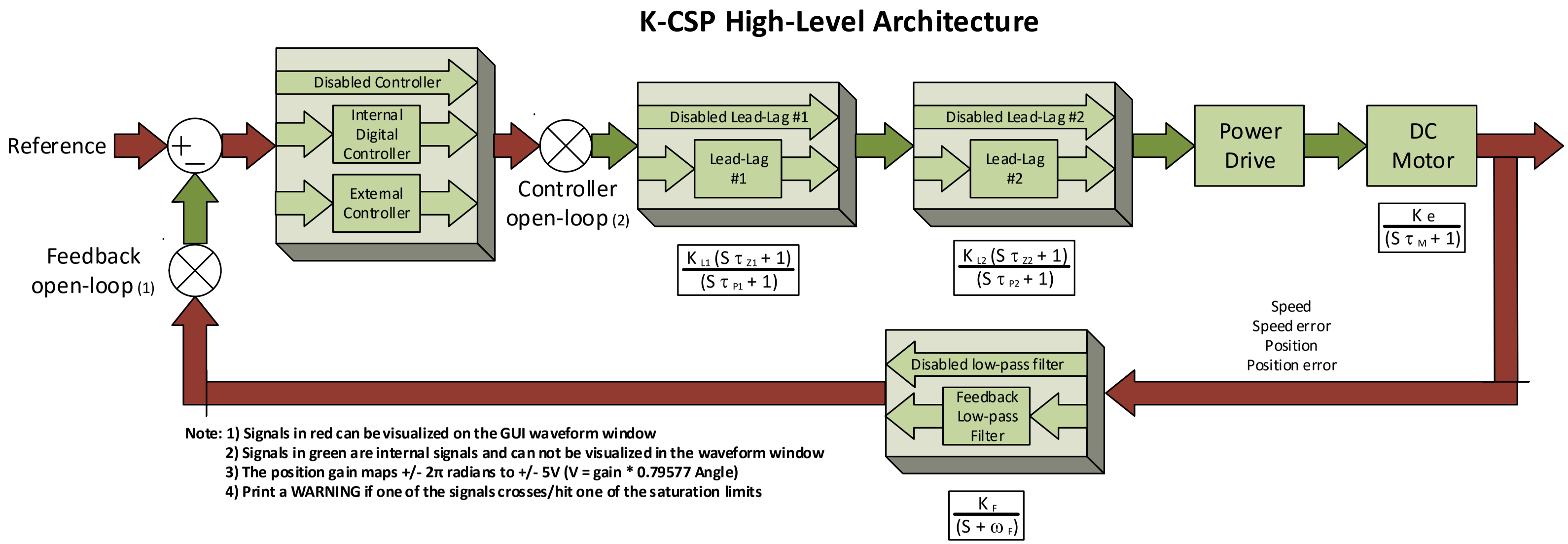


Figure I-1.3: Overall system's high-level block diagram



REFERENCE
Enable:
external generator
internal generators
Signal range: +/- 5 V

FEEDBACK
Enable:
enable
open-loop at feedback (1)
open-loop at controller (2)
Signal range: +/- 5 V

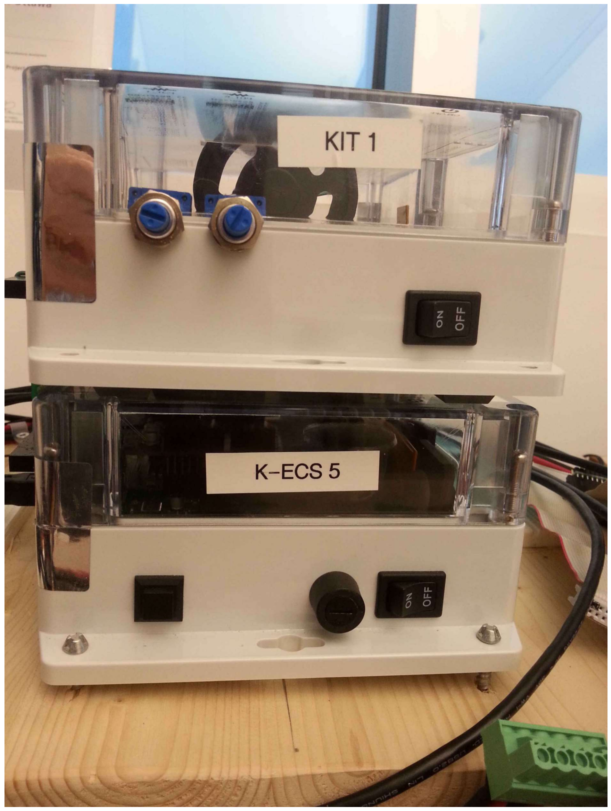
CONTROLLER
Enable:
none (bypass mode)
digital
external
Internal controller:
type: digital
range: programmable
input range default:
output range default:
External Controller:
type: analog or digital
input range: +/- 5 V
output range: +/- 15 V

LEAD-LAG
Enable:
none (bypass mode)
lead-lag
Type:
digital
Ranges:
programmable
input default:
output default:

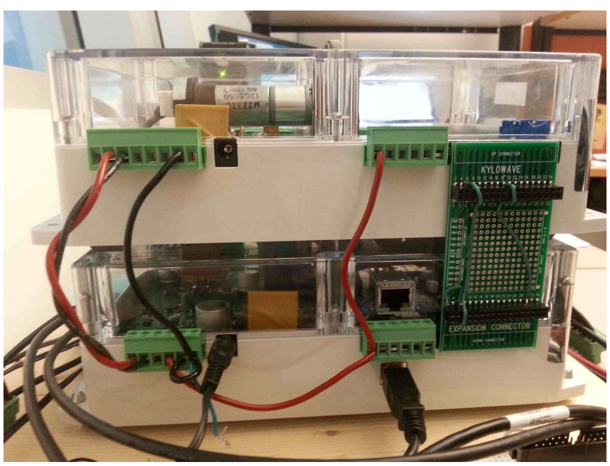
FEEDBACK FILTER
Enable:
none (bypass mode)
first-order low-pass
Type:
digital
Ranges:
programmable
input default:
output default:

ACTUATOR
Power drive:
supply voltage: 12 Vdc
max. voltage: 11.4 Vdc
max. current: 2A
full-bridge
gain: 1:1
DC motor:
max. voltage: 11.4 Vdc
min. voltage: 0.6 Vdc
no load current: 0.09A
max. current: 0.5 A
gear ratio: 1:84
bidirectional rotation
shaft speed: 7.2 KRPM
gear speed: 85.7 RPM
Signal range: +/- 5 V

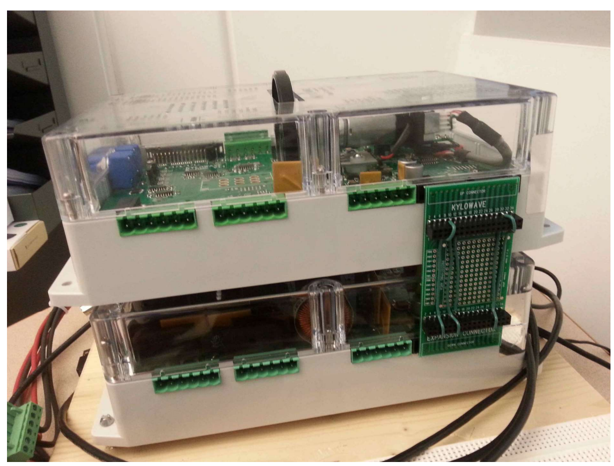
Figure I-1.4: Block diagram (courtesy of Kylowave Inc.)



(a)



(b)



(c)

Figure I-1.5: Kit Setup

Chapter I-2

Software

I-2.1 Software Description

The interface with the hardware is done through a graphical user interface (K-CSP GUI) which is the main window to show up when you launch the application (Fig. I-2.1). The following is a brief description of the K-CSP features.

The following is a brief description of the different components of the K-CSP GUI, as labeled on the figure.

1. Total duration time of the experiment.
2. The button to start running the experiment.
3. The button to stop the experiment.
4. Pull-down menu to select the communication port with the hardware.
5. Refresh button to rescan the open communication ports.
6. Path/name of the file where the experiment result's raw data is to be stored. Logging data in this file enables users to process it later (if needed) using Matlab, for example.

   Note: Log your data on the C: drive for best performance. Avoid real-time data logging onto remote or usb drives as it is a relatively time consuming process.

7. This button enables users to change the path/name of the data file.
8. Selects whether to run the motor system in an open or closed-loop mode (see Fig. I-2.4).
9. Selects the type and source of input to the system (V_r in Fig. I-1.3 and Fig. I-2.4). The user can choose to excite the system with an external signal (from an external device such as a signal generator) or with one of several automatically generated signals which are described in Fig. I-2.2.

   Note: In all cases, this signal must be between $-5\,\mathrm{V}$ and $5\,\mathrm{V}$.

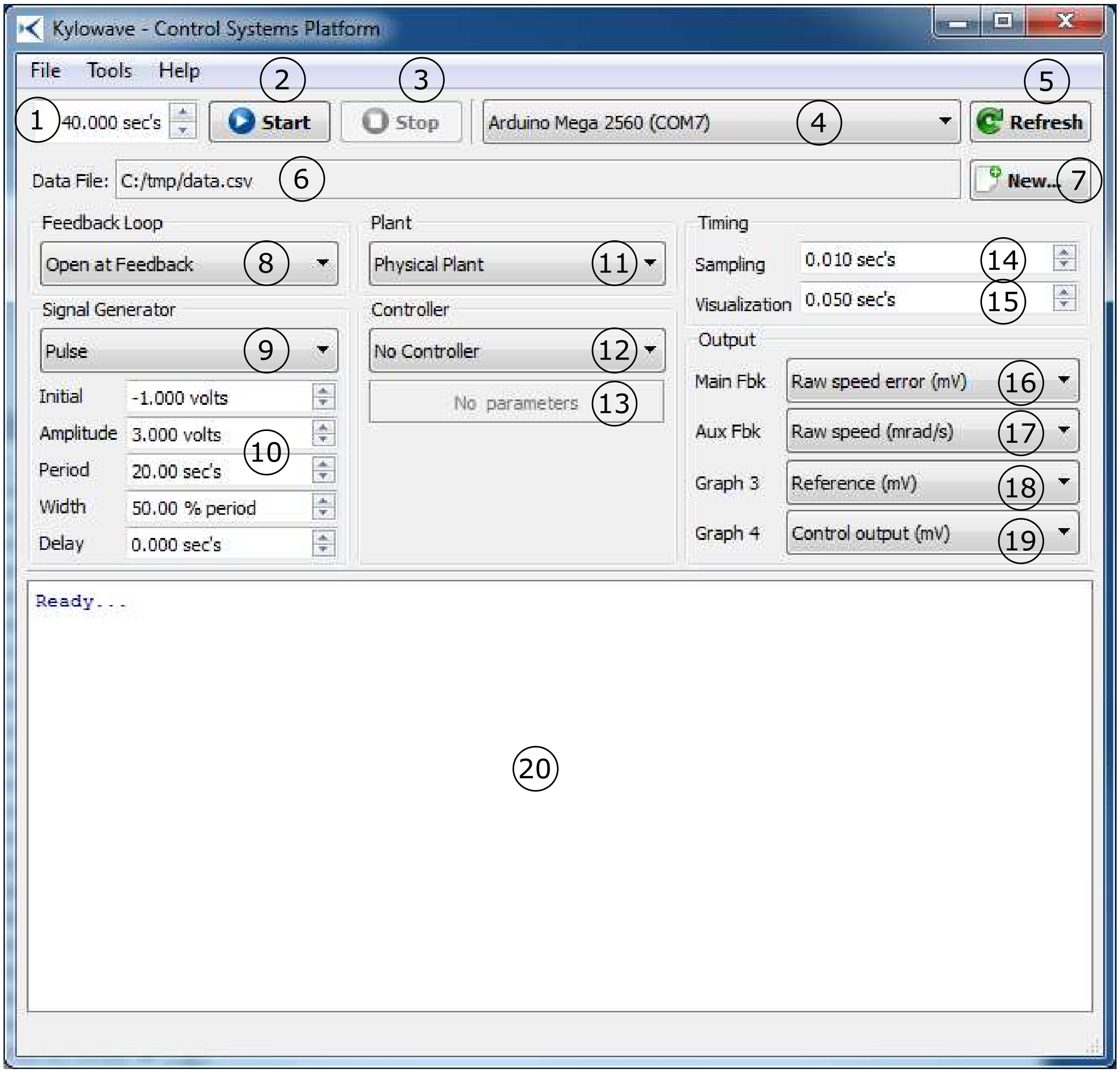


Figure I-2.1: K-CSP's graphical user interface

10. The parameters of the input signal (see Fig. I-2.2).

11. Pull-down menu to select whether to carry out a real experiment by applying the control algorithm on the physical system, or to emulate the algorithm by applying it on a pre-defined theoretical model.

12. Pull-down menu to select the type of controller to implement.

13. The parameters of the chosen controller type.

14. The time period at which analog signals are sampled for processing purpose.

15. The time period at which analog signals are sampled for data visualization, as for plotting the four generated graphs for example (shown in Fig. I-2.3). This period must be a multiple of the signal sampling period above.

16–19. These are the signals to be displayed in the graph matrix (Fig. I-2.3).

Main Fbk: This signal is always displayed as the top left graph. It can be any of the system's signals in the case of open-loop mode. However, in the case of closed-loop system, it has to be chosen as the actuation error, as shown in Fig. I-2.4. In other words, in closed-loop mode, this is always the signal passed to the input of the controller, if one is used, or to the input of the motor, otherwise.

Aux Fbk: This signal is always displayed as the top right graph. In general, it can be any of the system's signals. However, in the case of closed-loop system where the type of controller chosen is PI+PD, it is used as a second feedback signal.

Graph 3: Signal to be displayed at the bottom left graph. It is only used for display purpose, so it can be any of the system's signals.

Graph 4: Signal to be displayed at the bottom right graph. It is only used for display purpose, so it can be any of the system's signals.

The graph matrix is built on top of the KST visualization tool[1]. For the full description of how to use the menu, shortcuts, and any of its large consortium of features, readers are encouraged to consult its official website: https://kst-plot.kde.org

20. Text box where the system's status and activities are reported.

Fig. I-2.4 illustrates the relationship between the system's signals and the K-CSP settings.

I-2.1.1 System's Signals

Several of the system's signals come in two versions:

- Raw signal, meaning the signal as is. This is usually a relatively noisy signal.
- Filtered signal, meaning the value of the signal after undergoing a filtering process to remove noise.

For example, the raw speed is the motor speed as delivered by the speed sensor. The filtered speed is the result of passing the raw speed through a filter to remove noise. Note that although filtered signals are usually less noisy than the raw signals, they still carry some noise.

The following are definitions of some of the system signals:

Reference: the system's input (V_r in Fig. I-1.3 and Fig. I-2.4).

Speed: motor's angular speed.

Position: motor's angular position.

Control output: the input signal to the motor (V_m in Fig. I-1.3 and Fig. I-2.4).

[1] https://kst-plot.kde.org

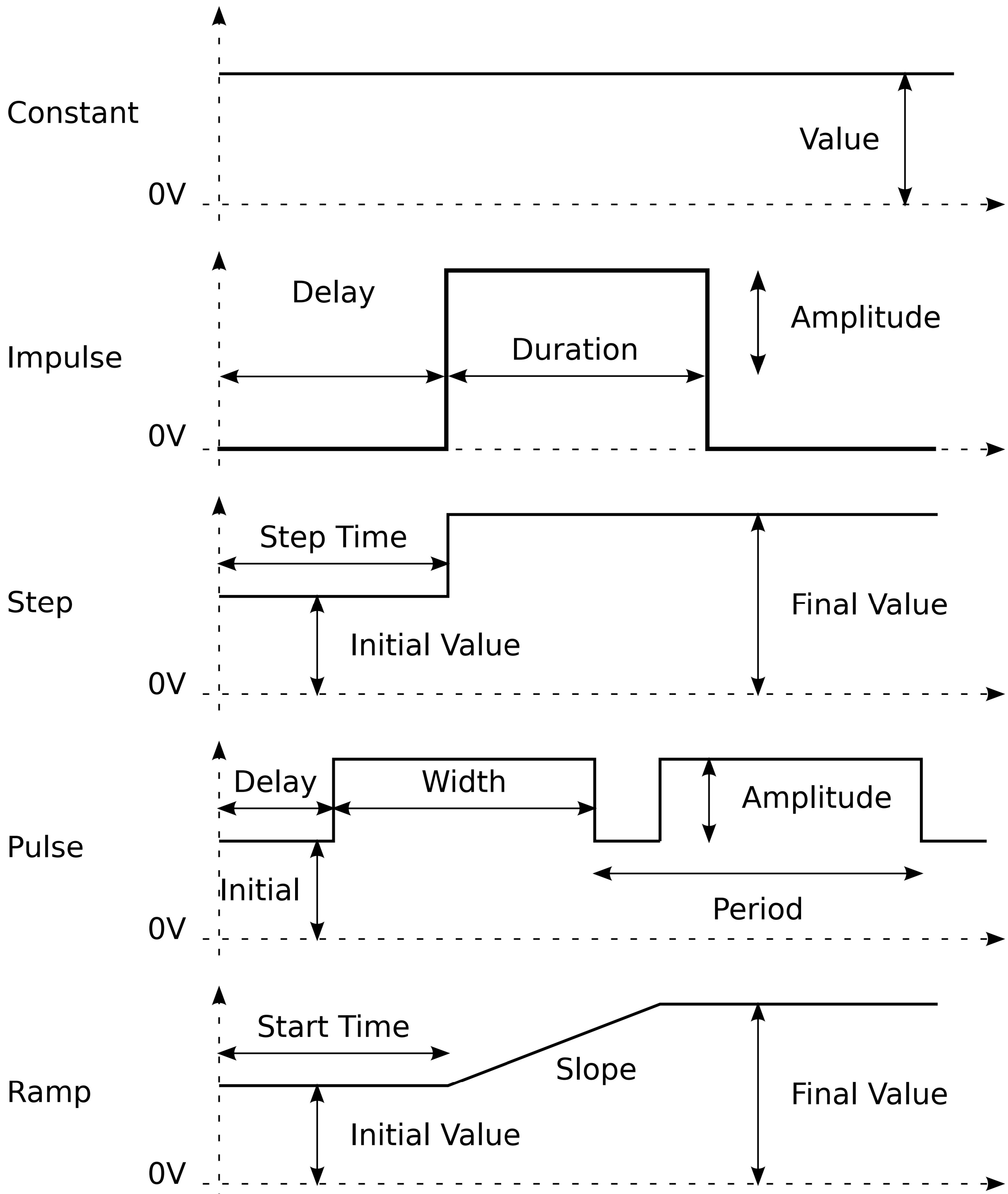


Figure I-2.2: Signals that can automatically be generated in software

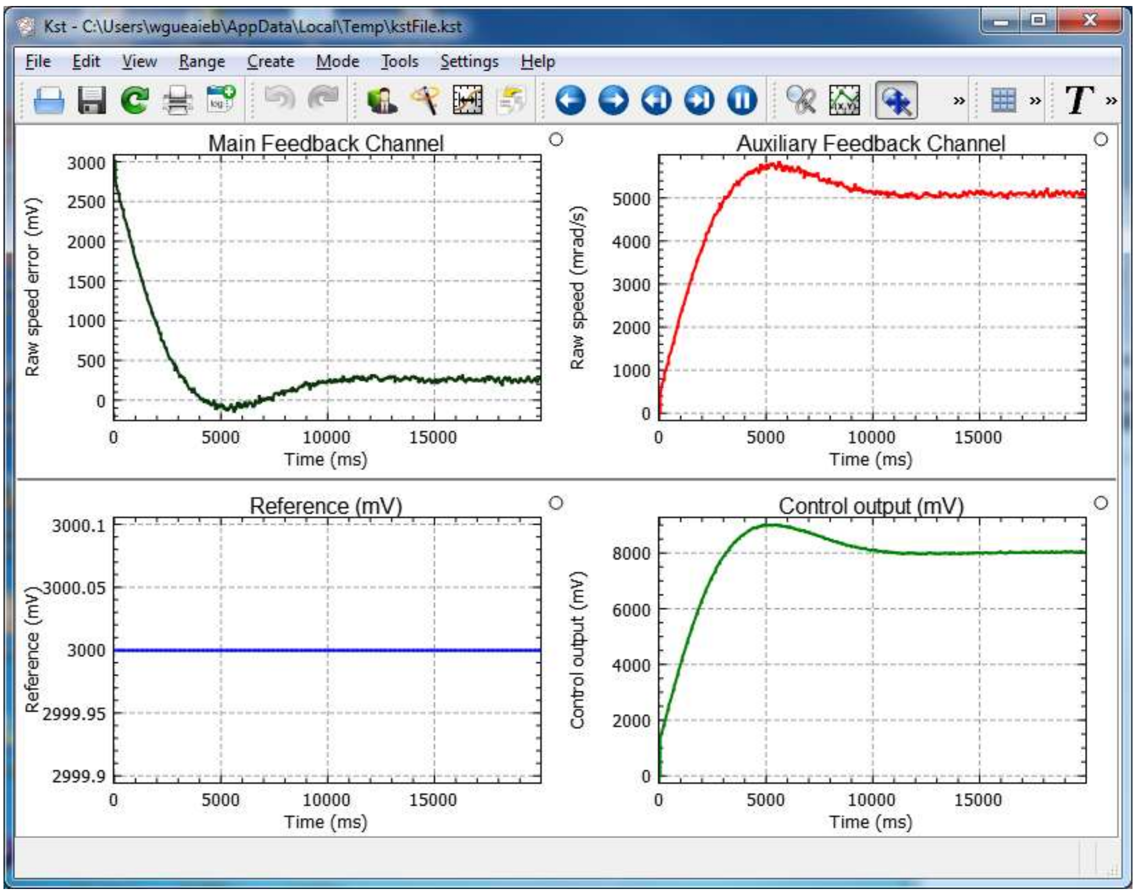


Figure I-2.3: K-CSP's graph matrix

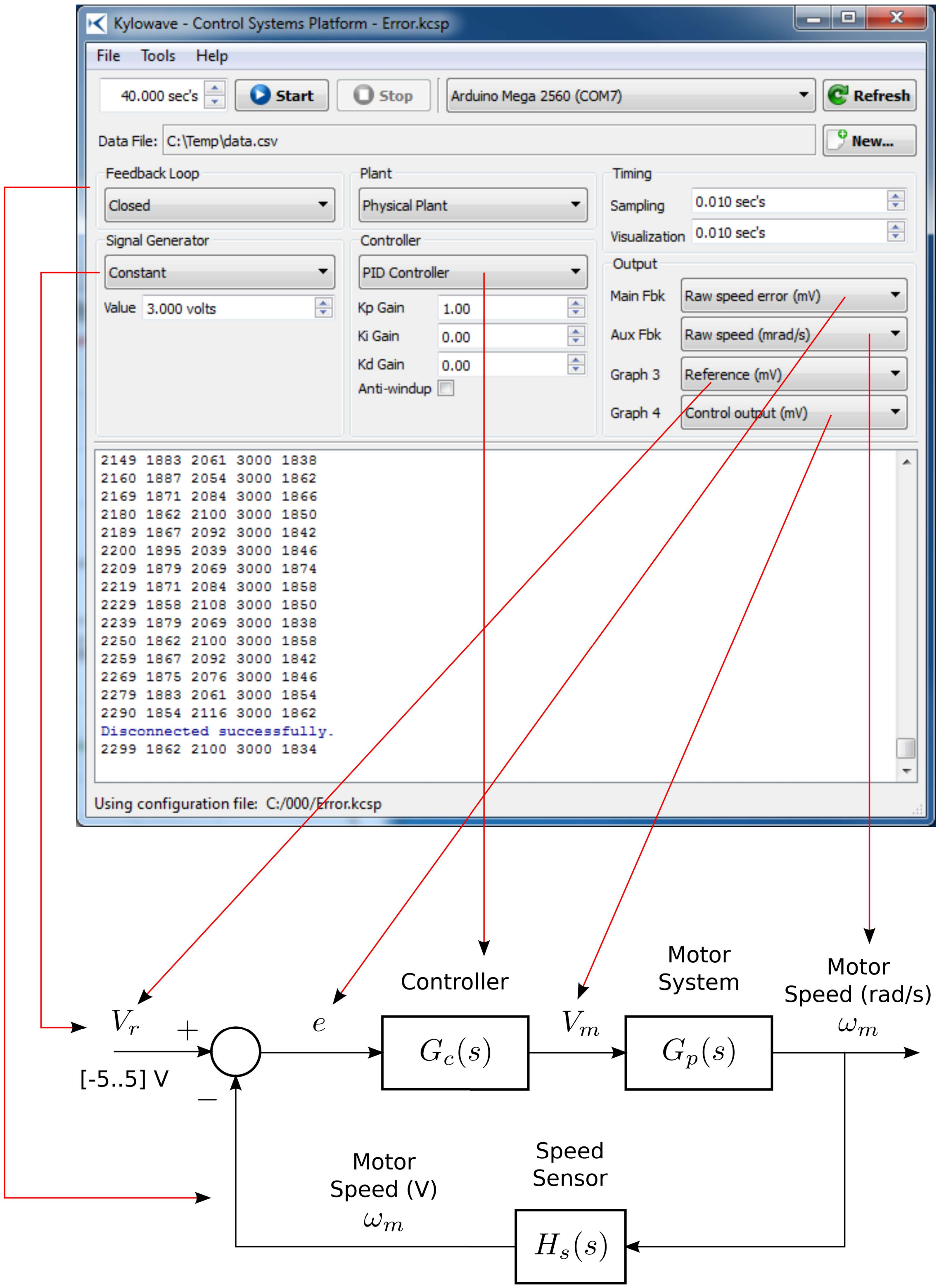


Figure I-2.4: Relationship between the system signals and the K-CSP settings

I-2.2 Using the Software

The following procedures have to be done each time the computer is rebooted. Hence, they have to be done at the beginning of each lab session. They also have to be done in the same order they are decribed below.

I-2.2.1 Driver Installation

Before you can run the software, you have to install the proper drivers that enable the hardware to communicate with the computer through the USB cable. For that, you need to follow the steps below.

1. Plugin the K-ECS board to the wall power outlet using the provided 12-V DC power adapter. Read the label to make sure that it actually provides 12V DC. Keep the switch turned OFF for now.

2. Plugin the K-ECS to the computer through the provided USB cable. Don't use your own!

3. Now, power ON the K-ECS using its own switch. The computer will probably try to automatically search for its drivers. Wait until it is done. Do not be alarmed if it fails (this is most likely).

4. Go to the following menu item (through Windows main menu)

    - Menu > Software Development > Arduino > Install Arduino Drivers

    and follow the intuitive instructions after that. Eventually, a confirmation message should come up on the screen confirming that the drivers installation was successful.

I-2.2.2 Running the K-CSP GUI

The software is to be run **only after** properly connecting the K-ECS and motor boards, as explained in Chapter I-1.

To run the K-CSP GUI on a lab computer, click on

- Menu > Software Development > Arduino > KCSP

The GUI of Fig. I-2.1 will show up on the screen.

Select the proper communication port (label 4 on Fig. I-2.1) if it was not already automatically selected by the software. The port number may vary from computer to another, but you may identify the right port by the label "Arduino Mega 2560".

I-2.2.3 Uploading the Firmware

The firmware is the software running inside the K-ECS. It is sometime referred to as the sketch. Because the K-ECS may be shared by several courses, it is recommended to upload the right firmware each time you want to use the K-ECS. This can be accomplished as follows:

1. Make sure the K-ECS is connected to the computer through the provided USB cable, then turn it ON.
2. Launch the K-CSP GUI.
3. Select the proper communication port (label 4 on Fig. I-2.1) if it was not already automatically selected by the software. The port number may vary from computer to another, but you may identify the right port by the label "Arduino Mega 2560".
4. In the K-CSP GUI menu, go to File > Upload Sketch, and upload the provided sketch (hex file). Make sure you get a success confirmation message at the end of this process. Otherwise, you may have to try again.

I-2.2.4 Uploading the Configuration File

The K-CSP is so flexible that it allows to use different configurations on the same hardware. Many parameters of the control system can be configured through configuration files.

To upload a configuration file from the menu of the K-CSP GUI, click on

- File > Open

Then select the configuration (kcsp) file you want to upload.

Chapter I-3

General Lab Instructions

The following are instructions and good common practice which students must follow in order to avoid losing points.

1. Present all the steps of your derivations. Missing out details may affect the clarity/integrity of your work and hence may cost you points.

2. All plots must be clear and serve their purpose. They have to have clear labels, units, and legends (whenever applicable). They must be well explained and referred to in the text (not just "dumped" in the document).

3. All plots in the lab reports must be generated from the experimental raw data and not from screen shots. For instance, you may do so using Matlab, Mathematica, or other neat data plotting software.
   Note that for them to serve their purpose, they may need to be processed by the plotting software of your choice (e.g., zoomed in/out, truncated, shifted, etc.) For example, if the purpose of the plot is to show the time constant of the response, that information must be visually easy to extract from the figure.

4. A participation sheet must be filled, signed by all group members, and submitted electronically as a separate document with the submission of the lab report. It is accepted to submit a scanned copy of the participation sheet.

5. Groups not storing their hardware kits neatly at the end of the lab or not keeping them in good order may receive a penalty.

6. The faculty forbids having food or drinks (except water) inside its labs.