Progress Report

CCTVal
CENTRO CIENTÍFICO
TECNOLÓGICO
DE VALPARAÍSO

December 5, 2022

Revision 4.0

1 User guide

The developed user interface is shown in Fig. 1, where it has been divided into different sections to be explained independently in this document. Each of the following subsections describe the corresponding number on the GUI figure.

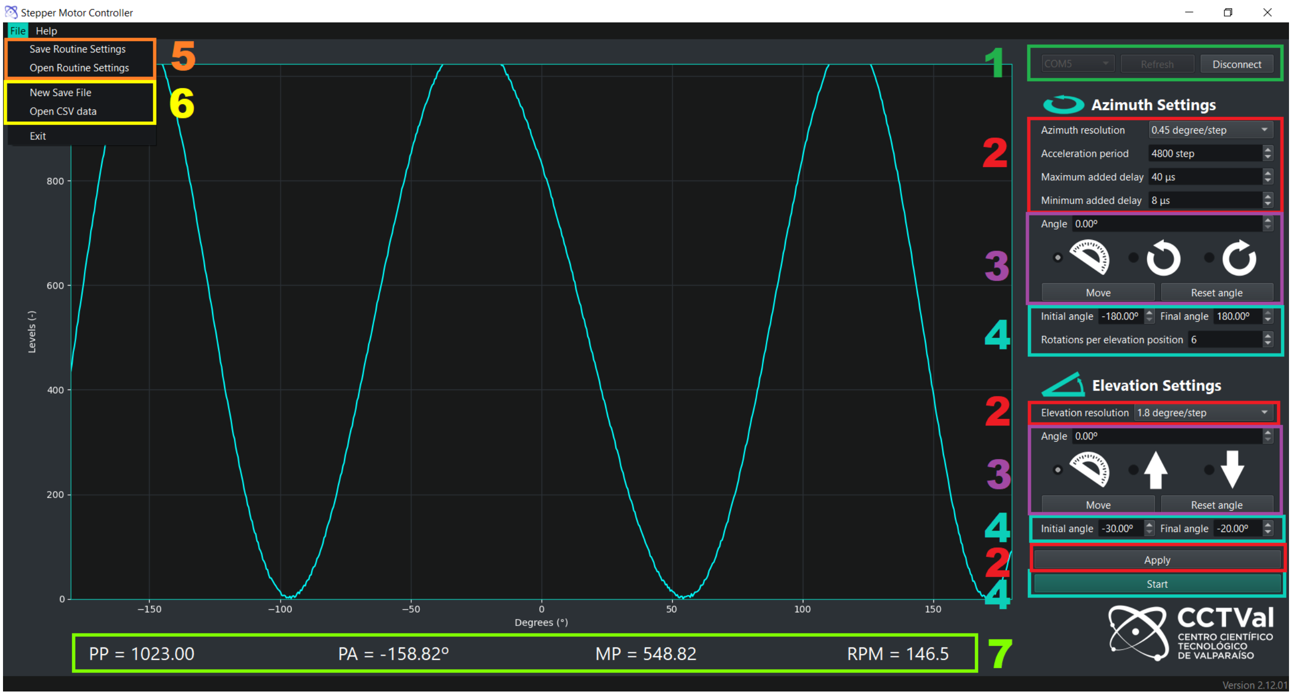

Fig. 1: Graphical User Interface mock-up

1.1 Connection

The connection section provides a list of available serial ports, which can be updated by clicking the **Refresh** button. Clicking the **Connect** button triggers the dispatch of a message with default parameters to the Portenta board, which responds with an acknowledgement message if the connection is established successfully. The absolute angle of azimuth and elevation and also set to zero. By default, the message sent has the following structure: "c−6400−4800−40−2−200−4800−40−8", where the first four numerical values correspond to azimuth resolution (in steps/rev), acceleration period (in steps), maximum and minimum added delay (in microseconds). The next four values follow the same pattern, but corresponding to the elevation stepper motor.

Once the connection is established by receiving the acknowledgement message correctly, the **Connect** button changes its name to **Disconnect**, and all widgets are enabled. The **Disconnect** button sets the variable related to the port to `None`, and widgets are disabled.

1.2 Movement Parameters

These widgets receive different parameter values from the user to be used in the movement of the motor. The first required parameter for both the azimuth and elevation settings is the movement/measurement resolution, measured in degrees/step. This parameter has a number of fixed values given by the driver resolution options, ranging from 0.05625 to 1.8 degrees/step. As an example, if the 1.8 degrees/step resolution is chosen, a measurement will be made every 1.8 degrees. In order not to change the driver settings manually, these resolutions are emulated in software by always having the maximum resolution set at the driver, and skipping measurements accordingly in powers of two.

The rest of the parameters are related to the acceleration/deceleration routine. This is a necessary step to avoid motor malfunctioning in certain resolutions. The idea is to gradually change the motor velocity at the beginning and end of its movement, as seen in Fig. 2, where the time intervals (which can be thought as time delays induced to change the velocity of the motor) between movements are shown. To implement this, three different parameters are used:

- Acceleration Period (P_a): amount of steps used to gradually change the velocity at the beginning and end of the movement.

- Maximum Added Delay (T_{as}): maximum induced delay between movements.
- Minimum Added Delay (T_{ai}): minimum induced delay between movements.

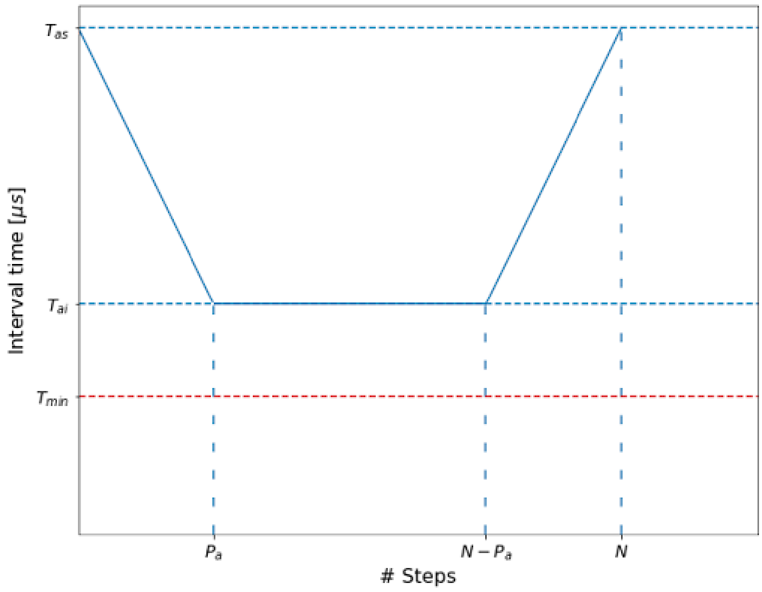

Fig. 2: Acceleration/Deceleration exmaple

To apply a change to any of these parameters, the **Apply** button needs to be used. This will send a message to the Portenta with for following structure: "p-6400-4800-40-2-200-4800-40-8", where the first four numerical values correspond to the azimuth parameters (resolution and acceleration), and the last 4 values correspond to the elevation parameters (except for the elevation resolution, by design choice, these parameters are currently not displayed on the GUI, but if needed they could be easily implemented). Note that there is always a default value already set in the Portenta for all of these parameters. Also, if no acceleration/deceleration routine is required, the aforementioned parameters can simply be set to zero.

1.3 Independent Movements

An option for independent movements both in azimuth and elevation is provided to move the antenna to an arbitrary position independent of the measurement routine. The leftmost radio button with a protractor icon is used to move each stepper motor to an *absolute* position, relative to the global reference set by software in the microcontroller. The **Reset angle** button sets the current position as angle zero. The remaining two radio buttons (counterclockwise/clockwise for azimuth and up/down for elevation) are designed to move the stepper motor an arbitrary angle independently of the global resolution set.

The string sent to the Portenta board has the structure `x000`, where `x` is a letter that determines whether an absolute, counterclockwise or clockwise movement is commanded in azimuth or elevation. The number `000` represents the number of steps dependent on the angle set at the azimuth/elevation spin box. An angle/step conversion is computed before sending the string according to the angular resolution set. For instance, if the counterclockwise radio button is checked, 360º is put at the Angle spin box, azimuth resolution is set at 200 step/rev, and the azimuth **Move** button is clicked, then `l200` is the message sent to the microcontroller, corresponding to a counterclockwise (leftward) movement. The remaining characters for the other radio buttons are `a` for absolute azimuth movements, `r` for clockwise (rightward) movements, `e` for absolute elevation movements, `u` for upward movements and `d` for downward movements.

1.4 Routine

Once the global angle reference is set, a measurement routine may be executed by clicking the **Start** button. The routine is designed so that a minimum of -180º and a maximum of 180º can be chosen for azimuth movements, and a minimum of -30º and a maximum of 60º for elevation movements.

A string with the format "`n-aIII-(l/r)FFF-eIII-eFFF-N`" is sent to the microcontroller. The first character `n` lets the microcontroller know that the following information is related to the routine. This is the same idea with `p` for parameters and `c` for connection. The first two sections "`aIII-(l/r)FFF`" are related to the initial and final azimuth angle spin boxes. These use the same format as with independent movements, that is, `aIII` represents an absolute azimuth movement to an absolute step which depends on the resolution set, and (l/r)FFF represents a relative movement counterclockwise or clockwise in order to reach the final angle set at the Final angle spin box. Likewise, the following two sections `eIII-eFFF` represent an absolute elevation movement related to the angles set at the Initial and Final elevation angle spin boxes. The last section `N` corresponds to the Rotations per elevation position spin box, which is the number of repetitions of azimuth movements for each elevation angle.

At the microcontroller, the routine consists of the following:

1. **Initialisation:** The azimuth and elevation motors move to the absolute position given by their respective Initial angle spin boxes.
2. **Repetitions:** The number of elevation movements to go from `eIII` to `eFFF` is computed according to the current elevation resolution. This is independent of the number of rotations per elevation position.
3. **For cycle:** For each elevation position, the azimuth motor moves to the final position set at the Final angle spin box. During this movement, captured data at each step is stored in the microcontroller RAM for it to be sent when the movement is complete. Then, the motor returns to the initial position without storing data travelling the shortest angle. This pair of movements is repeated `N` times for each elevation position. When the initial and final azimuth angle are set to -180º and 180º respectively, the second movement does not occur since the motor is already at its initial position. In this case the rotation and data transmission is continuous over 360 degrees. The data transfer time is negligible in any case.

1.4.1 Data sending

As already mentioned, data is sent once for each complete azimuth movement. This is done with `Serial.write()` function, which receives a pointer to memory where data is allocated. Metadata which includes times taken at each step, final position, and direction of movement is sent at the end of each data array. All data is sent as integers, whose size is of 4 bytes. Since the maximum value of each data is going to use at most 2 bytes (the rightmost bytes), the two leftmost bytes are always zero. This is taken advantage of to define a 4 byte arrangement which lets know the PC all info has been correctly sent for each movement, by setting an arbitrary number of bits to one at the leftmost bytes.

1.5 Save/Open Settings

Values set at each spin box and combo box can be stored as a JSON file by clicking `File > Save Routine Settings`. By clicking `Open Routine Settings`, a JSON file can be read to set previously chosen parameters.

1.6 Save/Open Data

IN PROGRESS. At the moment, clicking `File > New Save File` prompts the starting of saving data. The problem is that data is not classified. For instance, five azimuth movements at the same elevation position stores the same data as one azimuth movement for five elevation positions. Open CSV data does nothing at the moment, but its intended purpose is to plot previously saved data.

1.7 Data Information

Below the plot area, four labels are added which display characteristics of the measurements as well as the motor angular speed. From left to right, these are Peak Power, Peak Angle, Mean Power and Revolutions Per Minute.

Table 1 shows a comparison of angular velocities in RPM for different resolutions and acceleration parameters. Note that the maximum possible velocity reached at 6400 step/rev is 142.0 RPM without added delays. Delays are added for robustness, to prevent a drift at the absolute angular reference caused at the stepper motor in the development end.

Resolution (step/rev)	Accel. Period (rev)	Sup. Accel. Time (μs)	Inf. Accel. Time (μs)	RPM
6400	0.75	40	2	101.9
6400	0	0	0	142.0
3200	0.75	40	4	125.0
1600	0.75	40	8	137.9
800	0.75	40	8	146.5
400	0.75	40	8	153.7
200	0.75	40	8	156.2

Table 1: Angular velocities comparison.