Snake Charmer - User Guide

Gary Lee

September 2015

Git repository can be found in `https://github.com/KM_RoBoTa/SnakeCharmer.git`.

1 Core Control

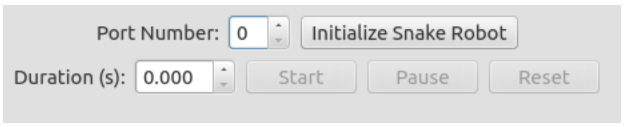
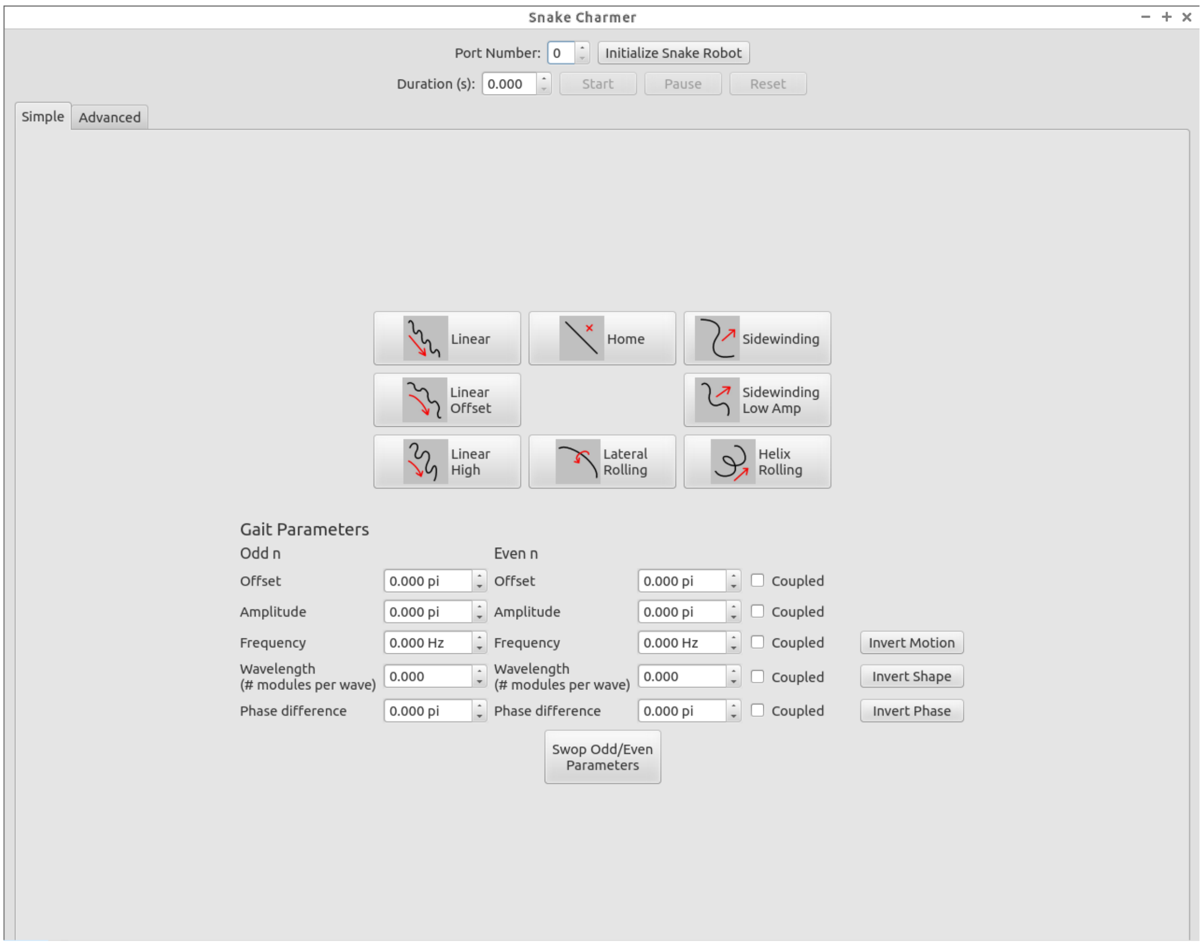


Figure 1: Top bar of the GUI - Core Control

The core control of the GUI can be found at the top of the window. This includes system initialization (requiring the port number to connect to), the duration box (to move for fixed duration; 0 for continuous movement) and the start/pause/reset buttons.
Note that the initialization button should be toggled whenever the system is unplugged or a new system is connected, to ensure proper initialization.
The three main buttons are 'Start' (to start the movement), 'Pause' (to pause the system) and 'Reset' (to stop and set back to home position). After pausing, clicking start will resume movement from the last time step. Whenever 'Start' is clicked, the system will adopt the initial configuration at a slow speed (1/10th motor speed) before performing the selected gait. The duration box allows users to set a fixed duration for which the system moves; when the duration has elapsed, the system will automatically be paused. To allow for continuous motion, the duration should be set to 0.
There are two different interfaces that can be used, presented in a tabbed format - the basic and the advanced interface (further described below). After clicking 'Start' (start moving), the tabs cannot be switched so as to avoid sudden changes in parameters or settings.
(Note: The maximum angles achievable by the actuators are $\pm 90°$ for each joint. This has been hardcoded within the program.)

2 Basic Interface



Snake Charmer

Port Number: 0

Duration (s): 0.000

Gait Parameters

Odd n		Even n		
Offset	0.000 pi	Offset	0.000 pi	☐ Coupled
Amplitude	0.000 pi	Amplitude	0.000 pi	☐ Coupled
Frequency	0.000 Hz	Frequency	0.000 Hz	☐ Coupled
Wavelength (# modules per wave)	0.000	Wavelength (# modules per wave)	0.000	☐ Coupled
Phase difference	0.000 pi	Phase difference	0.000 pi	☐ Coupled

Figure 2: GUI for Basic Settings

The basic interface consists of eight modes, each represented by their respective buttons (with corresponding icons). The five parameters are accompanied at the bottom, reflecting the current values set for each mode and allowing for manual changes by the user. The “Invert” buttons at the side and the “Swop” button below allows for the parameters to be inverted in sign and swapped between the two columns respectively. (Avoid doing so in the middle of the run, as it will cause a sudden jerk in motion.)

3 Advanced Interface

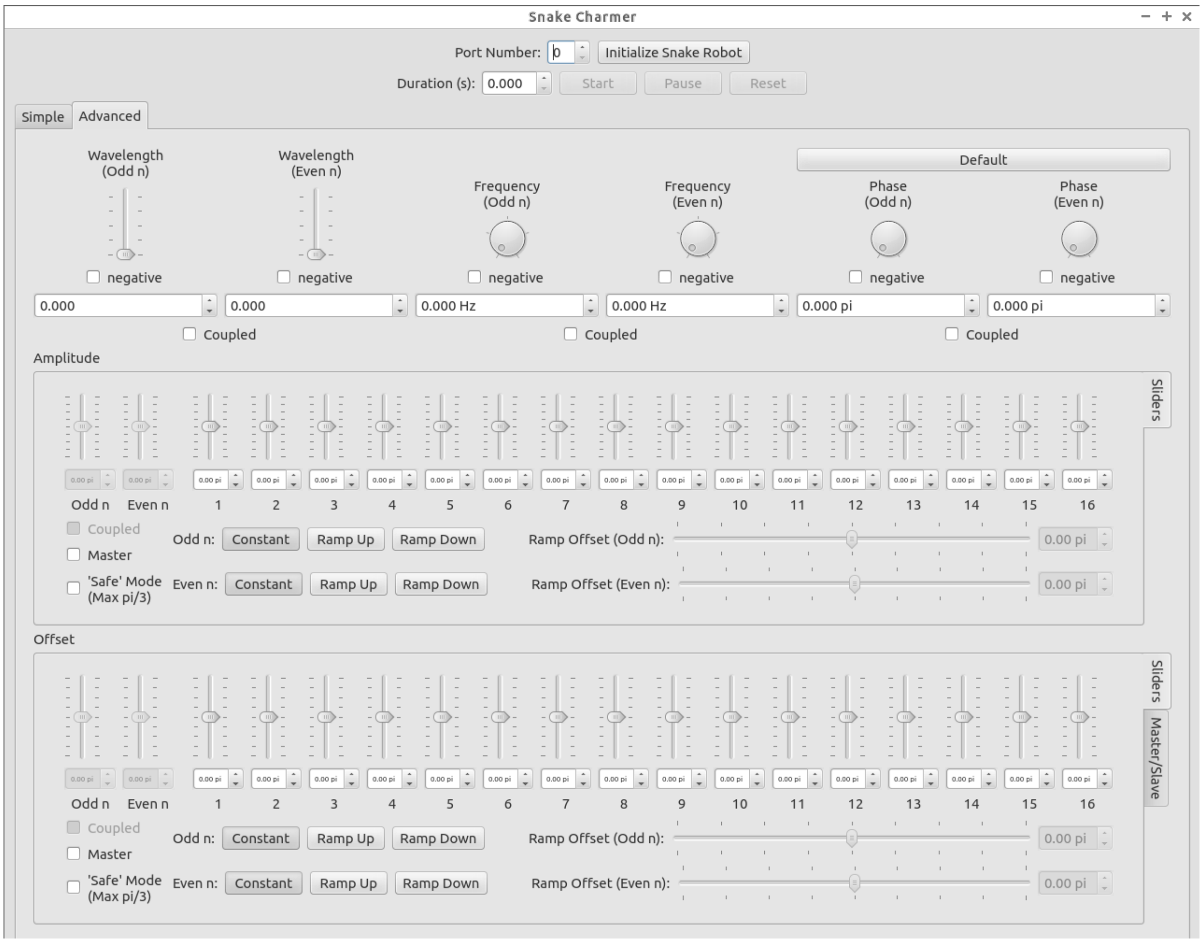
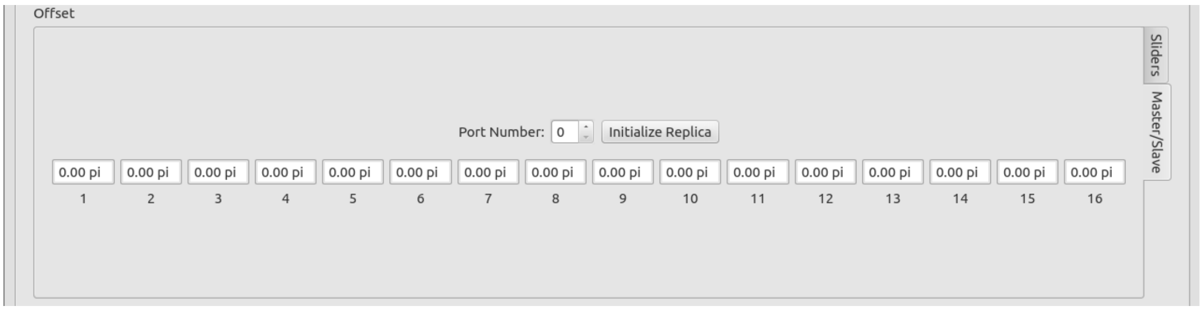

Port Number: 0
Duration (s): 0.000
Wavelength (Odd n): 0.000
Wavelength (Even n): 0.000
Frequency (Odd n): 0.000 Hz
Frequency (Even n): 0.000 Hz
Phase (Odd n): 0.000 pi
Phase (Even n): 0.000 pi
Ramp Offset (Odd n): 0.00 pi
Ramp Offset (Even n): 0.00 pi

Figure 3: GUI for Advanced Settings

When the advanced interface is toggled, the basic interface settings are carried over (and vice versa, although individually set amplitude and offset cannot be carried over). In this interface, there are more parameters for fine control.
The advanced interface features sliders and dials for more convenient control over the parameters. Furthermore, the amplitude and offset can be set individually (by de-selecting the 'master' checkbox). A 'Ramp Up' and 'Ramp Down' option is also provided for a linear increase and decrease in amplitude/offset, together with the option of adjusting a ramp offset to the resultant values.
A 'safe mode' for amplitude and offset is also offered to safeguard against overshooting the angular limit. In this safe mode, the maximum magnitude for amplitudes and offsets is $\pi/3$ rather than $\pi/2$.


Port Number: 0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi	0.00 pi

Figure 4: Bilateral Control of Offset for Advanced Settings

In addition to individual sliders, the offset parameters can also be controlled using a kinesthetic

interface through a master/slave system. Within the Master/Slave tab, another system (the 'Replica') can be initialized. The offset values will be based on the position data from the Replica. (E.g., at Home position, the main snake robot will have the same joint angles as the replica.)
Note that this option still possesses several issues, occasionally causing memory corruption and segmentation faults. It is suspected that the rapid rate of transferring (and possibly destroying) copies of vectors/data may be the cause. This issue has to be further examined and rectified.

4 Extra (separate implementation): Visualization

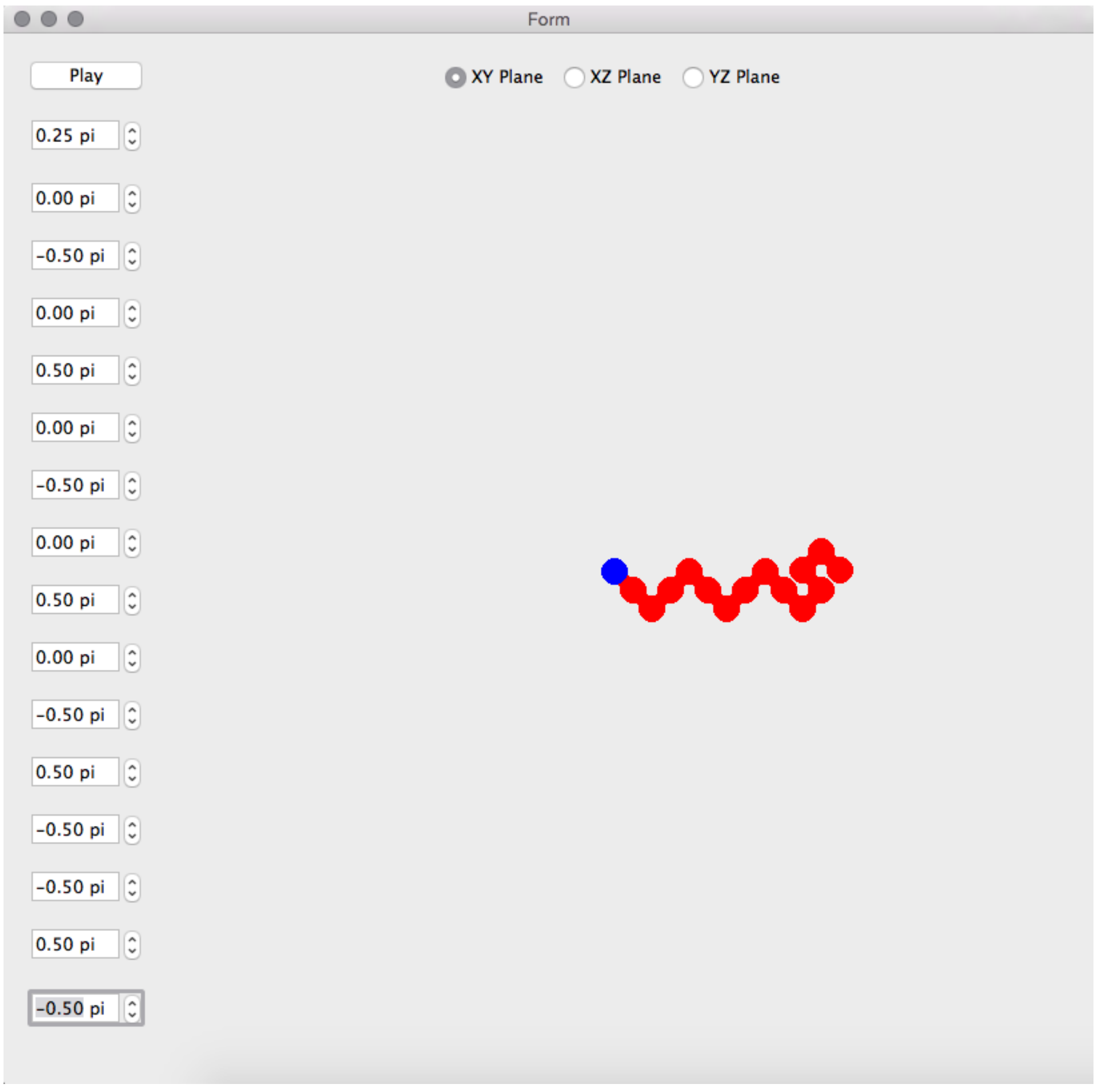


Figure 5: Preliminary GUI for Visualization

A simple Qt-based visualization widget has been created. However, as the visualization is a preliminary implementation, not all features are robust (e.g. lack of z-ordering on joints/objects). Hence, this is offered as a separate interface, which may be incorporated into the project in the near future.
The widget takes in a vector of angles, which will subsequently be used to visualize the configuration of the system (projected on XY, XZ or YZ plane). An interactive display taking in 16 angle values is provided as a preliminary proof of concept.
As the joints were stored as `QVector3D`, there is the potential of expanding the implementation to include **OpenGL** and add skinning/texturing, enabling a more advanced 3D visualization.
(Note: While the visualization is made for a Snake robot, the DSkeleton/DJoint classes should be useable for other forms of mobile robots (that only uses revolute joints). The DJoints have be the defined properly based on its position, axis of rotation, and children joints assigned.