

Simni Short Manual

Stefan Bethge
Neurorobotics Research Laboratory, Humboldt-Universität zu Berlin

11th June 2013

Contents

1	Introduction	1
2	Requirements	1
3	Usage	1
3.1	Data plotters	1
3.2	Buttons, Toggles and Shortcuts	1
4	Morphologies	2
4.1	Included	2
4.2	Custom	2
5	Simulator Interna	2

1 Introduction

The Simni simulator is a testbed for simple robots that work in a two-dimensional world (restricted). The built in morphology is imitating the Semni morphology of the Neurorobotics Research Laboratory, Berlin, (see <http://neurorobotics.de>). On top of the physics simulation, it contains an implementation of the ABC learning framework to explore the morphologies own fixpoints and therefore movement possibilities.

It provides phase space and motor voltage plots and a graph representation of the interesting points and connection between them in the explored manifold.

2 Requirements

- A computer, duh.
- A browser, preferably with a fast JavaScript implementation like V8. Recommended at the time of writing is Google Chrome or Chromium.
- For compiling the CoffeeScript sources: CoffeeScript 1.3.3, maybe newer Versions

3 Usage

3.1 Data plotters

The data plotters are in the top right of the page. To see current data and a history of the last 1000 time steps (96 Hz control loop), click the enable boxes in the top left of each graph. Phase trajectory displays in red, with the hip angle on the x-axis and the knee angle on the y-axis (it might be hard to find if there is no movement). The motor torque is displayed in green for the hip and blue for the knee torque.

3.2 Buttons, Toggles and Shortcuts

There are a number of controls in the interface that are grouped by usage. The "Simulation Controls" allow the selection of morphologies, at the moment single and double pendulum and Semni. "Simulate in Reatime" is intended to make the physics move the robot at about 60 fps which corresponds to about the same speed as realtime. Otherwise the simulation will run as fast as possible. "Pause/Run Simulation" toggles between halting and running again. The graph layouting and other processes independent from the physics will not be paused. While paused, progressing one step can be achieved by pressing "Next step(s)". This means however 10 physics simulation steps and one controller step, c.f. Section 5. The buttons "toggle CSL" and "toggle bounce controller" each enable or disable motor controllers for both joints. The CSL controller will use the paramters below the buttons, the bounce controller (using constant velocity and changing direction on stall) has default parameters.

The "ABC learning" group contains controls related to the learning framework. "toggle explore" will also toggle the CSL controller and will start exploring. It will use the set "next mode strategy" setting to determine what next csl mode to set when a fixpoint was detected. This change will be reflected in the select boxes and the CSL parameters on the left. The buttons "Save graph as SVG", "Save graph as JSON" and "Load graph from JSON" should be self explanatory. All progress will be displayed in the graph below all of the controls after postures are detected. The group "Graph properties" holds settings for this graph. Repulsion and stiffness relate to the spring simulation between nodes and changes the way the layouting looks and behaves. These will need a browser with support for html range elements.

"Animate graph" will toggle if the graph is continuously redrawn or only once when a new posture is detected or when the user moves nodes in the graph with the mouse. "Pause graph layouting between new poses" will make the layout algorithm be stopped after a few seconds of processing whenever a posture was found. Otherwise it will run continuously and might slow down the physics simulation. If the page was loaded from a remote url, web workers will be used for the layouting which in effect runs on another CPU core than the physics (if possible) and reduce the drag. "Show node activation" will show the numerical value that indicates how much there is to explore or to still be learned at each node. This is used with the unseen mode strategy to determine which edge to follow when for one node, all possible directions have been taken already. The same information is shown by colour if "Show node activation colors" is activated. "Show Semni postures" will trigger the display of small semni images above each node in the graph. "Show transition labels" will show the CSL mode that was set during a transition from one node to another, in addition to the mode on the node, as these may differ. Finally, "Save graph with every new posture" will save the whole graph as JSON file whenever a new pose is found to have an automatic history of what happened.

The more frequent buttons have one letter shortcuts as depicted by the [] brackets in their button label.

4 Morphologies

4.1 Included

- Single Pendulum
- Double Pendulum
- Semni

4.2 Custom

Import Data, Preparation, Creating Box2D entity

5 Simulator Interna

Uses Box2D (2.1a3) port to JavaScript, patched so torque can be applied to a joint. The morphology is drawn on a html5 canvas, the graph is drawn directly with svg. The html5 range element provides sliders for certain values.

Apart from mere physics simulation, there is simple motor model and a simple fluid friction model that is currently switched off however as fluid friction does not play a big role in real experiments.

Friction

Joint Motor von Box2D hat Paramter maxTorque und setVelocity 0, so dass dieser versucht, die Geschwindigkeit auf 0 zu halten, so lange die eingehenden Kräfte nicht größer als maxTorque werden, wir haben eine Haftreibung. Wenn die eingehende Kraft größer ist, wird die maxTorque effektiv abgezogen und dadurch eine trockene/konstante Reibung erzeugt. Der Übergang weist aber damit keinen Stribeck-Effekt auf, der Übergang ist (wie beim Legomotor) sehr sauber.

RE-MAX 17 ist der Name des Motors, das sind also Konstanten aus dem Datenblatt. Da der Servo mit Getriebe usw. kommt, kann man die aber nicht direkt nehmen. Zusätzlich haben wir aber auch ein paar Daten gemessen und damit bessere Daten als die aus dem Datenblatt. In der Tat ist es wichtig,

die Formeln auch mal zu sehen bzw. zu benennen, die den Rechnungen da zu Grunde liegen, Teilweise sind die aber auch im Maxon-Material enthalten. k_m des Motors wird umgerechnet, da die Einheit im Datenblatt nicht brauchbar ist ($mNm * A^{-1}$, die Physik will Nm), dann mit dem Übersetzungsfaktor des Getriebes (hat Mario ermittelt, ich glaube durch Zahnradzählen) multipliziert und dem Wirkungsgrad (der auch aus Messungen von Torsten kommt). (Wirkungsgrad ist Messgröße, die sich ergibt. Kann aber so nicht reinmultipliziert werden.). k_b muss umgerechnet werden, da es im Datenblatt in $rpm/V = (r/min)/V$ angegeben ist und wir aber rad/sec wollen. Umdrehung ist Einheitslos und eine entspricht 2π , eine Minute hat 60 Sekunden, dann noch den Übersetzungsfaktor. Die Zahl die Vorne steht, ist auf Korsika experimentell ermittelt worden (allerdings mit dem Spannungsmodus, nicht PWM. Das müsste/könnte man jetzt nochmal bei Torsten erfragen, ob sich die Werte geändert haben, aber viel wirds nicht sein). R ist der Widerstand aus dem Datenblatt (der für 25C gilt) mit einer Formel aus dem Maxon-Material und der Temperaturkonstante für Kupfer für 65C umgerechnet. Und ja, Semni-akkus haben nur drei Zellen, also wie ich heute gelernt hab kommen da maximal genau $4,2V * 3 = 12,6V$ raus.

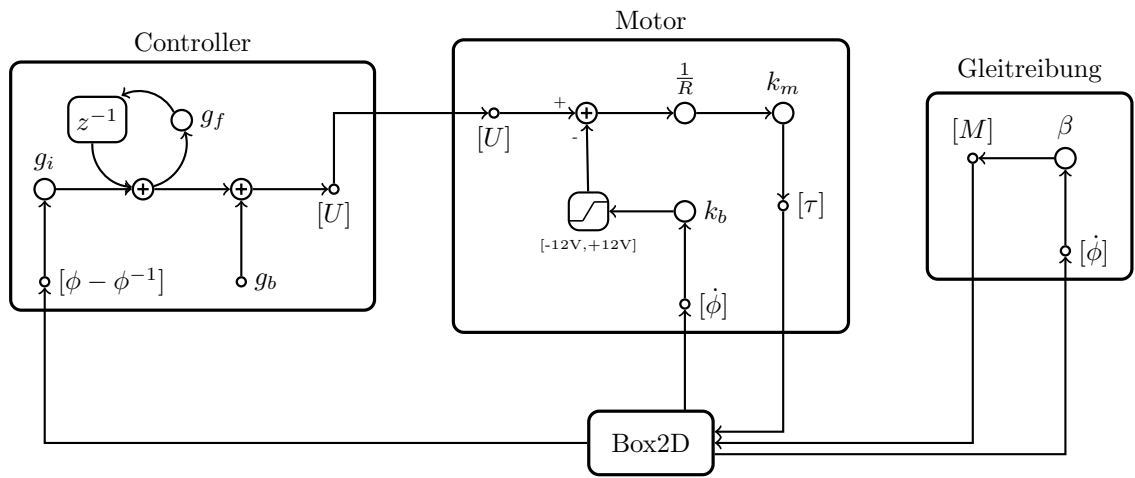


Figure 1: Controller model: Fluid friction was added, Box2D provides dry and sticky friction with a simple controller on physics update rate that tries to maintain a zero velocity and has a very small maximum torque set. This stiction and dry friction has no Stribeck effect.

Add limit and voltage multiplier