SUNFEST Technical Report TR-CST01NOV04, Center for Sensor Technologies, Univ. of Pennsylvania, Philadelphia, PA, 2004

University of Pennsylvania
Center for Sensor Technologies

SUNFEST

NSF REU Program
Summer 2004

VISUALIZATION OF REACHABILITY GRAPHS IN HYBRID SYSTEMS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Seth Charlip-Blumlein, Electrical Engineering – University of Pennsylvania
Microsoft Sunfest Fellow
Advisor: Professor Vijay Kumar

ABSTRACT

Hybrid Systems is a control technique that is used in robotics. It combines sets of continuous differential equations with a discrete system that adjusts them based on multiple criteria, and uses this to simulate or control a system. The BIOCharon team takes this technique and uses it to simulate biological systems. These systems can contain many variables, and the end result of the simulation is a state space that is not easily understandable or viewable. GGV (short for Gridded Graph Viewer) is a program that allows a user to do specific "reachability analysis" on the system as well as presents a good visualization for it.

Table of Contents

1. INTRODUCTION

The hybrid systems technique is built around the concept of combining discrete modeling techniques with continuous ones. Not all systems can be successfully modeled with only one of these techniques: some systems will be accurate for a time but will break down and lose their accuracy after a certain amount of time. The hybrid systems approach solves these problems by combining continuous and discrete systems into a powerful modeling technique that is capable of handling most any type of system.

A hybrid system is constructed from several elements. The most important element is the set of N variables that make up an N-dimensional state space. These variables are chosen to be significant in terms of the simulation (for example, in a biological simulation, they would be chosen as substance concentrations or other important values). The system also contains a finite state machine, defined in order to transition between states when one or more variables satisfy certain conditions. Lastly, each state contains a set of differential equations that govern the behavior of the system in that particular state.

The system is simulated as follows. Each variable in the state space is partitioned into slices, making N-dimensional surfaces. The sets of differential equations can be thought of as vector fields throughout the state space, and thus the flux of the vector fields through the surfaces can be calculated. If we consider the spaces bounded by the surfaces as nodes in a directed graph, a directed edge can be made between nodes based of how much flux there is between adjacent surfaces. Doing this for all of the surfaces in the state space results in a directed graph, which is the input to GGV (the graph analysis program).

2. DEFINITIONS

This section is devoted to defining some key terms that are needed to explain the functionality of GGV.

2.1 N-tangle

An N-tangle is an extension of a rectangle into N dimensions. An N-tangle is defined by two vectors: one designating the smallest point of the rectangle and one designating the biggest. The N-tangle itself is the set of all points that are both greater than or equal to the small vector and less than or equal to the big vector.

2.2 Directed Graph

A directed graph is composed of two sets: a set of nodes and edges. A node is a point in the graph, and in practical applications, it has a piece of information attached to it (sometimes a name, sometimes something more informative). An edge in a directed graph has a starting node and an ending node, and may only be traversed in one direction.

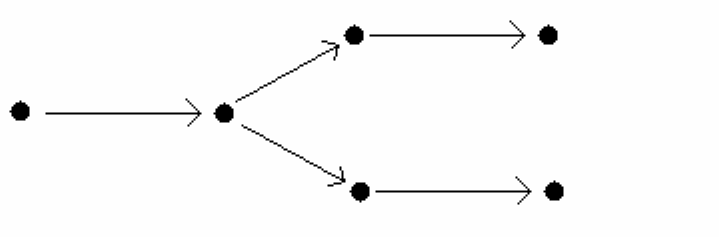


Figure 1: Example of a Directed Graph

2.3 Gridded Graph

A gridded graph is a directed graph for which each node is associated with a different N-tangle. Edges between nodes may only exist if the N-tangles of the start and end nodes are adjacent to each other.

A gridded graph is given as input into GGV.

2.4 Reachability Graph

A reachability graph is a directed graph that is created based on a starting graph and a starting node. A reachability graph contains the subgraph of all nodes and all edges that can be reached from the starting node. There are several algorithms that can be run to extract a reachability graph.

Breadth first search is an algorithm that is run on a graph in order to determine what nodes are reachable from a particular starting node. The breadth first search (BFS from now on) that GGV uses is different from the usual definition of breadth first search. It takes 3 parameters: the graph to be searched, a starting node, and an integer to designate the maximum number of links in the graph.

The algorithm runs as follows. Beginning with the starting node, the algorithm adds the starting node, all outgoing edges and the end nodes of those edges to a new graph. It continues to do this for the newly added nodes until the new nodes are further away from the starting node than the threshold allows or that there are no new nodes to add. Once either of these conditions have been fulfilled, the algorithm terminates and returns the newly created graph.

This algorithm can be retrofitted to search in reverse quite simply and efficiently. Instead of using the outgoing edges of newly added nodes, the algorithm can use the incoming edges. This allows BFS to create the reverse graph.

3. GGV

3.1 Reachability Analysis

One of GGVs main functions is to do reachability analysis. At startup, GGV reads in a gridded graph from a file. It then allows the user to select the starting node and, once this is done, it generates the new reachability graph using the selected base node.

Since the graphs can sometimes be difficult to visualize, GGV allows you to ask reachability questions of the form “Is node X reachable from the current base node?” or “Is it possible to reach the current base node from node Y?”.

3.2 Visualization

The other major function of GGV is to visualize these gridded graphs in a helpful way. The best way to do this is to project all of the N-tangle nodes down into 2-dimensions (based on 2 variables chosen by the user), plot the nodes in these 2 dimensions, and look at many of these plots at the same time. This is how GGV displays the gridded graphs. It also allows the ability to mask nodes, gain node information by mousing over a node, as well as other possibilities.

3.3 The GGV Graphical User Interface

The following images were captured based on a simulation run by the BIOCharon team. It is a system with 5 variables, 3125 nodes and approximately 15000 edges.

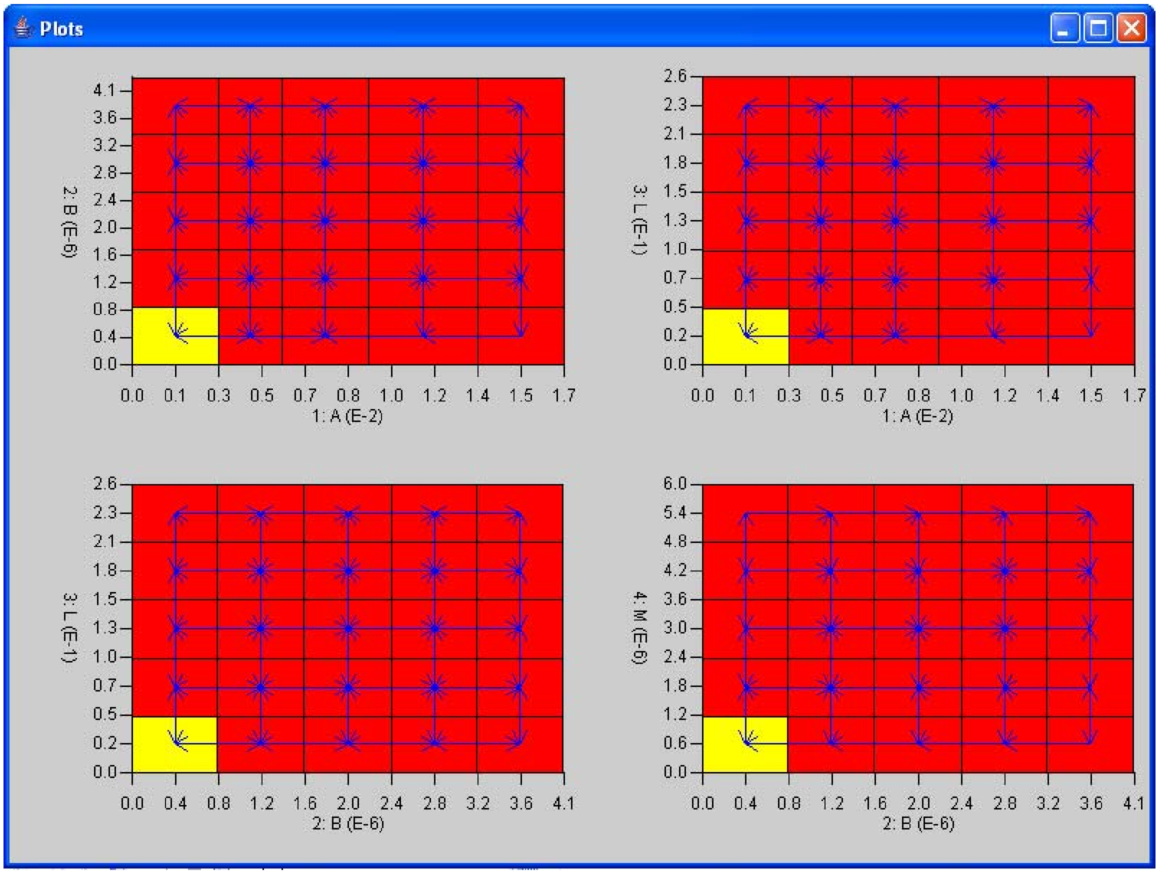
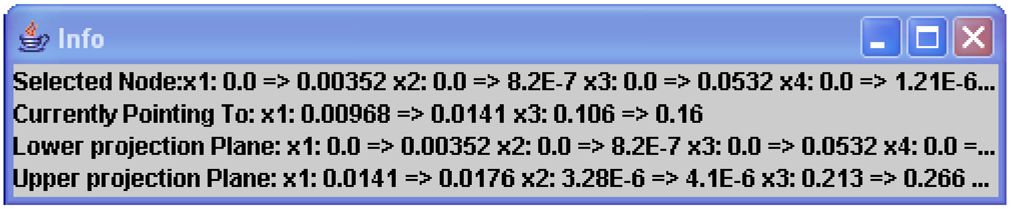


Figure 2: GGV Plots Window

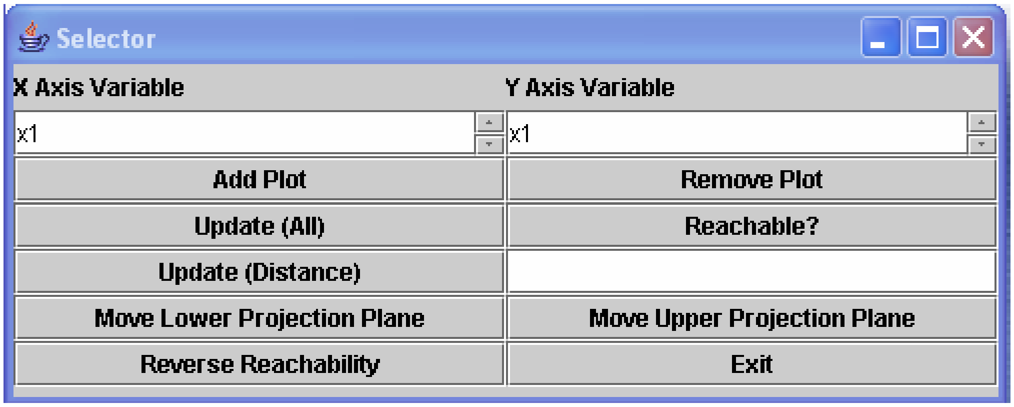
Figure 2 is the GGV plots window, where the plots are displayed. The yellow rectangles represent the N-tangle of the current base node and the red rectangles represent the reachable nodes. The blue arrows are the visible edges (but only the ones where the start and end nodes differ only in the plotted dimension).



```
Selected Node:x1: 0.0 => 0.00352 x2: 0.0 => 8.2E-7 x3: 0.0 => 0.0532 x4: 0.0 => 1.21E-6...
Currently Pointing To: x1: 0.00968 => 0.0141 x3: 0.106 => 0.16
Lower projection Plane: x1: 0.0 => 0.00352 x2: 0.0 => 8.2E-7 x3: 0.0 => 0.0532 x4: 0.0 =...
Upper projection Plane: x1: 0.0141 => 0.0176 x2: 3.28E-6 => 4.1E-6 x3: 0.213 => 0.266 ...
```

Figure 3: GGV Information Window

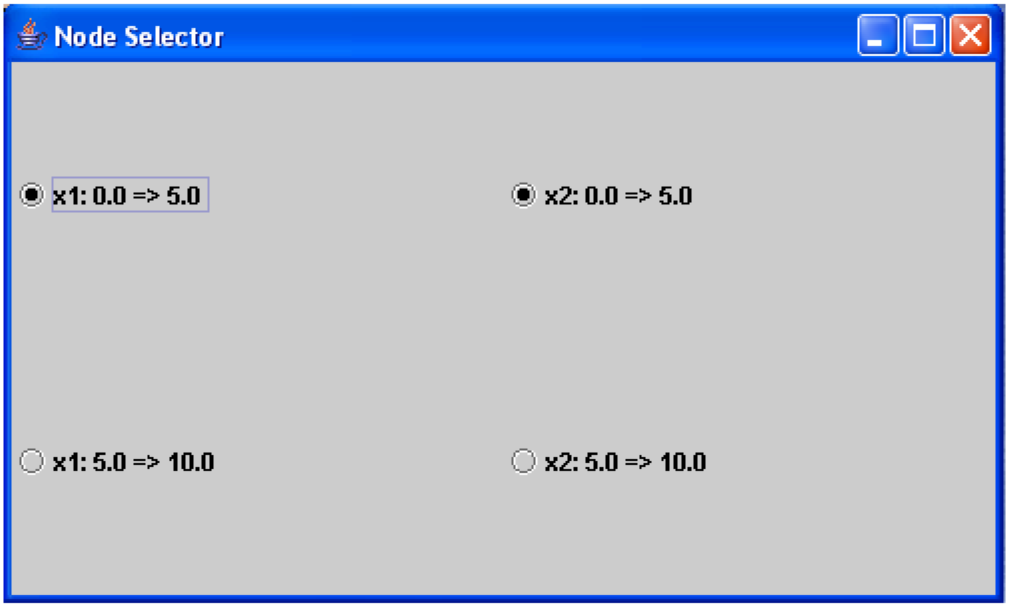
Figure 3 is the GGV Information window. It lists the currently selected node, the node that the user is currently pointing to with the mouse, as well as the upper and lower projection planes. Any node that is greater than the upper projection plane or less than the lower projection plane is not drawn on the plot (these planes are unique to the individual plots).



X Axis Variable: x1
Y Axis Variable: x1

Figure 4: GGV Plot Selector and Manipulation Window

Figure 4 is where most of the user interaction takes place. Here, the user can add and remove plots from the plot window, update the graph for a new base node or limit the number of links in the reachability graph (as mentioned in section 2.4) as well as moving the projection planes and reversing the reachbility graphs.



● x1: 0.0 => 5.0

● x2: 0.0 => 5.0

○ x1: 5.0 => 10.0

○ x2: 5.0 => 10.0

Figure 5: GGV Node Selector Window

Figure 5 is the GGV node selector window (the figure is from a different, much smaller, 2 variable model than the 3125 node model). It is used to select new base nodes, ask the reachability questions (from section 3.1), as well as move the projection planes.

4. ACKNOWLEDGMENTS

Several people made this project possible and helped me in its realization. Peter Finin helped me a great deal in understanding hybrid systems and reachability analysis, as well as helped me with general coding and input issues. Adam Halsasz and Oleg Sokolsky were also very helpful in pointing me in the right direction on the project. I would also like to thank Professors Vijay Kumar and Jan Van der Spiegel for making this internship possible, as well as the generous people at the Microsoft Corporation for funding my project. They have all been very supportive and helpful over the course of this project and I extend my most sincere thanks to all of them.