

SSL Freeform Generator v1.00

Michael Kuchnik

CAIDA Lab

NASA KSC

CAIDA Lab Background

- Customer Avionics Interface Development & Analysis (CAIDA) lab simulates data for Launch Control Systems (LCS) to increase confidence in systems which are dependent on Orion hardware.
- Focuses on performance testing LCS systems and channelization of displays.
- Hopes for the future include simulating entire systems.

Purpose

- Too time consuming to write SSL code by hand for the large simulation requirements of the CAIDA lab.
- Previous versions of the CAIDA lab SSL generators were limited in waveform display capability.
- CAIDA lab would benefit from higher fidelity simulations.

Purpose

- NOT to mimic the functionality of OrionSIG. KMR(NI3

Slide 4

KMR(NI3 OrionSIG has functionality which is not supported by the Freeform Generator. OrionSIG is waveform aware in the sense that it scales waveforms to individual CUIs and is, in the end, easier to use.

OrionSIG is better for use in quick performance and channelization testing.

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

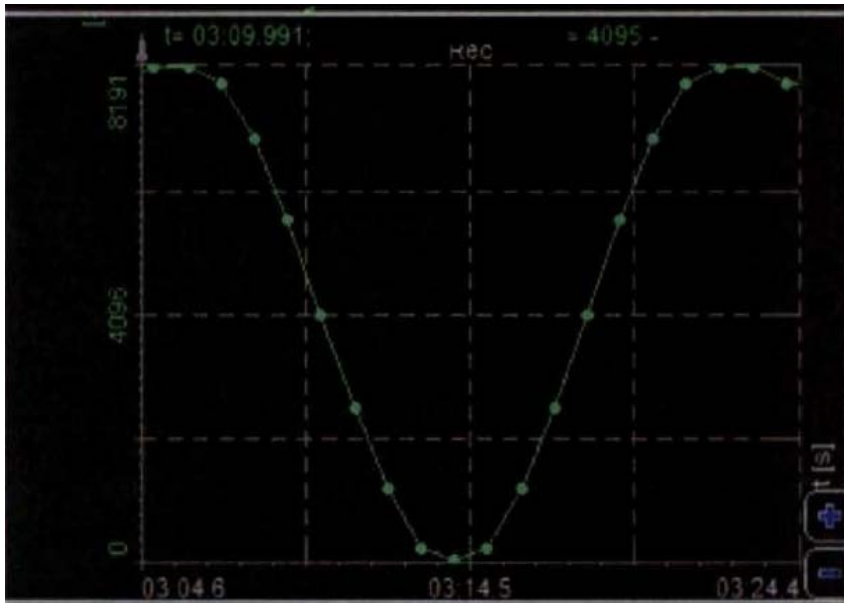
OrionSIG

- Presented software routines for generating waveforms.
- Any new waveforms required software modifications.
- Supported waveforms:
 1. Ascending sawtooth
 2. Cosine
 3. Descending sawtooth
 4. Flip bits
 5. Horizontal line
 6. Middle triangular wave
 7. Sine wave
 8. Square wave
 9. Triangular wave

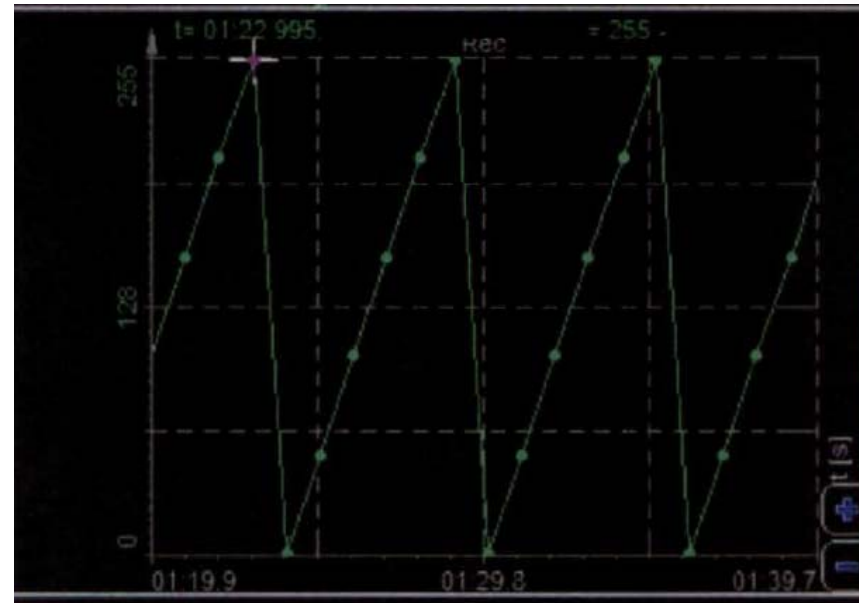
OrionSIG information taken from:

Robert J. Dooling. (2013, July 29) Orion Scripted Interface Generator (OrionSIG) [Online].

OrionSIG Examples



Cosine Waveform



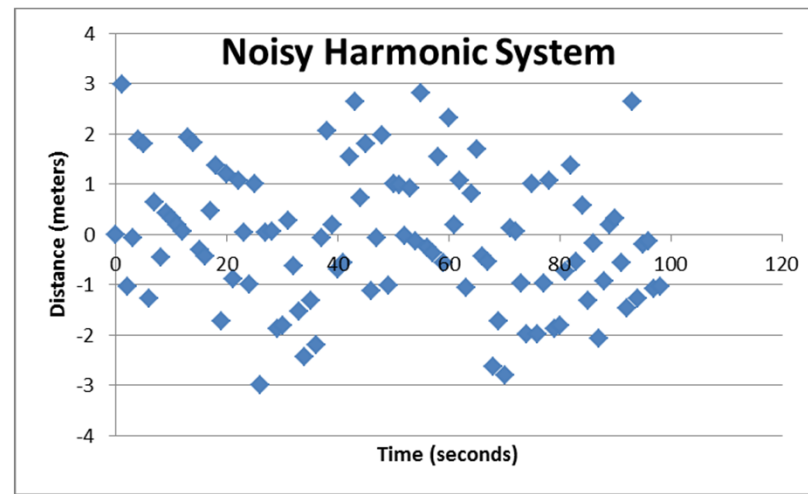
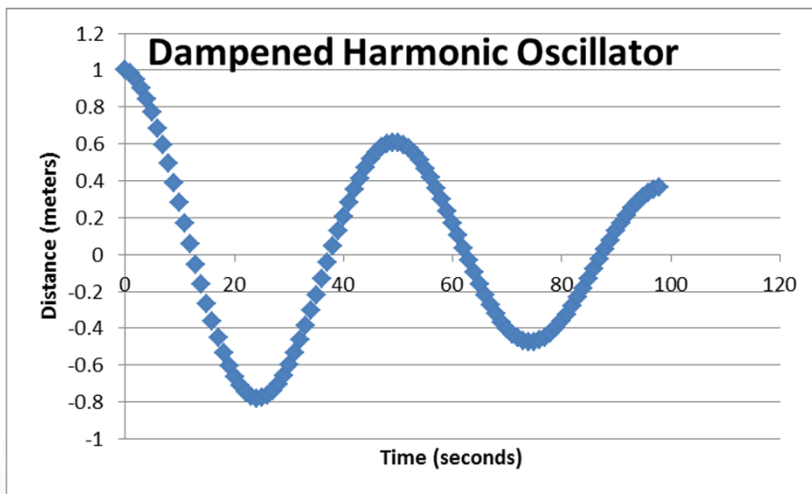
Ascending Sawtooth Waveform

Images from:
Robert J. Dooling. (2013, July 29) Orion Scripted Interface Generator (OrionSIG) [Online].

OrionSIG Feasibility

- What can be modeled with the available waveforms? KMR(NI4)
- Should software be updated to increase simulation options?
- Will new routines work on all CUIs? KMR(NI6)

KMR(NI5)



Slide 7

KMR(NI4) Very few systems can be modeled by sinusoids and variants of square or sawtooth waves.

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

KMR(NI5) Even systems considered sinusoidal are not capable of being modeled with these waveforms

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

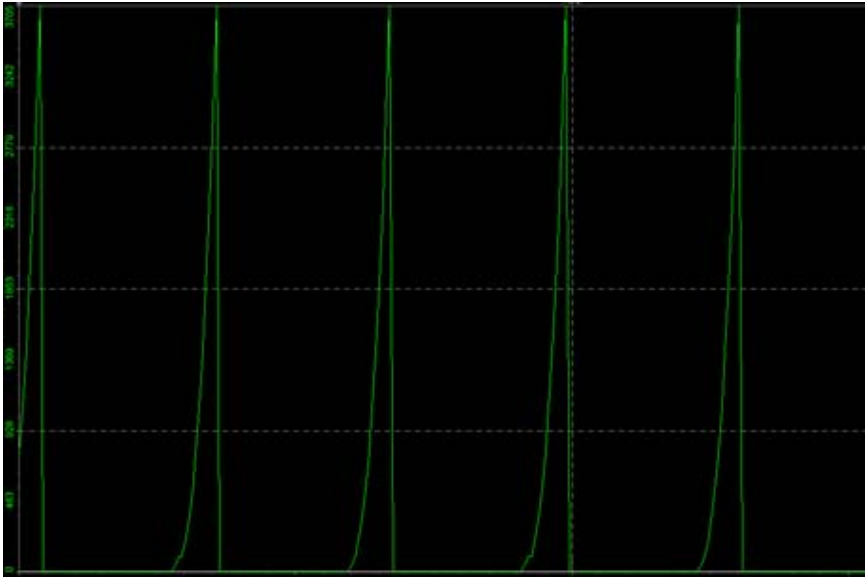
KMR(NI6) Different types and precision

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

SSL Freeform Generator

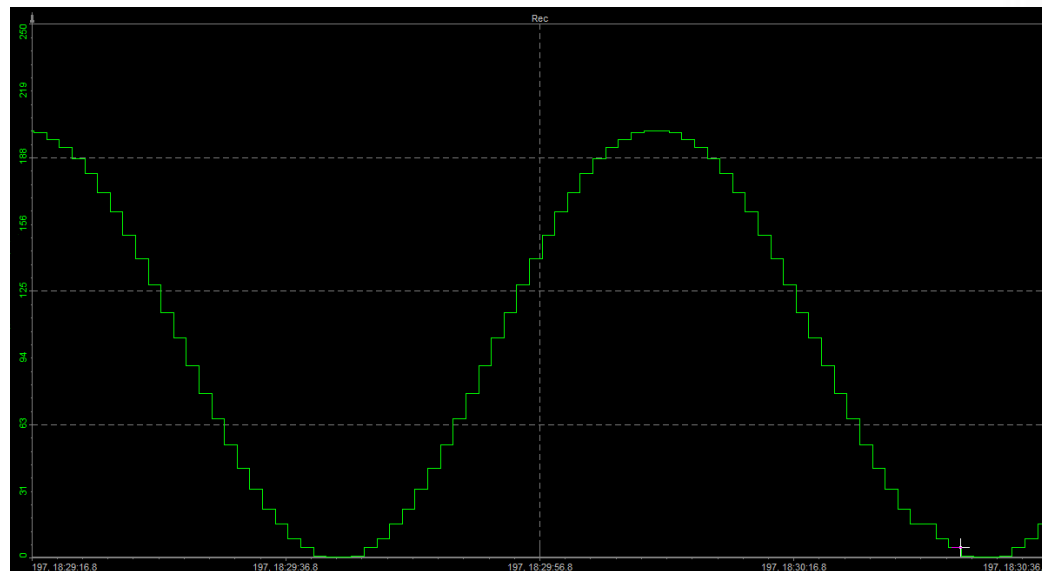
- Could improve on OrionSIG in allowing user defined waveforms.
- Waveforms could be added with no software modifications.
- Allows for a near unlimited amount of flexibility.

SSL Freeform Examples

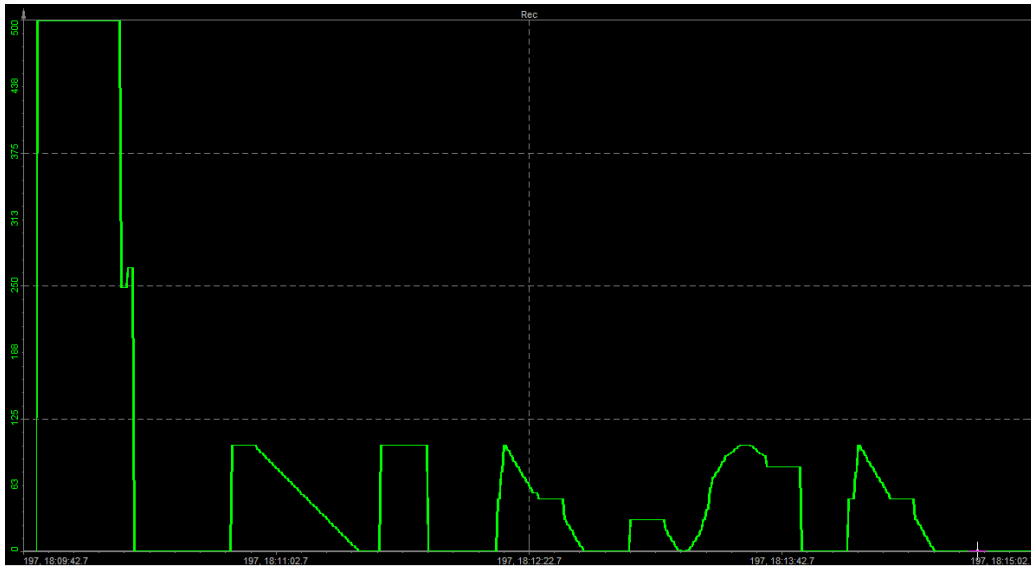


$$F(x) = \text{ABS}(-1 X^3 + \text{COS}(X^3) - 9 * X^2 + X)$$

$$F(x) = \text{COS}(\theta)$$

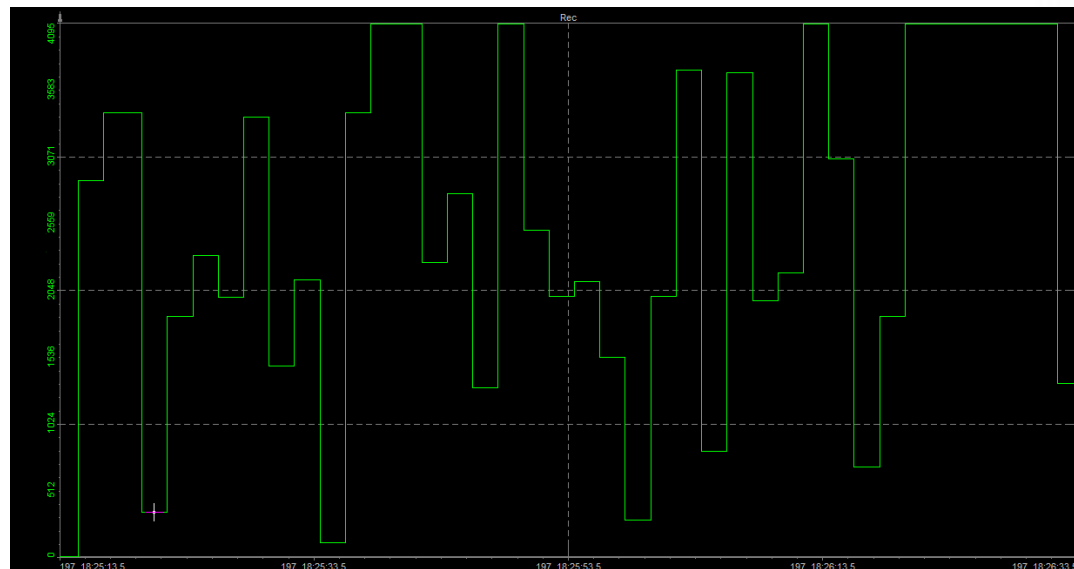


SSL Freeform Examples



Drawing NASA

Random data



What Stayed the Same

- The style of input interface is very similar.
- High level program structure is the same.
- End product SSL script is almost the same.

What Changed

- Low level implementation details.
- A few SSL script level optimizations.
- More expected from the user.
- More runtime checks and warnings.
- Large increase in simulation flexibility.

OrionSIG vs. FreeForm

KMR(N18)

Functionality	OrionSIG	FreeForm
Waveform and CUI aware	Scales waveforms to CUIs	Checks for user errors
User Inputs	1 Input File	1 Input File <small>KMR(N17)</small> Vs
Simulation capabilities	9 Waveforms	Any discrete function
Correctness Safety	Few chances for error	Error detection and warnings

Slide 13

KMR(NI7 Where N is the amount of CSVs
Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

KMR(NI8 FreeForm is slightly harder to use
Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

How to Use

- The CUI database file should be in the current directory.
- Create input file and all the waveform CSVs required and place in current directory.
- Read warnings and logs.
- Use output file.

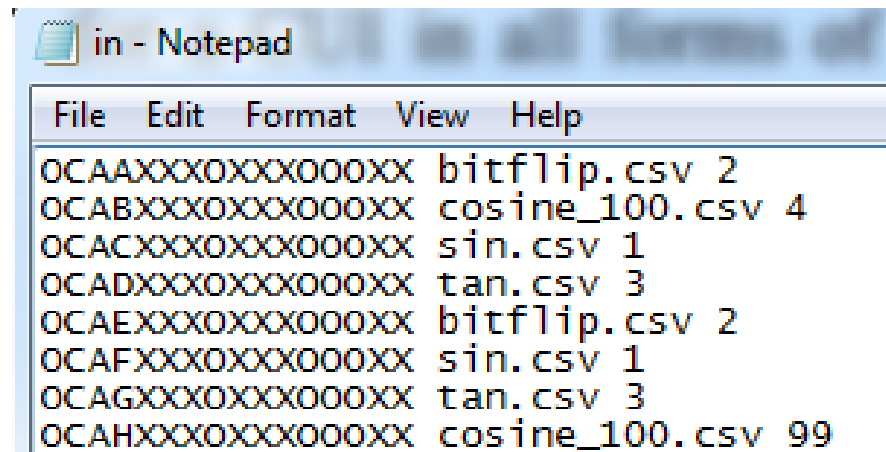
KMR(N110)

Slide 14

KMR(NI10 This is used to look up all references CUIs
Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

Input File

- The input configuration text file is used to enter all user-defined input.
- The file takes the format <CUI> <waveform CSV> <timestep>.

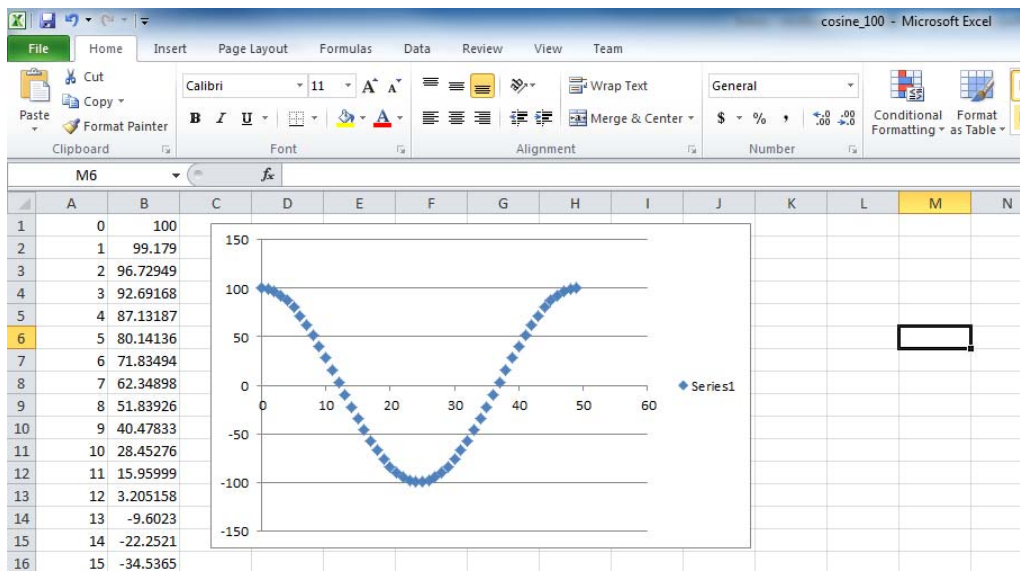


The image shows a screenshot of a Notepad window titled "in - Notepad". The window contains a list of input configurations, each consisting of a unique CUI (Case User ID), a waveform CSV file name, and a timestep value. The configurations are as follows:

```
OCAAXXXXXXXXXXXXXX bitflip.csv 2
OCABXXXXXXXXXXXXX cosine_100.csv 4
OCACXXXXXXXXXXXXX sin.csv 1
OCADXXXXXXXXXXXXX tan.csv 3
OCAEXXXXXXXXXXXXX bitflip.csv 2
OCAFXXXXXXXXXXXXXX sin.csv 1
OCAGXXXXXXXXXXXXX tan.csv 3
OCAHXXXXXXXXXXXXX cosine_100.csv 99
```

Waveform CSV

- The waveform CSV is stores the discrete points that make a waveform.
- Can export from Excel to CSV easily.



The screenshot shows a Notepad window with the following CSV data:

```
100,0
99.17900138,1
96.7294863,2
92.69167573,3
87.13187041,4
80.14136219,5
71.83493501,6
62.34898019,7
51.83925683,8
40.47833431,9
28.45275866,10
15.9599895,11
3.205157757,12
-9.602302591,13
-22.2520934,14
-34.53650544,15
```

Errors

- Errors are problems which are considered too severe to attempt to correct.
- Errors cause complete or partial program termination.
- Problematic inputs can be ignored.
- Common causes of errors:
 1. CUI lookup failure
 2. CUI read only
 3. Input file failure

Warnings

- Warnings are suggestions or notifications of problematic or corrected user input.
- Warnings are handled at runtime and require no user input.
- They should be inspected after program execution to make sure program behaved as intended.
- Common causes of warnings:
 1. Timestep smaller than 1 Hz
 2. Data type significant loss of precision
 3. Dangerous assignments of data

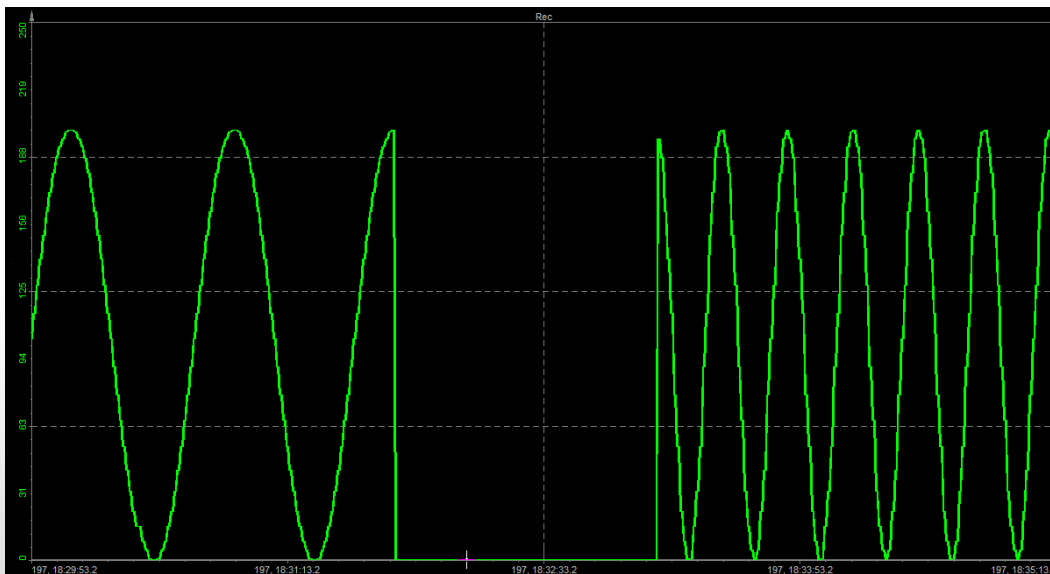
User Exposed Problems

- Undersampling
- Saturation
- Loss of precision



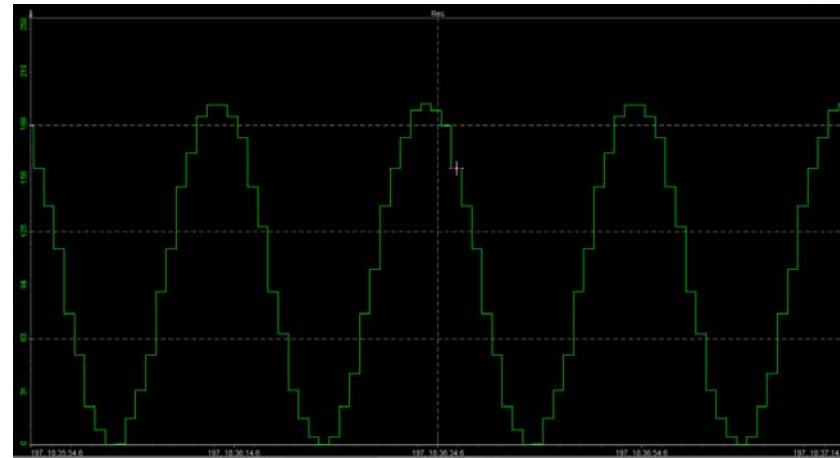
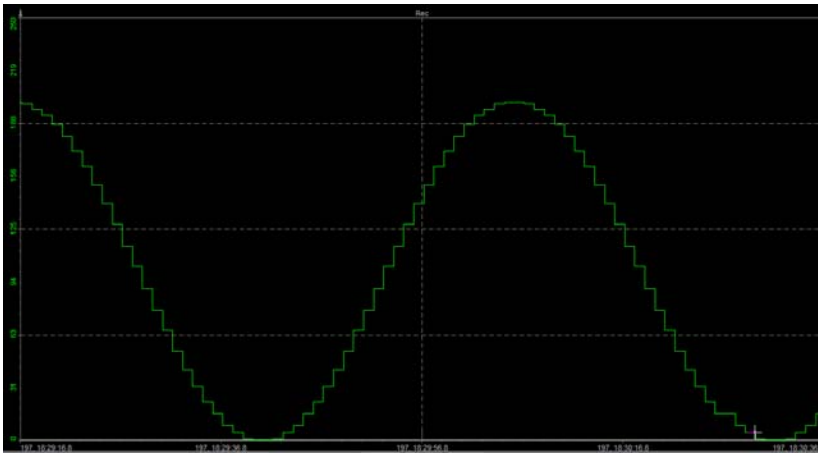
Sampling Example

Saturation Example



User Exposed Problems

- Loss of precision and data
- Aliasing



Changing the timestep to a rate less than the sampling rate causes points to be skipped KMR(N19)

Slide 20

KMR(NI9 Skipping enough points will make a waveform look like another
Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/17/2014

Logs

- Check logs!
- Present information useful for debugging and checking the integrity of output.
- Will be overwritten if the program runs again and the filenames remain unchanged in the current directory.

Ideas for the Future

- Implement a GUI frontend for controlling user input to increase program accessibility.
- Increase simulation fidelity to improve simulKMR(NI1) quality.
- Decrease the program's input dependency on user knowledge to improve ease of use.
- Create a format for waveforms that is both easy to use and robust to facilitate waveform generation.
- Collect data on various systems with the intent of creating accurate simulation models.

Slide 22

KMR(NI1) Double data tyoe

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/14/2014

KMR(NI2) User must understand CUIs and SSL Generator

Create front end to shape CUIs with error checking

Kuchnik, Michael R. (KSC)[KSC NASA Intern], 7/14/2014

Questions