

# Feedback Simulator Users Guide

## Table of Contents

Introduction.....	3
Program Details.....	3
Options.....	4
Feedback Loop.....	5
Feedback Loop Options.....	6

# Feedback Simulator Users Guide

## Introduction

The Feedback Simulator utility is a small program to allow exploring the effects of a close loop inspection system where the inspection equipment is used to provide feedback to the machine tool. The goal of such a system is to produce perfect parts where errors are corrected automatically without intervention.

The Feedback Simulator program is cross platform and can be run on OSX, Linux, and Windows.

## Program Details

The goal of the program is to mimic the effect of having a machine tool produce a part with an adjustable error, proceed through inspection that produces a correction value which is returned to the machine tool.

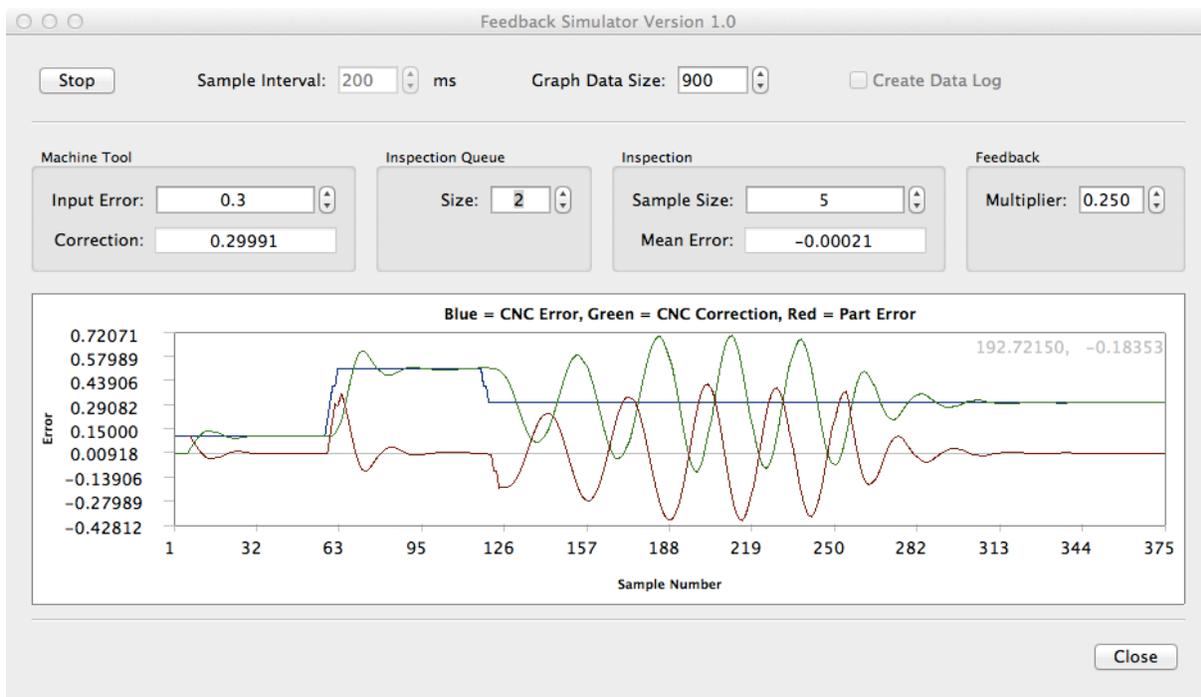


Illustration 1: Main window of Feedback Simulator Program

The graph shows a representation of the different parameters as they interact over time. The three parameters shown on the graph are the following:

- **Blue.** This represents the value of the input error parameter from the machine tool. A change in this value will appear immediately on the graph.

*Although the change is technically a perfect step from one level to the next the graphical representation will only show as a slope representing the value from one sample to the next.*

- **Green.** This data shows the correction value applied against the error. When stable the

# Feedback Simulator Users Guide

---

correction value should match that of the current input error.

- **Red.** This data is the results of the last part inspected. This is the newest part measurement from the inspection samples; this is the part the was most recently created by the machine tool.

The four main areas of the utility program are this:

## Machine Tool

This represents the machine tool that produces the part. A user adjustable input field called 'Input Error' allows a simulated part to be produced with an error ranging in value from -1 to 1. The correction field shows the compensation that is applied against the input error field which is the end result of the feedback.

## Inspection Queue

This section represents the number of parts that are held between the machine tool and before inspection. In some cases it may be possible to have the parts produced by the machine tool inspected immediately but not always. It may be necessary to allow the parts time to thermally stabilize for example. This field represents how many parts are produced before reaching inspection.

## Inspection

The inspection section is where the part measurement is performed. The sample size represents the number of parts used to produce the mean error shown in this section. A sample size of five parts means that the average error from the last five parts is used for the correction value.

*Currently the measurement error is calculated as only the mean error of all samples contained within inspection. The number of parts is set by the variable Sample Size.*

## Feedback

The feedback section allows the user to set a multiplier of the measurement error that is added to the current correction value from the machine tool. This is the last step and effectively closes the loop.

## Options

The following is a description of the various options use by the Feedback Simulator utility program.

<i>Option</i>	<i>Description</i>
Start Stop	Start or stop the simulation.
Sample Interval	The time in milliseconds between simulated sample parts.

## Feedback Simulator Users Guide

Graph Data Size	This sets the maximum number of samples shown on the data graph. When the data exceeds this limit the oldest graph data is removed.
Create Data Log	If checked an output log file is created showing the input error, mean inspection error, correction, and other relevant simulation parameters.
Close	Close the program.

### Feedback Loop

The feedback loop used is based on the idea of using the average value of a series of measurements done by inspection. Since this is the most common method used to implement this kind of closed loop feedback system it was decided to use this option.

When testing it is evident that the feedback loop can become very unstable with only minor and seemingly innocent changes. Considering how sensitive the loop can be this method is probably not the best method to implement this kind of closed loop control. It is the first option typically proposed by anyone who had tried to implement this type of feedback control.

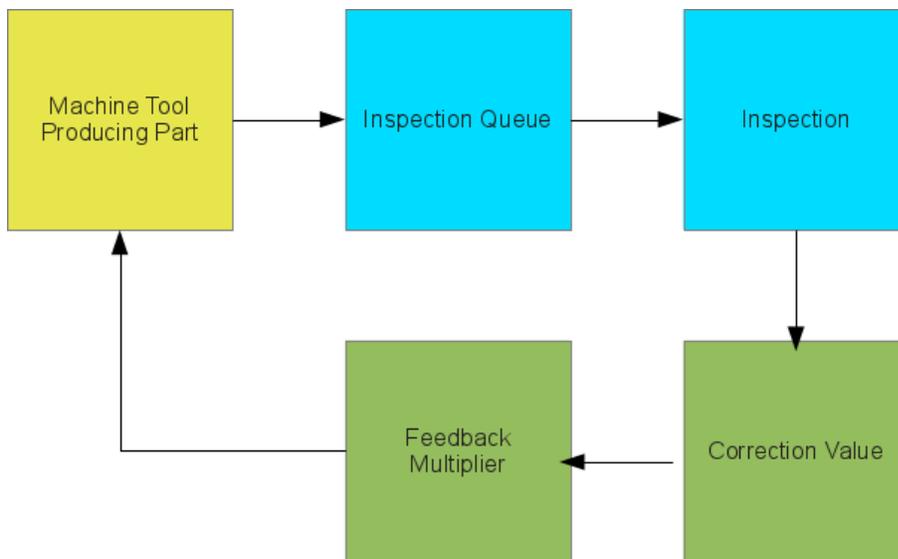


Illustration 2: Feedback Loop diagram

The machine tool produces parts based on this formula:

$$part_{error} = input_{error} - correction_{current}$$

The inspection section calculates the part error as the average of all currently held samples:

$$inspection_{error} = \sum (part_{error}) / samples$$

The correction value that is returned to the machine tool is this:

$$Correction_{new} = correction_{current} + inspection_{error} * feedback_{multiplier}$$

# Feedback Simulator Users Guide

---

## Feedback Loop Options

Based on testing it appears some options can be implemented to improve the stability of the feedback loop. These are ideas and not currently implemented in the Feedback Simulator program but may help improve the stability of this type of system.

- Limit changes to small incremental updates only and only apply these if the change exceeds a minimum value.
- Provide feed forward information from the machine tool when a change has been made so that the next correction values take into account the new change blindly for a minimum number of parts (the number of parts in the queue and average).
- Block all future changes until a minimum number of parts have passed through the inspection process. Essentially this opens the control loop to clear the effect of the change until the full effect is realized.
- Use of a weighted filter on the inspection side where the last parts produced by the machine tool are given a higher priority than the earliest parts produced. Currently the Feedback Simulator program uses the average of all parts with equal weight for each part.
- Use a very small feedback multiplier. This would slow down the rate of change and stabilize the loop but will increase the number of parts produced with any given error.