Penn State
Reactivity Computer Upgrade

TRTR 2017

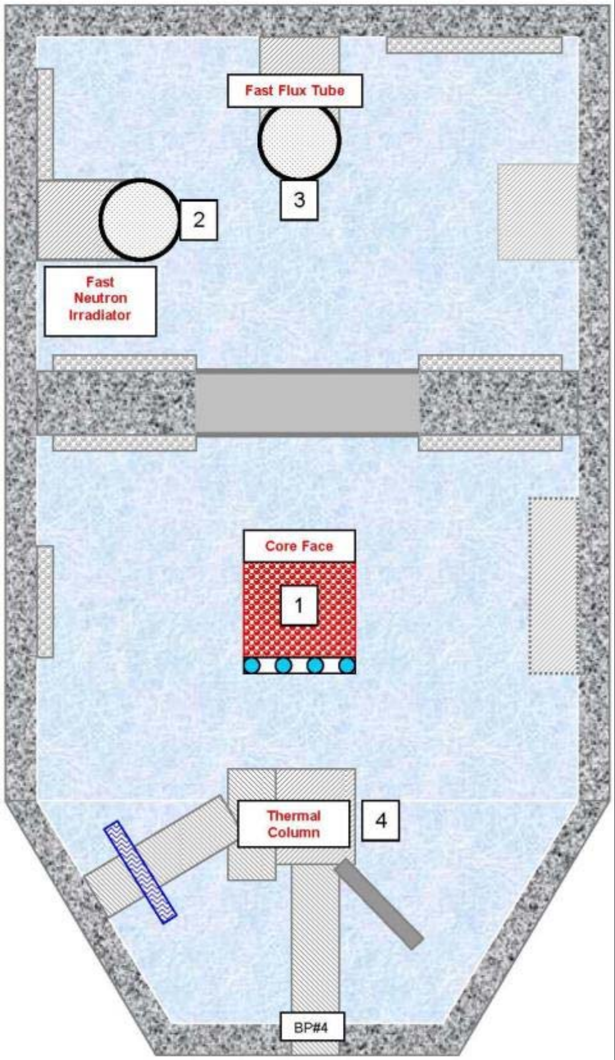
PennState

James Turso
Assistant Director
Radiation Science and Engineering Center

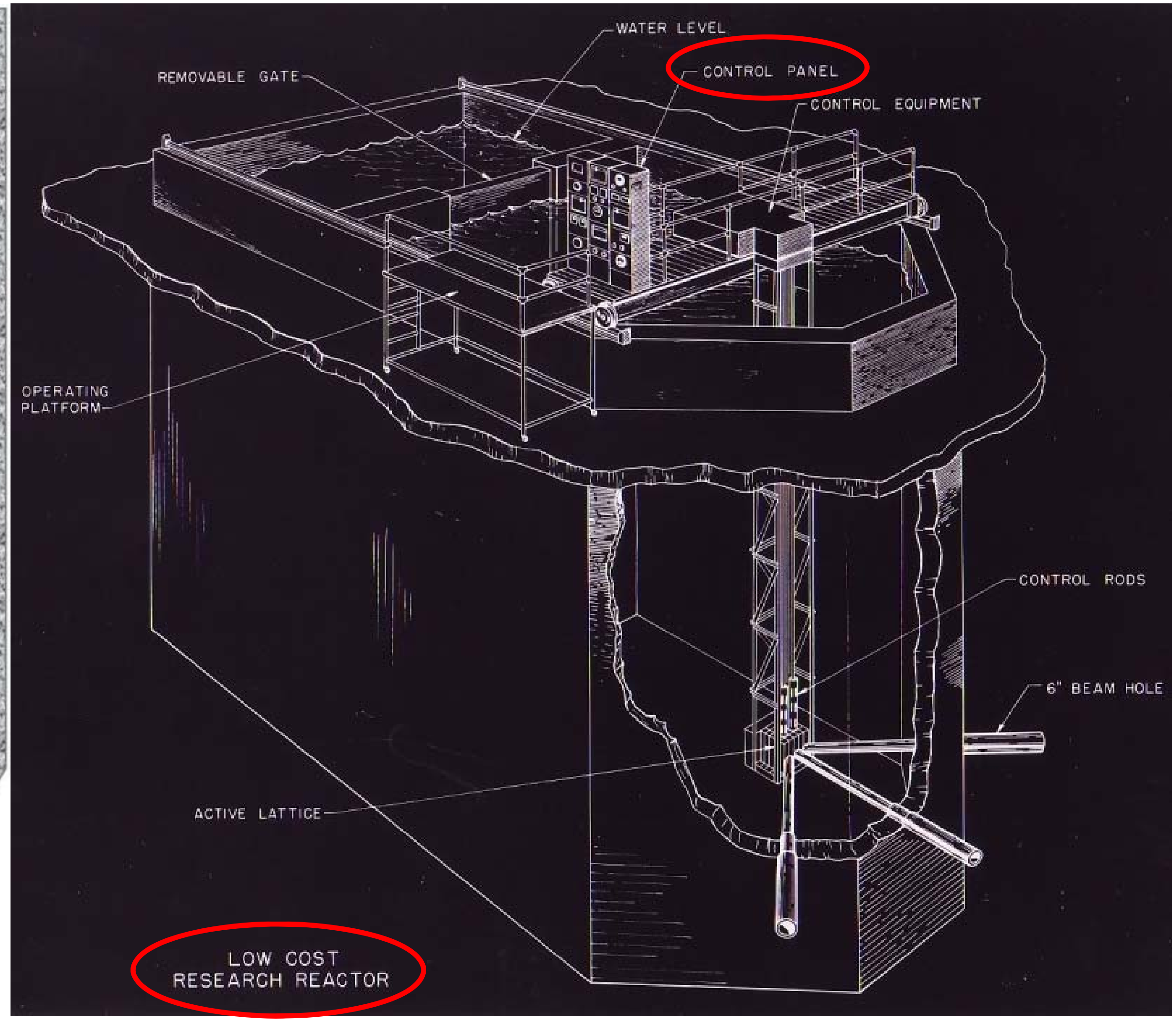
Overview

- PSBR TRIGA Description
- History…If it ain't broke….Motivation
- Existing Rod Cal Procedure
- RC PKE-Based Calculation
- LabView Implementation (highlights) - placement guidance for 3 CR not being calibrated
- Results – comparison to previous RC
- Conclusion

PSU TRIGA



Fast Flux Tube
3
2
Fast Neutron Irradiator
Core Face
1
Thermal Column
4
BP#4

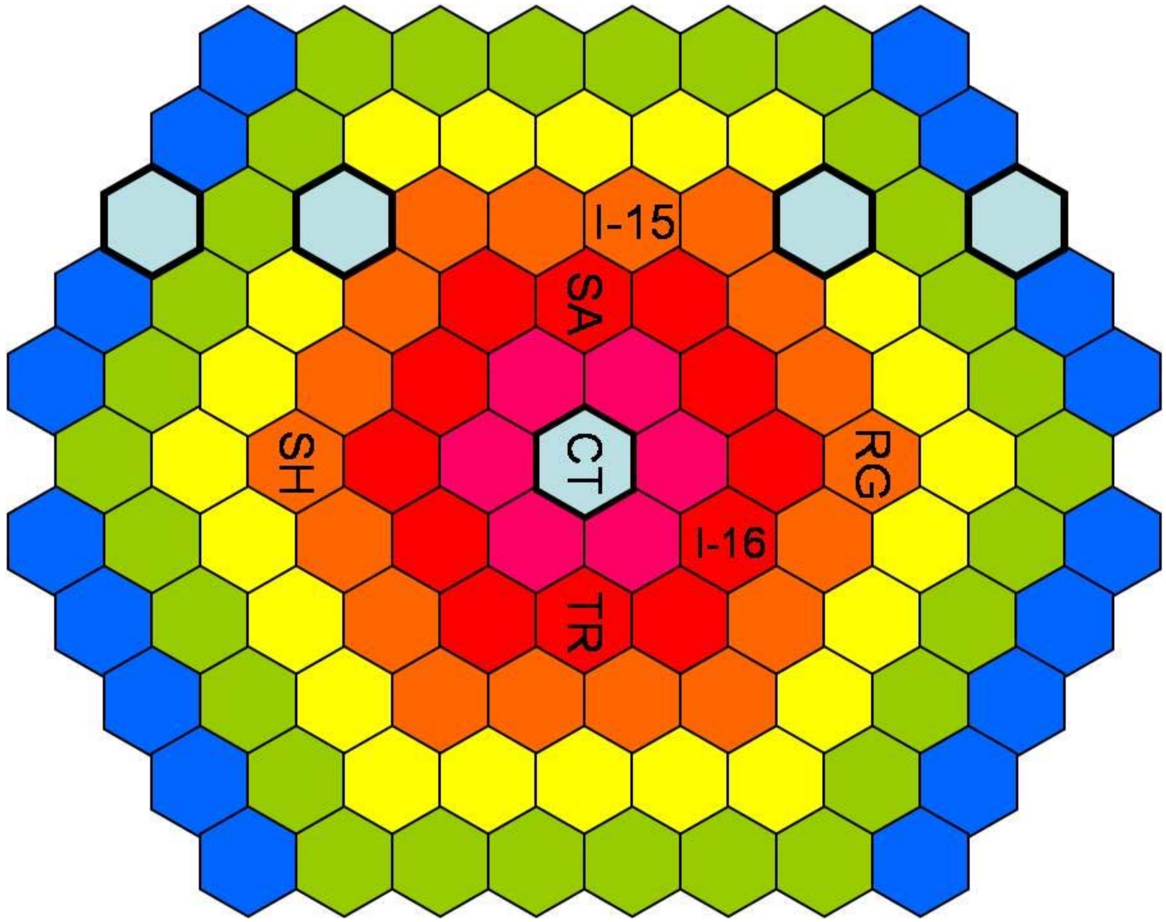


WATER LEVEL
CONTROL PANEL
REMOVABLE GATE
CONTROL EQUIPMENT
OPERATING PLATFORM
CONTROL RODS
6" BEAM HOLE
ACTIVE LATTICE
LOW COST
RESEARCH REACTOR

PSU TRIGA CORE MAP

2 Dry irradiation tubes
1 wet irradiation tube
1 source location

4 Control Rods
2 inst. Elements



Background / History

- Stable Period Method (Inhour) used at PSU until RC developed - took 2 days to do yearly calibration of all rods.
- Legacy RC developed approx. 15 years ago – now takes <4 hours for all rods.
- Legacy RC only runs on one, older computer. Update rate (and results) linked to processor speed of host computer. Source code unavailable.
- Developed new version in “language” that’s easily understood by reactor staff (i.e., LabView).

Rod Worth Procedure with Reactivity Computer

- Spare CIC Positioned Next To Core

- Power supplies/pico-ammeter setup

- Go critical w/3 rods

- Linearity checks performed between 100W-1kW

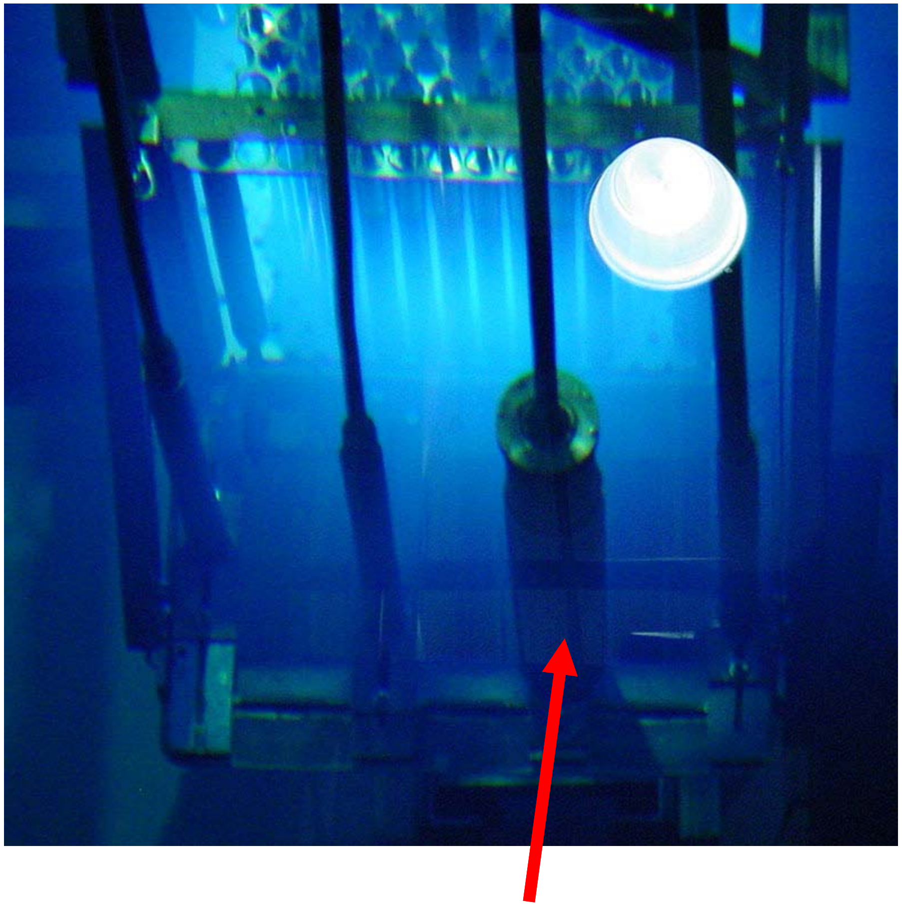
- Start test 100W, put 3 rods into manual

1) Shim-out cal'ed rod

2) Wait for RC to converge on constant delta-rho

3) With power < 1kW, shim in-other 3 rods (in manual) wait for stable negative reactivity

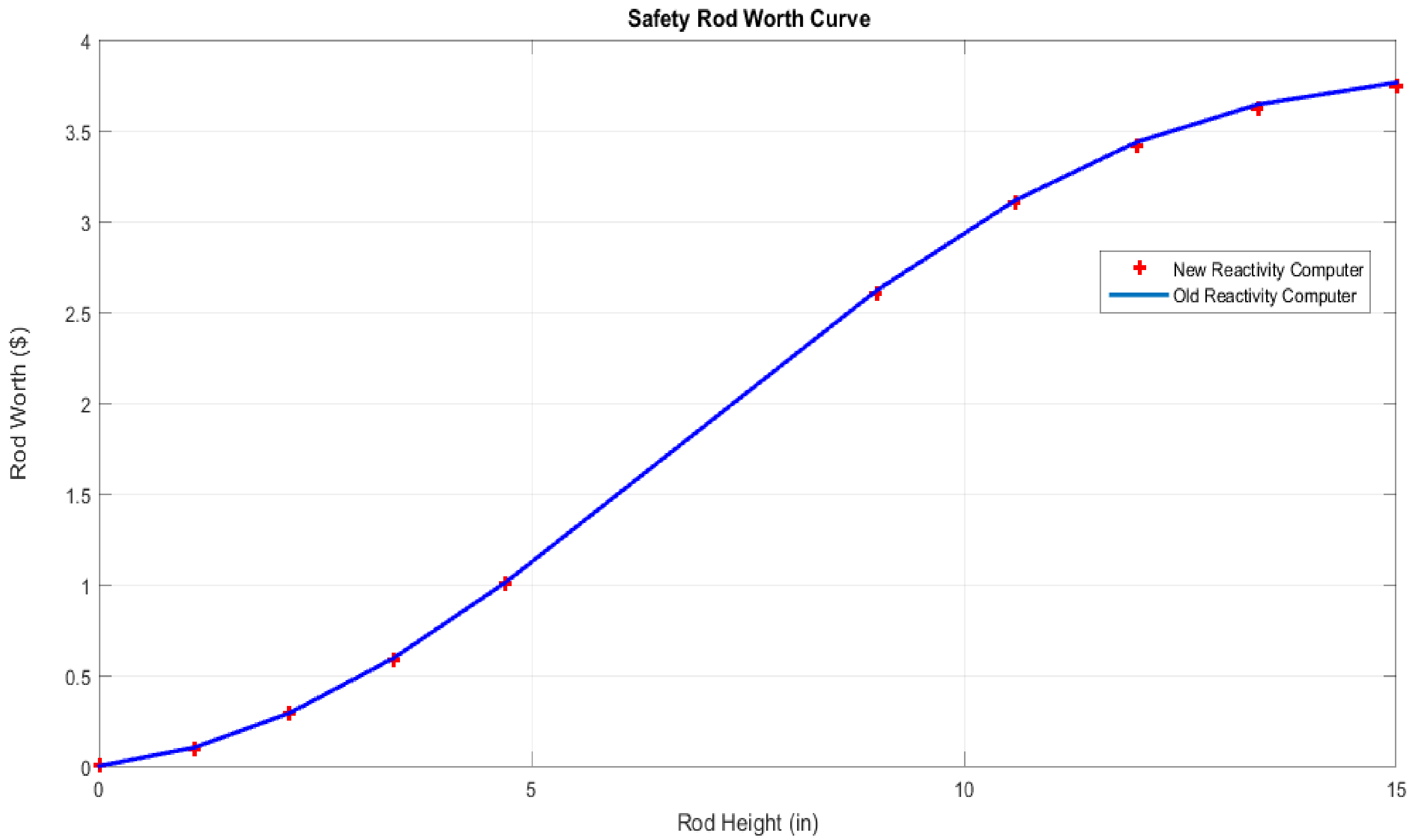
4) Repeat 1) - 3) until cal'ed rod fully withdrawn



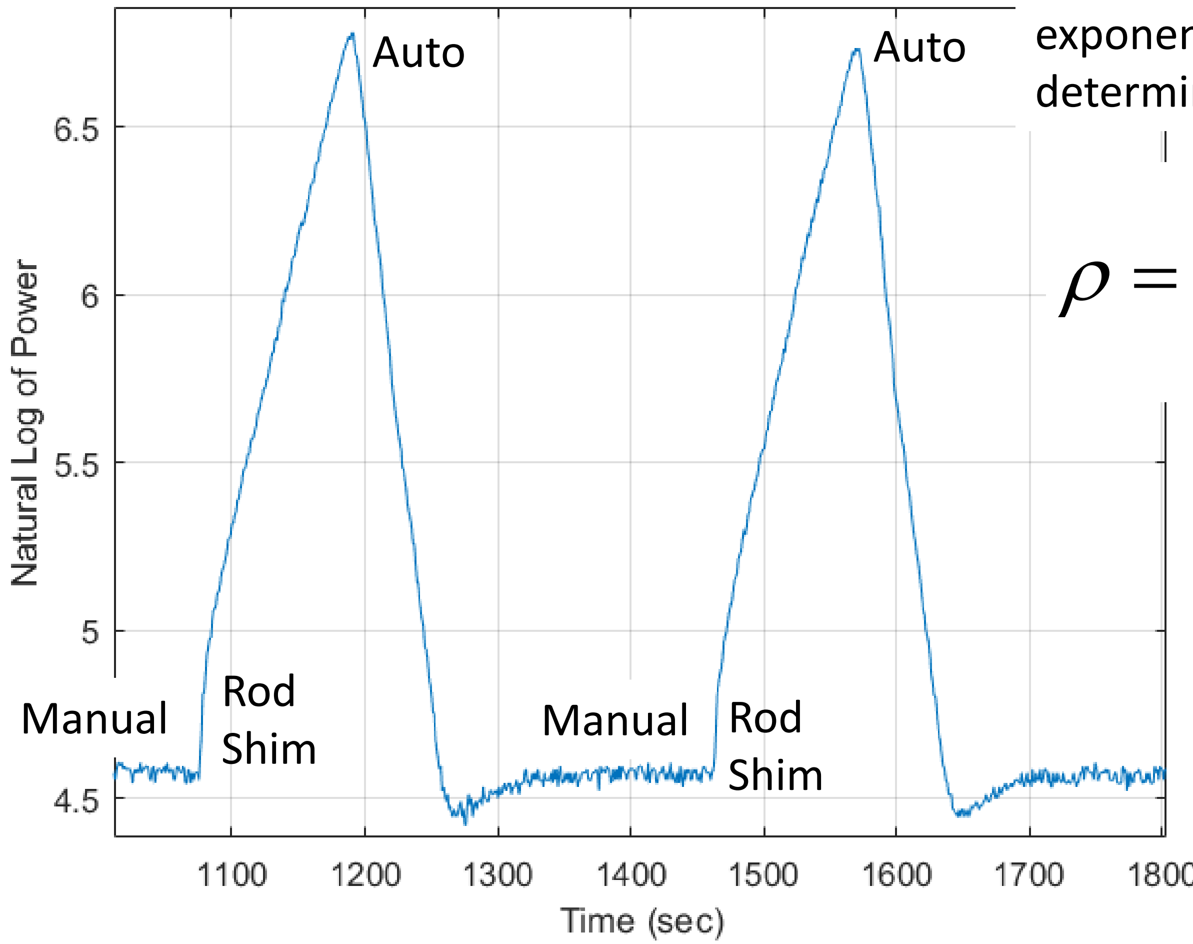
Compensated Ion Chamber

CR Worth Curve….Curve Fit Provides rho($) = f(Z)

$rho = 0.00044 + 0.00028*x + 2.50106E\text{-}6*x^2 + 6.73458E\text{-}10*x^3 - 1.99381E\text{-}12*x^4 + 5.83857E\text{-}16*x^5$



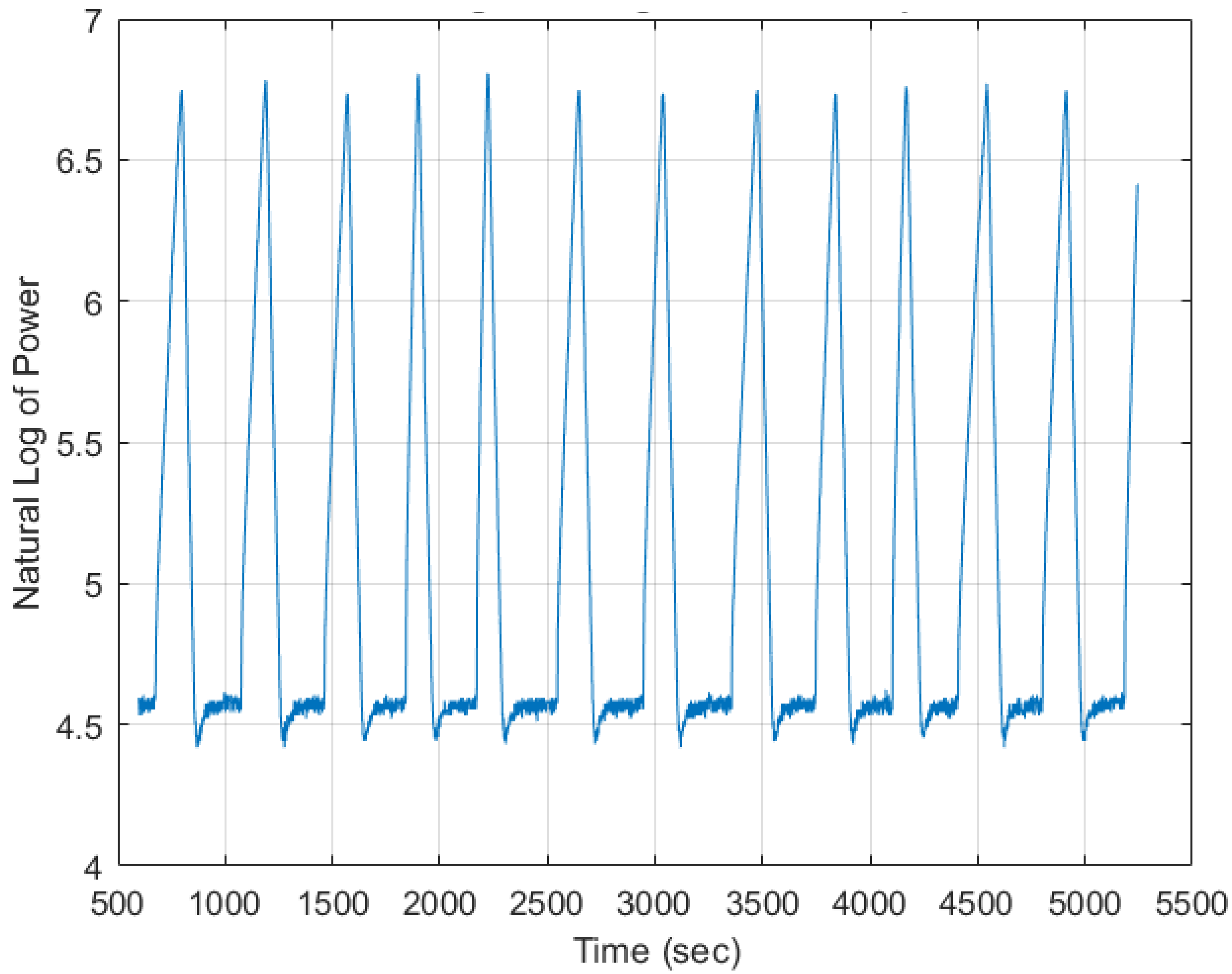
Rod Worth Measurement: Stable Period Method



In-Hour method uses points on exponential part of curve to determine reactor period/reactivity

$$\rho = \frac{\ell^*}{\tau} + \sum_{i=1}^{6} \frac{\beta_i}{1 + \lambda_i \tau}$$

Rod Worth Measurement: Stable Period Method



Reactivity Computer uses Normalized Version of Point Kinetics Equations

In-Hour method uses points on exponential part of curve to determine reactor period/reactivity

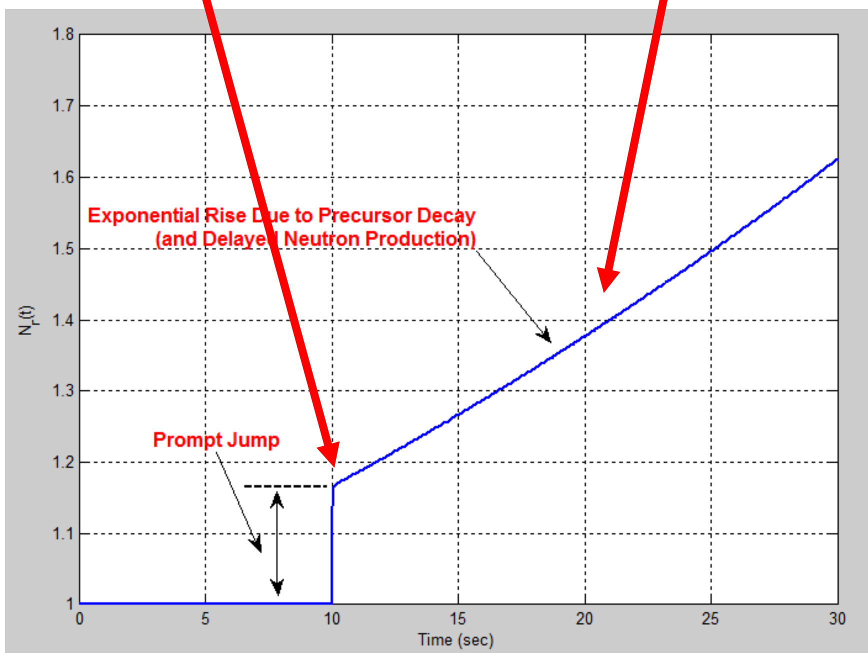
Point Kinetics Equations

$$\frac{dn}{dt} = \frac{\rho(t) - \beta}{\Lambda} n(t) + \sum_{i=1}^{i=6} \lambda_i c_i(t)$$

$$\frac{dc_i(t)}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_k c_i(t)$$

Good for small Rx below the point of adding heat…

Response of Point Kinetics Equations to Step Reactivity Insertion



Could use "Inverse" Point Kinetics Equations, BUT …

…noisy measurements result in noisier derivatives

$$\rho(t) = \beta + \frac{\Lambda}{n(t)} * \{\frac{dn(t)}{dt} - \sum_{i=1}^{i=6} \lambda_i c_i(t)\}$$

$$\frac{dc_i(t)}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i c_i(t)$$

Reactivity Computer uses Normalized Version of Point Kinetics Equations

- Use Normalized Point Kinetics Equations: $n_r(t), cri(t)$ = 1.0 at beginning of test
- This translates into using normalized, measured voltage in calculation
- Precursor concentration DE numerically integrated

$$\frac{dn_r(t)}{dt} = \frac{(\rho_{net} - \beta) * nr(t)}{\Lambda} + \frac{\sum_{i=1}^{i=6} \beta_i c_{ri}(t)}{\Lambda} \qquad n_r(t) = \frac{n(t)}{n(0)}$$

$$\frac{dc_{ri}(t)}{dt} = \lambda_i\big(n_r(t) - c_{ri}(t)\big) \qquad c_{ri}(t) = \frac{c_{ri}(t)}{c_{ri}(0)}$$

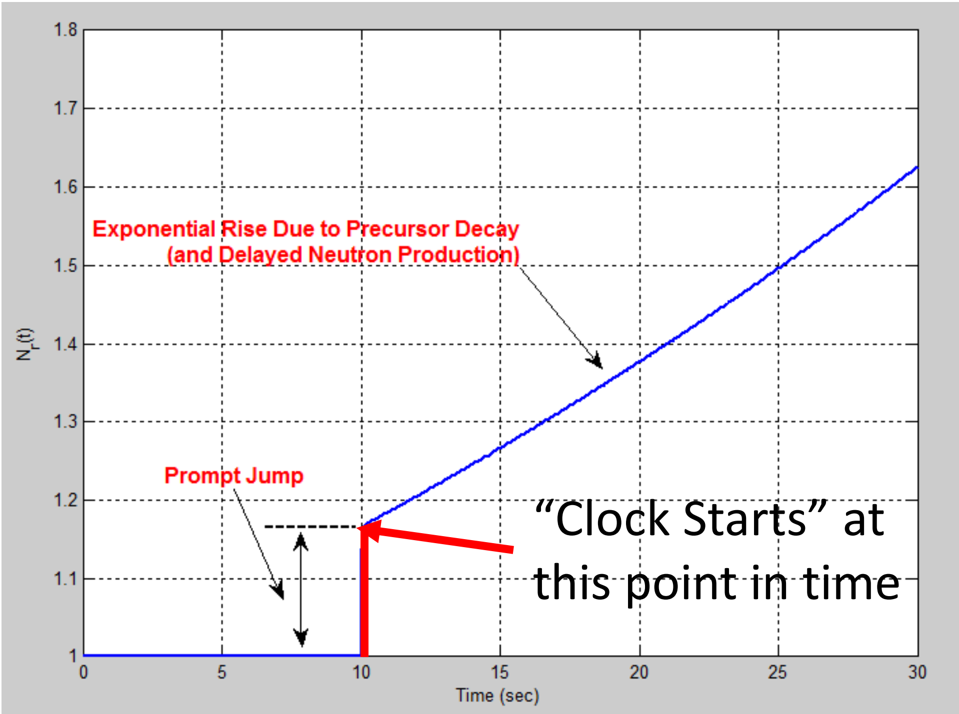
Reactivity Computer uses Normalized Version of Point Kinetics Equations/Prompt Jump Approximation

Traditional Use of Prompt-Jump Approximation

$$\frac{dn_r(t)}{dt} = 0$$

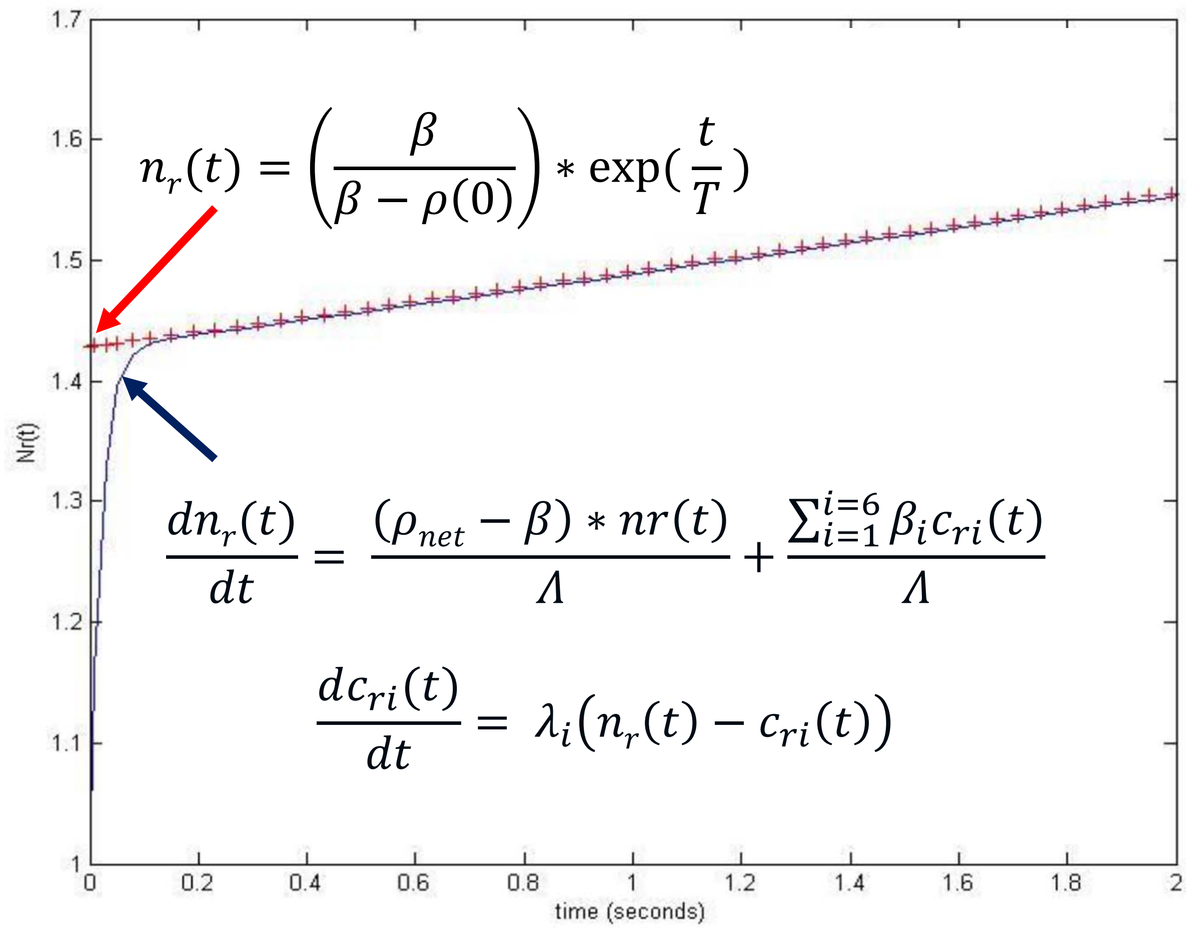
$$n_r(t) = \left(\frac{\beta}{\beta - \rho(t)}\right) * \exp\left(\frac{t}{T}\right)$$

$$T \rightarrow Reactor\ Period$$



Need to know $\rho(t)$ a-priori, and only good for a step input up to $0.25

Comparison of Normalized Version of Complete Point Kinetics Equations and Prompt Jump Approximation



Reactivity Computer uses Normalized Version of Point Kinetics Equations/Prompt Jump Approximation

Don't know Reactor Period beforehand, so apply Prompt-Jump Approximation at time of reactivity insertion to Normalized PKE and solve in LabView for reactivity corresponding to stable-period power rise observed….

$$\frac{dn_r(t)}{dt} = \frac{(\rho_{net} - \beta) * nr(t)}{\Lambda} + \frac{\sum_{i=1}^{i=6} \beta_i c_{ri}(t)}{\Lambda} \longrightarrow \frac{dn_r(t)}{dt} = 0$$

$$\frac{dc_{ri}(t)}{dt} = \lambda_i\left(n_r(t) - c_{ri}(t)\right)$$

$$n_r(t) = Input$$

$$\rho(t) = Output$$

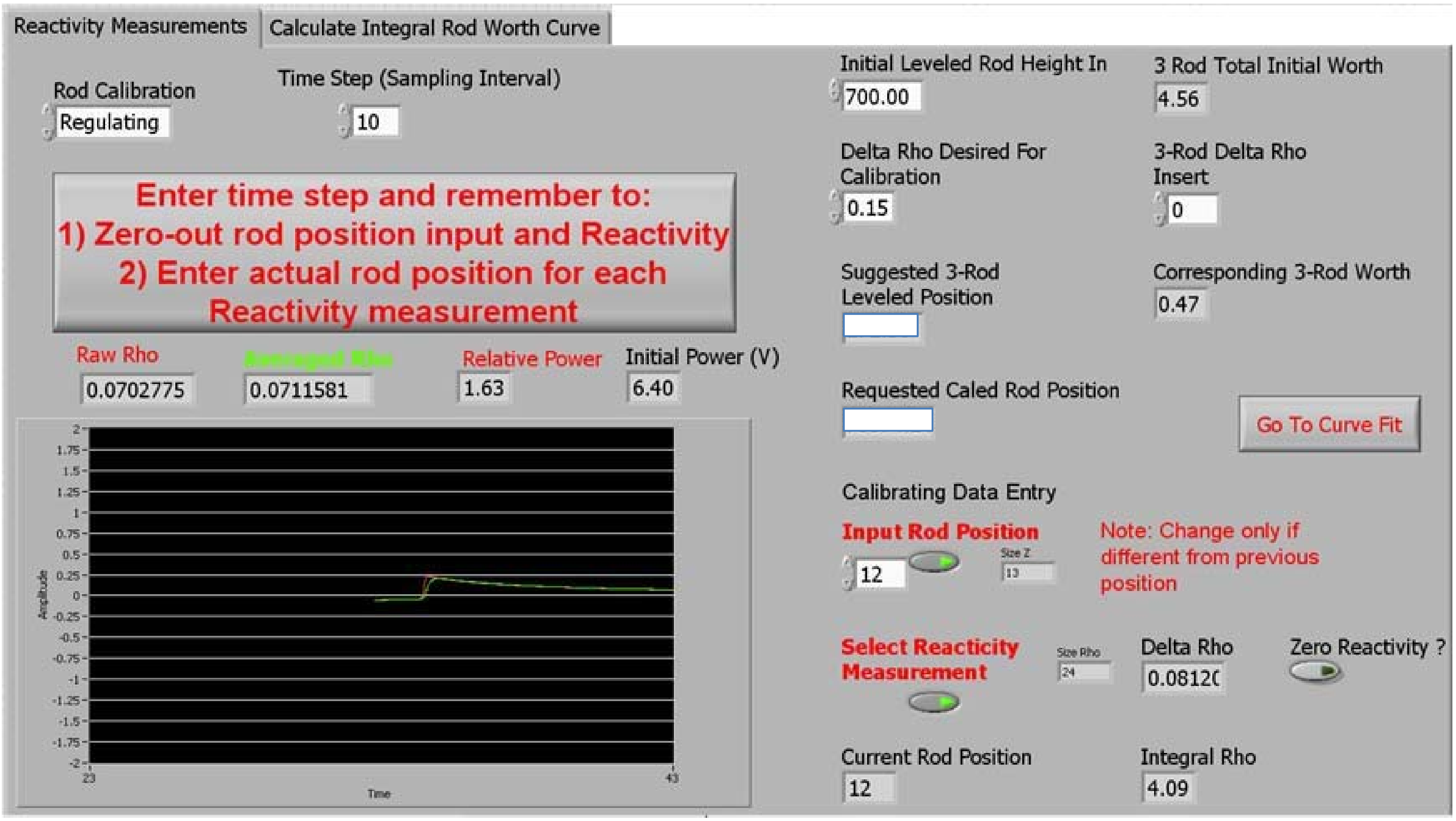
$$n_r(t) = \frac{V(t)}{V(0)}$$

Numerically Integrate in LabView

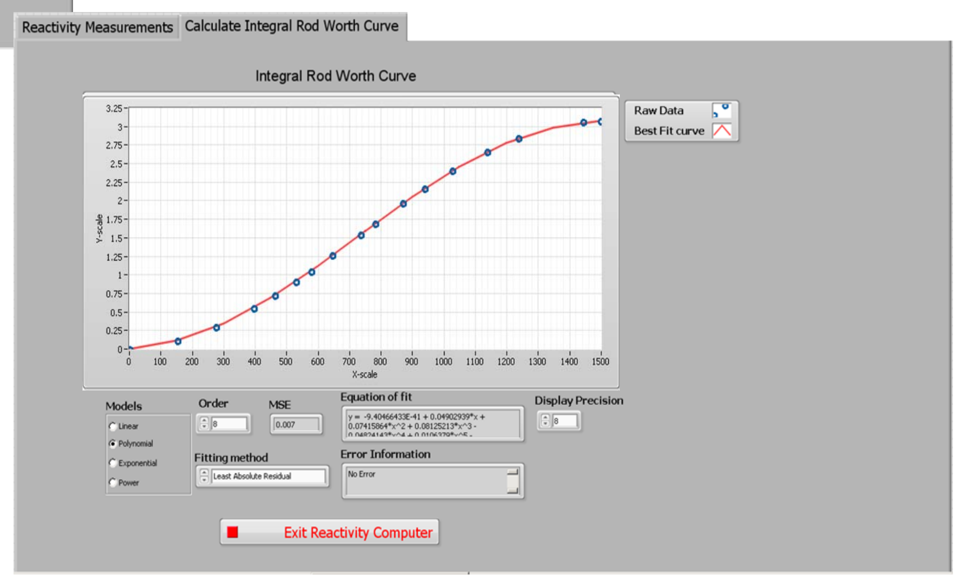
$$\rho(t) = \beta - \frac{\sum_{i=1}^{i=6} \beta_i c_{ri}(t)}{n_r(t)}$$

$$\frac{dc_i(t)}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i c_i(t)$$

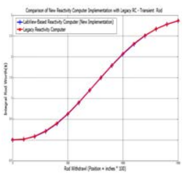
Implementation in LabView: Front Panel



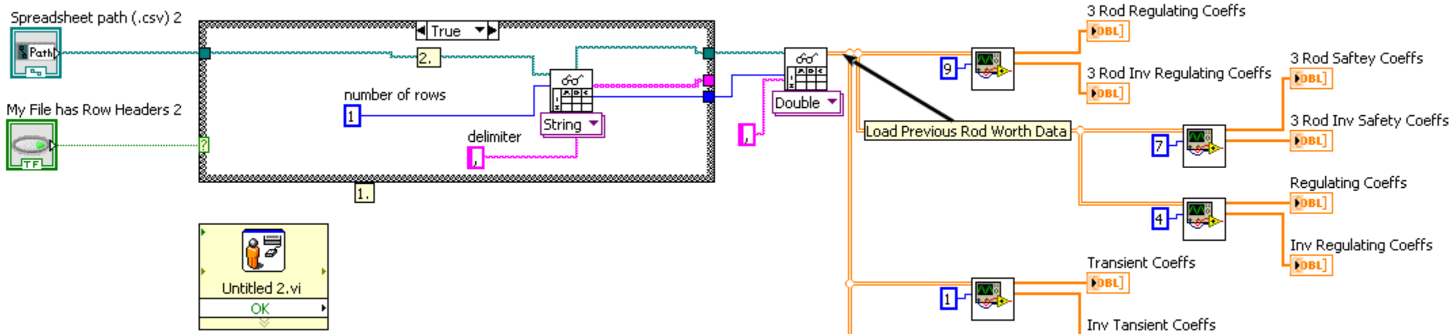
Implementation in LabView: Front Panel



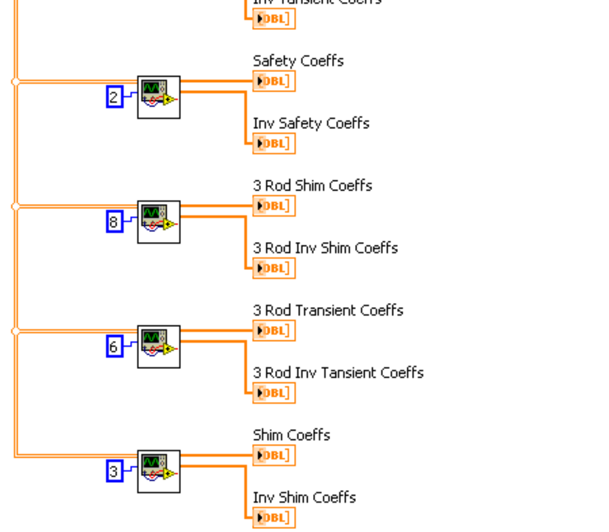
Implementation in LabView: Rod Position Estimator - Load Previous Rod Worth Data



Rod Worth Data In

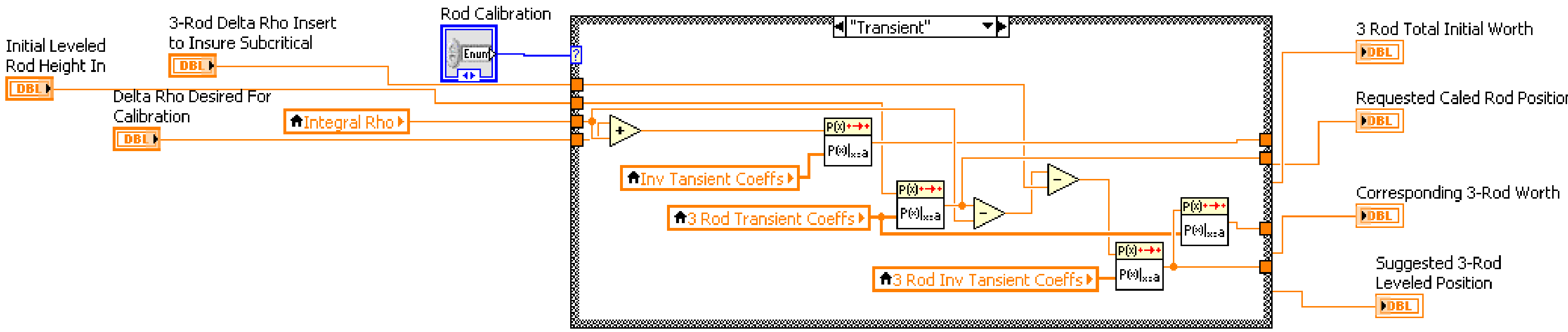


- “Forward” RW Curve Fit (rho = f(z))
- “Backward” RW Curve Fit (z = f(rho))
- Used to provide operator with position placement guidance for 3 CR used to offset calibrated rod reactivity insertion
- Only calculated once during initialization of RC



Implementation In LabView: Rod Position Estimator

- Used to provide operator with position placement guidance for 3 CR used to offset calibrated rod reactivity insertion



For 3 CR used to offset calibrated CR reactivity inserted:

1) Operator inputs 3 CR leveled initial position (only once) - “Forward” RW Curve gives total worth of 3 DR.
2) Integral RW of calibrated CR subtracted from 1).
3) Subtract an additional amount to ensure sub-criticality upon insertion.
4) Use “Backward” RW Curve to give new, suggested 3 CR estimated position

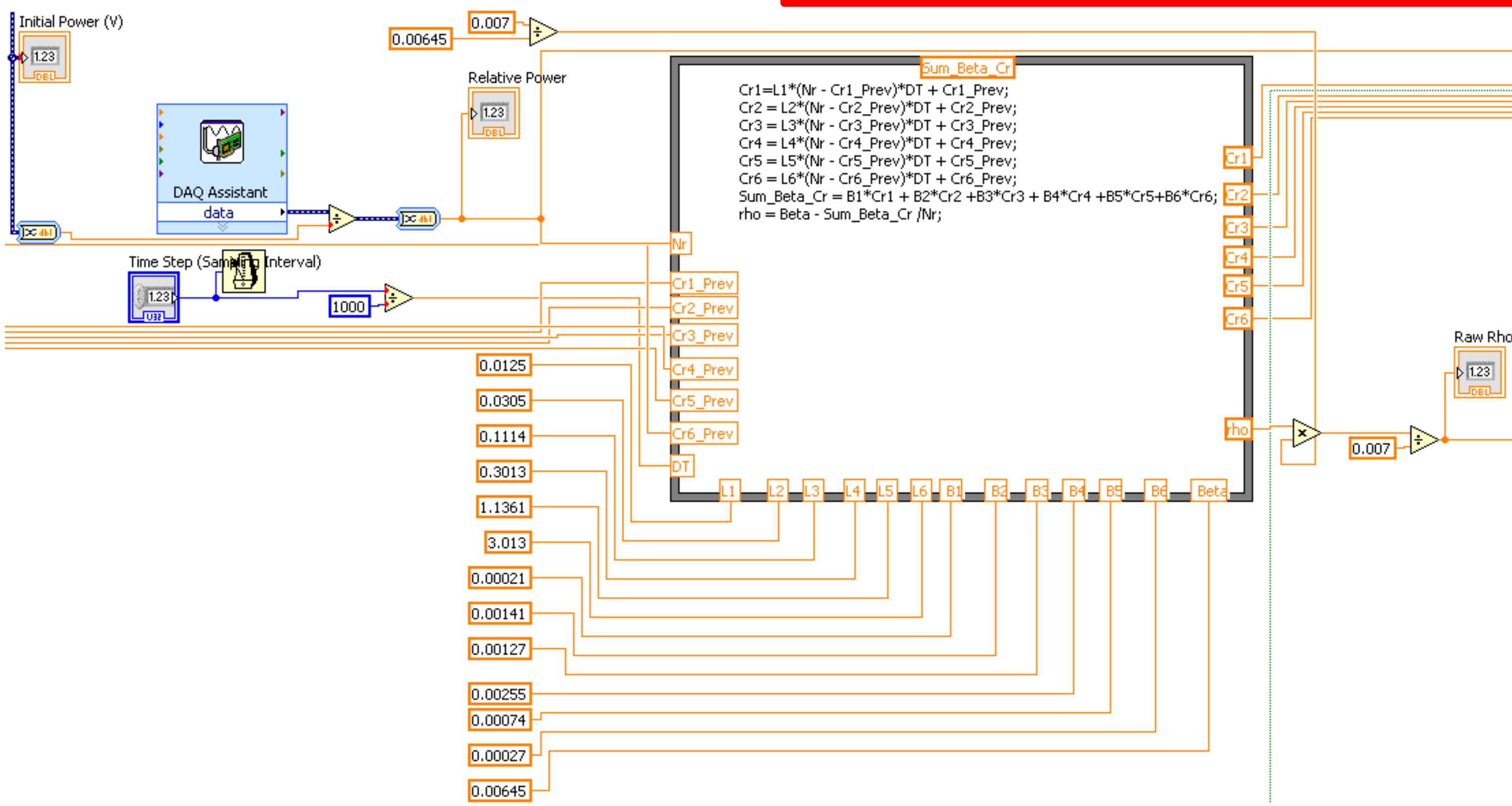
For calibrated CR:

1) Operator identifies desired Delta-Rho insertion from calibrated CR.
2) 1) gets added to previous pull’s integral CR worth.
3) 2) uses “Backward” CR curve to estimate where to position calibrated CR for next reactivity measurement.

Implementation in LabView:
Differential Rod Reactivity Calculation

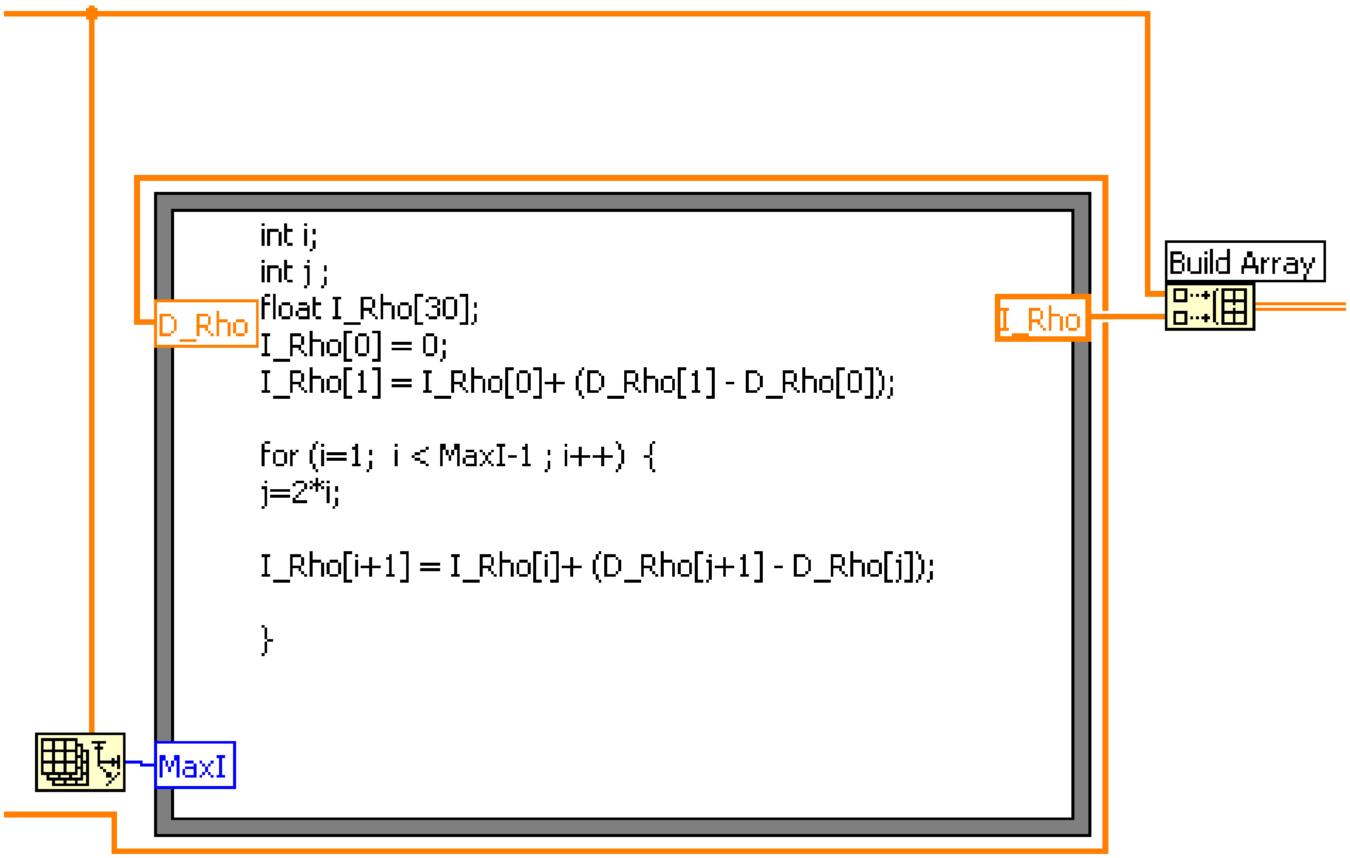
$$\rho(t) = \beta - \frac{\sum_{i=1}^{i=6} \beta_i c_{ri}(t)}{n(t)}$$

$$\frac{dc_i(t)}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i c_i(t)$$



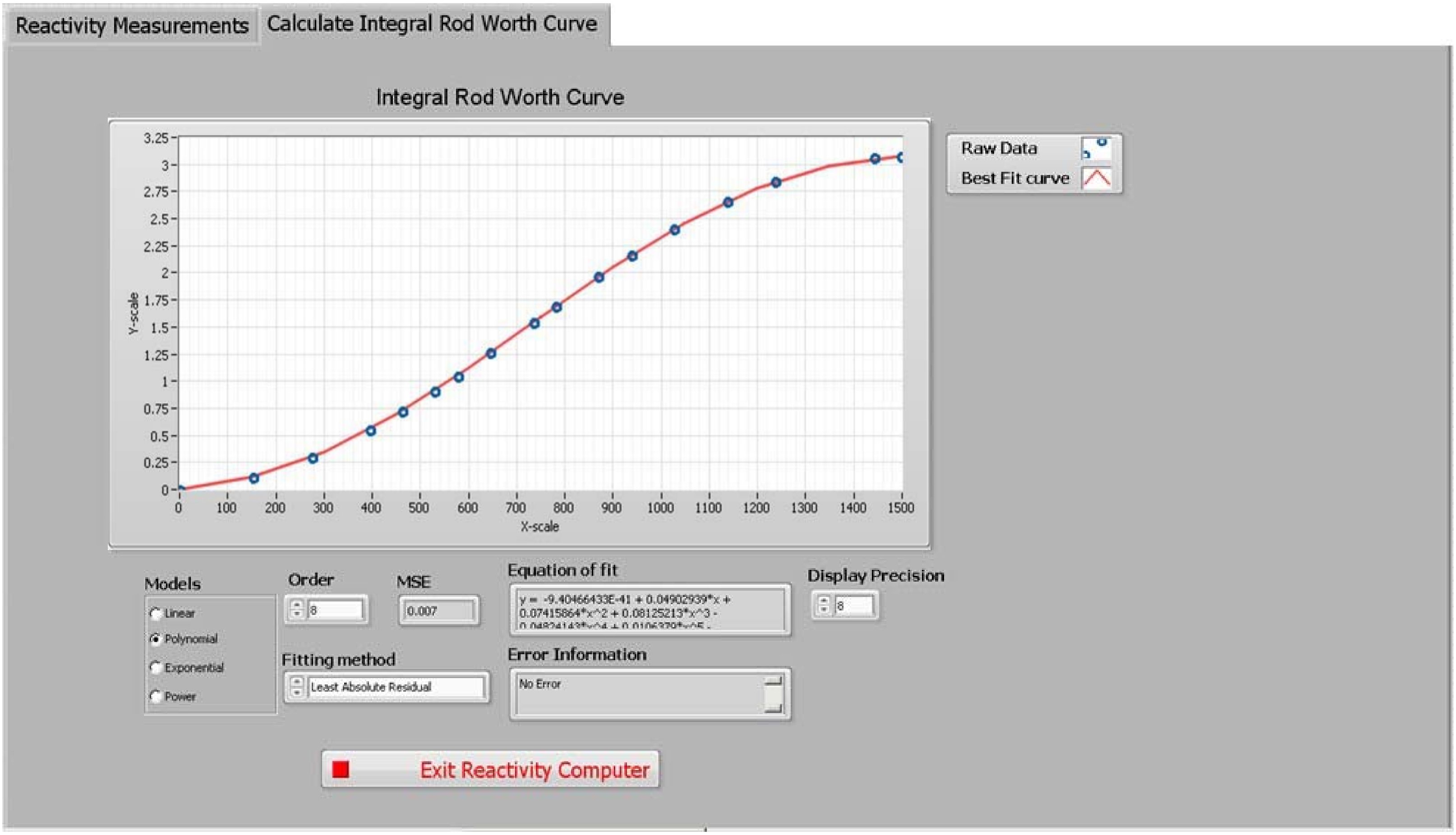
Implementation in LabView: Integral Rod Worth Calculation

- Integral RW is the summation of the current calculated differential CR worth and the previous iteration's integral CR worth

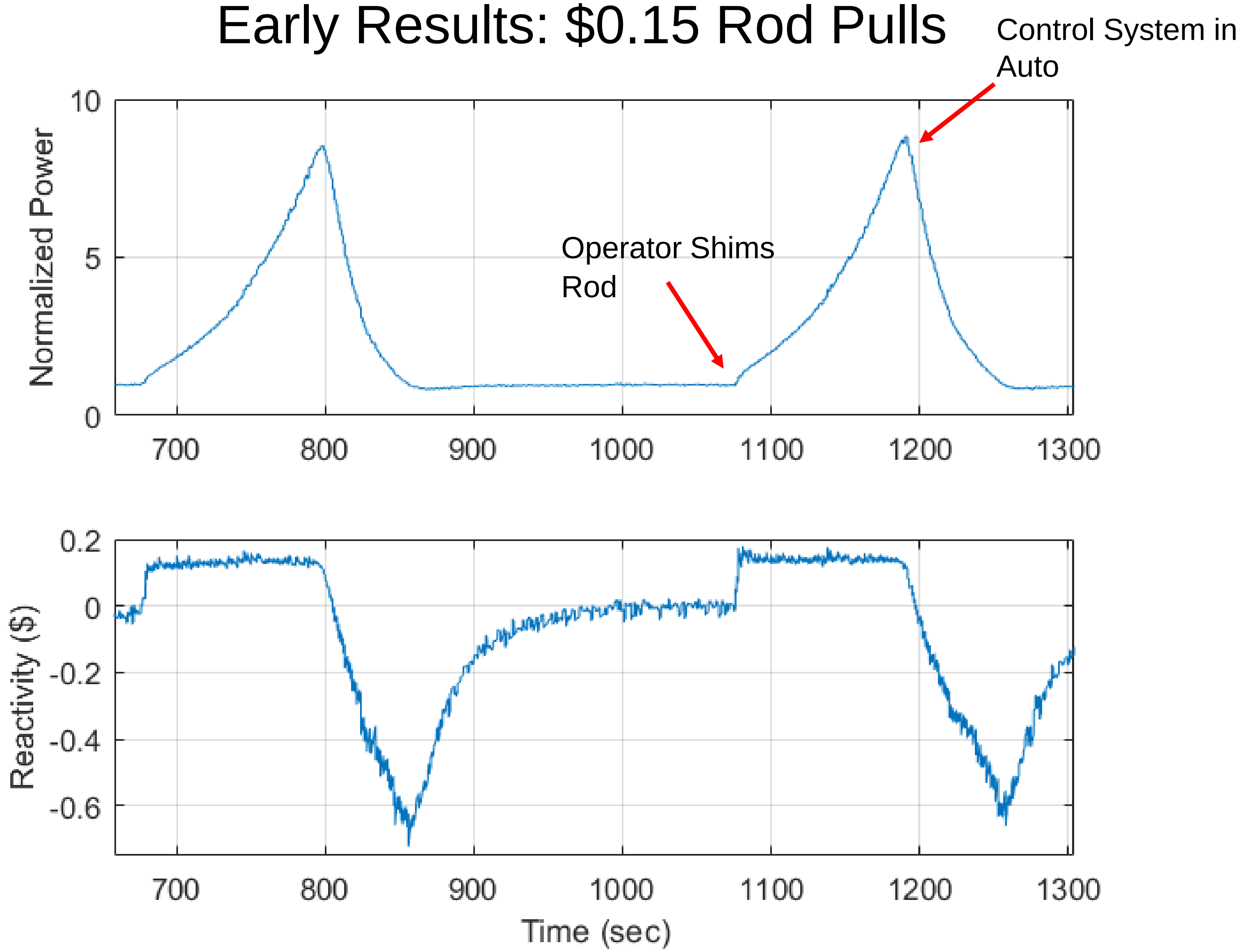


Implementation in LabView: CR Worth Curve Fit

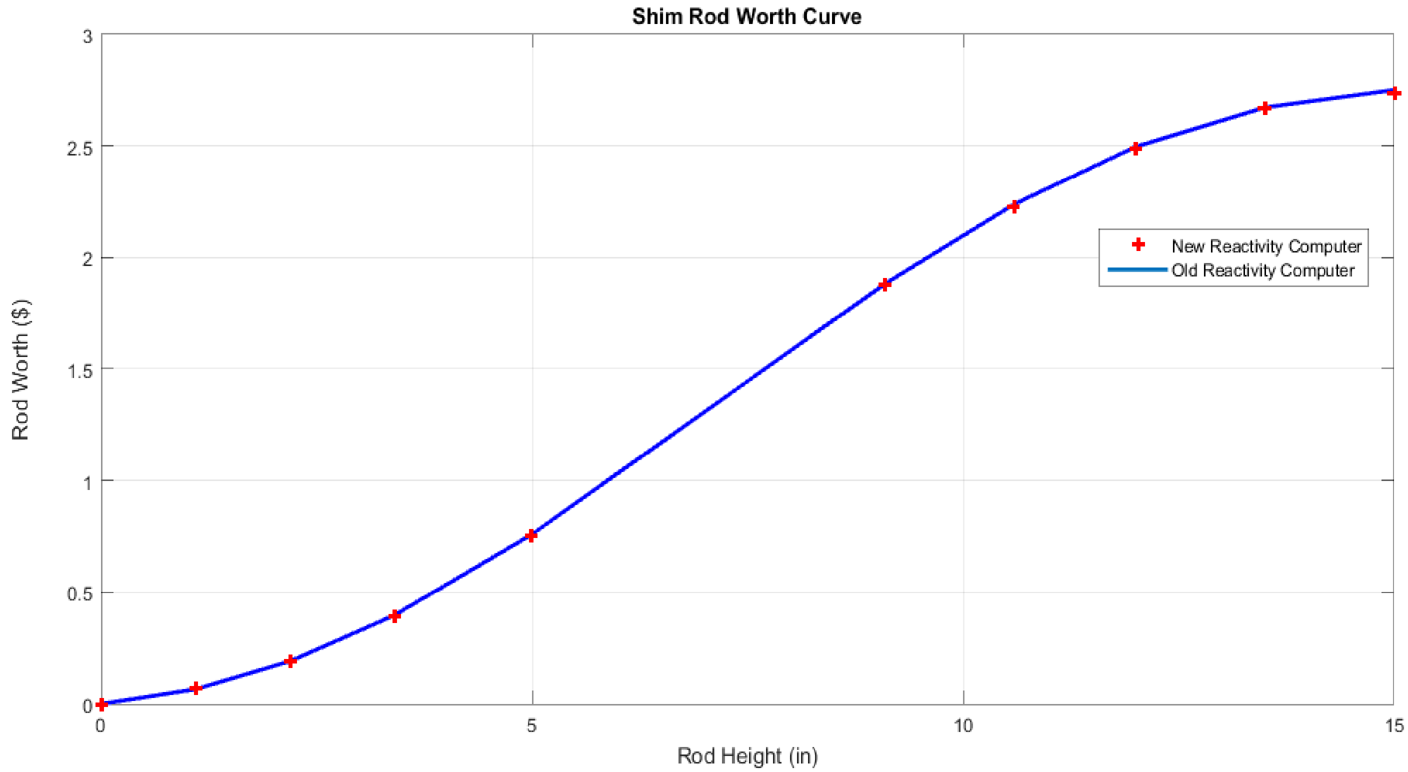
- Best understood by reviewing LabView program…



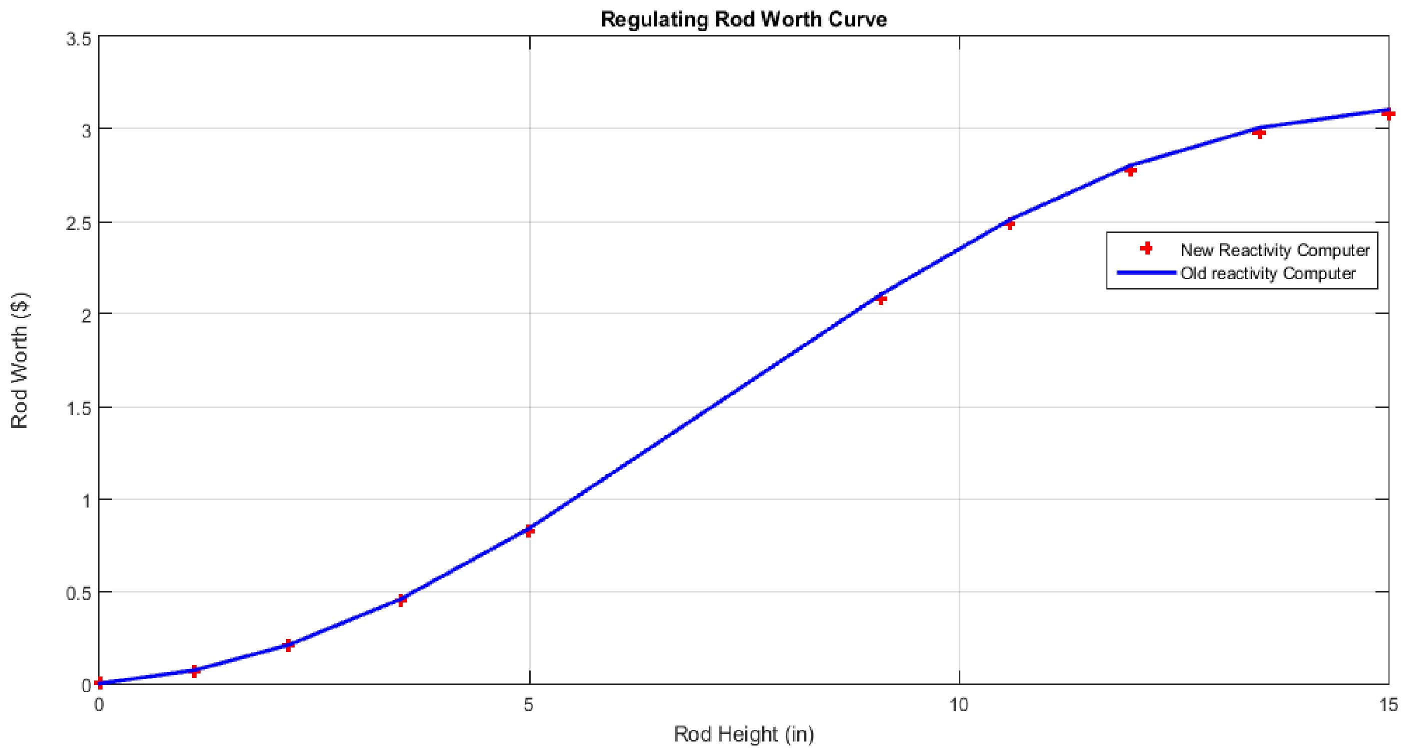
Early Results: $0.15 Rod Pulls



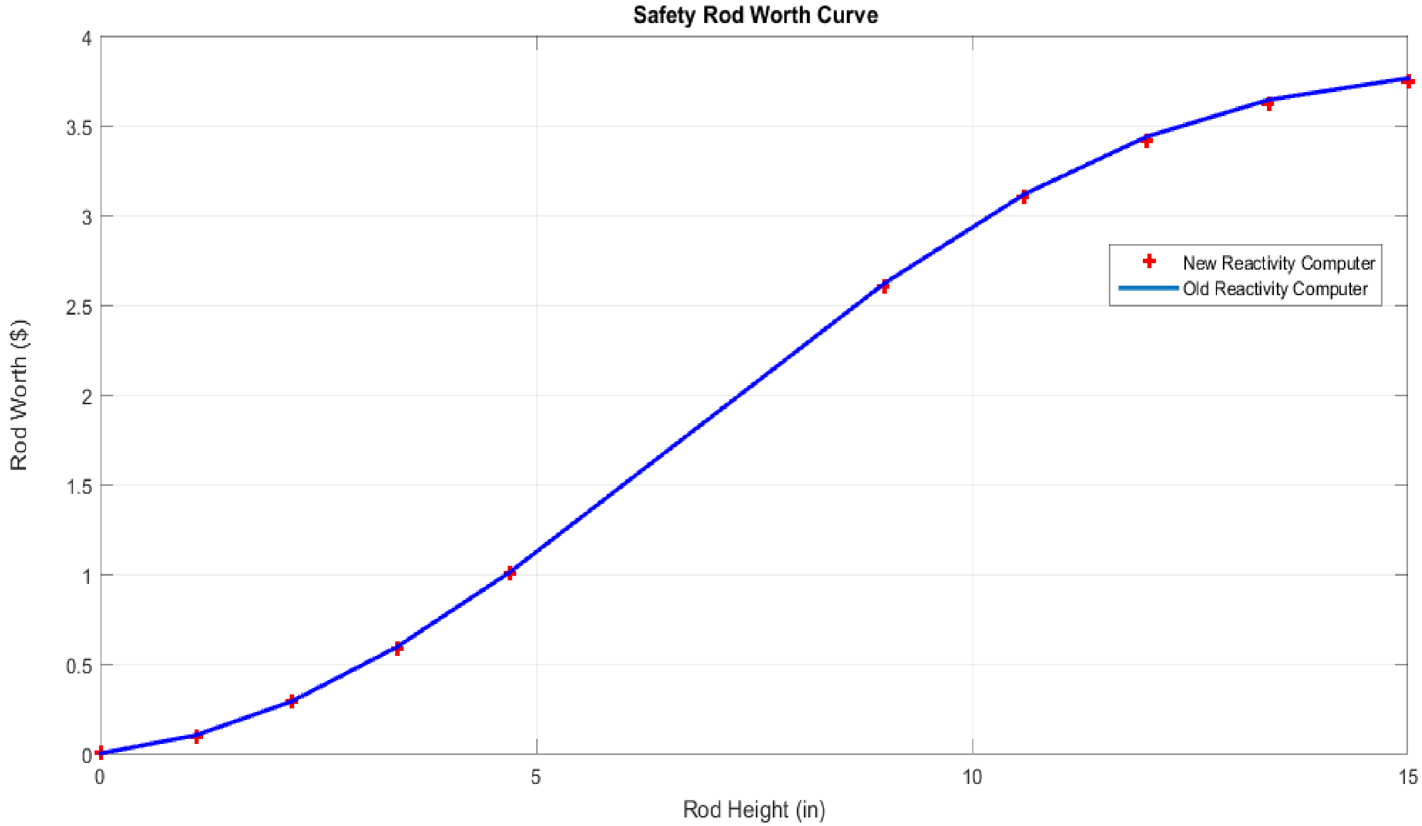
Test Data: Shim CR Worth Curve



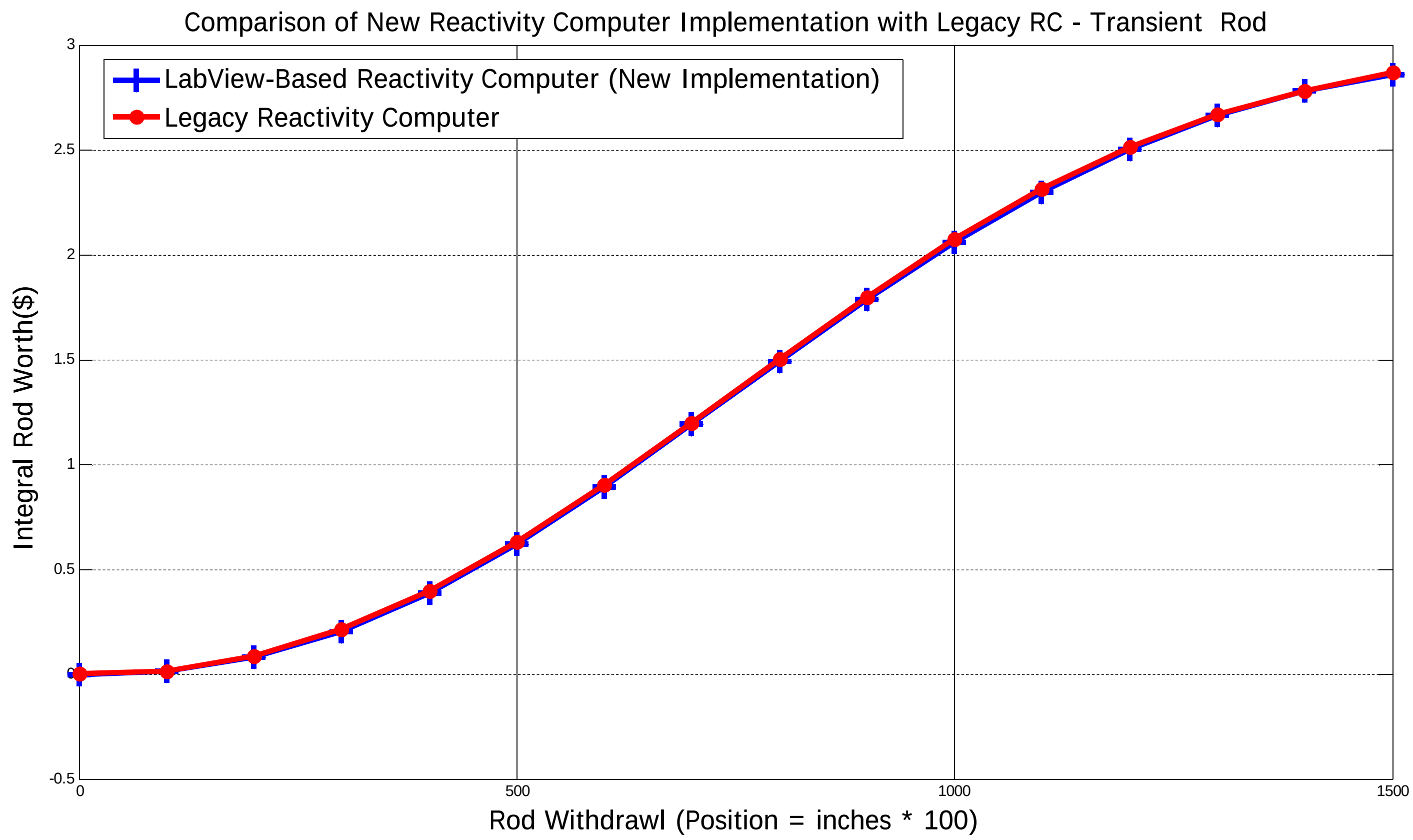
Test Data: Regulating CR Worth Curve



Test Data: Safety CR Worth Curve



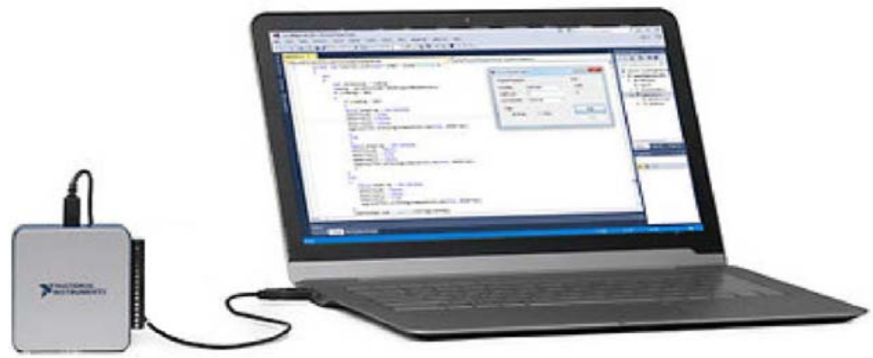
Test Data: Transient CR Worth Curve



Conclusion

- Validated against legacy RC – validated with In-hour Method

- Final packaging – HW (using $180 NI box for DAQ)



- Stand-Alone Deployment

- LabView – “easier to understand??”

- **AGARA:** Will share with members of TRTR Community