CALIFORNIA ISO
California Independent System Operator

Using PI-ACE to Calculate Regulation Performance Metrics

Prepared by

Craig Taylor and Don DeBerry

Presentation to
2002 OSIsoft T&D Users Conference

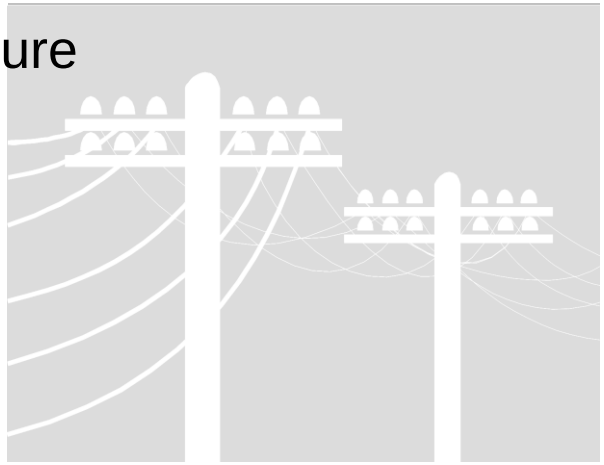
Agenda

Don Covering

- Description of Regulation
- Regulation Performance Metrics

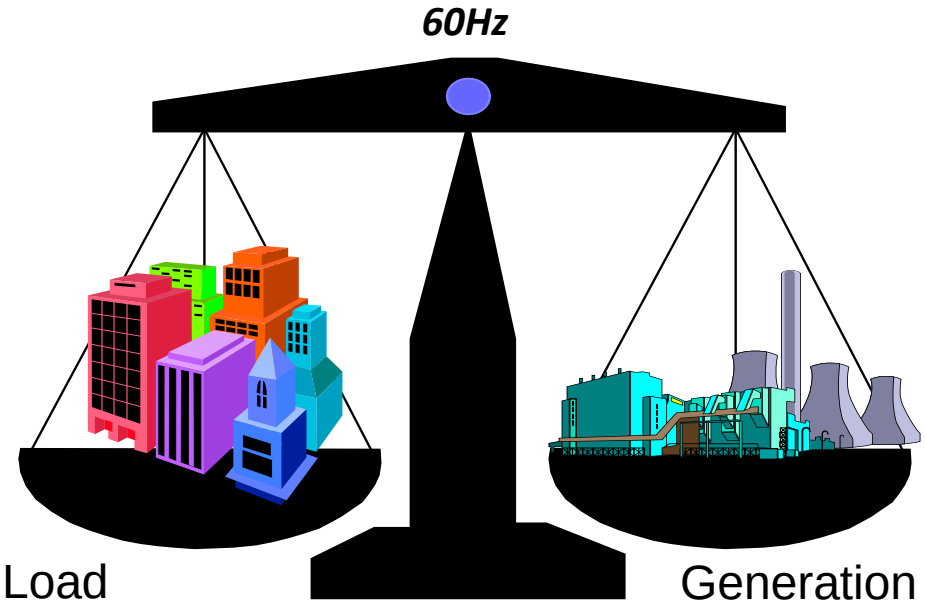
Craig Covering

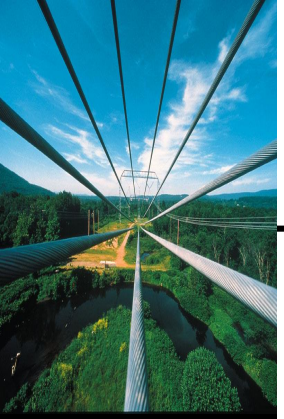
- Calculation Requirements
- Server and Module Database Structure
- Calculation Code Layout
- Issues and Solutions



What is Regulation?

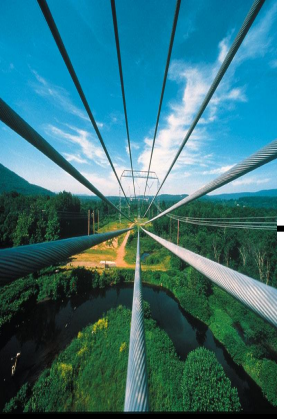
Regulation – Generators equipped with Automatic Generation Control (AGC) that can change output quickly to accommodate the fluctuations in system supply and demand.





What is Regulation?

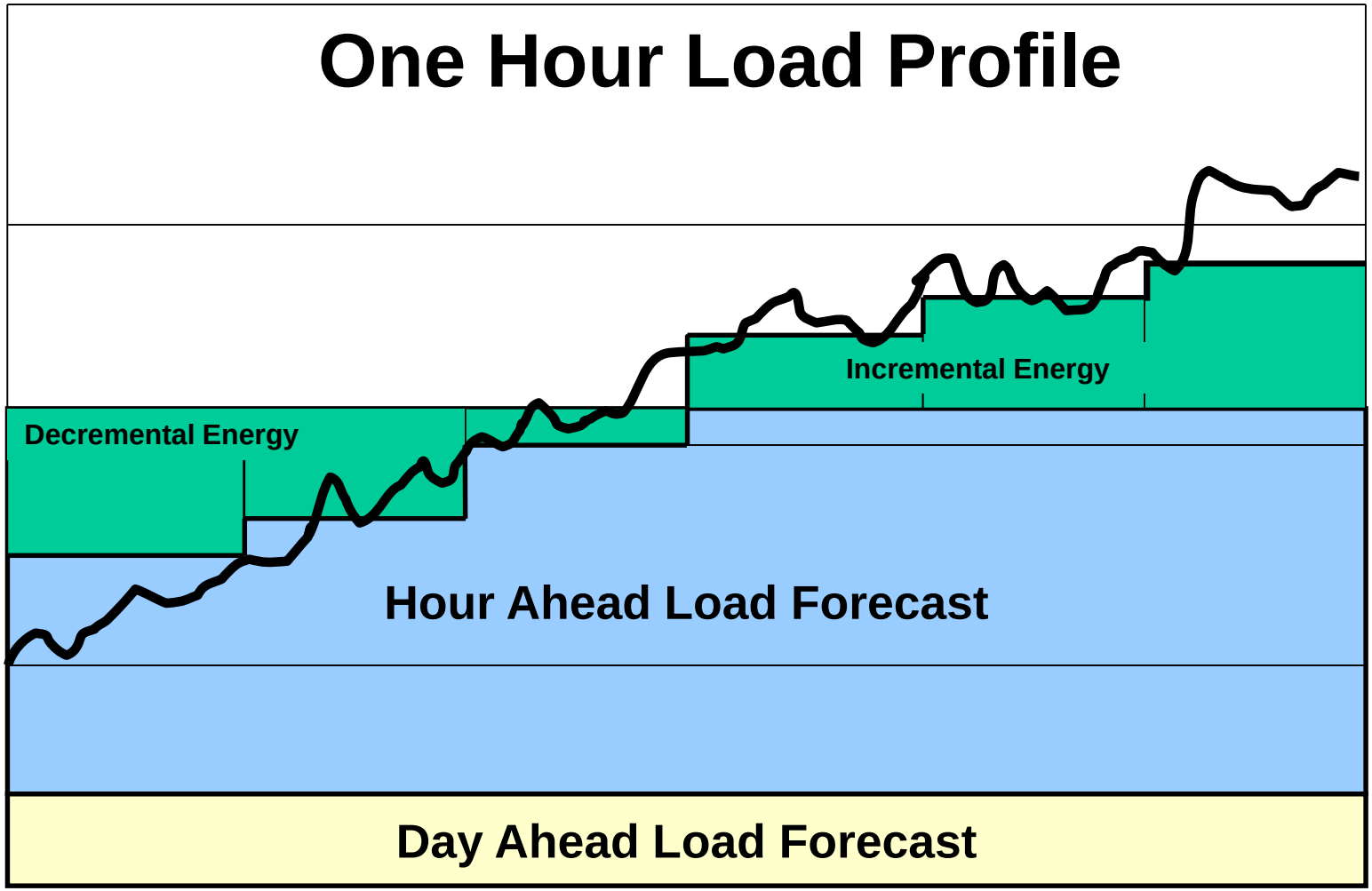
- Ancillary Service – Expensive
- Reserved Capacity
- Generators Require Certification
- Telemetry Requirements
  - Verify AGC Status
  - Two Way Communication
- Generators Controlled by CAISO EMS
- Typically 5-20 Units (CAISO system)



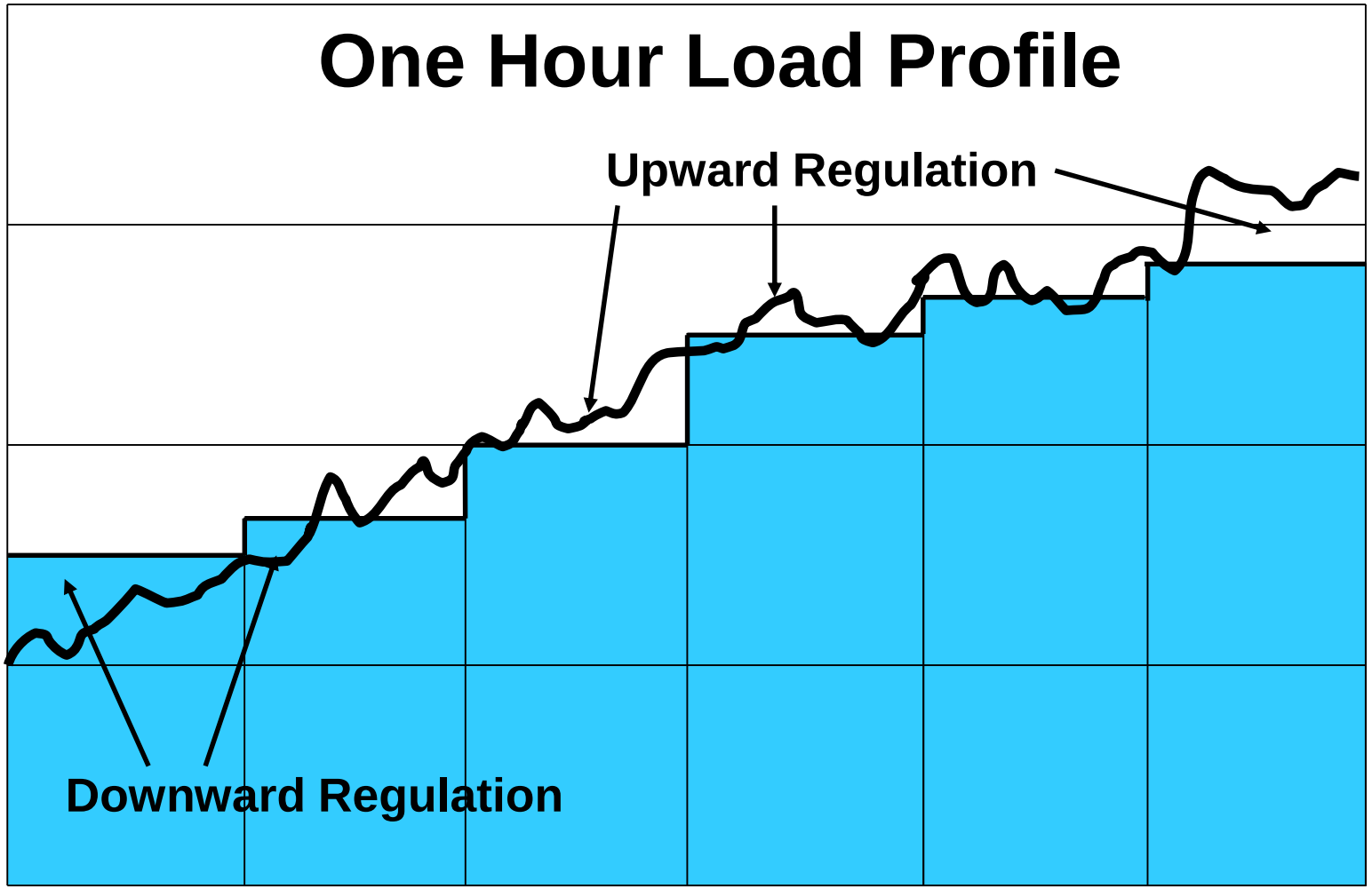
Why are Performance Metrics Needed?

- Improve the Quality of Regulation Service
  - Rank Regulation Performance
  - Share information with Generator Owners
  - Eventually Penalize Poor Performers
- Reduce Costs
  - Reduce the Amount of Regulation Required
  - Free up Capacity for Other Energy Markets
- Improve Reliability
  - Better Response to System Emergencies
  - NERC Control Performance Standards

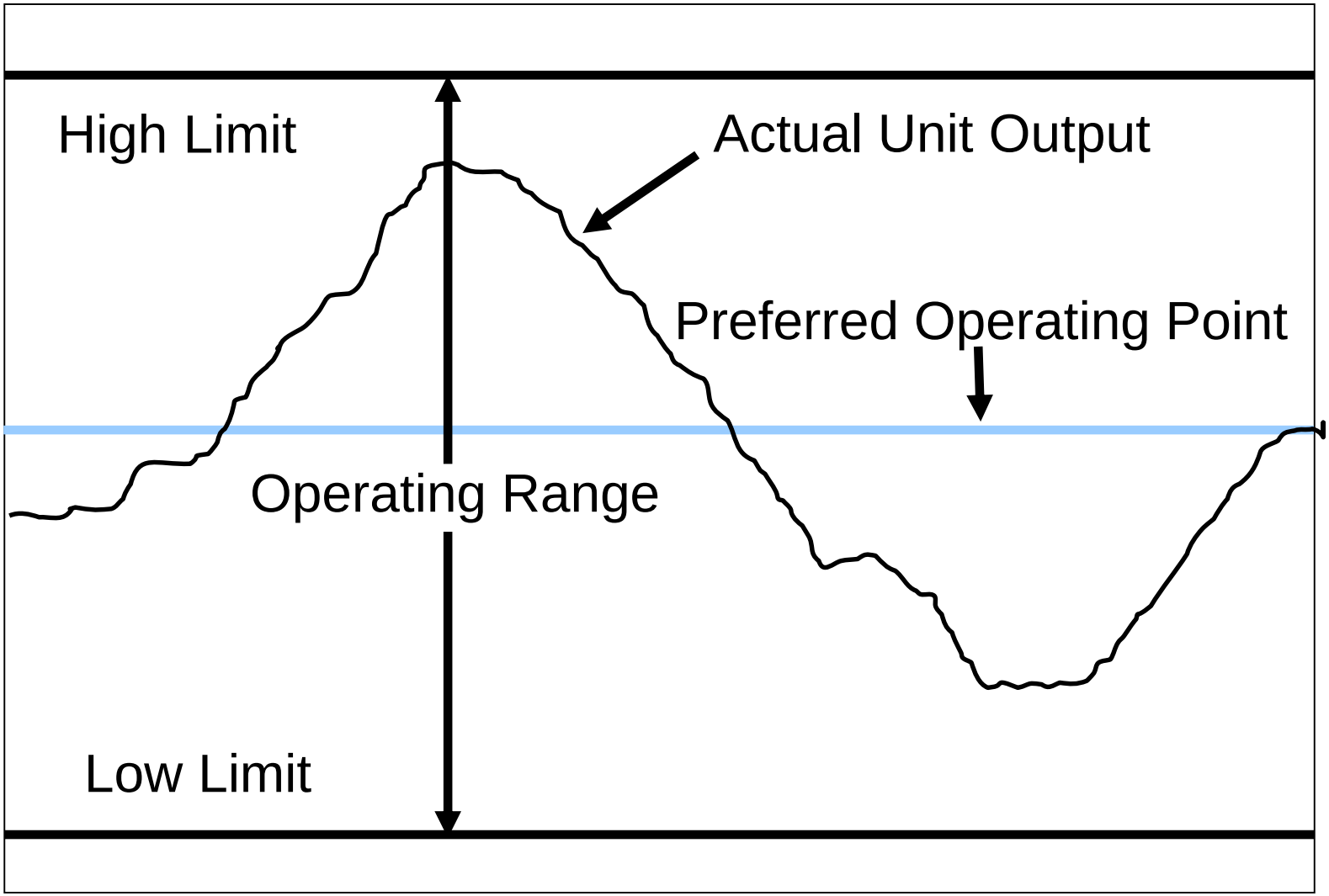
Regulation – System Perspective



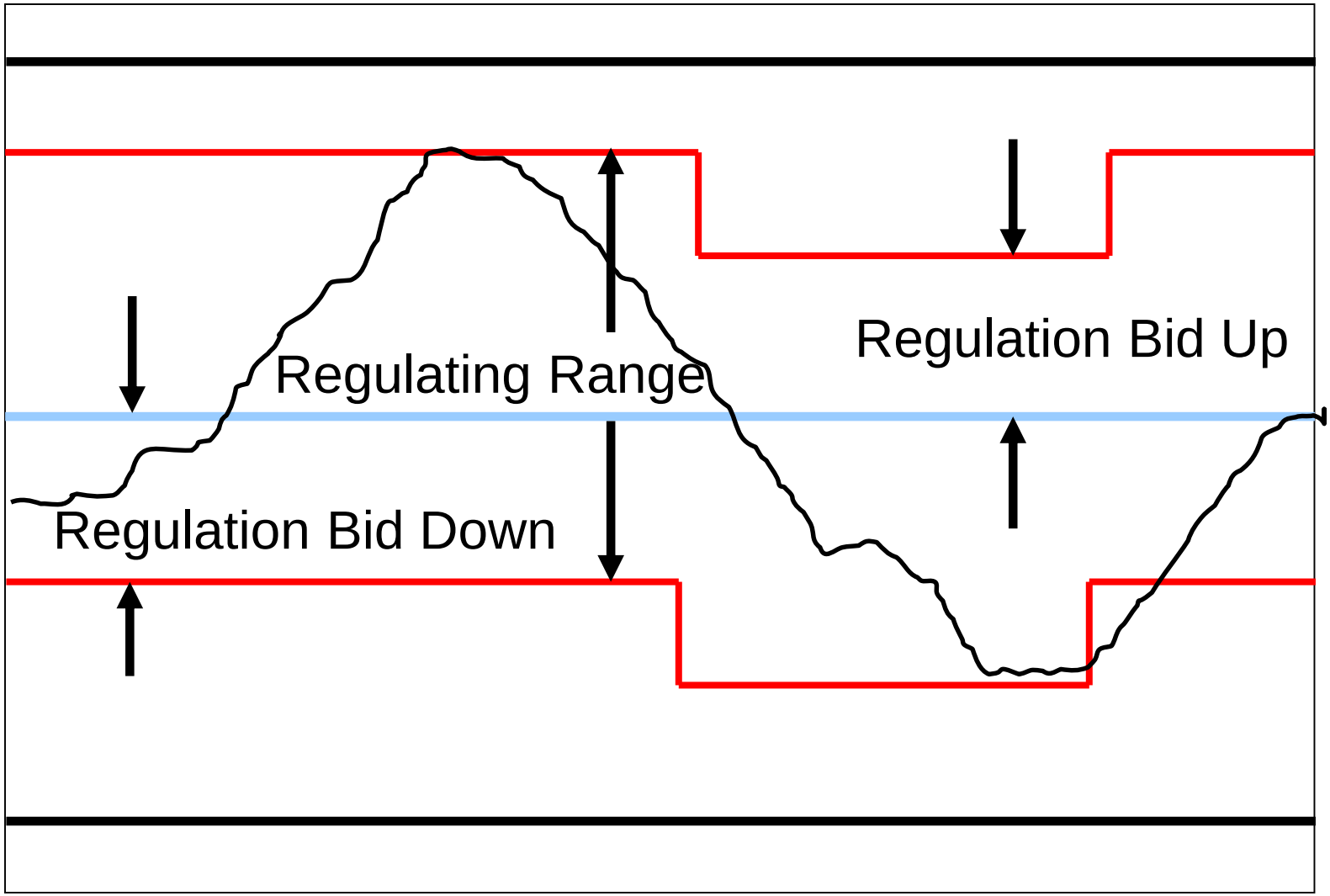
Regulation – System Perspective



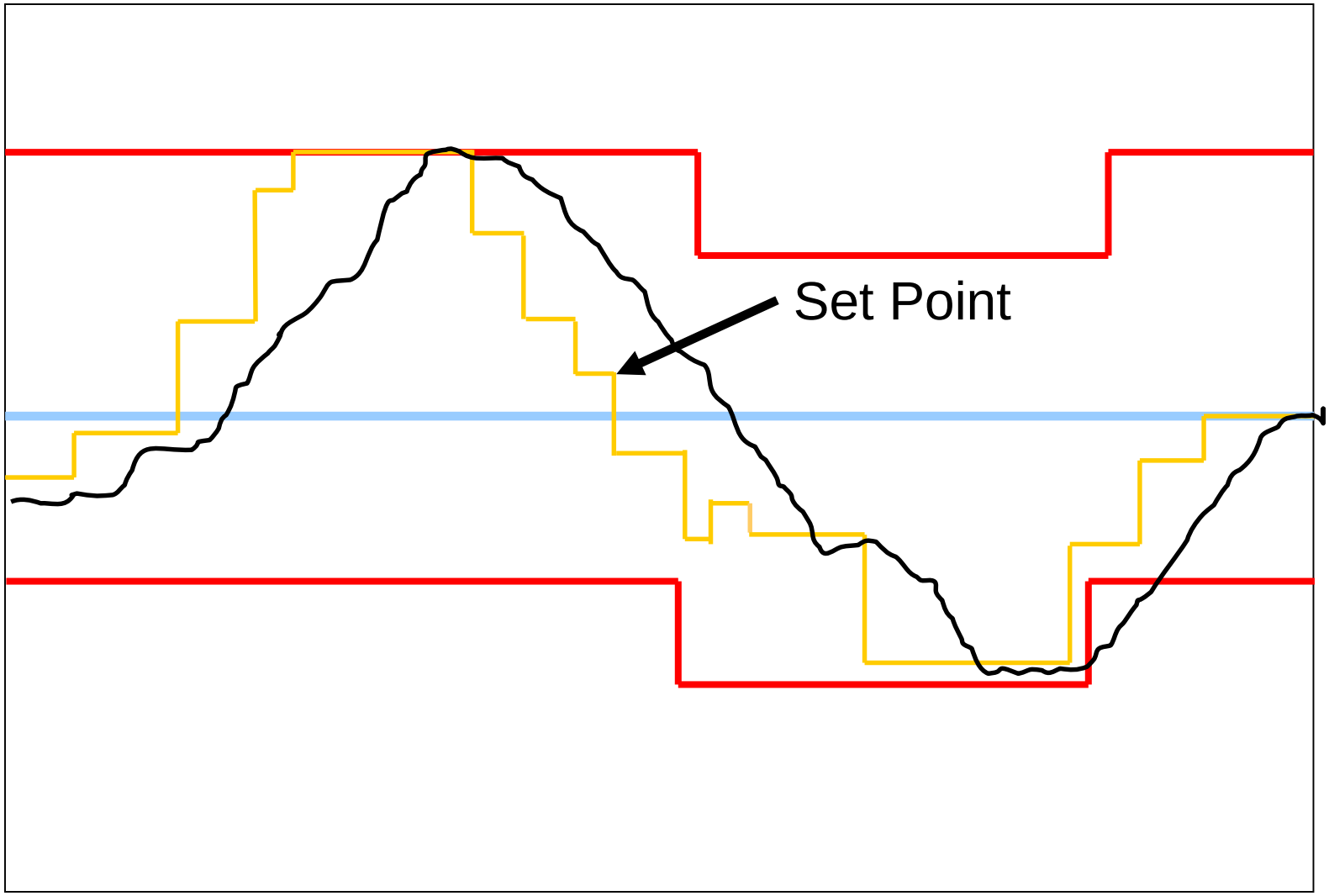
Regulation – Generator Parameters



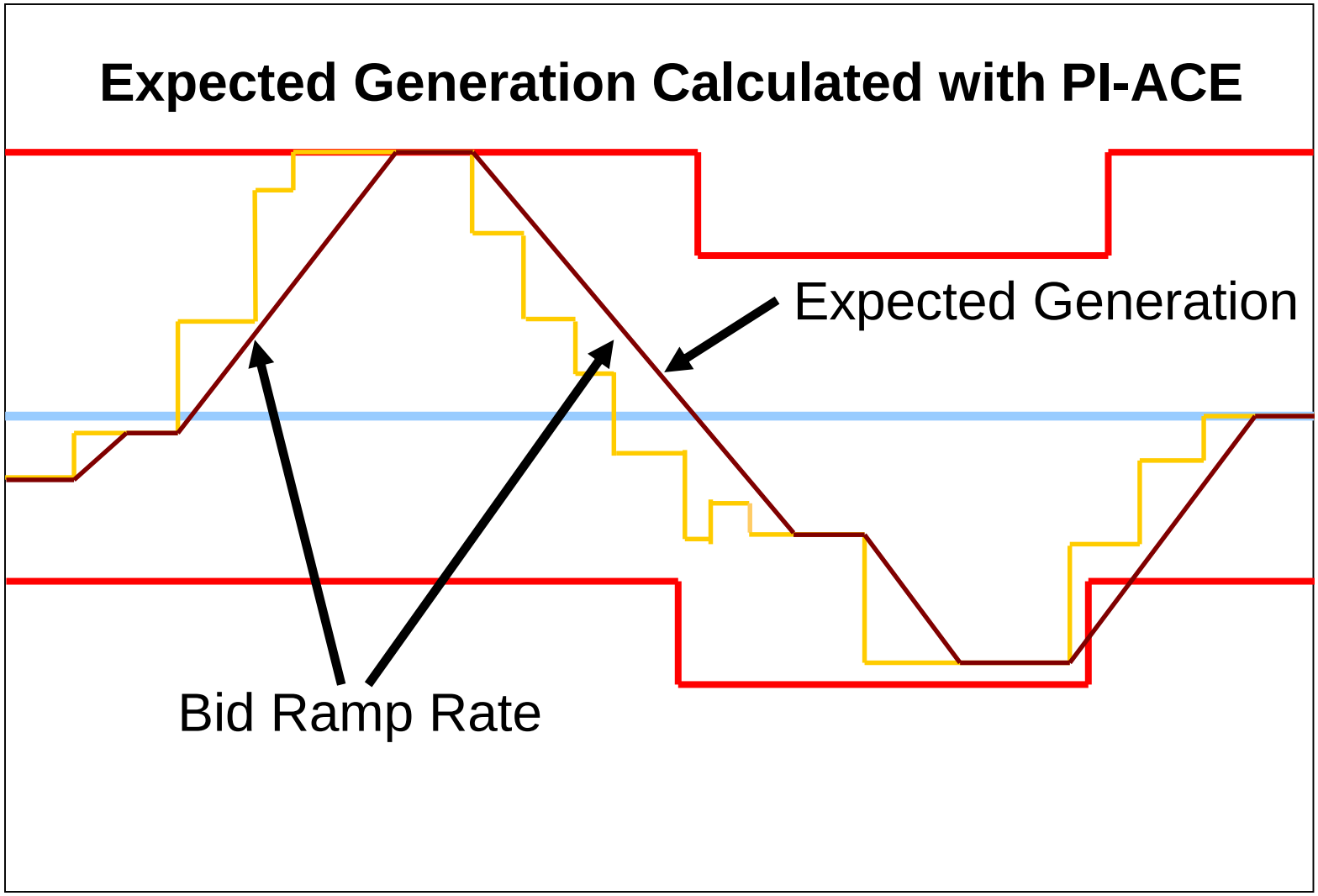
Regulation – Generator Parameters



Regulation – Unit Control

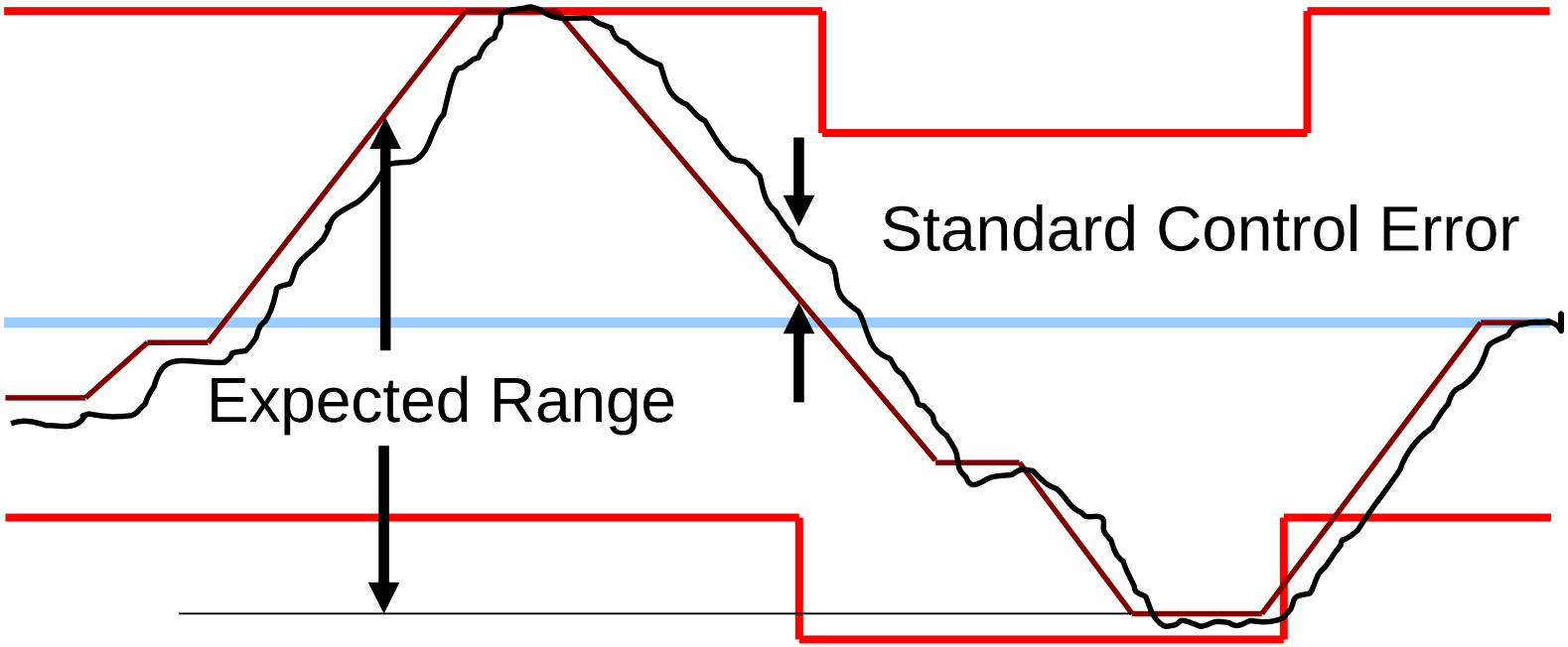


Expected Generation Calculation



Performance Metric Calculation

Calculated with PI-ACE



RMS Σ for 1 Hr and for 10 Min

Test Performance Metric Formulas

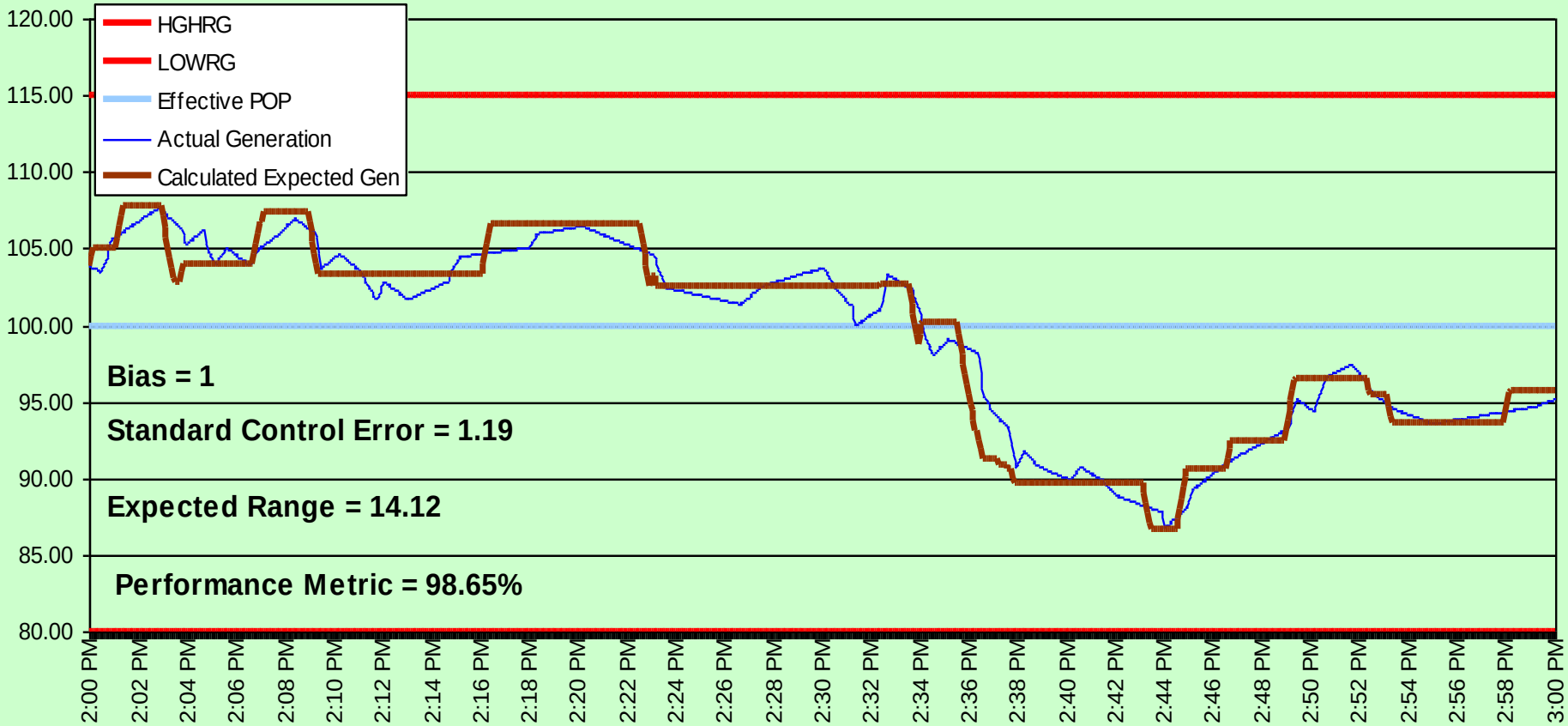
$$Error = \sqrt{\frac{\sum_{j=1}^{N}(Actual_j - Expected_j)^2}{N-1}}$$

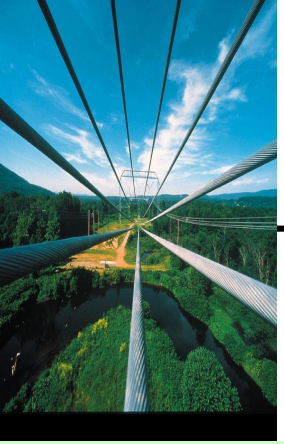
$$Range = \sqrt{\frac{\sum_{j=1}^{N}(Expected_j - Expected_{Min})^2}{N-1}}$$

$$\text{Performance Metric} = \left(1 - \frac{Error - bias}{Range}\right) * 100$$

Example - Good Performance

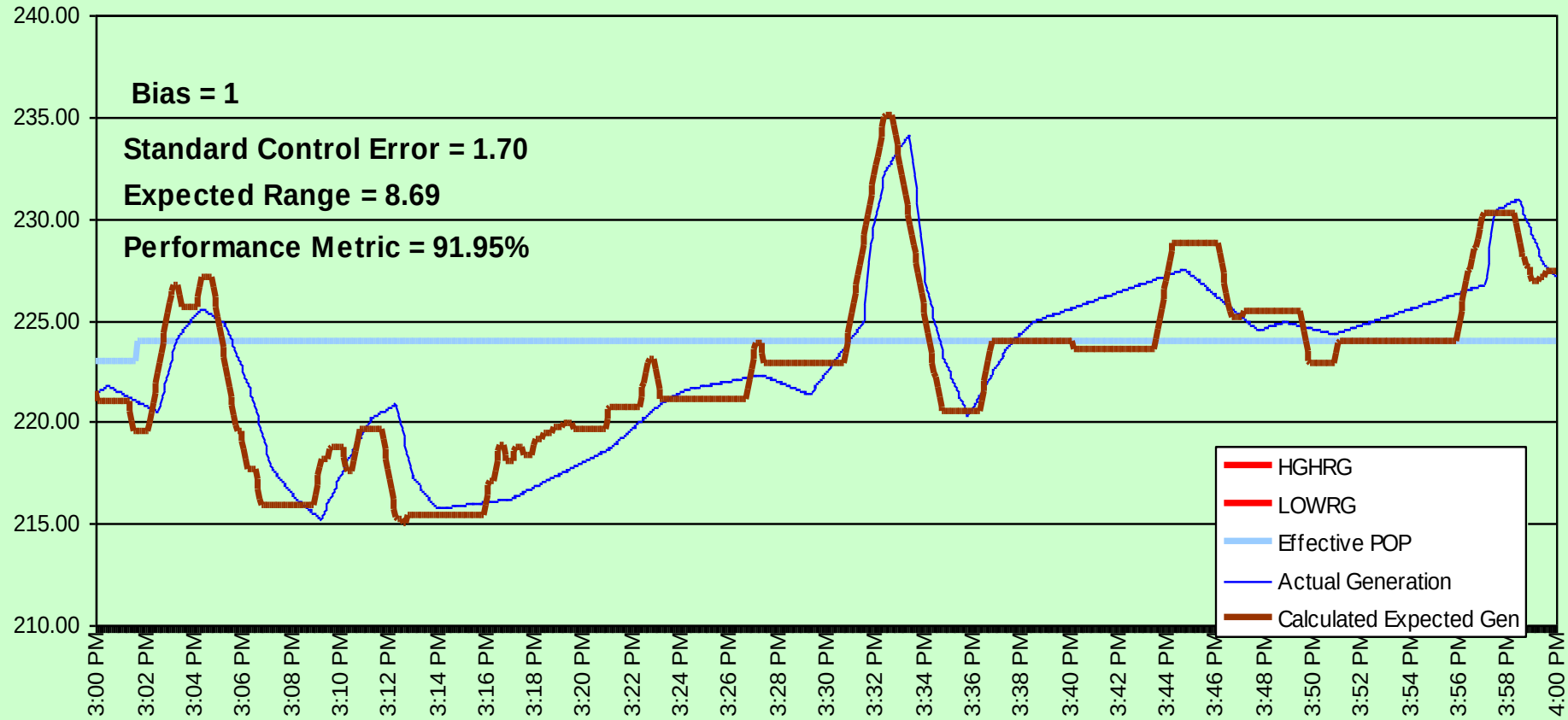
Performance Metric = $\left(1 - \frac{Error - bias}{Range}\right) * 100$





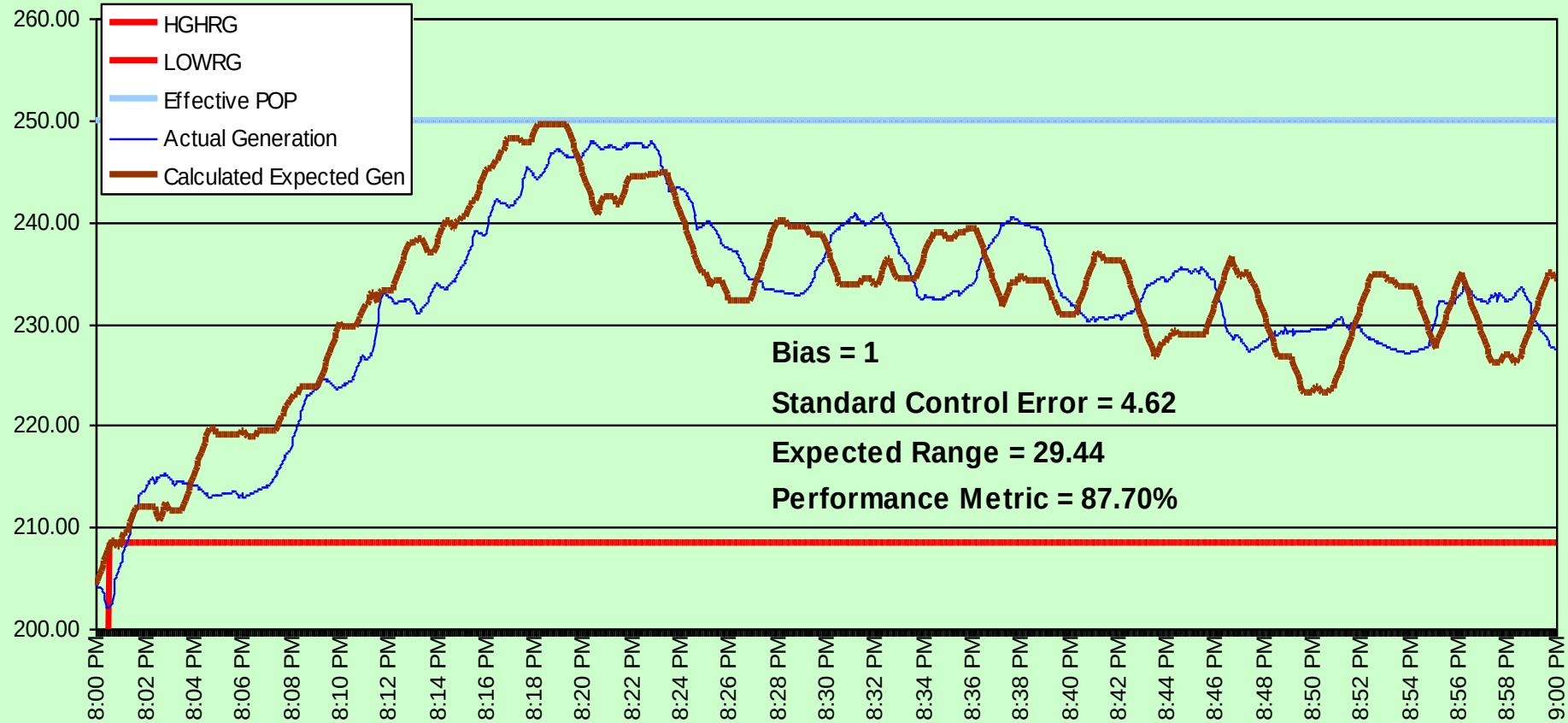
Example - Moderate Performance

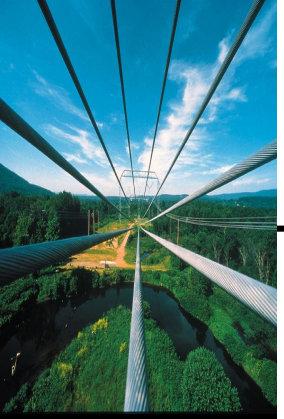
Performance Metric = $\left(1 - \frac{Error - bias}{Range}\right) * 100$



Example - Poor Performance

Performance Metric = $\left(1 - \frac{Error - bias}{Range}\right) * 100$



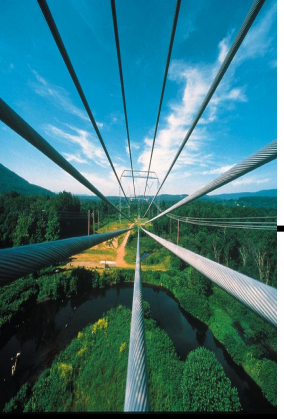


Performance Metric Implementation

- **Compliance**
  - Test Additional Performance Metrics
  - Develop 1 Hr Performance Metrics
  - Evaluate and Rank the Performance of Regulating Units
  - Share Performance Metrics with Regulation Providers
  - Eventually Penalize Poor Regulation Providers

- **Real-Time Operations**
  - PI Process Book Displays
  - 10 min Performance Metrics for NERC CPS2
  - 1 Hr Performance Metrics for Ranking Units

Calculation Requirements



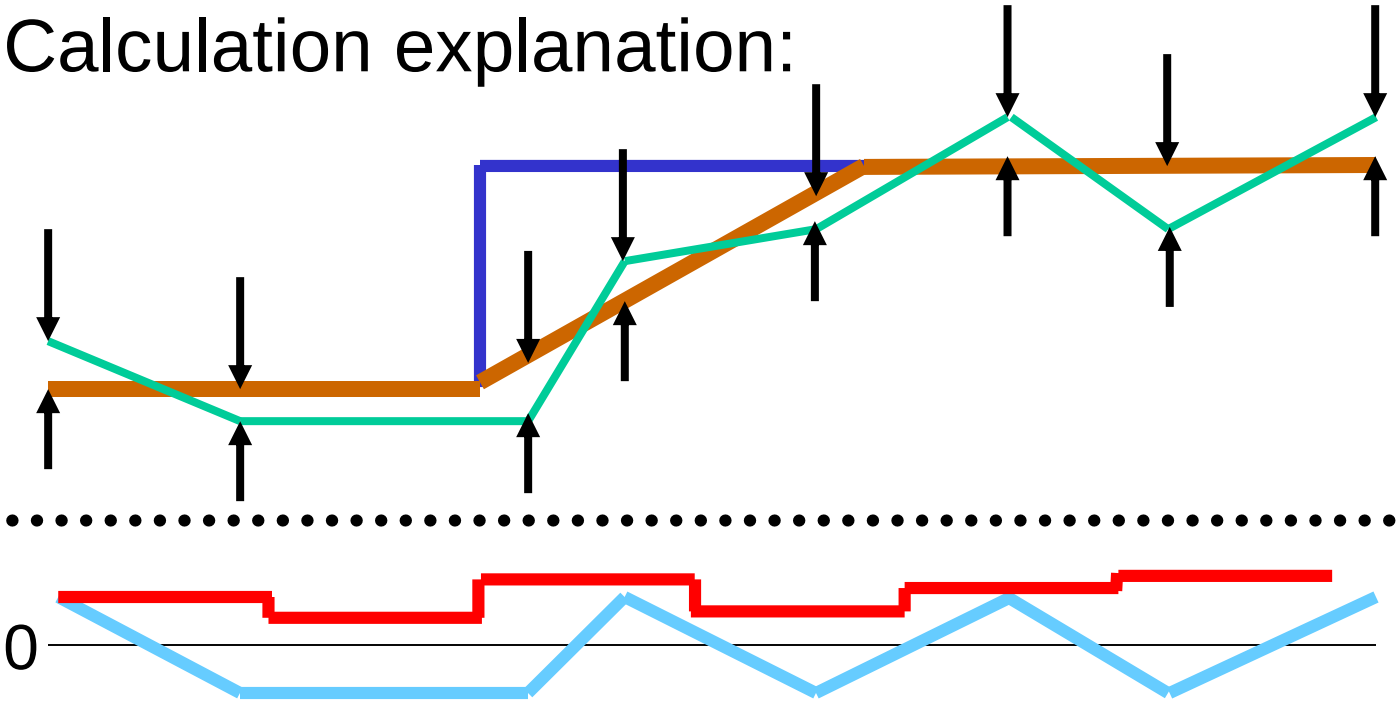
Calculation Requirements

We wanted:

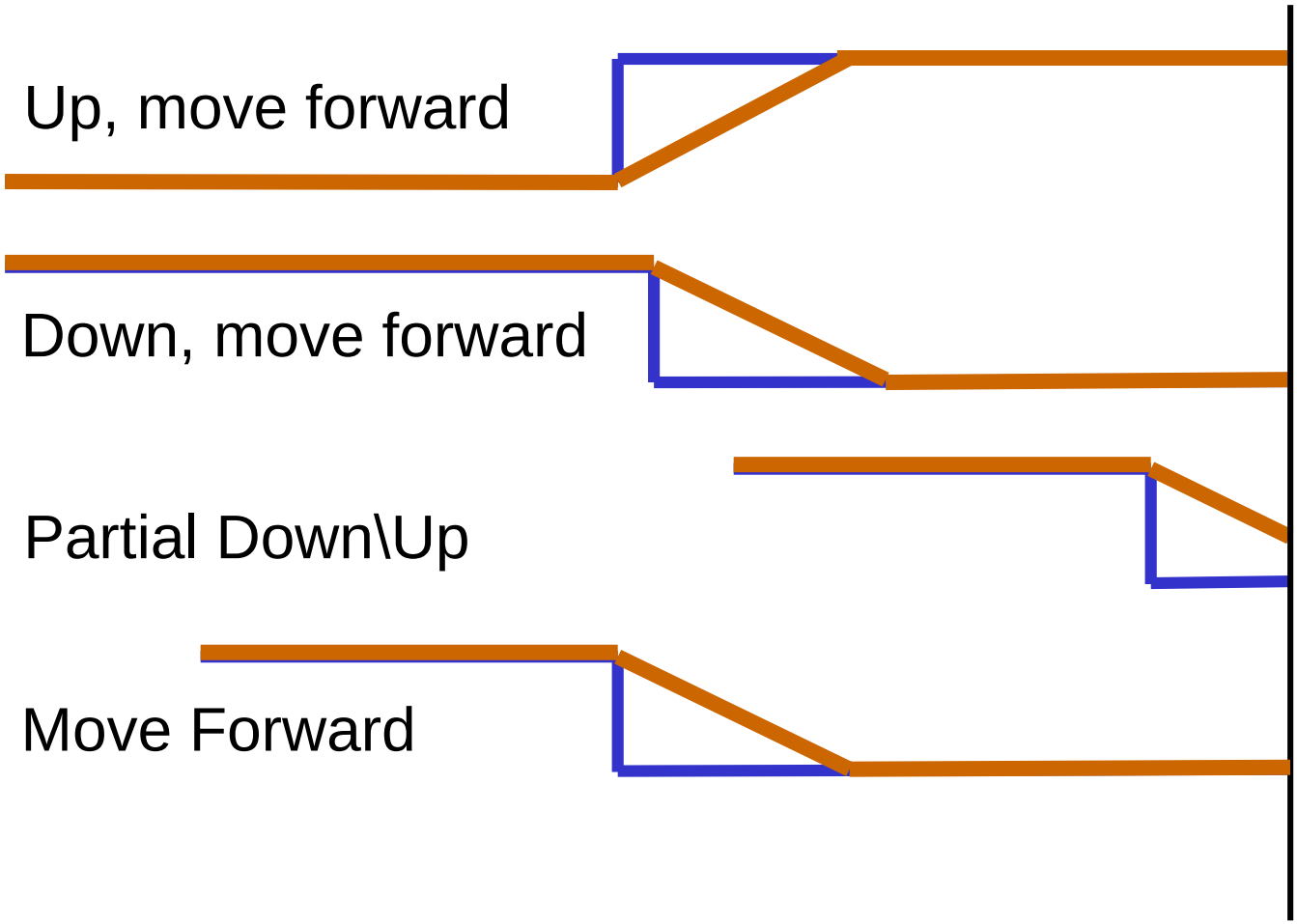
- Results posted close to real time (1 minute)
- Operational redundancy
- Single piece of code for all Units
- Calculations based on previous events
- Calculations backfilled in event of failure

Calculation Requirements

Calculation explanation:

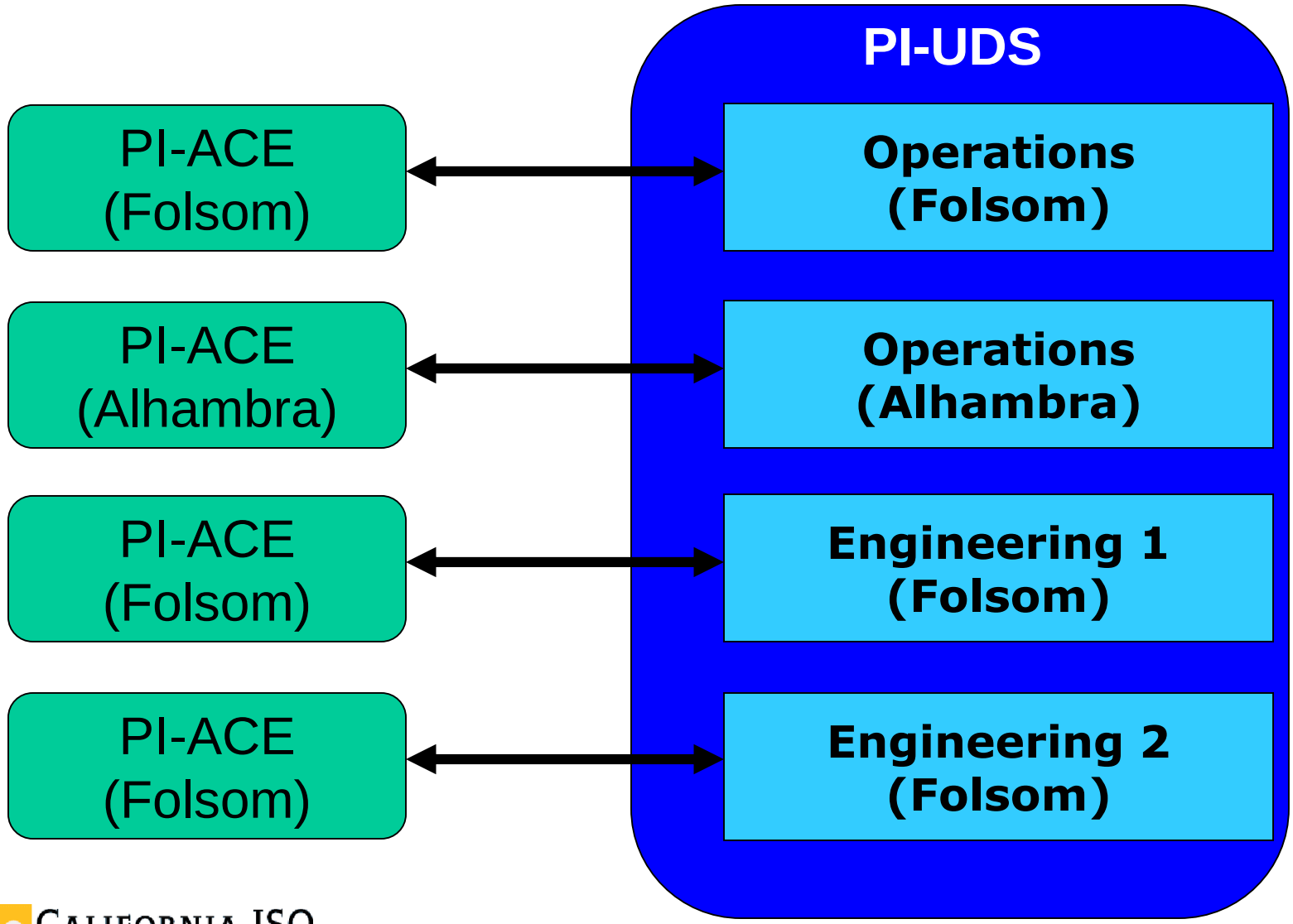


Calculation Requirements



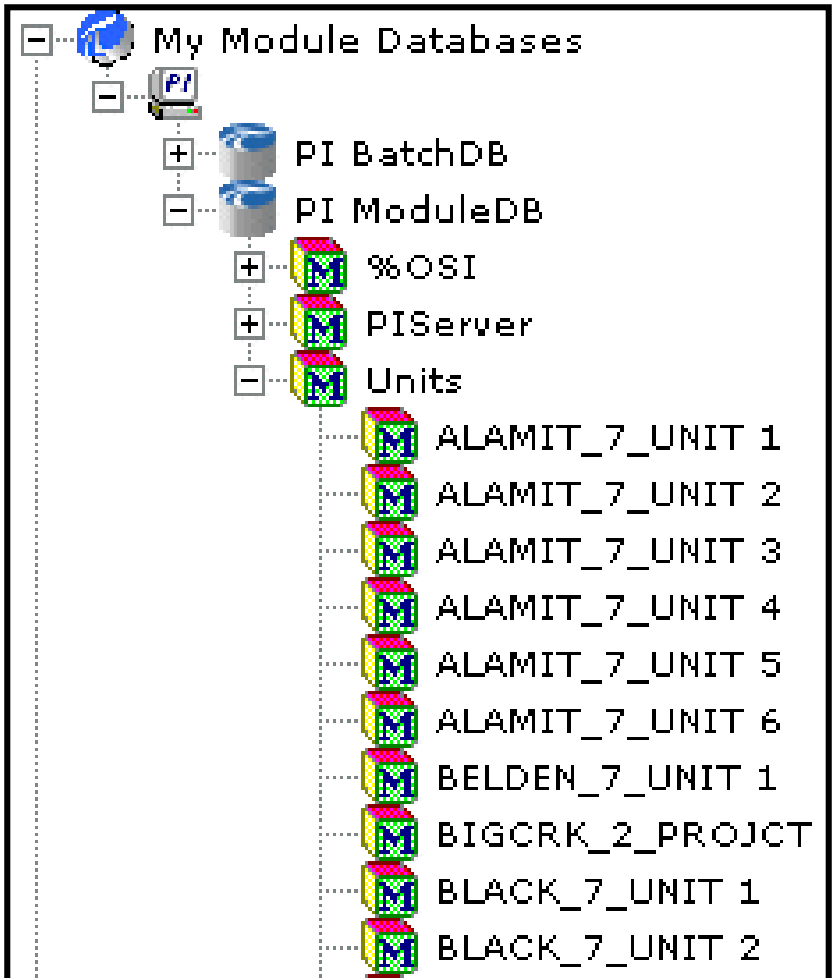
Server and Module Database Structure

PI-ACE Server Structure



PI-Module Database Structure

- 126 Units on Automatic Generation Control (AGC)
- Each unit modeled in PI-Module Database



PI-Module Database Structure

- Aliases created for key Unit measurements:

Sub-Modules | PI Aliases | PI Properties

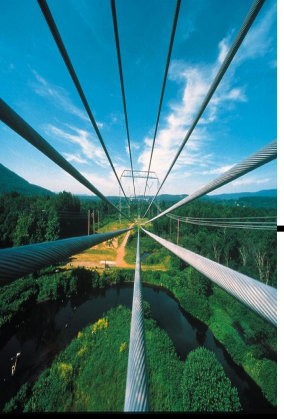
PIAlias Name	Tag Name
BSEPT_GEN_MW	Effective POP for ALAMIT_7_UNIT 1
Calculation Time	Calculation Time for ALAMIT_7_UNIT 1
Effective High Reg Limit	Effective High Reg Limit for ALAMIT_7_UNIT 1
Effective Low Reg Limit	Effective Low Reg Limit for ALAMIT_7_UNIT 1
Effective POP	Effective POP for ALAMIT_7_UNIT 1
Expected Generation	Expected Generation for ALAMIT_7_UNIT 1
Expected Generation-2	Expected Generation for ALAMIT_7_UNIT 1-2
HGHRG_GEN_MW	Effective High Reg Limit for ALAMIT_7_UNIT 1
Load Deviation	Load Deviation for ALAMIT_7_UNIT 1
Load Deviation 10min StdDev	Load Deviation 10min StdDev for ALAMIT_7_U...
Load Deviation 10min StdDev-2	Load Deviation 10min StdDev for ALAMIT_7_U...
Load Deviation 1hr StdDev	Load Deviation 1hr StdDev for ALAMIT_7_UNIT 1
Load Deviation 1hr StdDev-2	Load Deviation 1hr StdDev for ALAMIT_7_UNIT...
Load Deviation-2	Load Deviation for ALAMIT_7_UNIT 1-2
LOWRG_GEN_MW	Effective Low Reg Limit for ALAMIT_7_UNIT 1
Server	SYSFolsom

PI-Module Database Structure

- Aliases needed for our calculation:

Measurement		Module DB Alias
Generation	-	UNMW_GEN_MW
Set Point	-	SETPT_GEN_MW
On/Off AGC Control	-	UAGC_GEN
Down Ramp Rate	-	Unit Reg Ramp Down Rate
Up Ramp Rate	-	Unit Reg Ramp Up Rate
Expected Generation	-	Expected Generation
Standard Control Error	-	Standard Control Error
Standard Deviation (10min)	-	SCE 10min StdDev
Data Status	-	SYSDataUpToCurrentSecond

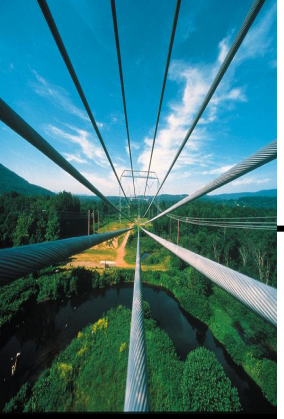
Calculation Code Layout



Typical Backfilling Calculation

Steps

1. Check current data flowing into PI Database
2. Set calculation start time equal to last event written to output tag
3. Set calculation end time as either:
   - Current time
   - Start time + 3 hours
4. Gather all data for defined time period
5. Calculate results from data
6. Write results to output tags

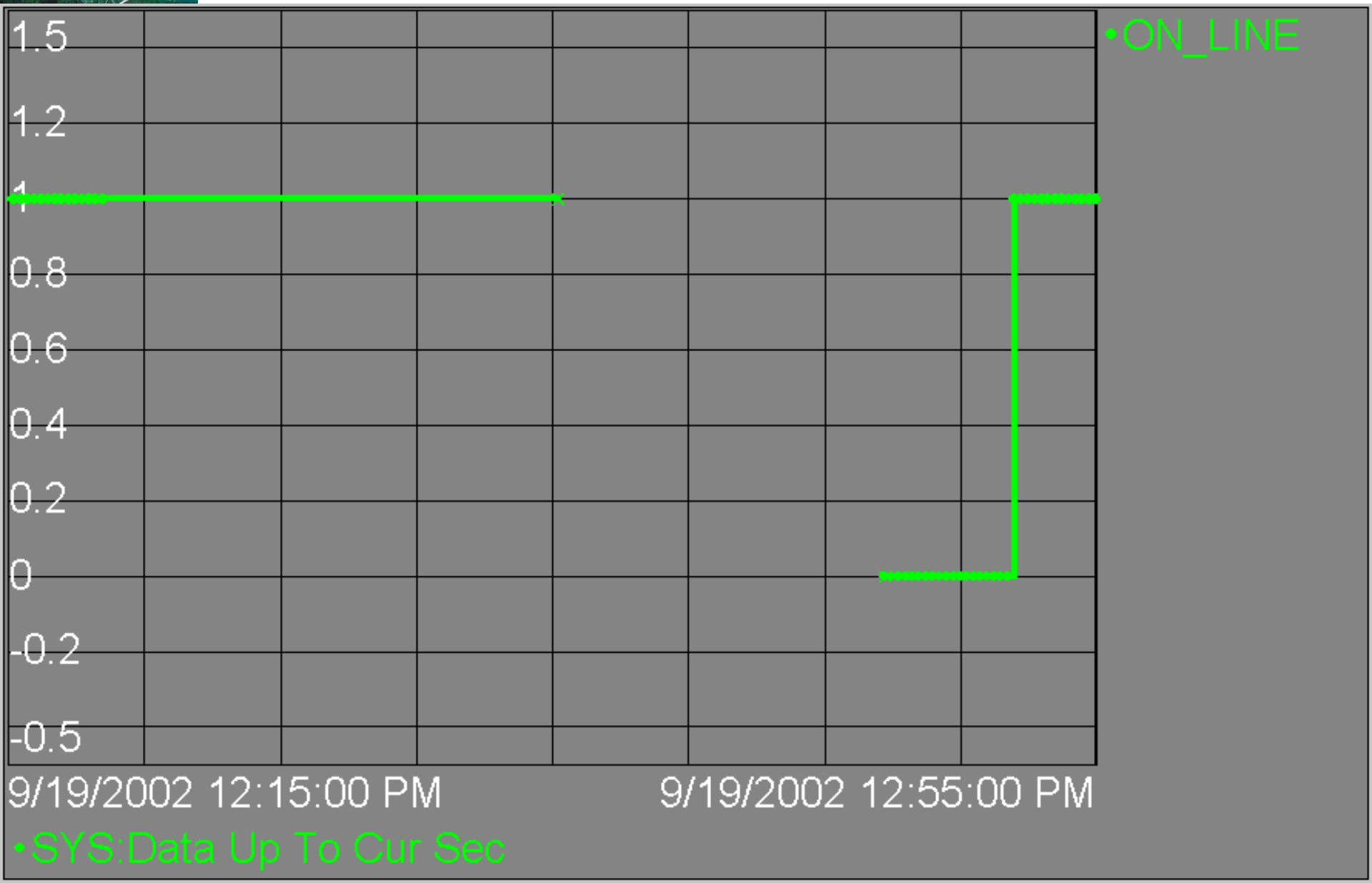


Data Up-To-Date Calculation

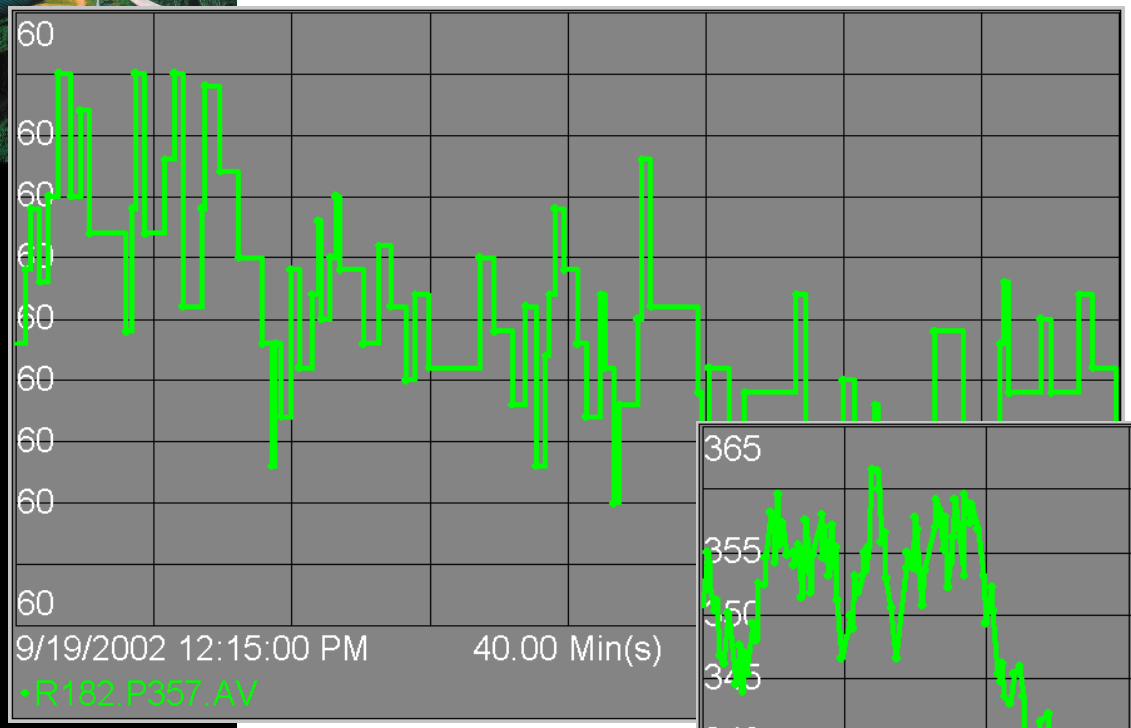
Needed flag to control backfilling calculations which indicated data up-to-date

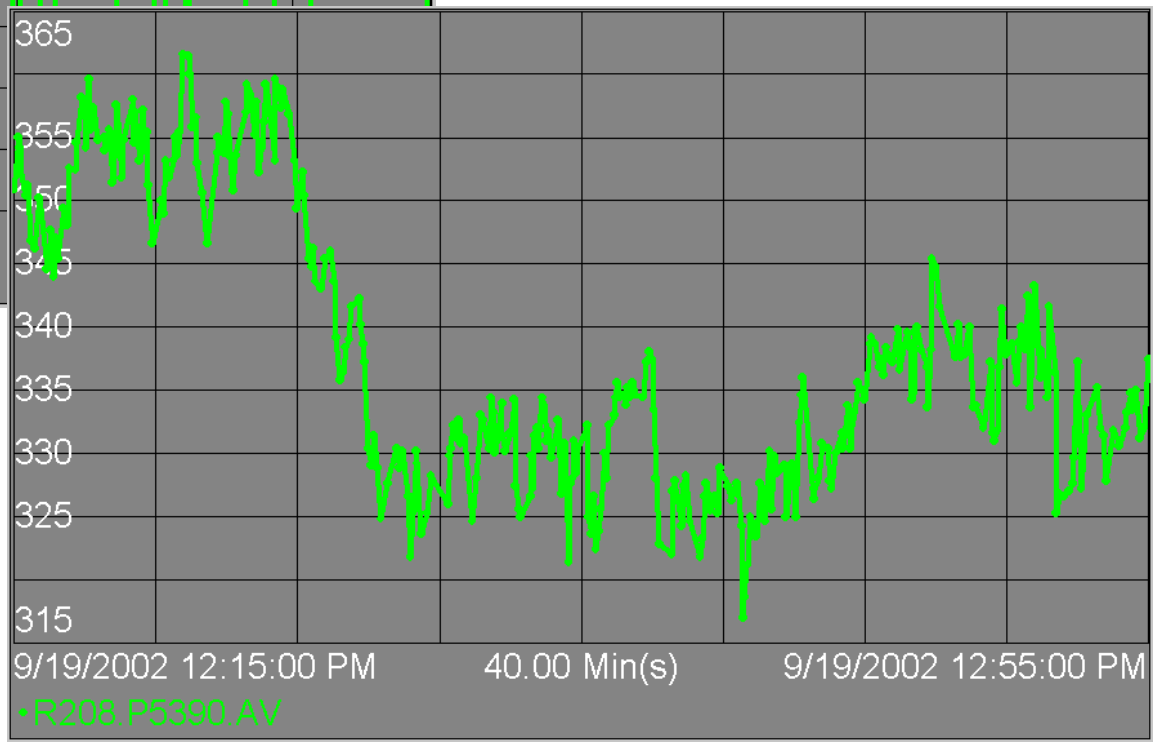
```
Dim bDataUpToSec As Boolean
bDataUpToSec = False
If FREQ01.PrevEvent > (mdblExeTime - 120) Then
   bDataUpToSec = True
Else
    If FREQ02.PrevEvent > (mdblExeTime - 120) Then
       bDataUpToSec = True
    Else
        'Data not up to date... :-(
    End If
End If

If bDataUpToSec = True Then
    SYSDataUpToCurrentSecond.Value = "ON_LINE"
Else
    SYSDataUpToCurrentSecond.Value = "OFF_LINE"
End If
```

Data Up-To-Date Calculation



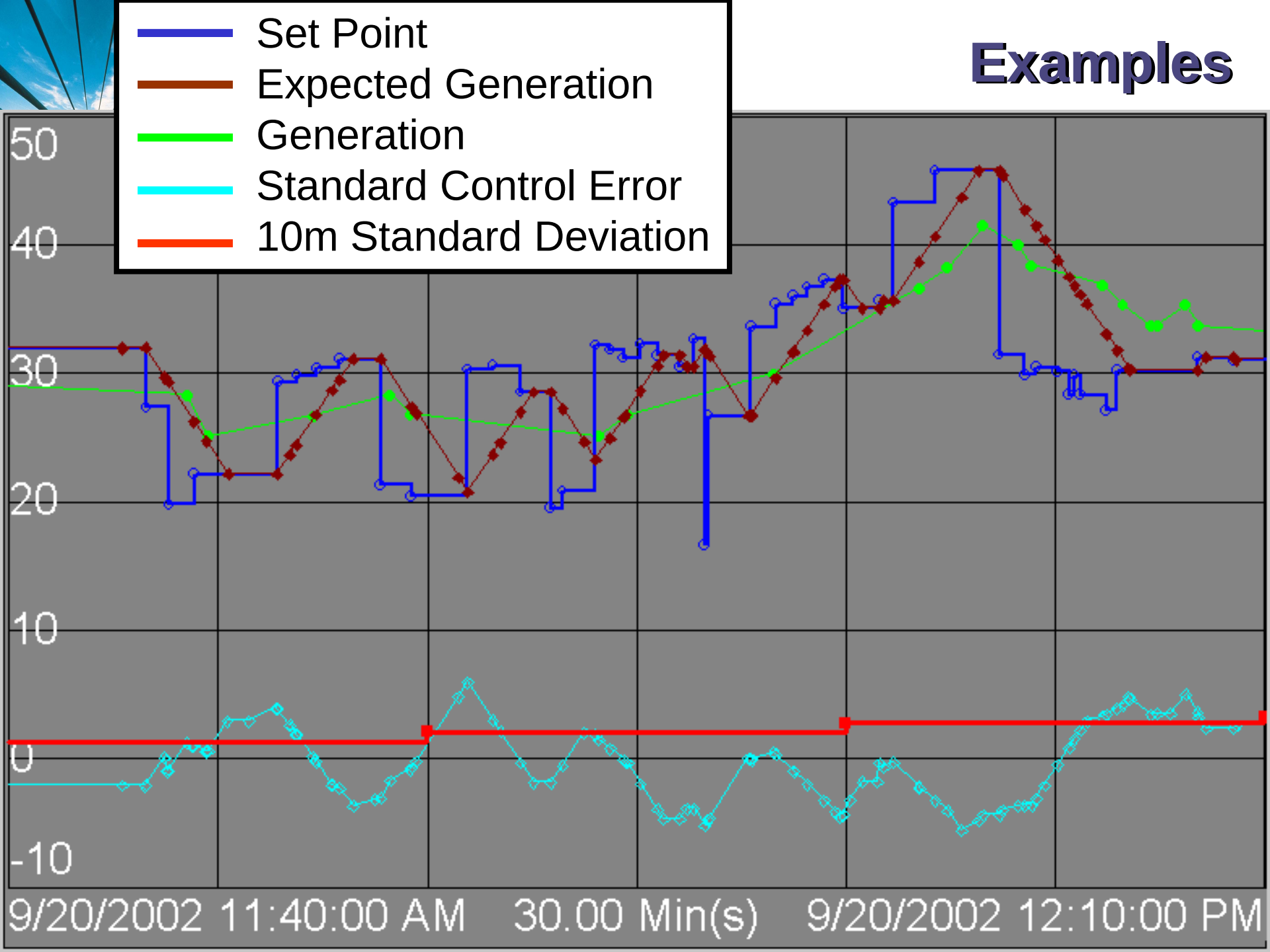
Typical Backfilling Calculation



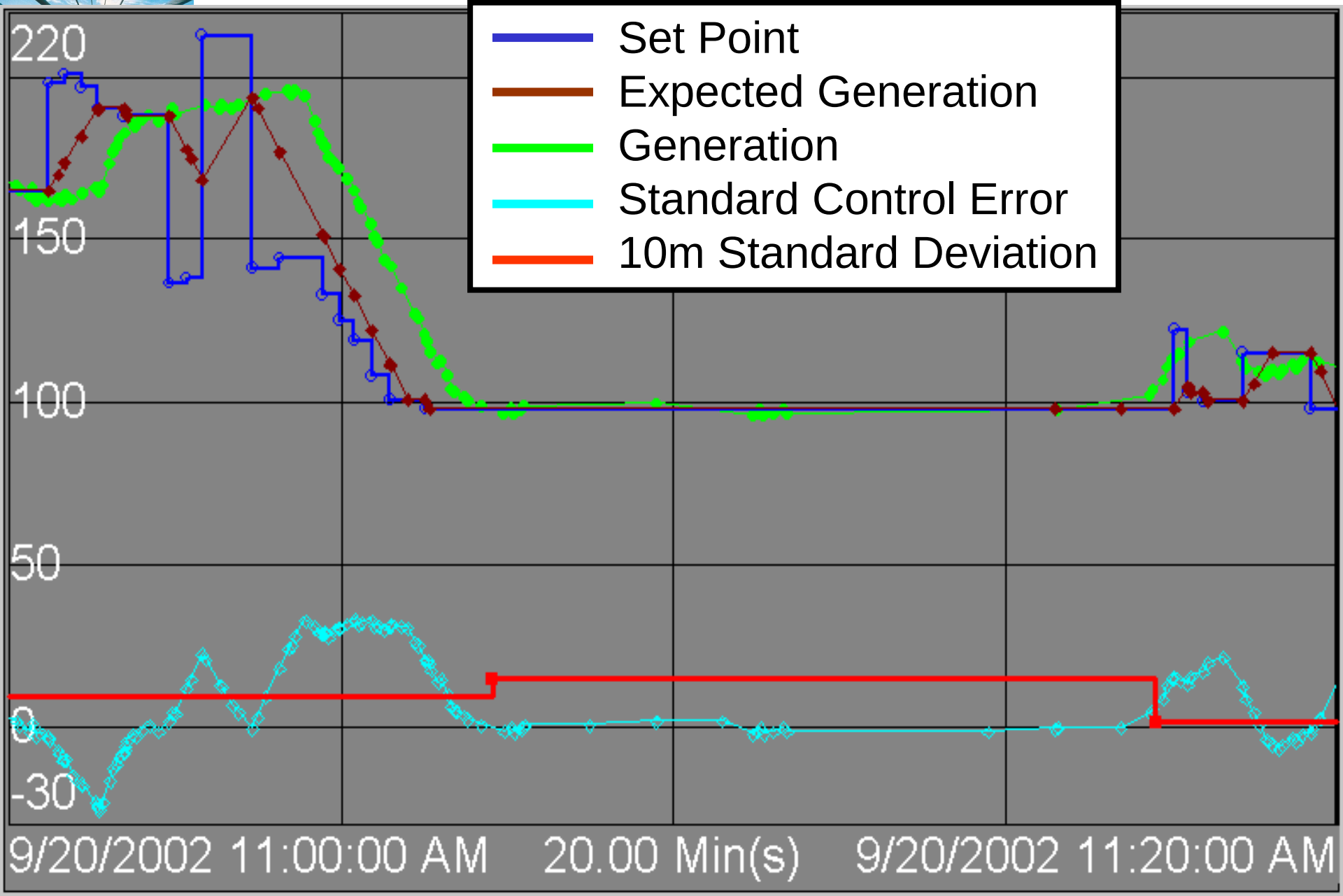


CALIFORNIA ISO
California Independent System Operator

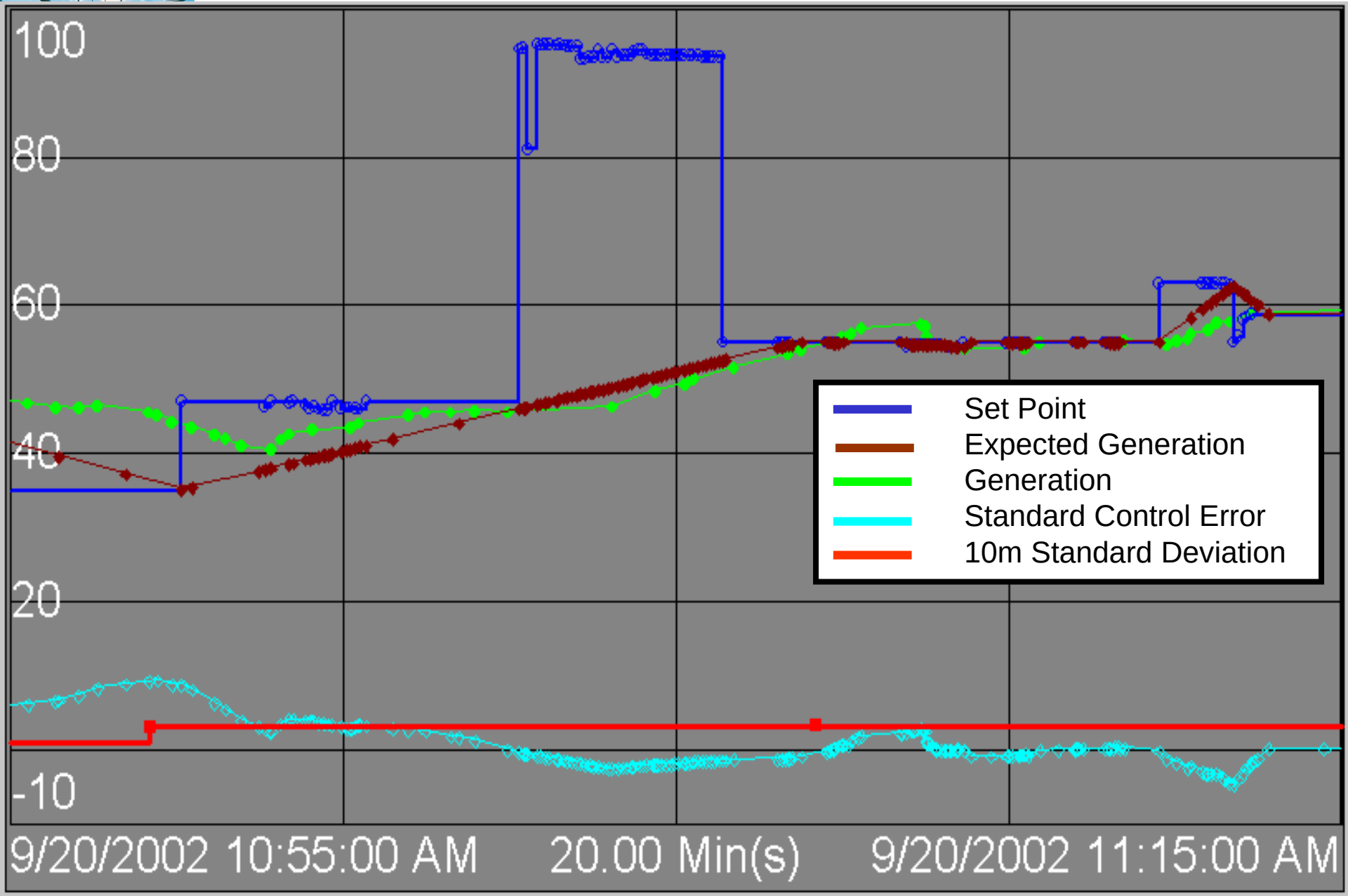
Examples


Set Point
Expected Generation
Generation
Standard Control Error
10m Standard Deviation
50
40
30
20
10
0
-10
9/20/2002 11:40:00 AM
30.00 Min(s)
9/20/2002 12:10:00 PM

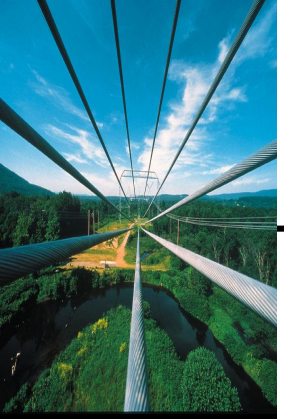
Examples


Set Point
Expected Generation
Generation
Standard Control Error
10m Standard Deviation
220
150
100
50
0
-30
9/20/2002 11:00:00 AM
20.00 Min(s)
9/20/2002 11:20:00 AM

Examples


100
80
60
40
20
-10
Set Point
Expected Generation
Generation
Standard Control Error
10m Standard Deviation
9/20/2002 10:55:00 AM
20.00 Min(s)
9/20/2002 11:15:00 AM

Issues and Solutions



(1) Dynamic Variable Sizing

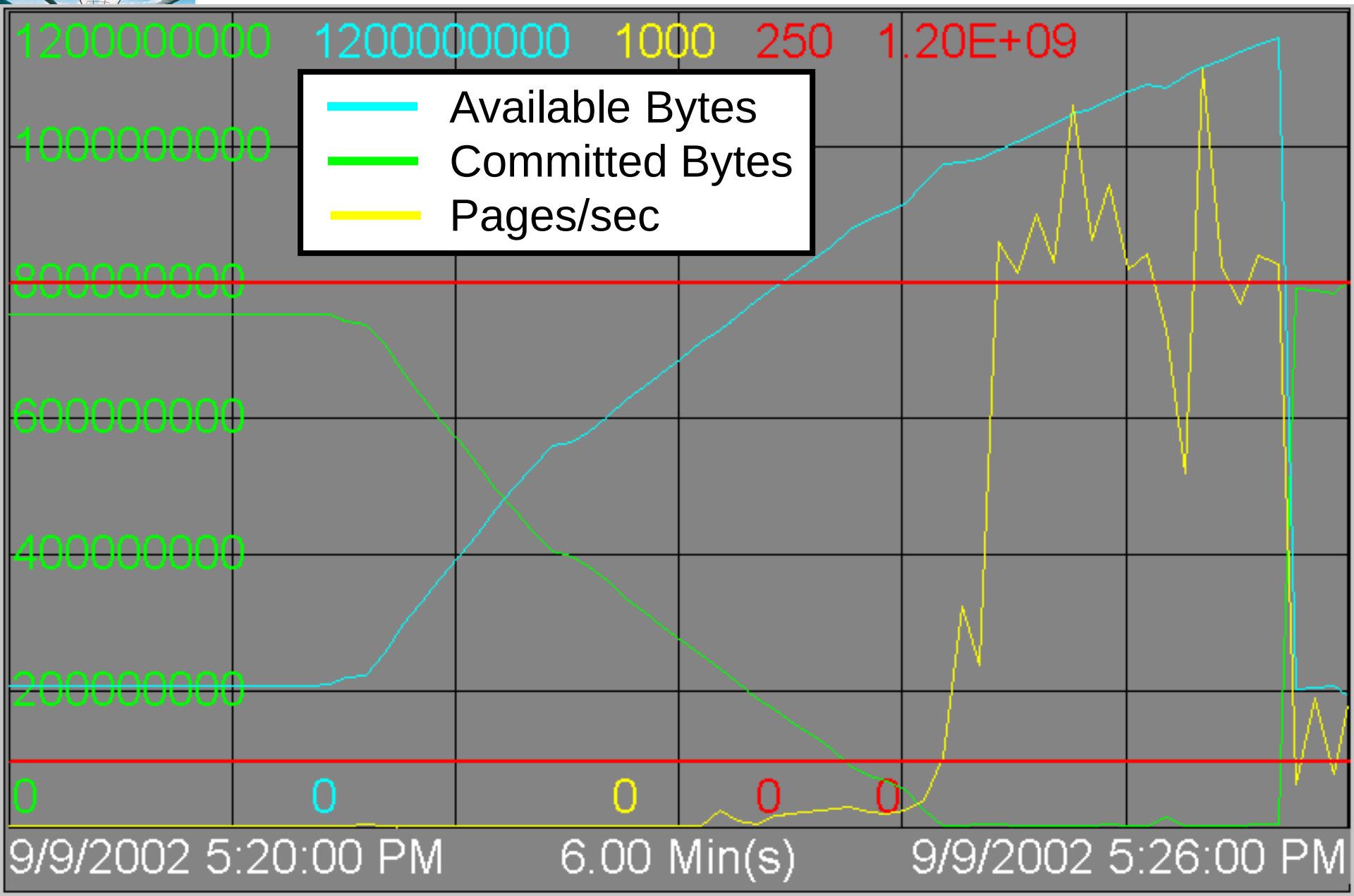
Problem

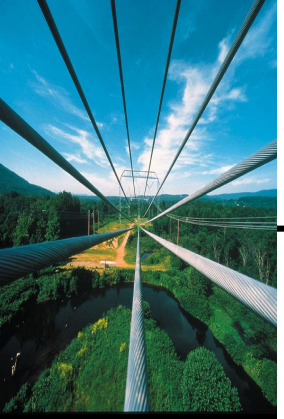
- Used variables to hold retrieved data
- Forced Available memory to 0
- Caused excessive page file use

Solution

- Used the values collection to dynamically increase/decrease storage size needed

(1) Dynamic Variable Sizing

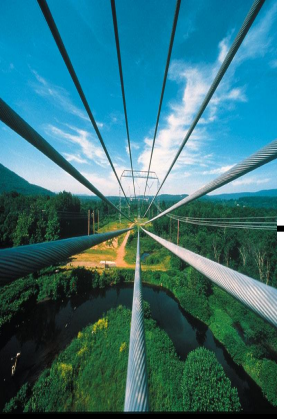




(1) Dynamic Variable Sizing

Values Collection Code Example

```
Dim piExpGenOutput As PISDK.PIValues
Dim piNamedValues As NamedValues
Dim dbTime As Double
Dim dbValue As Double

Set piExpGenOutput = New PISDK.PIValues
piExpGenOutput.ReadOnly = False
Set piNamedValues = New NamedValues
dbTime=<ENTER TIME VALUE>
dbValue=<ENTER RESULT VALUE>

piExpGenOutput.Add dbTime, dbValue, piNamedValues

For Each piVal In piExpGenOutput
   Expected_Generation.Value(piVal.TimeStamp)=piVal.Value
   Expected_Generation.PutValue
Next
```



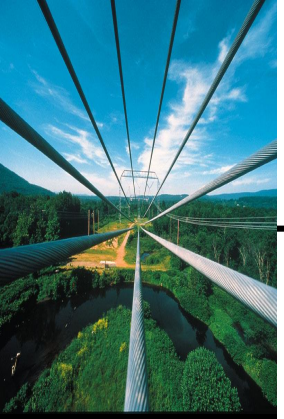
(2) Data Retrieval

Problem

- Accessed data one event at a time for long time periods (3-12 hours)
- Slowed calculation time

Solution

- Use .Values call to gather all data in one call into a values collection

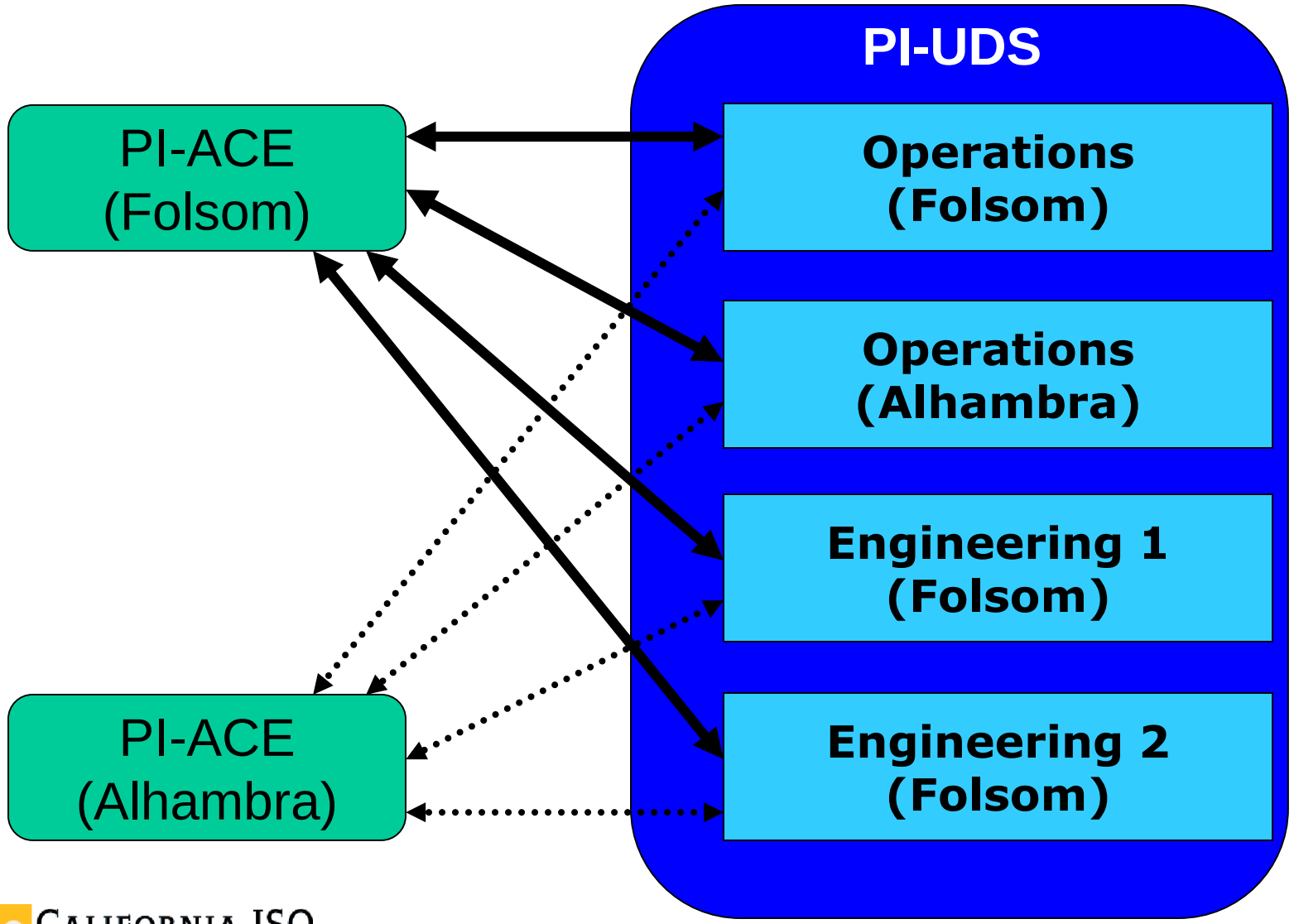


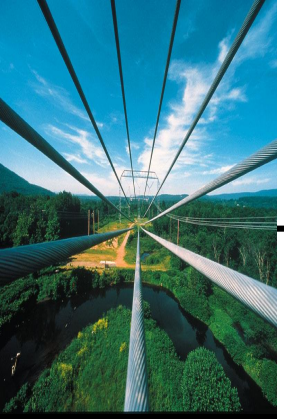
(2) Data Retrieval

.Values Call to Gather Data

```
Dim piUNMWValues As PISDK.PIValues
Dim dbWrkStartTime As Double
Dim dbWrkEndTime As Double

Set piUNMWValues = Nothing
Set piUNMWValues = UNMW_GEN_MW.Values(dbWrkStartTime,
   dbWrkEndTime, btInterp)
```

(3) 2 PI-ACE Redundant Servers





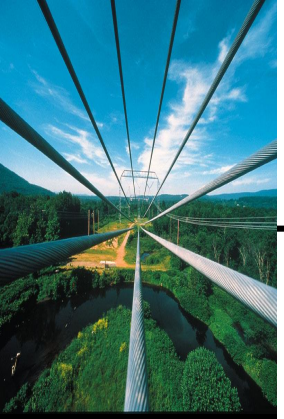
(3) 2 PI-ACE Redundant Servers

Problems

- Identified network stability as an issue
- Found that a single context could force all calculations to re-start, putting an exceptional load on server

Solutions

- Currently working with Alex Zheng to streamline calculations and test alternate configurations



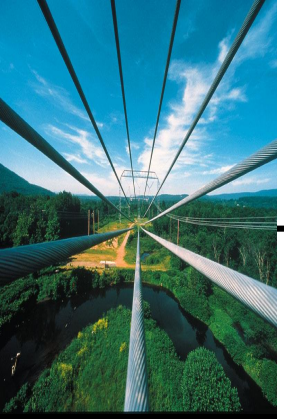
Acknowledgements

OSIsoft

- Tom Gilson
- Alex Zheng
- Don Smith
- Jon Peterson

California ISO

- Jim Hiebert
- Eric Whitley
- Tom Siegel
- Eric Leuze



Review

- Description of regulation and our metrics
- Calculation requirements and database structure
- Code layout and issue/solutions

CALIFORNIA ISO
California Independent System Operator

Discussion