OpenDSS Training Workshop - 2021

Controls in OpenDSS

Celso Rocha
EPRI Knoxville, TN

September 1, 2021

ELECTRIFICATION & SUSTAINABLE ENERGY STRATEGY
TECHNOLOGY INNOVATION
GENERATION & LOW-CARBON RESOURCES
INTEGRATED GRID & ENERGY SYSTEMS
TRANSMISSION & DISTRIBUTION INFRASTRUCTURE
NUCLEAR
EPRI

Instructor

- **Celso Rocha, *Member, IEEE***

Celso Rocha serves as Engineer Scientist II at the Electric Power Research Institute (EPRI) in Knoxville, Tennessee, USA. He holds the BSEE (2017) degree and the Master (2021) degree in Electrical Engineering with emphasis in energy and automation from University of Sao Paulo, Brazil. His work has been focused on Distribution Engineering, with a broad range of topics including DER integration, impacts and mitigation strategies assessments, active network management through optimization, DER modeling for QSTS and more recently on defining new planning methodologies for resiliency and distribution model generation, verification and validation from utility data repositories. He has 5 years of experience with OpenDSS, having taught several OpenDSS trainings at conferences, universities and industry and is part of the OpenDSS development team within EPRI.

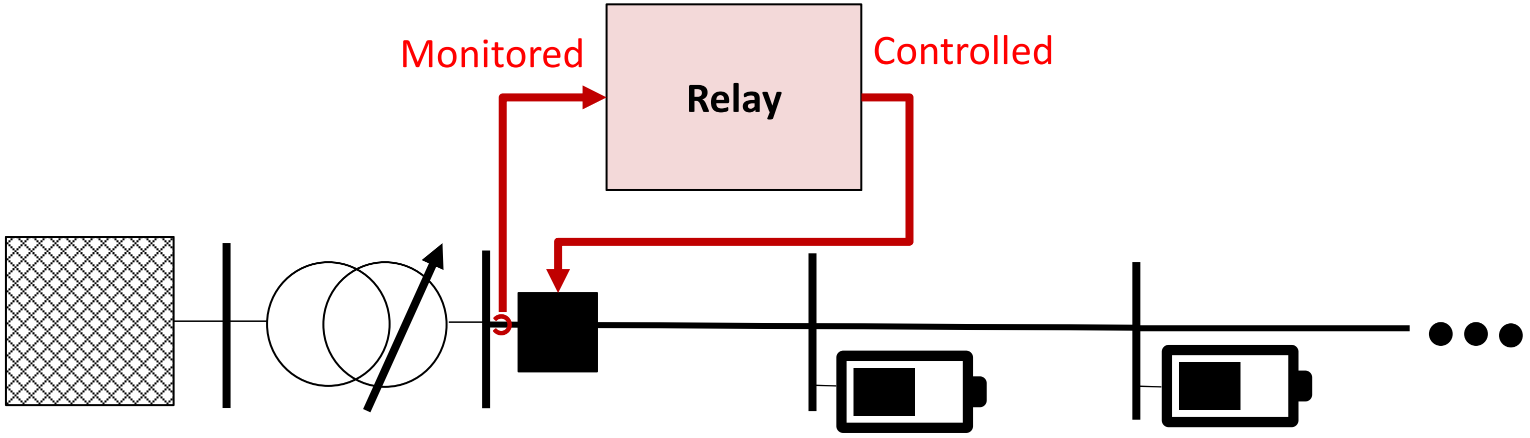
Agenda

- Controls in OpenDSS
- OpenDSS Solution with Controls and Control Modes
- Tracking Control Actions
- Customized Solution Process and Control Actions
- Examples

Controls in OpenDSS

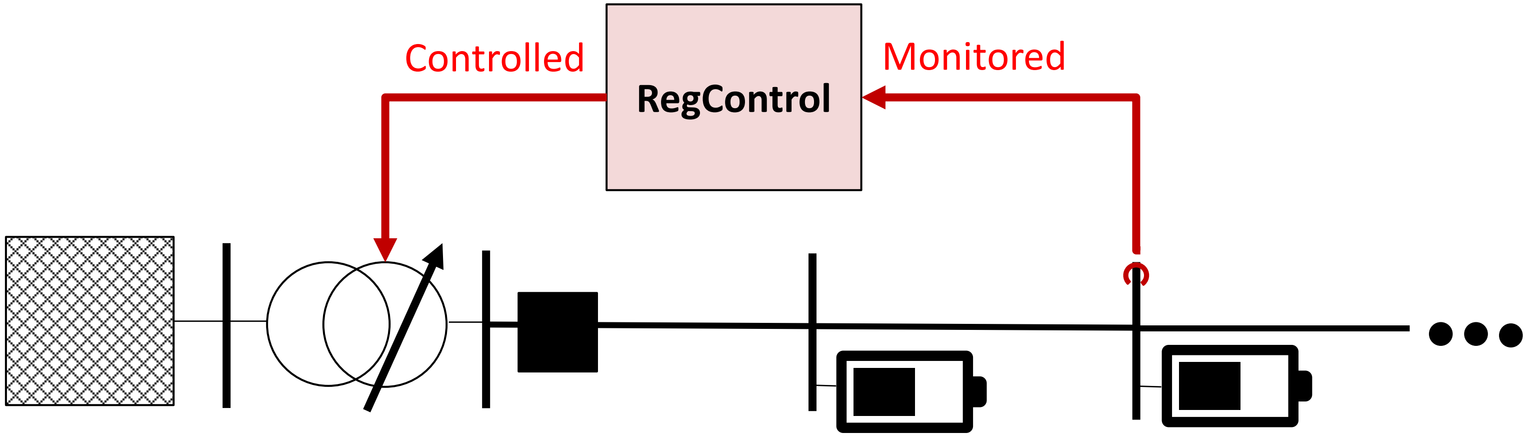
Controls in OpenDSS

- In OpenDSS, **control elements (CE)** are modeled separately from the **controlled element**
- Distinction between **controlled element** and **monitored element**



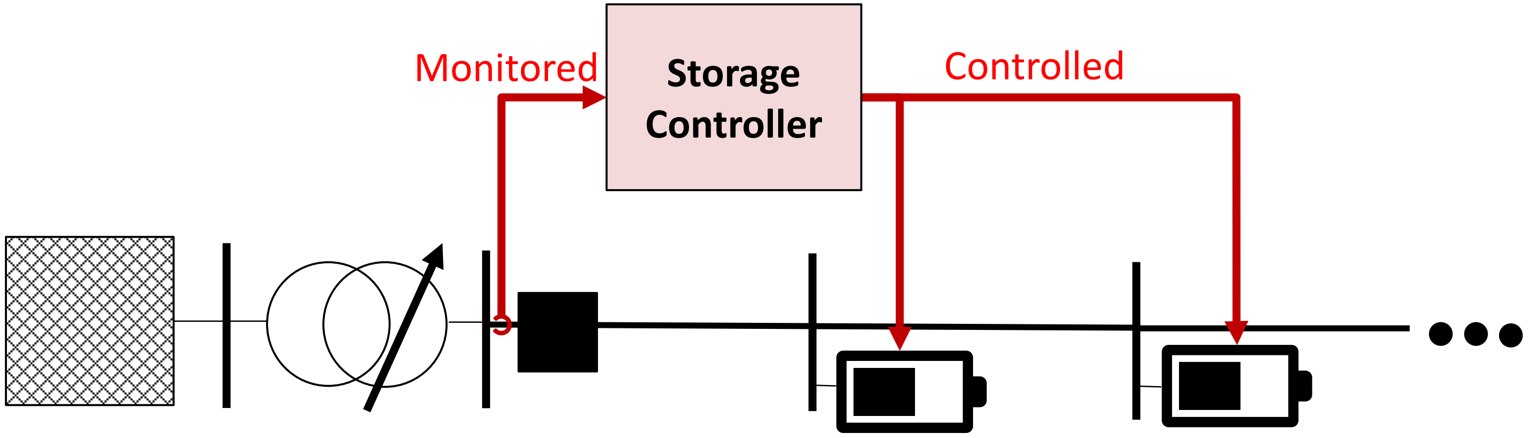
Controls in OpenDSS

- In OpenDSS, **control elements (CE)** are modeled separately from the **controlled element**
- Distinction between **controlled element** and **monitored element**
  - Locations may be different



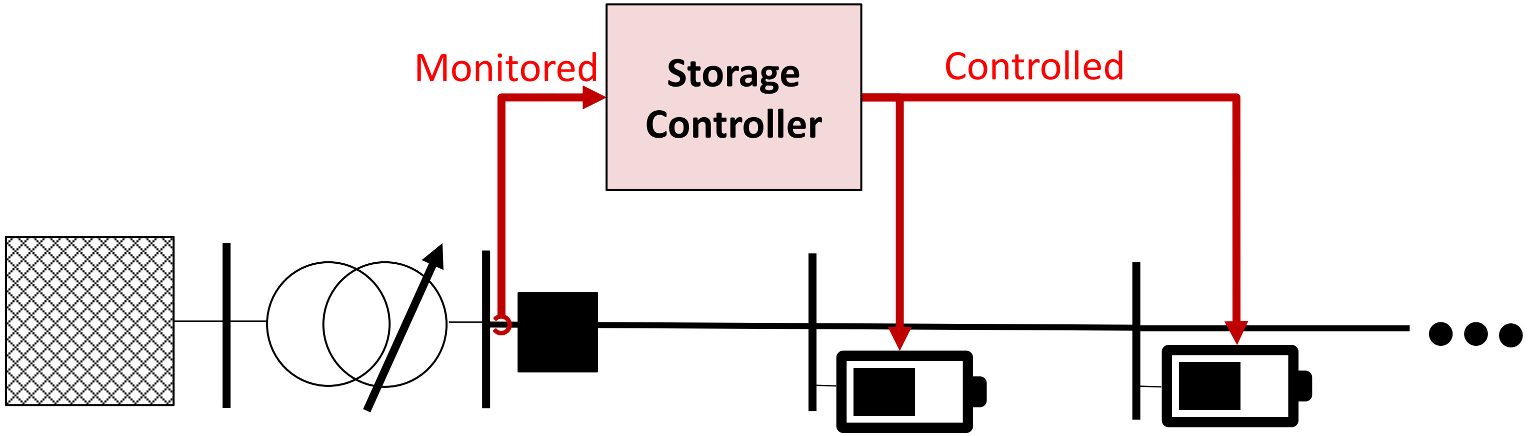
Controls in OpenDSS

- In OpenDSS, **control elements (CE)** are modeled separately from the **controlled element**
- Distinction between **controlled element** and **monitored element**
  - Locations may be different
  - May have multiple controlled elements



Controls in OpenDSS

- In OpenDSS, **control elements (CE)** are modeled separately from the **controlled element**
- Distinction between **controlled element** and **monitored element**
  - Locations may be different
  - May have multiple controlled elements
- Distinction between **controlled variable** and **monitored variable**



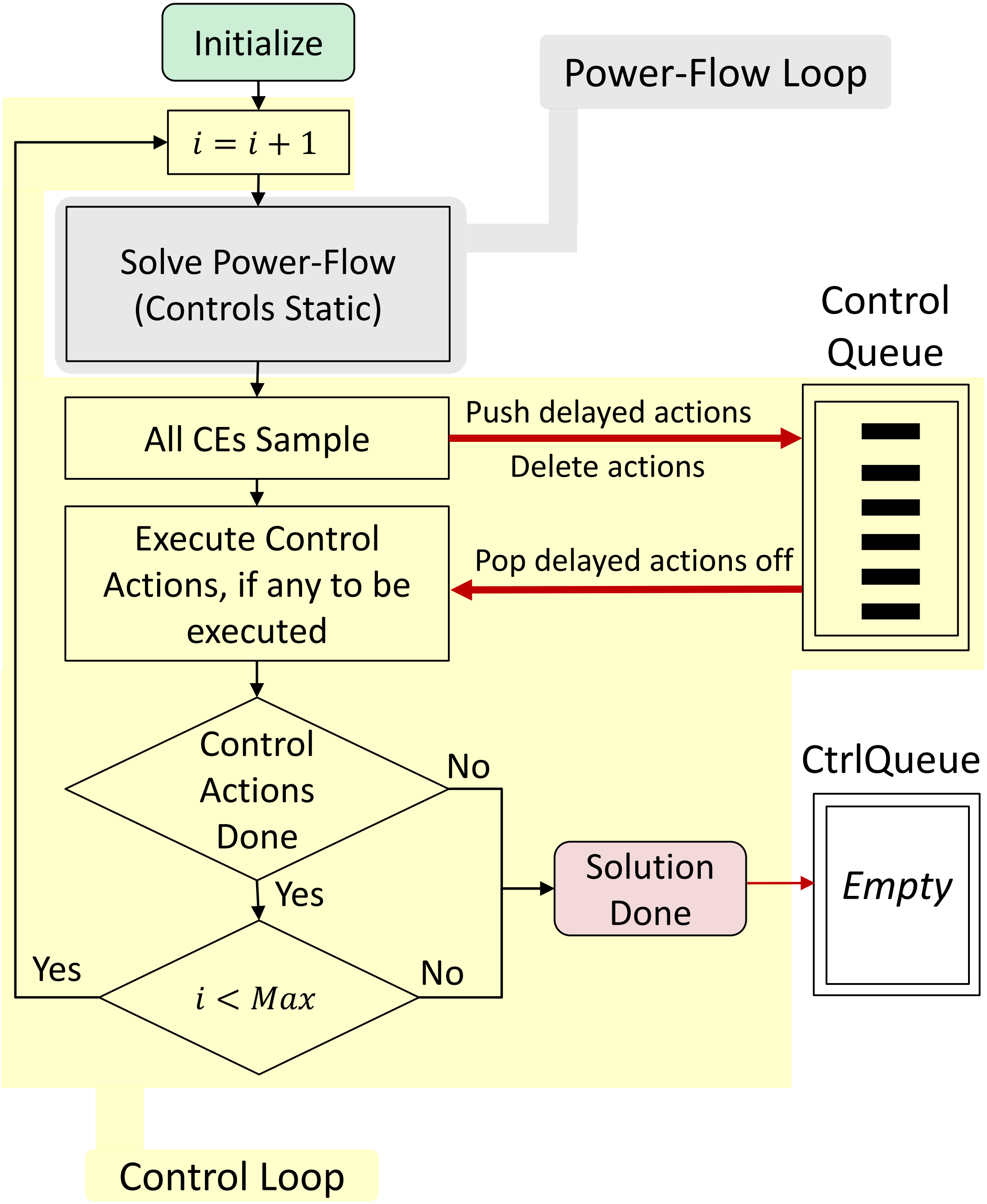
Controls in OpenDSS

Control Element	Monitored Element	Monitored Variable (s)	Controlled Element (s)	Controlled Variable (s)	Controlled Device (s) Multiplicity
CapControl	• Any PDE • Any PCE	• Current • Voltage • Reactive Power • PF • Time	Capacitors	Capacitor State	1..1
RegControl	• Transformer Winding Bus • Remote Bus	• Voltage • Current (LDC) • Active Power	Transformer (Winding)	Transformer Winding Tap Position	1..1
InvControl	• PVSystem • Storage • Remote Bus	• Voltage • Power	• PVSystem • Storage	• Active Power • Reactive Power	• 1..N (per CE Instance) • 1..1 (per Monitored Device)
StorageController	• Any PDE • Any PCE	• Active Power • Current	Storage	Active Power	1..N
Fuse, Recloser, Relay, SwtControl	• Any PDE • Any PCE	• Current (All) • Voltage (Relay)	• Any PDE • Any PCE	Switching Device State	1..1

OpenDSS Solution with Controls and Control Modes

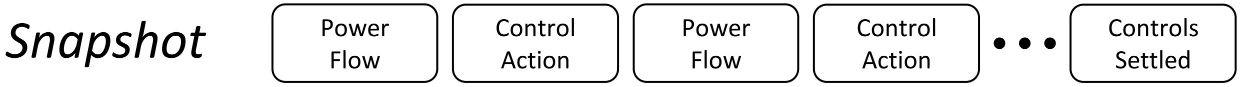
OpenDSS Solution with Controls

- All built-in Control Elements are *Autonomous*
  - *Which one should operate first?*
- Use of a Control Loop around the Power-Flow Solution
  - Controls are Sampled and Executed after a converged power-flow Solution
  - Control Queue stores actions to be *possibly executed*
  - The popping of actions from the Control Queue are dictated by different control modes
    - Static (figure), Time, Event

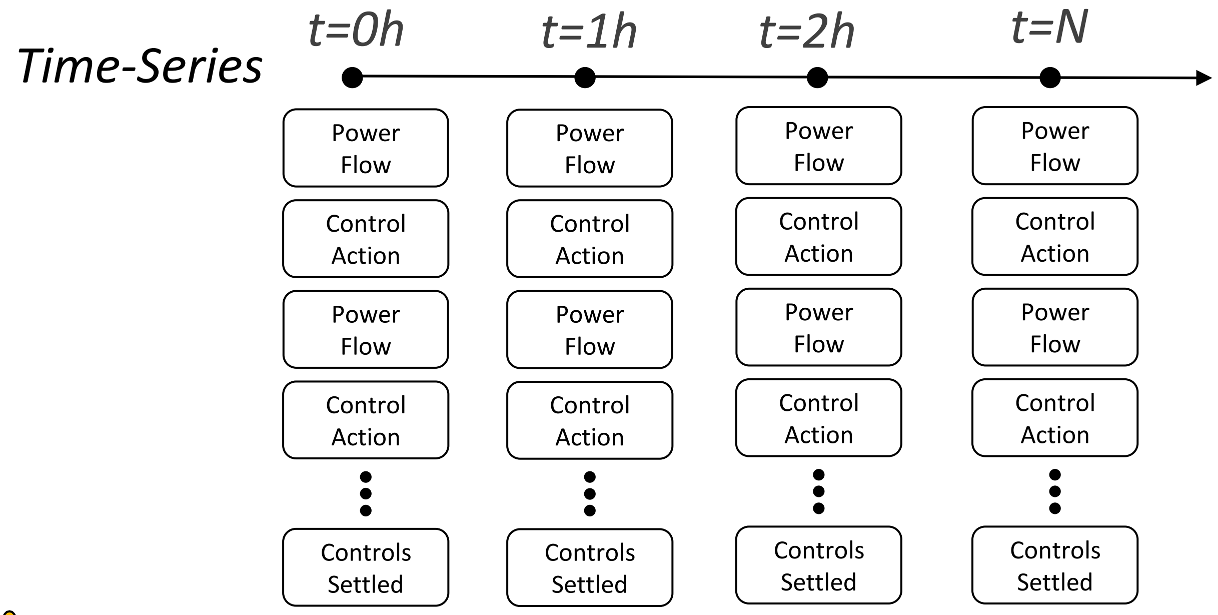


Control Modes - Static

- Solution is complete until all control actions from Control Queue settle out (empty control queue)
- At each control iteration, only the **closest control action in time** is executed
- **Solution time does not advance**
- Typically used to reach a steady state with the controls active
  - In *Snapshot* solution mode
  - In *Time-Series* solution modes (*daily, yearly, …*) when time step size is much greater than the control delays



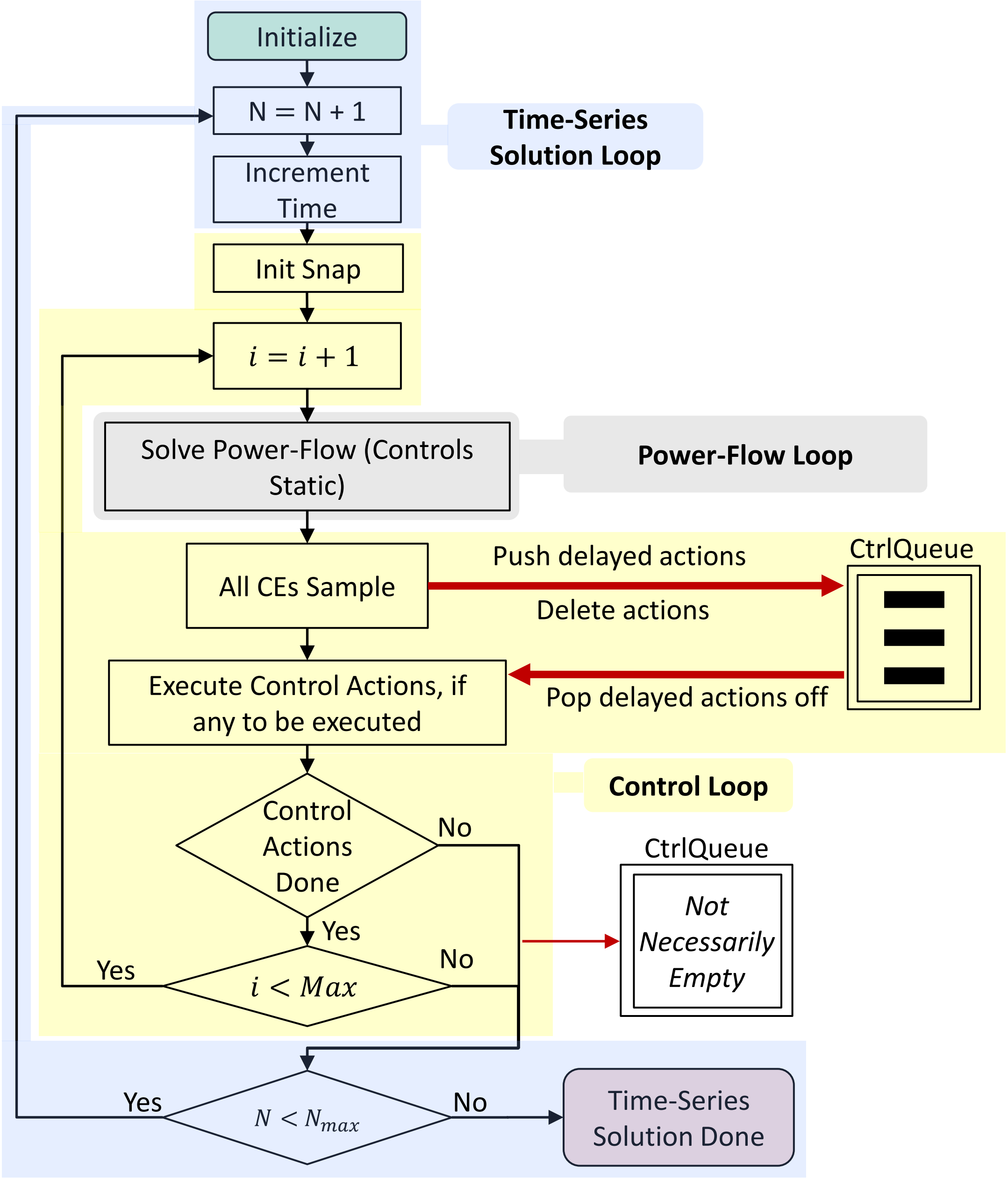
⚠ In snapshot mode, even though there is no solution time defined, the delays associated to each control device are utilized to determine which control acts first



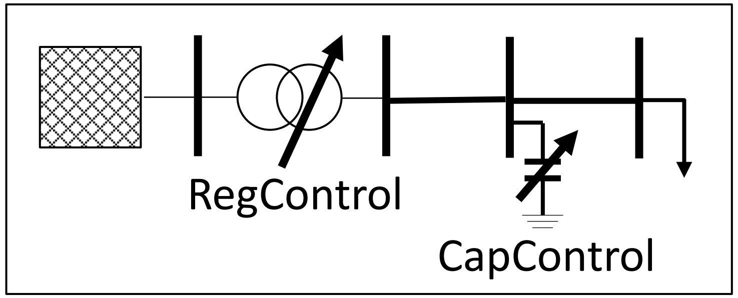
⚠ If there are devices with the same time delay, multiple control actions might be executed at the same control loop iteration, which may lead to the "hunting effect"

Control Modes - Time

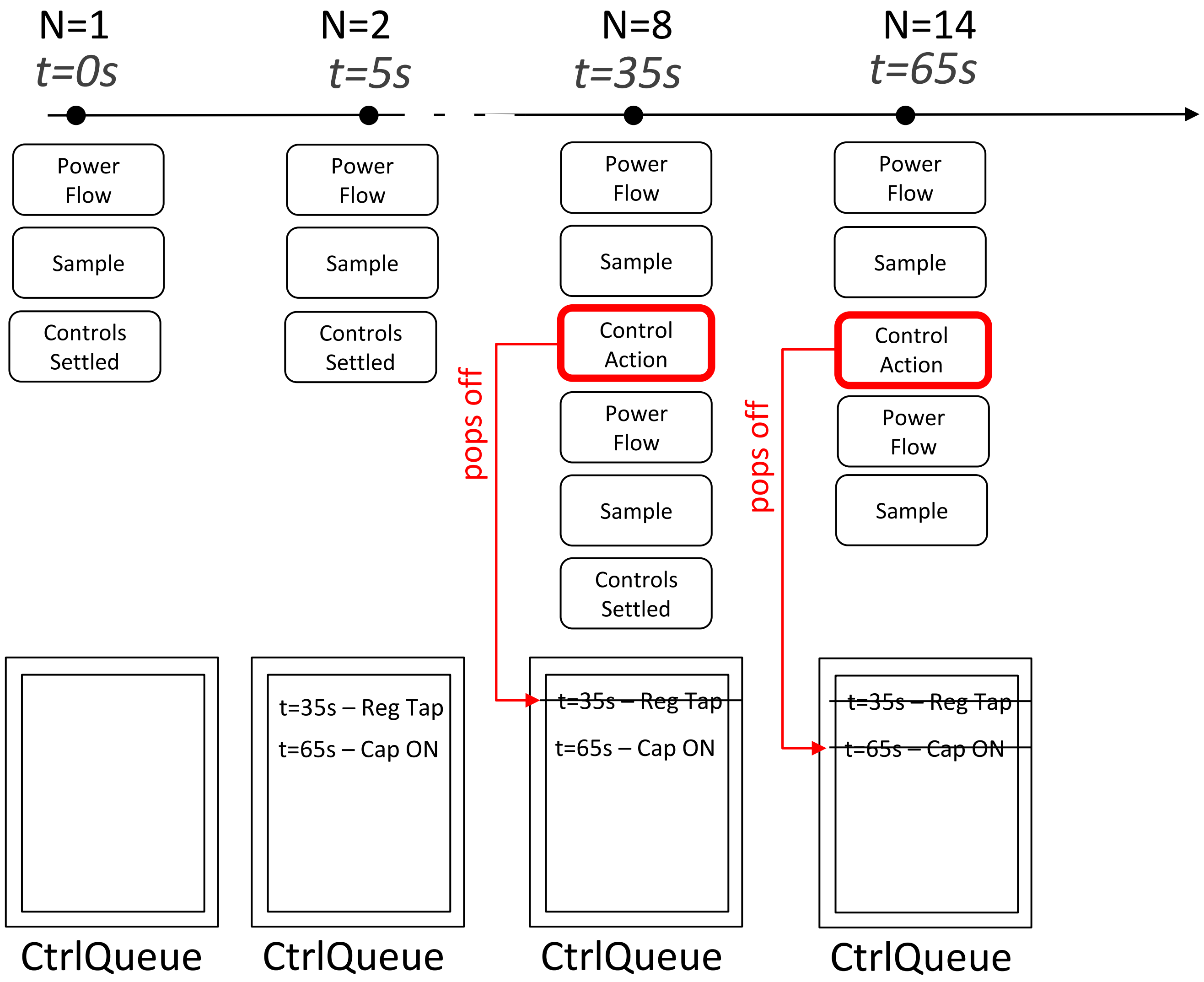
- Solution is **time-driven**
- Solution will proceed at a **fixed time step** (solution *stepsize*), executing any pending actions when their time is reached or surpassed
- **Solution time does not advance**
- Used when the phenomenon under study have time constants less than the control delays
  - In *Time-Series* solution modes (*daily*, *yearly*, …) when time step size is less than the control delays



Control Modes - Time



```
New RegControl.myRegControl
~transformer=Reg delay=30

New CapControl.myCapControl
~capacitor=Cap delay=60
~delayoff=60

Set mode=daily

Set controlmode=time

Set stepsize=5s

Set number=14

Solve
```



N=1 *t=0s*

Power Flow

Sample

Controls Settled

CtrlQueue

N=2 *t=5s*

Power Flow

Sample

Controls Settled

t=35s – Reg Tap

t=65s – Cap ON

CtrlQueue

N=8 *t=35s*

Power Flow

Sample

Control Action

pops off

Power Flow

Sample

Controls Settled

~~t=35s – Reg Tap~~

t=65s – Cap ON

CtrlQueue

N=14 *t=65s*

Power Flow

Sample

Control Action

pops off

Power Flow

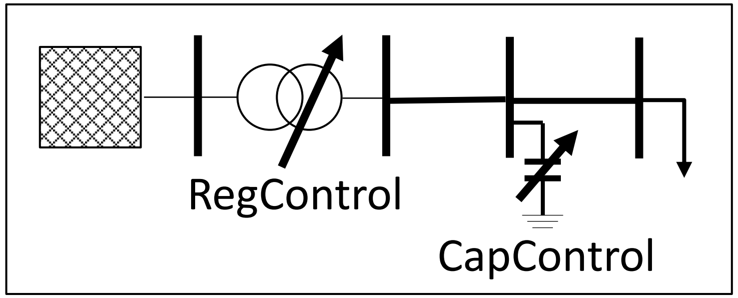
Sample

~~t=35s – Reg Tap~~

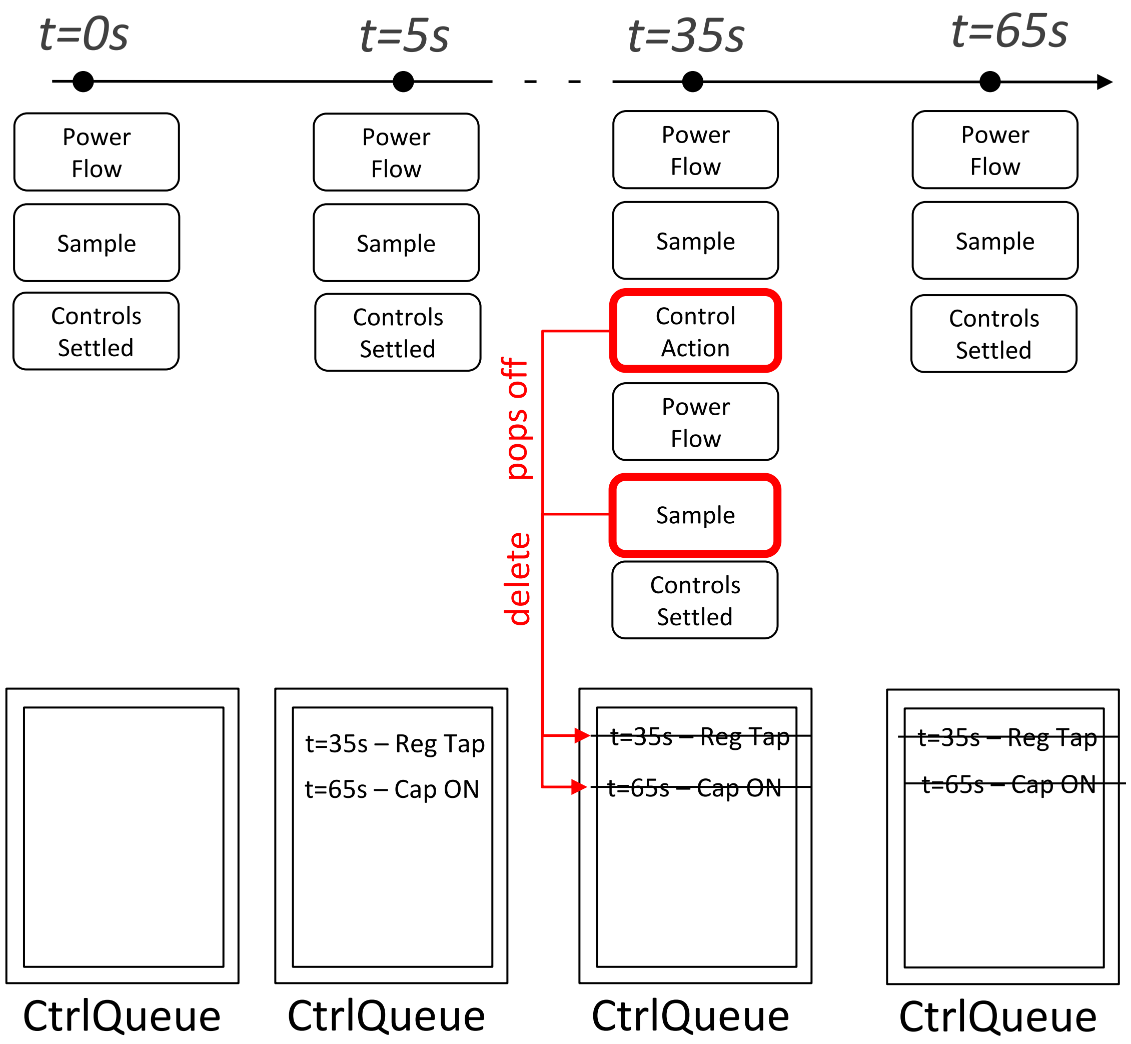
~~t=65s – Cap ON~~

CtrlQueue

Control Modes - Time

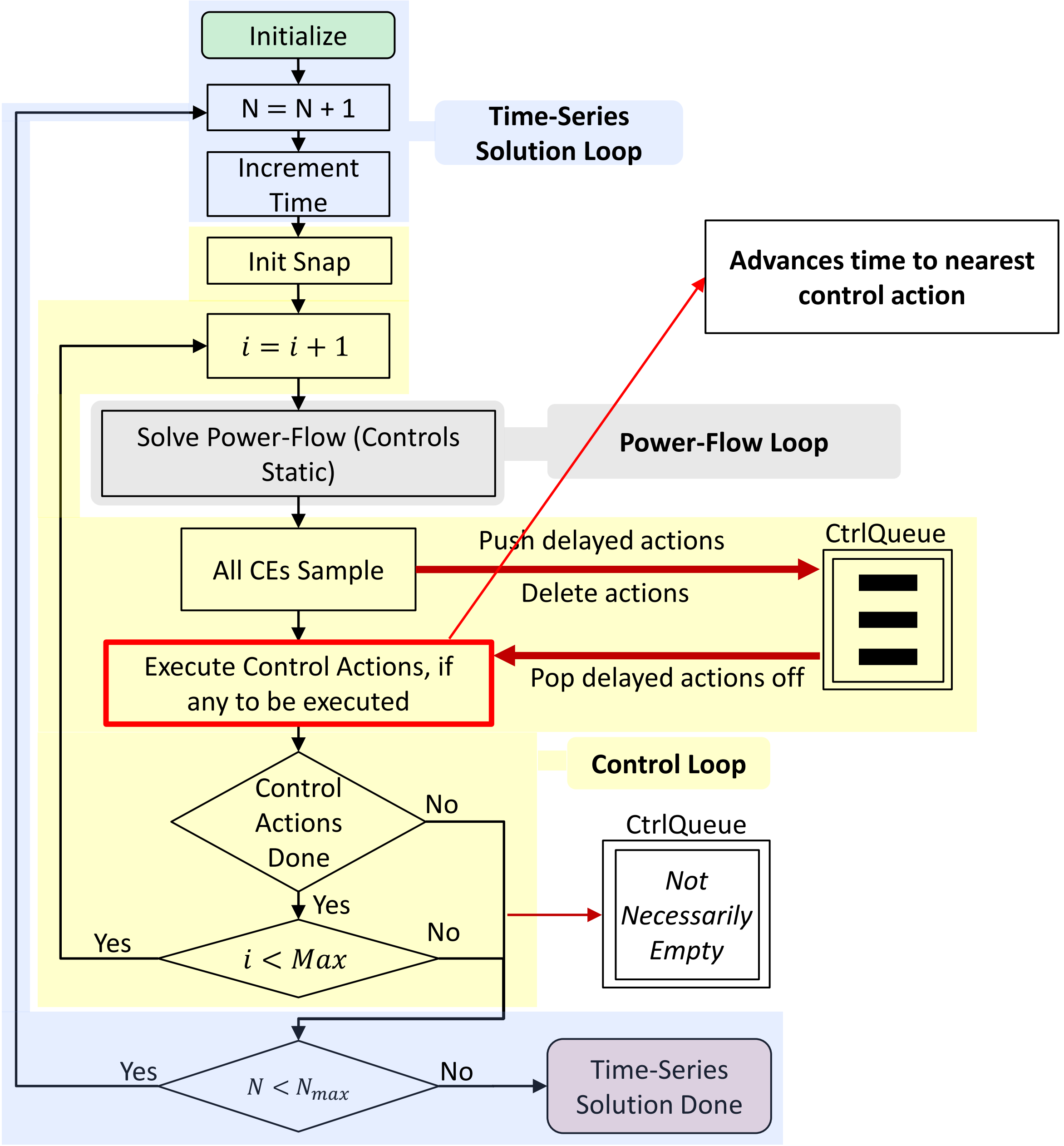


```
New RegControl.myRegControl
~transformer=Reg delay=30

New CapControl.myCapControl
~capacitor=Cap delay=60
~delayoff=60

Set mode=daily

Set controlmode=time

Set stepsize=5s

Set number=14

Solve
```

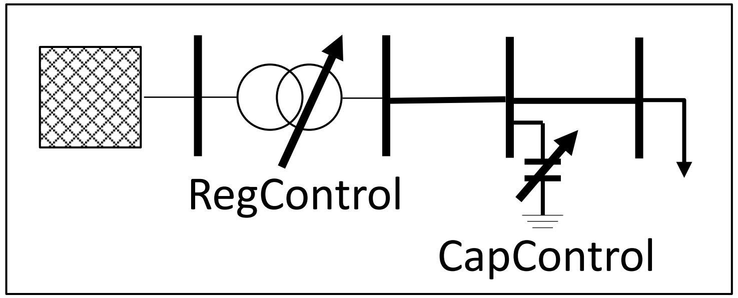


Control Modes - Event

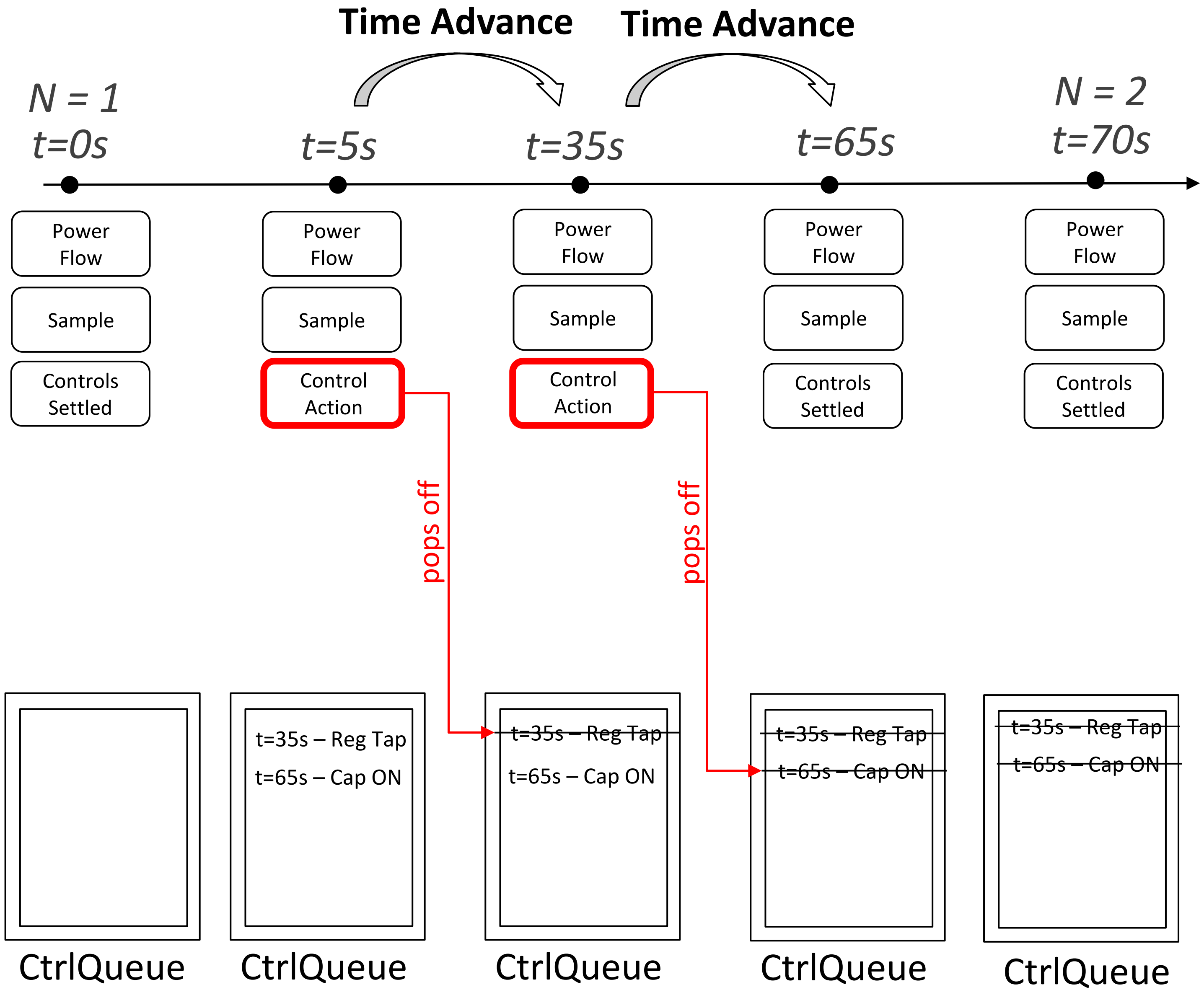
- Solution is event driven
- Solution proceeds at a **variable time step**, running from one control action to the next one
- Only the control action nearest in time are executed and the **time advances automatically for the time of the event**
- Use when the focus of the analysis in on the control events (e.g., sequence of events for protection coordination in dynamics mode)



Control Modes - Event



```
New RegControl.myRegControl
~transformer=Reg delay=30

New CapControl.myCapControl
~capacitor=Cap delay=60
~delayoff=60

Set mode=daily

Set controlmode=event

Set stepsize=5s

Set number=2

Solve
```



Time Advance

Time Advance

N = 1 t=0s: Power Flow; Sample; Controls Settled; CtrlQueue

t=5s: Power Flow; Sample; Control Action; pops off; CtrlQueue: t=35s – Reg Tap, t=65s – Cap ON

t=35s: Power Flow; Sample; Control Action; pops off; CtrlQueue: ~~t=35s – Reg Tap~~, t=65s – Cap ON

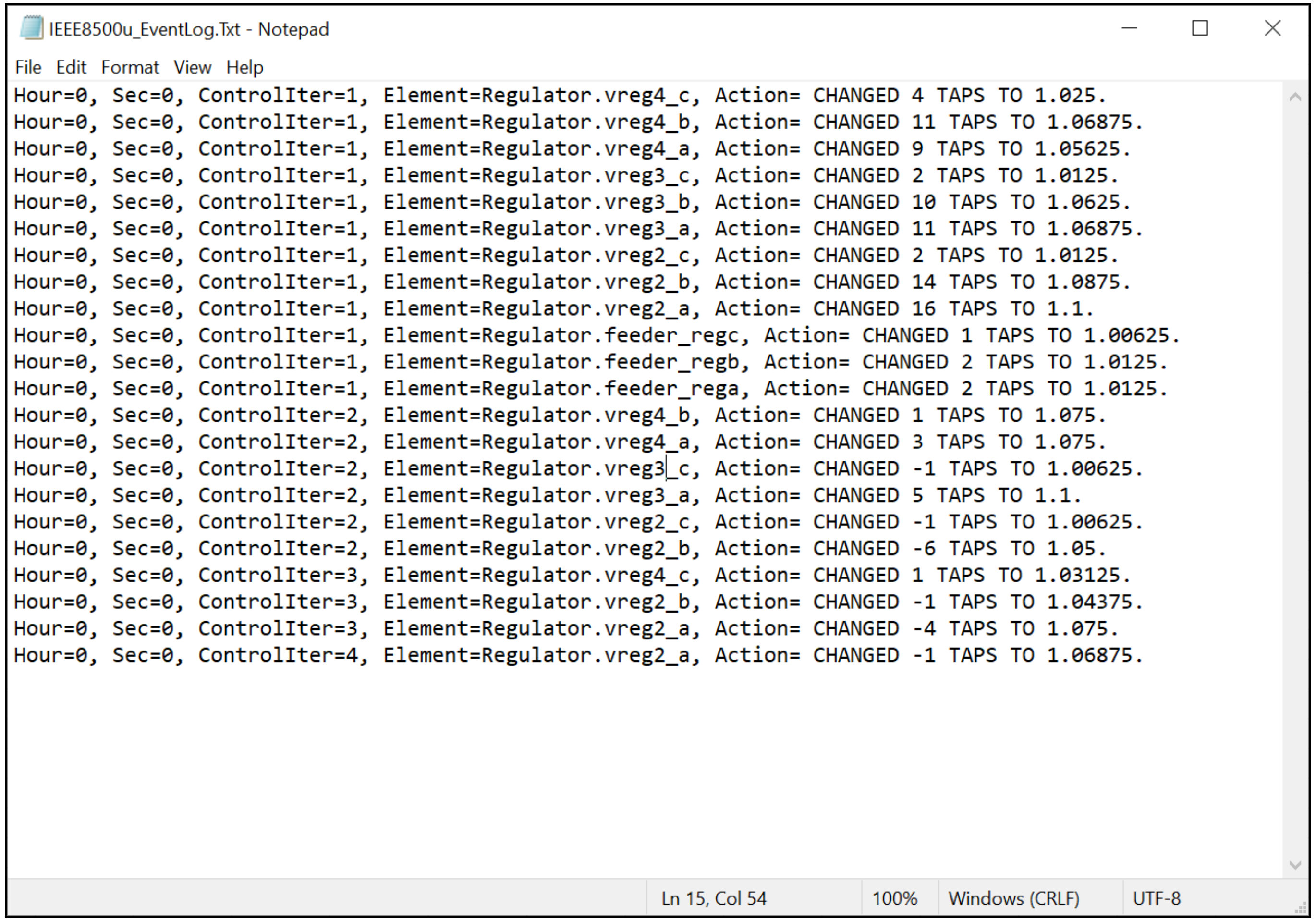
t=65s: Power Flow; Sample; Controls Settled; CtrlQueue: ~~t=35s – Reg Tap~~, ~~t=65s – Cap ON~~

N = 2 t=70s: Power Flow; Sample; Controls Settled; CtrlQueue: ~~t=35s – Reg Tap~~, ~~t=65s – Cap ON~~

Tracking Control Actions

Tracking Control Actions

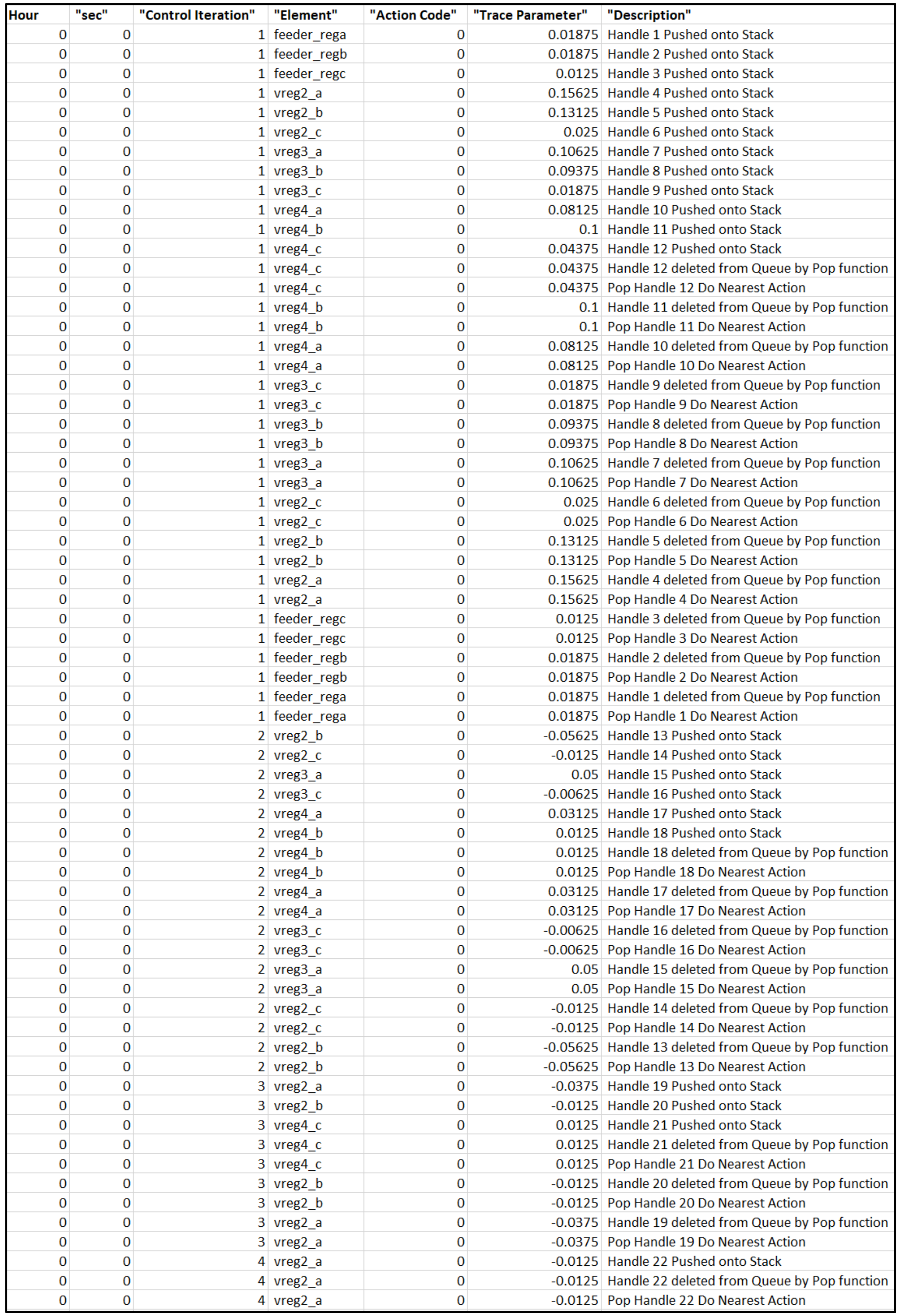
- Tracking control actions from a system-wide perspective:
  - **Show EventLog** ⟶
    - After executing a simulation
    - Make sure to enable logging in each control device through **eventlog=True**
  - **Set tracecontrol=True**
    - Before solving the model
    - After solving, “Trace_ControlQueue.csv” file will be generated



```
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg4_c, Action= CHANGED 4 TAPS TO 1.025.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg4_b, Action= CHANGED 11 TAPS TO 1.06875.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg4_a, Action= CHANGED 9 TAPS TO 1.05625.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg3_c, Action= CHANGED 2 TAPS TO 1.0125.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg3_b, Action= CHANGED 10 TAPS TO 1.0625.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg3_a, Action= CHANGED 11 TAPS TO 1.06875.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg2_c, Action= CHANGED 2 TAPS TO 1.0125.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg2_b, Action= CHANGED 14 TAPS TO 1.0875.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.vreg2_a, Action= CHANGED 16 TAPS TO 1.1.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.feeder_regc, Action= CHANGED 1 TAPS TO 1.00625.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.feeder_regb, Action= CHANGED 2 TAPS TO 1.0125.
Hour=0, Sec=0, ControlIter=1, Element=Regulator.feeder_rega, Action= CHANGED 2 TAPS TO 1.0125.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg4_b, Action= CHANGED 1 TAPS TO 1.075.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg4_a, Action= CHANGED 3 TAPS TO 1.075.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg3_c, Action= CHANGED -1 TAPS TO 1.00625.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg3_a, Action= CHANGED 5 TAPS TO 1.1.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg2_c, Action= CHANGED -1 TAPS TO 1.00625.
Hour=0, Sec=0, ControlIter=2, Element=Regulator.vreg2_b, Action= CHANGED -6 TAPS TO 1.05.
Hour=0, Sec=0, ControlIter=3, Element=Regulator.vreg4_c, Action= CHANGED 1 TAPS TO 1.03125.
Hour=0, Sec=0, ControlIter=3, Element=Regulator.vreg2_b, Action= CHANGED -1 TAPS TO 1.04375.
Hour=0, Sec=0, ControlIter=3, Element=Regulator.vreg2_a, Action= CHANGED -4 TAPS TO 1.075.
Hour=0, Sec=0, ControlIter=4, Element=Regulator.vreg2_a, Action= CHANGED -1 TAPS TO 1.06875.
```

Tracking Control Actions

- Tracking control actions from a system-wide perspective:
  - Show EventLog
    - After executing a simulation
    - Make sure to enable logging in each control device through eventlog=True
  - Set tracecontrol=True →
    - Before solving the model
    - After solving, "Trace_ControlQueue.csv" file will be generated

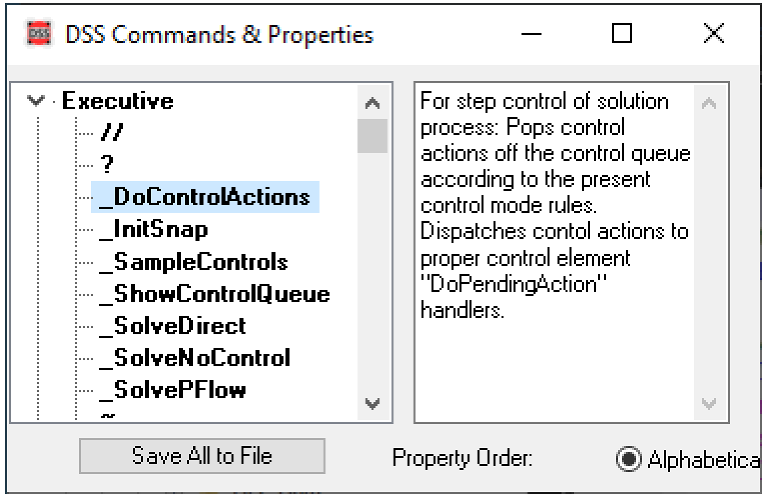


Hour	"sec"	"Control Iteration"	"Element"	"Action Code"	"Trace Parameter"	"Description"
0	0	1	feeder_rega	0	0.01875	Handle 1 Pushed onto Stack
0	0	1	feeder_regb	0	0.01875	Handle 2 Pushed onto Stack
0	0	1	feeder_regc	0	0.0125	Handle 3 Pushed onto Stack
0	0	1	vreg2_a	0	0.15625	Handle 4 Pushed onto Stack
0	0	1	vreg2_b	0	0.13125	Handle 5 Pushed onto Stack
0	0	1	vreg2_c	0	0.025	Handle 6 Pushed onto Stack
0	0	1	vreg3_a	0	0.10625	Handle 7 Pushed onto Stack
0	0	1	vreg3_b	0	0.09375	Handle 8 Pushed onto Stack
0	0	1	vreg3_c	0	0.01875	Handle 9 Pushed onto Stack
0	0	1	vreg4_a	0	0.08125	Handle 10 Pushed onto Stack
0	0	1	vreg4_b	0	0.1	Handle 11 Pushed onto Stack
0	0	1	vreg4_c	0	0.04375	Handle 12 Pushed onto Stack
0	0	1	vreg4_c	0	0.04375	Handle 12 deleted from Queue by Pop function
0	0	1	vreg4_c	0	0.04375	Pop Handle 12 Do Nearest Action
0	0	1	vreg4_b	0	0.1	Handle 11 deleted from Queue by Pop function
0	0	1	vreg4_b	0	0.1	Pop Handle 11 Do Nearest Action
0	0	1	vreg4_a	0	0.08125	Handle 10 deleted from Queue by Pop function
0	0	1	vreg4_a	0	0.08125	Pop Handle 10 Do Nearest Action
0	0	1	vreg3_c	0	0.01875	Handle 9 deleted from Queue by Pop function
0	0	1	vreg3_c	0	0.01875	Pop Handle 9 Do Nearest Action
0	0	1	vreg3_b	0	0.09375	Handle 8 deleted from Queue by Pop function
0	0	1	vreg3_b	0	0.09375	Pop Handle 8 Do Nearest Action
0	0	1	vreg3_a	0	0.10625	Handle 7 deleted from Queue by Pop function
0	0	1	vreg3_a	0	0.10625	Pop Handle 7 Do Nearest Action
0	0	1	vreg2_c	0	0.025	Handle 6 deleted from Queue by Pop function
0	0	1	vreg2_c	0	0.025	Pop Handle 6 Do Nearest Action
0	0	1	vreg2_b	0	0.13125	Handle 5 deleted from Queue by Pop function
0	0	1	vreg2_b	0	0.13125	Pop Handle 5 Do Nearest Action
0	0	1	vreg2_a	0	0.15625	Handle 4 deleted from Queue by Pop function
0	0	1	vreg2_a	0	0.15625	Pop Handle 4 Do Nearest Action
0	0	1	feeder_regc	0	0.0125	Handle 3 deleted from Queue by Pop function
0	0	1	feeder_regc	0	0.0125	Pop Handle 3 Do Nearest Action
0	0	1	feeder_regb	0	0.01875	Handle 2 deleted from Queue by Pop function
0	0	1	feeder_regb	0	0.01875	Pop Handle 2 Do Nearest Action
0	0	1	feeder_rega	0	0.01875	Handle 1 deleted from Queue by Pop function
0	0	1	feeder_rega	0	0.01875	Pop Handle 1 Do Nearest Action
0	0	2	vreg2_b	0	-0.05625	Handle 13 Pushed onto Stack
0	0	2	vreg2_c	0	-0.0125	Handle 14 Pushed onto Stack
0	0	2	vreg3_a	0	0.05	Handle 15 Pushed onto Stack
0	0	2	vreg3_c	0	-0.00625	Handle 16 Pushed onto Stack
0	0	2	vreg4_a	0	0.03125	Handle 17 Pushed onto Stack
0	0	2	vreg4_b	0	0.0125	Handle 18 Pushed onto Stack
0	0	2	vreg4_b	0	0.0125	Handle 18 deleted from Queue by Pop function
0	0	2	vreg4_b	0	0.0125	Pop Handle 18 Do Nearest Action
0	0	2	vreg4_a	0	0.03125	Handle 17 deleted from Queue by Pop function
0	0	2	vreg4_a	0	0.03125	Pop Handle 17 Do Nearest Action
0	0	2	vreg3_c	0	-0.00625	Handle 16 deleted from Queue by Pop function
0	0	2	vreg3_c	0	-0.00625	Pop Handle 16 Do Nearest Action
0	0	2	vreg3_a	0	0.05	Handle 15 deleted from Queue by Pop function
0	0	2	vreg3_a	0	0.05	Pop Handle 15 Do Nearest Action
0	0	2	vreg2_c	0	-0.0125	Handle 14 deleted from Queue by Pop function
0	0	2	vreg2_c	0	-0.0125	Pop Handle 14 Do Nearest Action
0	0	2	vreg2_b	0	-0.05625	Handle 13 deleted from Queue by Pop function
0	0	2	vreg2_b	0	-0.05625	Pop Handle 13 Do Nearest Action
0	0	3	vreg2_a	0	-0.0375	Handle 19 Pushed onto Stack
0	0	3	vreg2_b	0	-0.0125	Handle 20 Pushed onto Stack
0	0	3	vreg4_c	0	0.0125	Handle 21 Pushed onto Stack
0	0	3	vreg4_c	0	0.0125	Handle 21 deleted from Queue by Pop function
0	0	3	vreg4_c	0	0.0125	Pop Handle 21 Do Nearest Action
0	0	3	vreg2_b	0	-0.0125	Handle 20 deleted from Queue by Pop function
0	0	3	vreg2_b	0	-0.0125	Pop Handle 20 Do Nearest Action
0	0	3	vreg2_a	0	-0.0375	Handle 19 deleted from Queue by Pop function
0	0	3	vreg2_a	0	-0.0375	Pop Handle 19 Do Nearest Action
0	0	4	vreg2_a	0	-0.0125	Handle 22 Pushed onto Stack
0	0	4	vreg2_a	0	-0.0125	Handle 22 deleted from Queue by Pop function
0	0	4	vreg2_a	0	-0.0125	Pop Handle 22 Do Nearest Action

Customized Solution Process and Control Actions

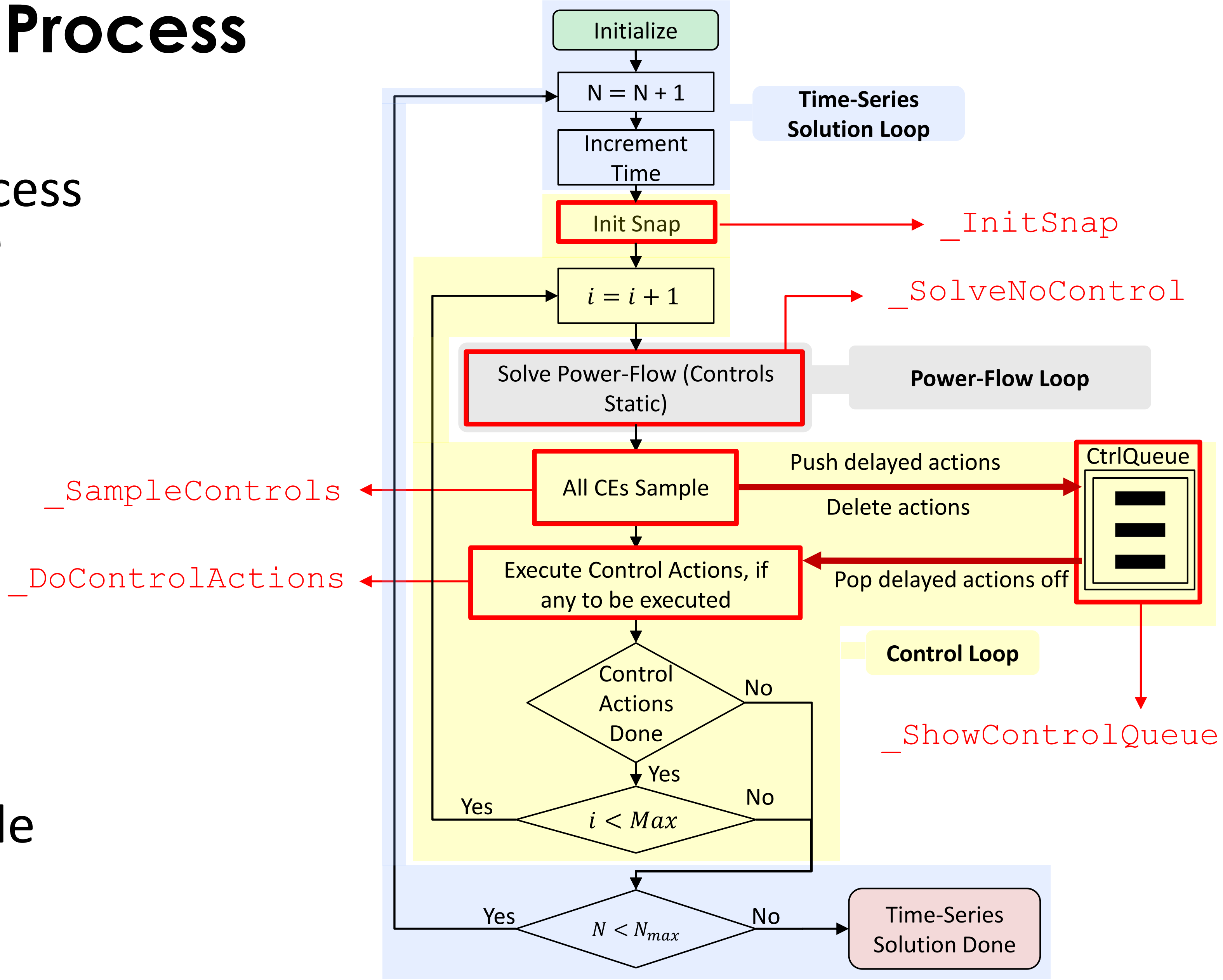
Customized Solution Process and Control Actions

- OpenDSS provides users access to the different steps of the solution process



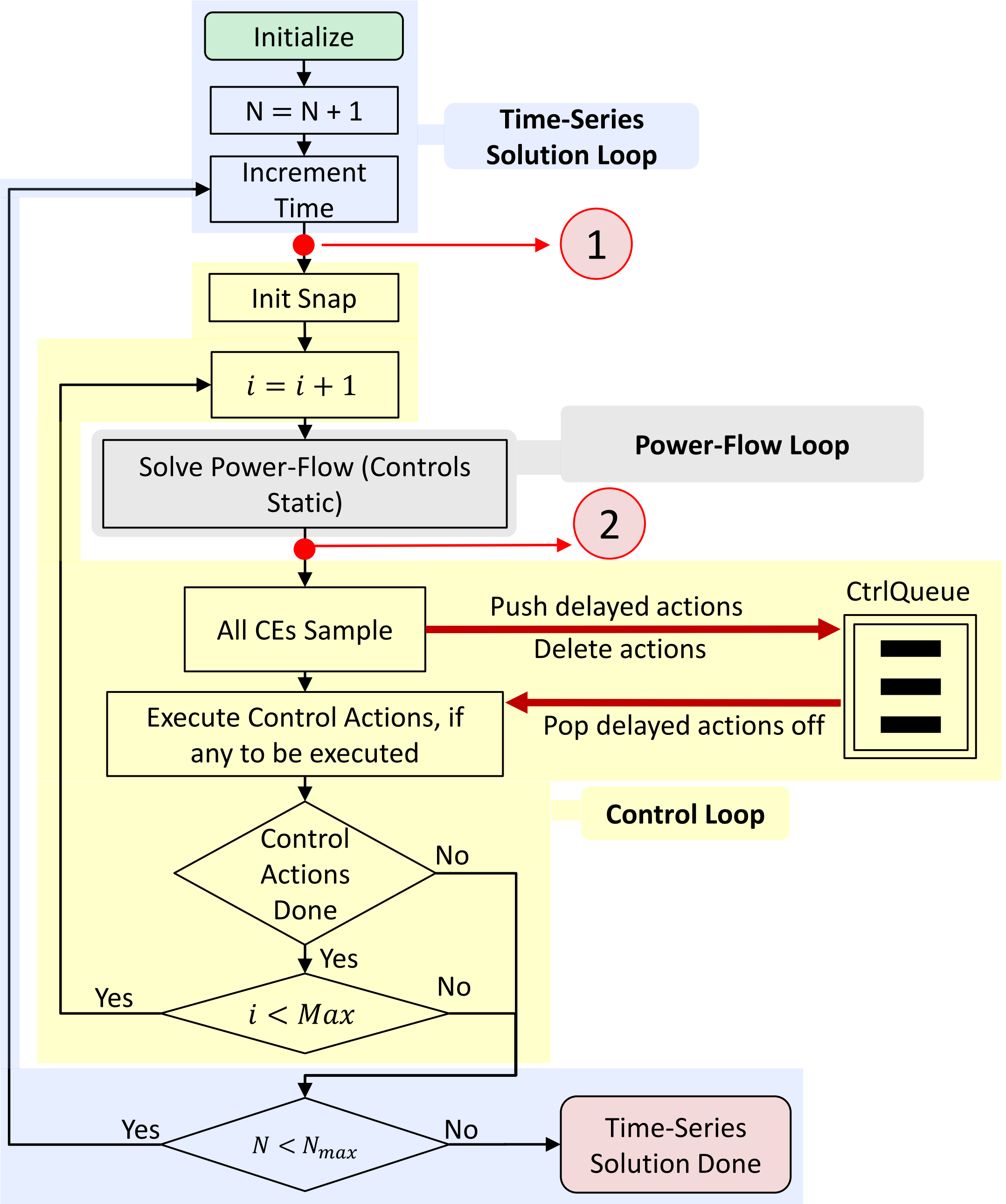
For step control of solution process: Pops control actions off the control queue according to the present control mode rules. Dispatches contol actions to proper control element "DoPendingAction" handlers.

- Commands are also available through COM and DDLL interfaces



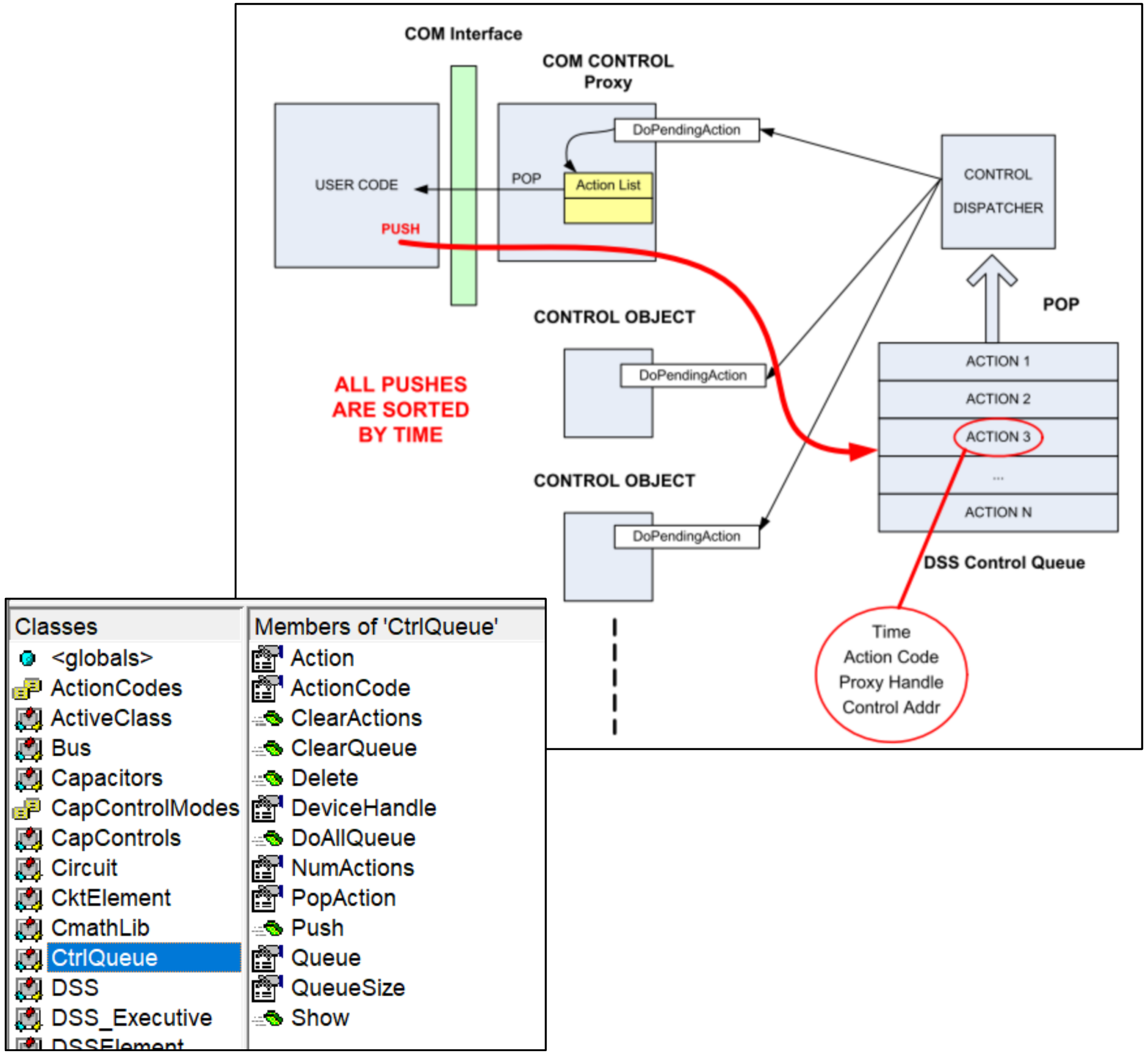
Customized Solution Process and Control Actions

- Customized control actions might be of two types:
  - **1**: DMS-like control: parameters for each control device are set
  - **2**: True customized delayed control actions, able to interact with other controllers and push/delete actions from Control Queue
    - E.g.: custom regulator logic, custom capacitor logic, …



Customized Solution Process and Control Actions

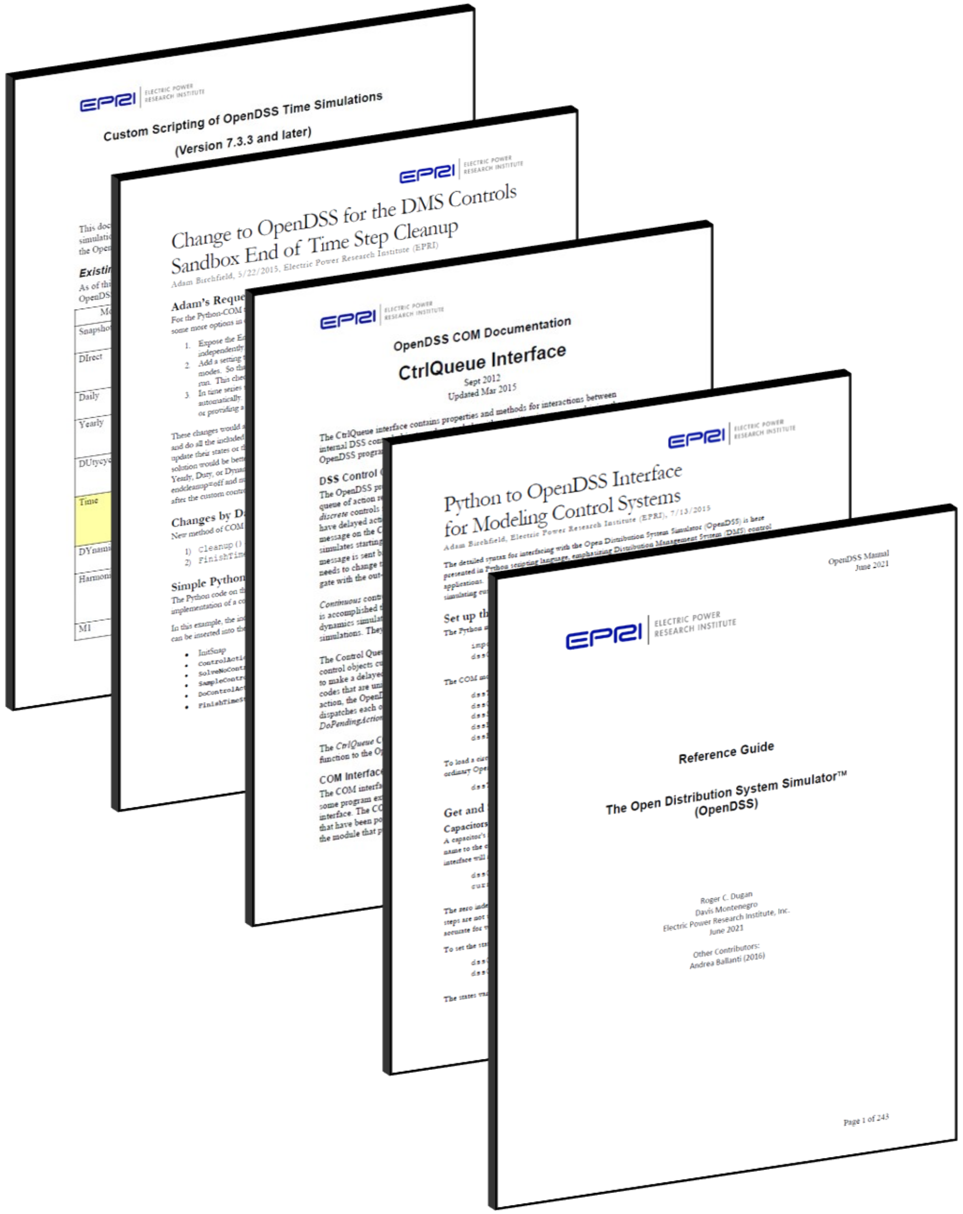
- The ControlQueue Interface
  - Provides a service to external controllers by managing pending control operations
  - Necessary to avoid that custom controls “jump the line” and executed before existing controls
  - User custom functions should emulate the OpenDSS “Sample”
  - Available at COM interface and Direct DLL version **only**



Classes	Members of 'CtrlQueue'
<globals>	Action
ActionCodes	ActionCode
ActiveClass	ClearActions
Bus	ClearQueue
Capacitors	Delete
CapControlModes	DeviceHandle
CapControls	DoAllQueue
Circuit	NumActions
CktElement	PopAction
CmathLib	Push
CtrlQueue	Queue
DSS	QueueSize
DSS_Executive	Show
DSSElement	

Where do I go from here?

- Technical notes available at your local OpenDSS installation Doc folder (*C:\Program Files\OpenDSS\Doc*)
  - Python to OpenDSS Interface for Modeling Control Systems
  - OpenDSS COM CtrlQueue Interface
  - OpenDSS Manual
  - Change to OpenDSS for the DMS Controls Sandbox End of Time Step Cleanup
  - OpenDSS Custom Scripting
- Official Forum at SourceForge
  - https://sourceforge.net/p/electricdss/discussion/



Custom Scripting of OpenDSS Time Simulations
(Version 7.3.3 and later)

Change to OpenDSS for the DMS Controls Sandbox End of Time Step Cleanup
Adam Birchfield, 3/22/2015, Electric Power Research Institute (EPRI)

OpenDSS COM Documentation
CtrlQueue Interface
Sept 2012
Updated Mar 2015

Python to OpenDSS Interface for Modeling Control Systems
Adam Birchfield, Electric Power Research Institute (EPRI), 7/13/2015

OpenDSS Manual
June 2021

ELECTRIC POWER RESEARCH INSTITUTE

Reference Guide

The Open Distribution System Simulator™
(OpenDSS)

Roger C. Dugan
Davis Montenegro
Electric Power Research Institute, Inc.
June 2021

Other Contributors:
Andrea Ballanti (2016)

Page 1 of 243

Examples

Questions?

Together…Shaping the Future of Energy™