Carnegie Mellon

Electrical & Computer ENGINEERING

Towards Feasible Power Delivery On AC Electric Networks Using Smart Wires

2015 Carnegie Mellon Electricity Conference: Preconference Workshop

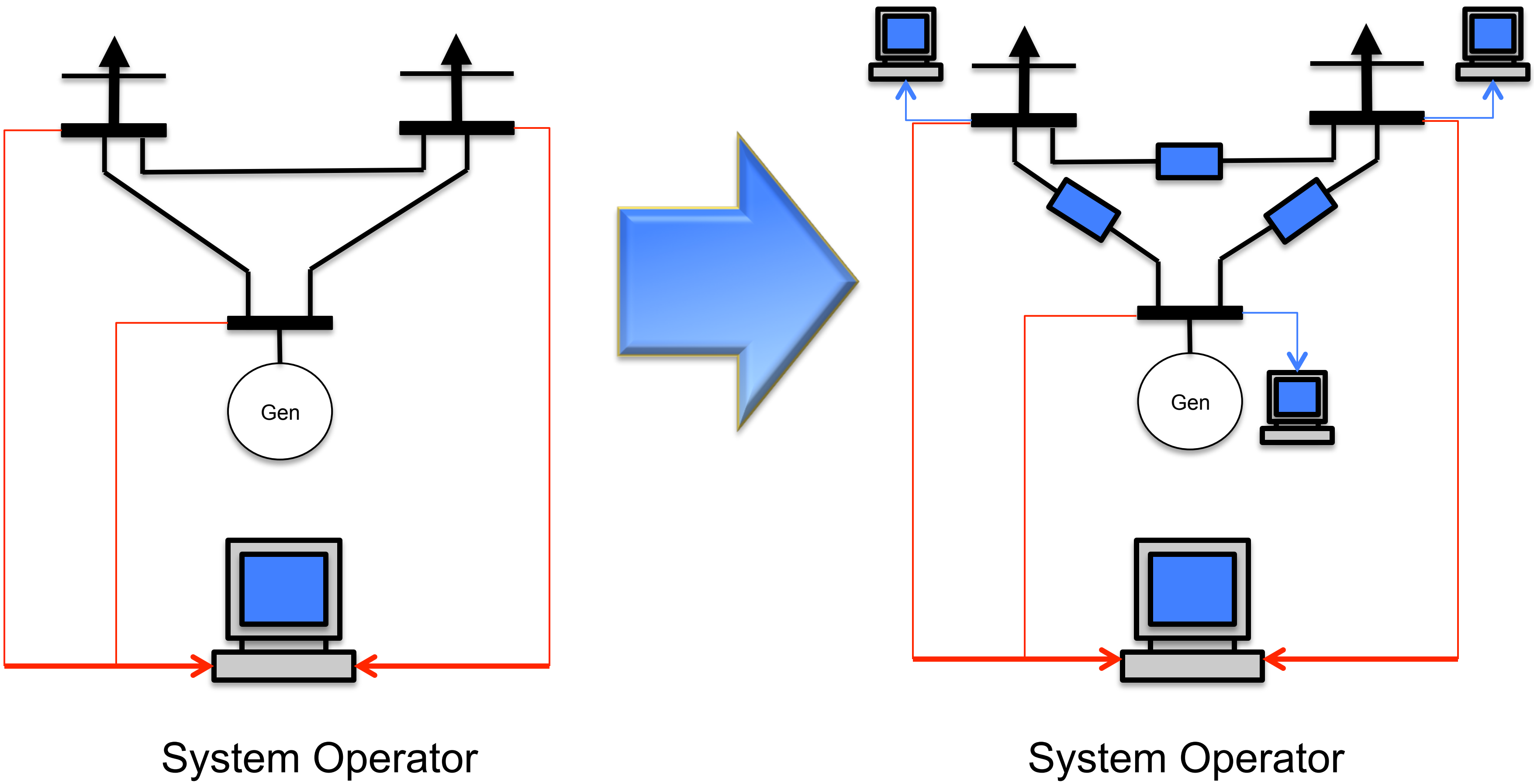
Andrew Hsu and Marija Ilić

Motivation

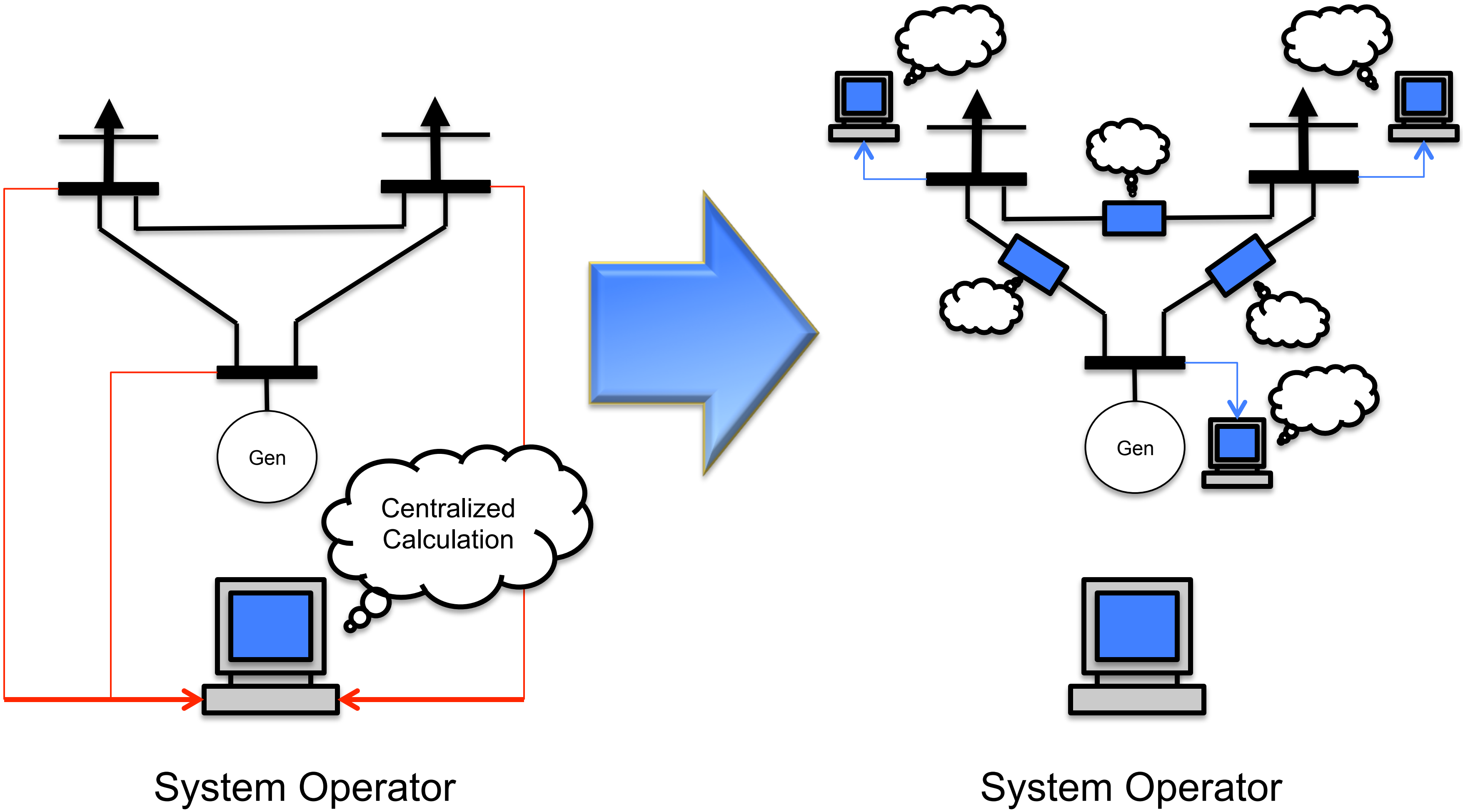
- **Many new wire-based devices that can measure, actuate, and calculate/store data**
  - Dynamic line rating units (DLRs)
  - Distributed series reactances (DSRs)
  - Flexible AC transmission system (FACTS) devices

- Can distributed, smart, wire-based devices cooperate to perform system calculations?

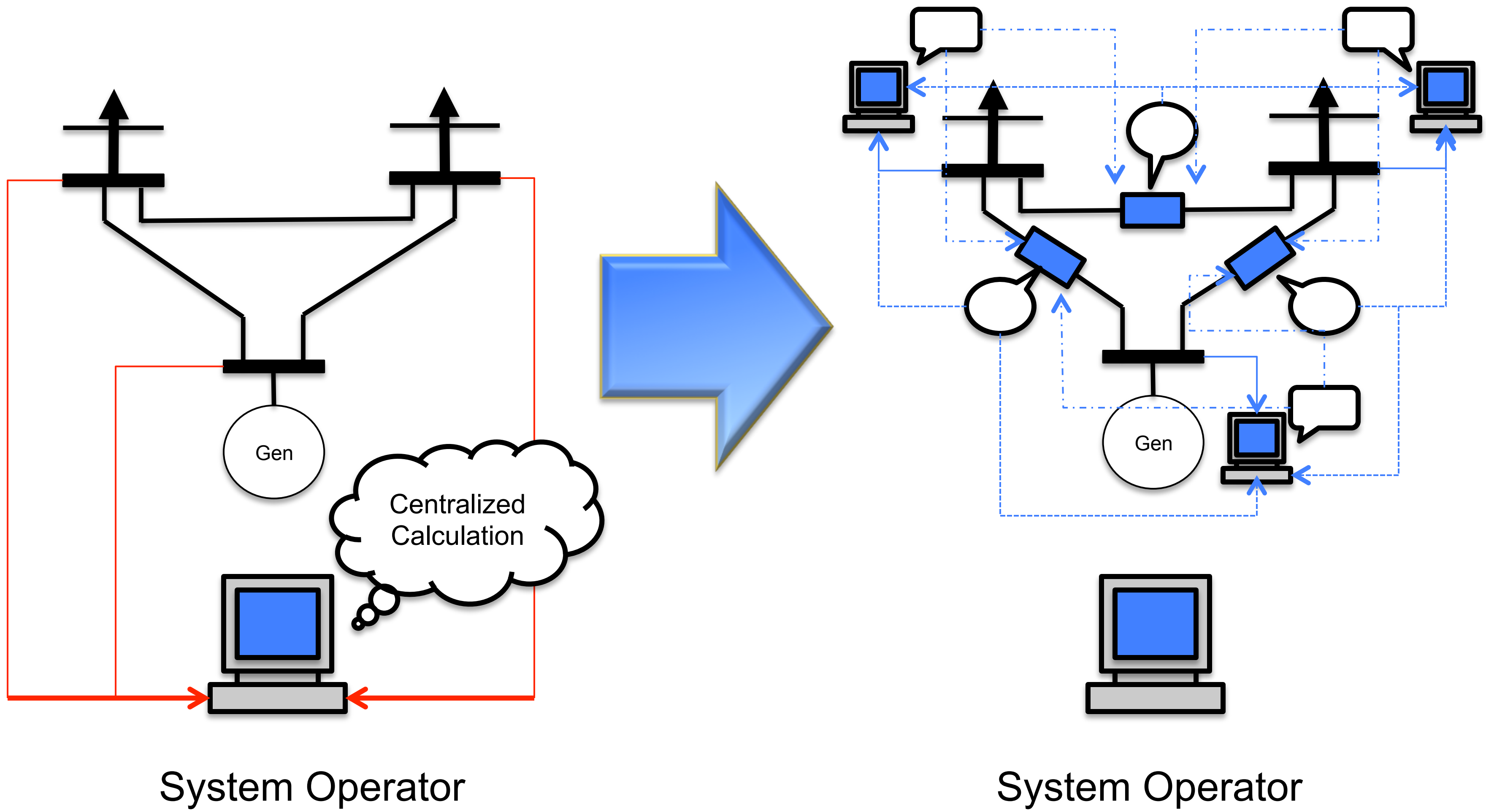
Grid Components Becoming Smarter



Distributed Power Flow



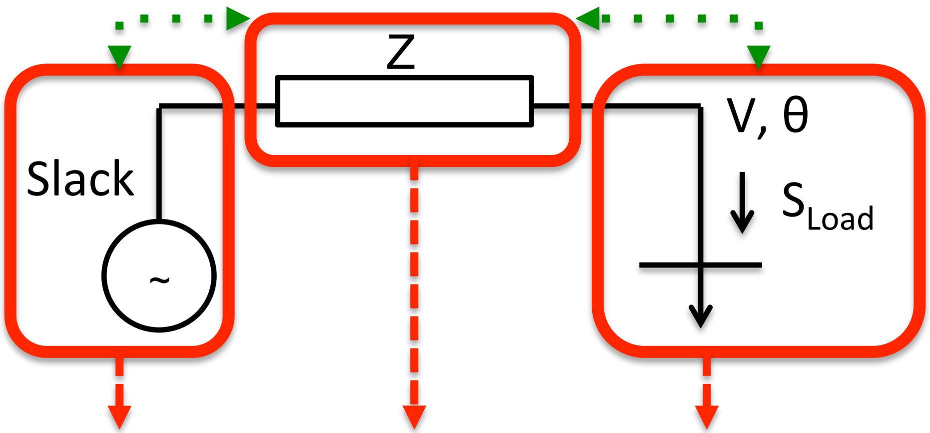
Distributed Power Flow



Potential Uses

- **Complement to centralized operator**
  - Calculate power flow independently[1,2,3]
  - Individual computations easy to carry out
- **Prevent Infeasible Power Transfer**
  - No voltage solution for given loads/network
  - Hard to calculate in centralized way (except 2 bus)[4]

DPF Algorithm 2 Bus Case



Component Name: Bus1 Line1 Bus2
Component Type: BusSlack Line BusPQ

Communications Setup Table:

From	Bus1	Line1	Bus2	Line1
To	Line1	Bus1	Line1	Bus2

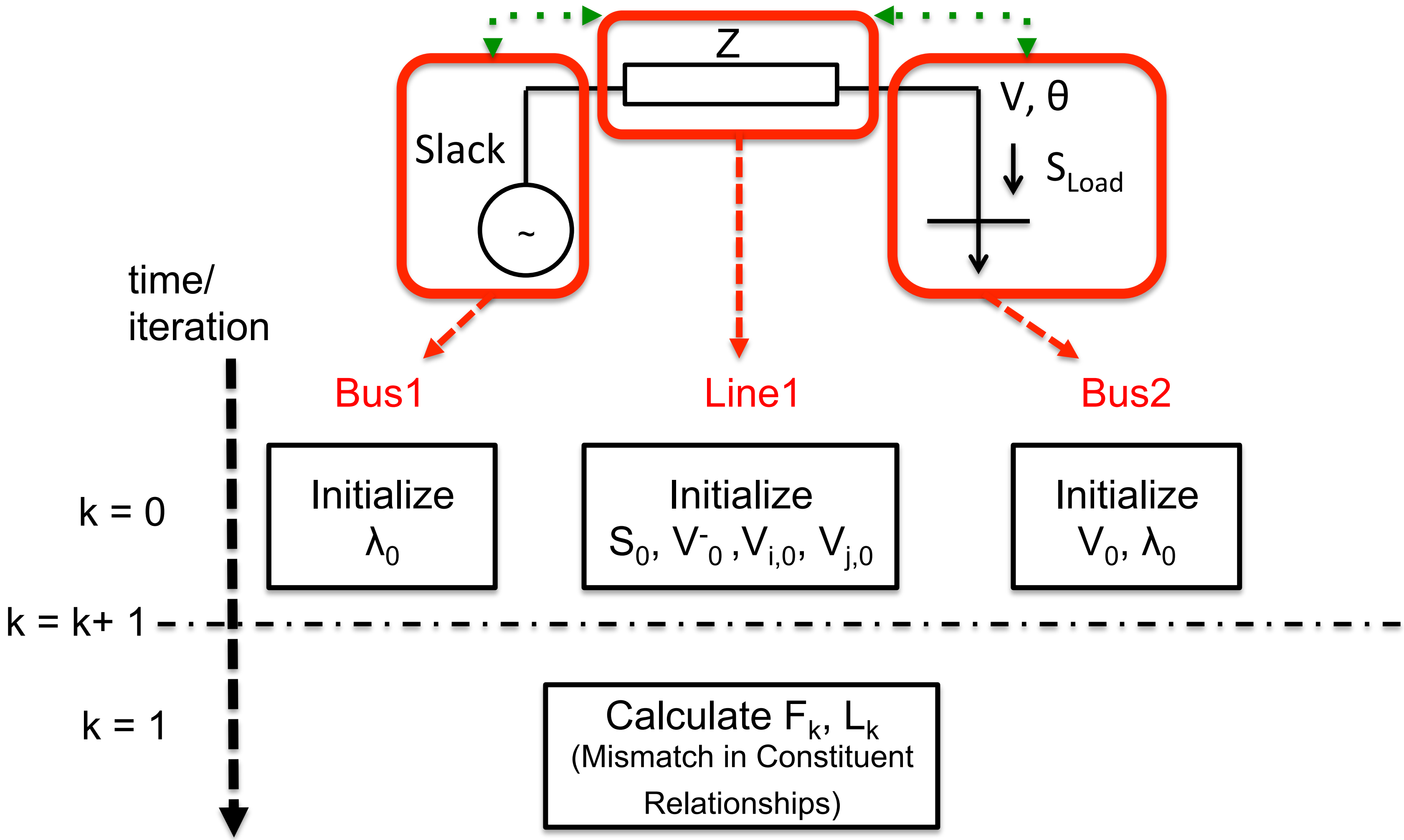
DPF Algorithm – Bus Modules

- **Bus Module**
- Properties
  - name, type, LinesIn, LinesOut, time, iter
  - Variables
    - Vbus, lambda, Dlambda
- Methods
  - Constructor:
    - read params from Matfile
  - Calculators:
    - calculate Dlambda, lambda
  - Comm. from lines (read()):
    - get mismatches, Sf, Sl, voltages, Vbus (tree status lines)
  - Comm. to lines (write()):
    - send Dlambda, lambda, Vbus
  - Run function (while loop)

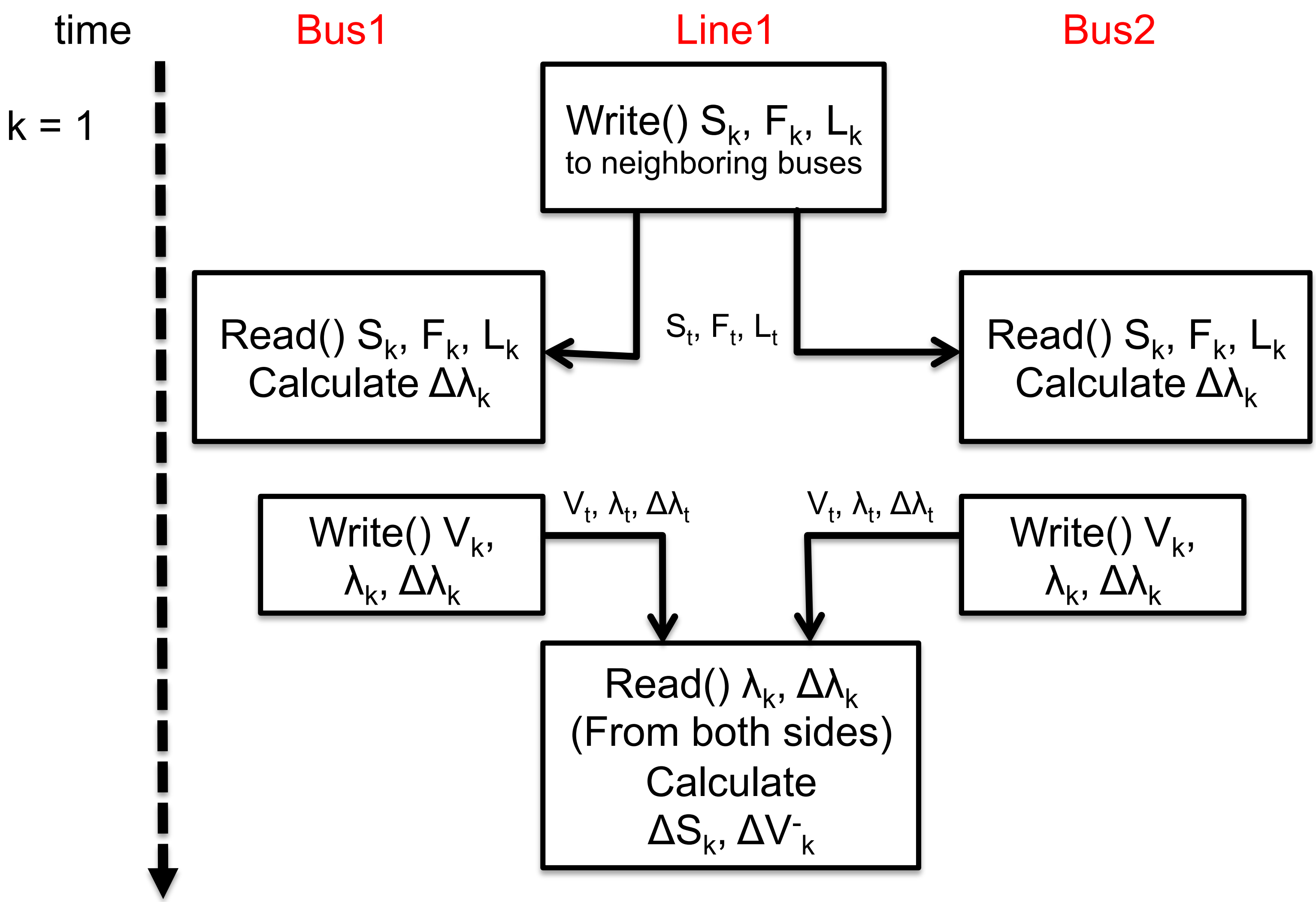
DPF Algorithm – Line Modules

- **Line Module**
- Properties
  - name, type, busSen, busRec, time, iter
  - Variables
    - Z, Sf, Sl, Vacross, VSend, VRec, Fmis, Lmis
- Methods
  - Constructor:
    - read params from Matfile
  - Calculators:
    - calculate mismatches,
    - update S, V
  - Comm. from lines (read()):
    - get lambda variables, voltages
  - Comm. to lines (write()):
    - send Fmis, Lmis, Sf, Sl, VRec
  - Run function (while loop)

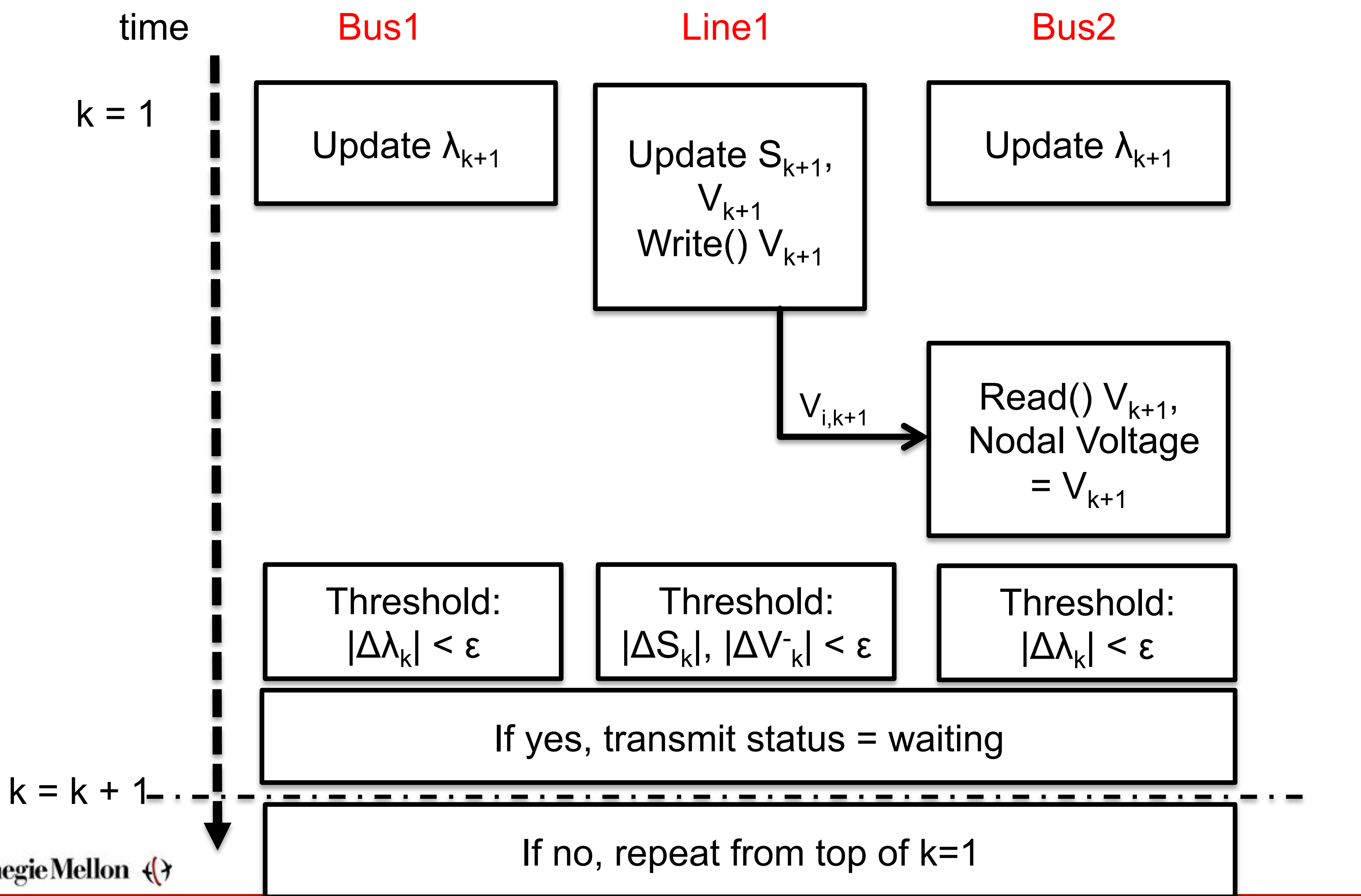
DPF Algorithm 2 Bus Case – Evolution Over Time 1



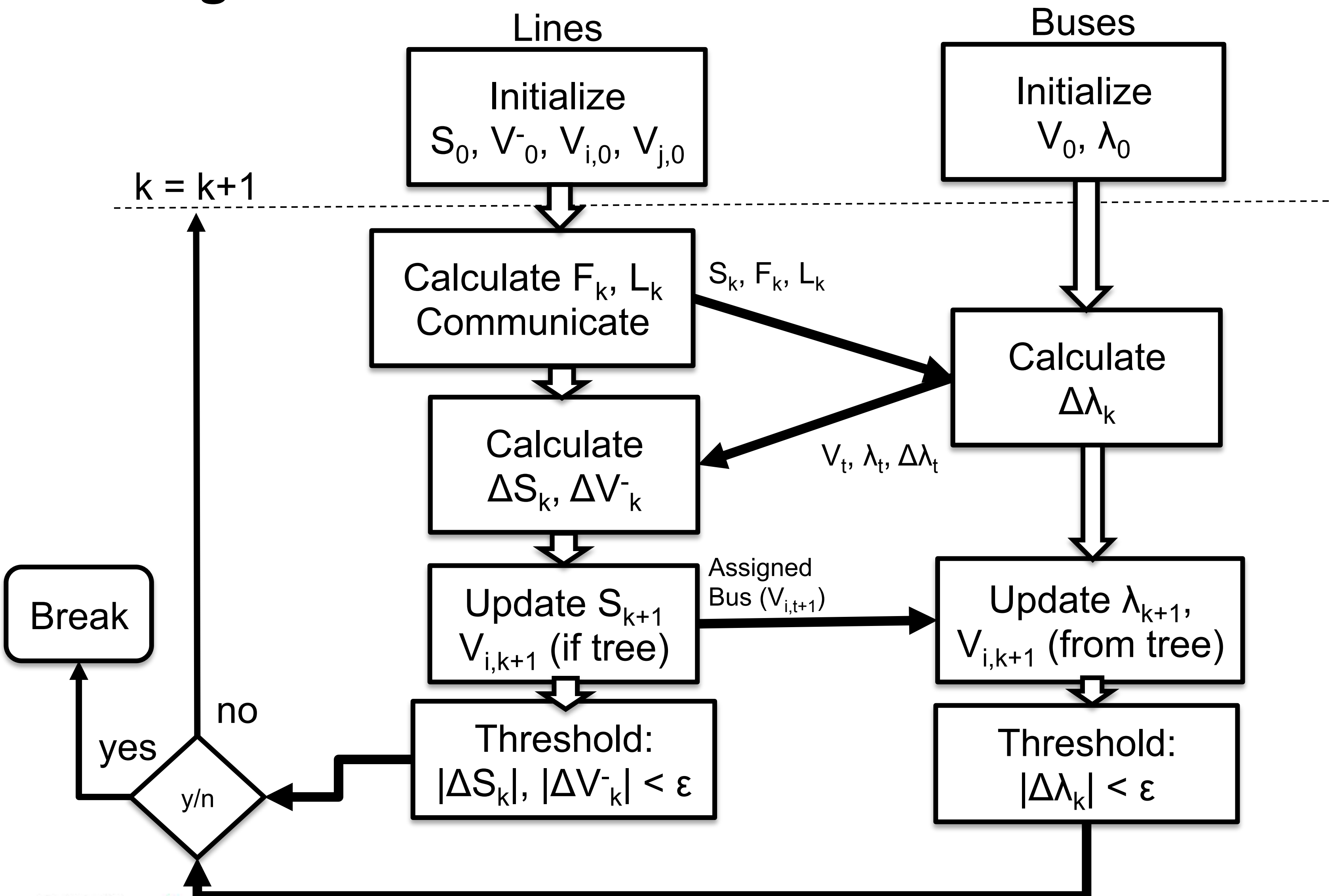
DPF Algorithm 2 Bus Case – Evolution Over Time 2



DPF Algorithm 2 Bus Case – Evolution Over Time 3

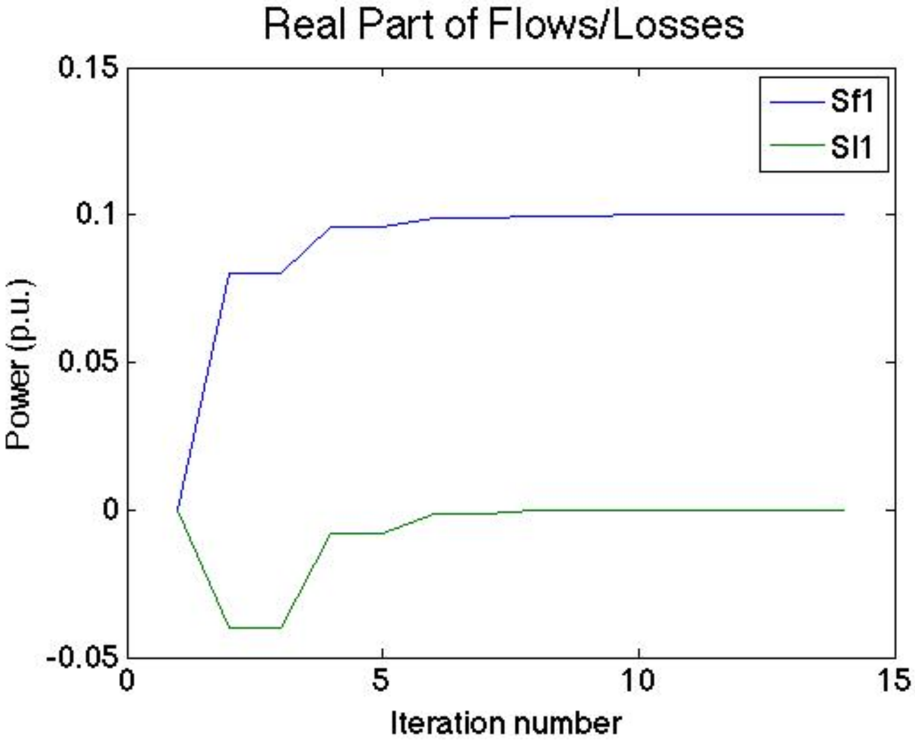


DPF Algorithm Flowchart

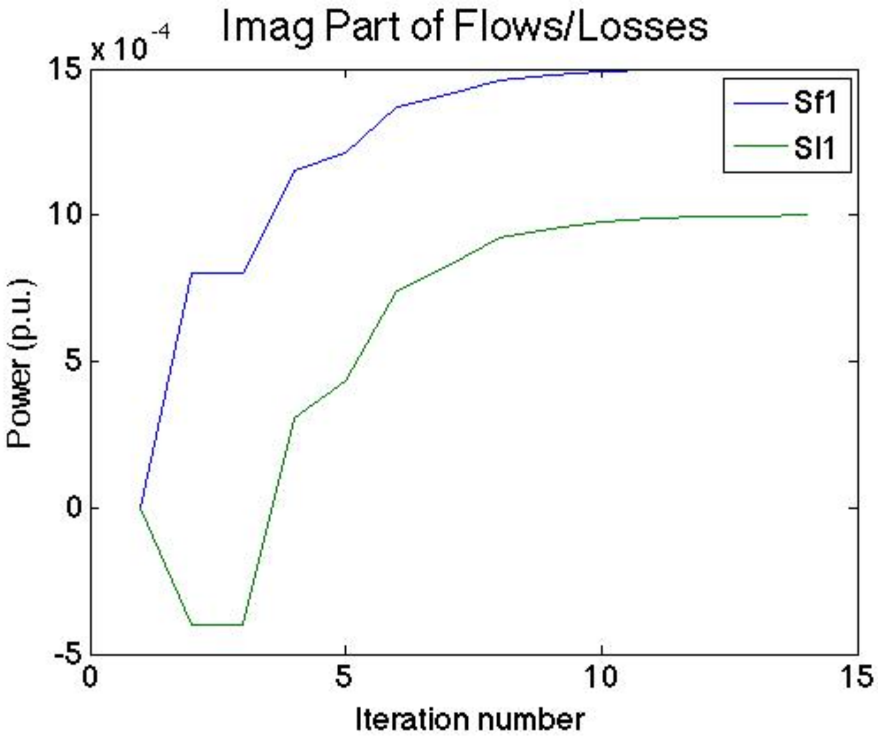


Expected Output

Real{Sf,Sl} per iteration



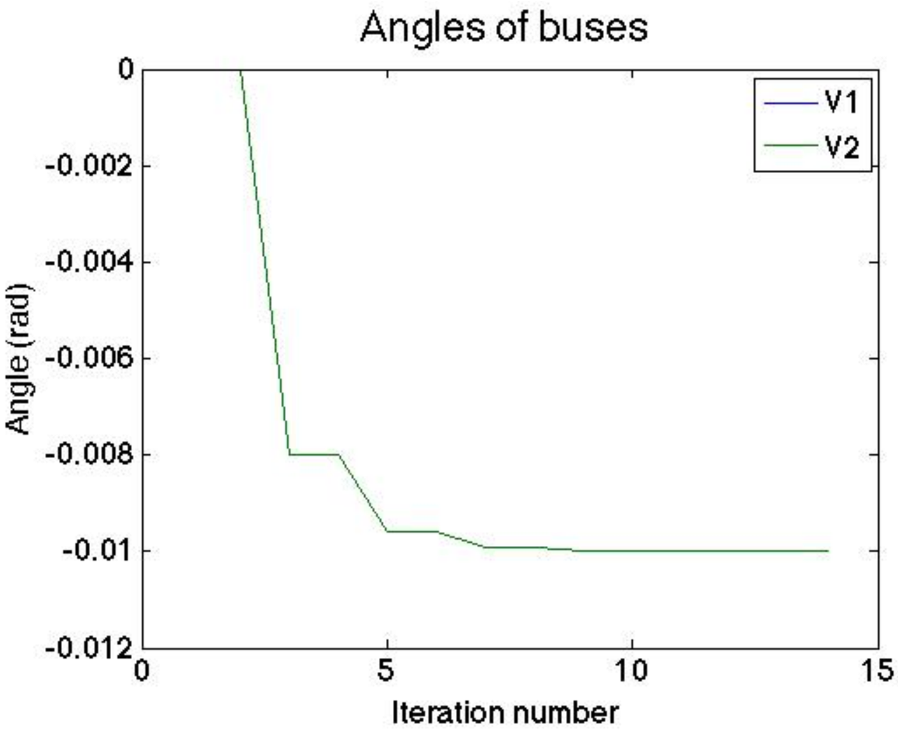
Imag{Sf,Sl} per iteration



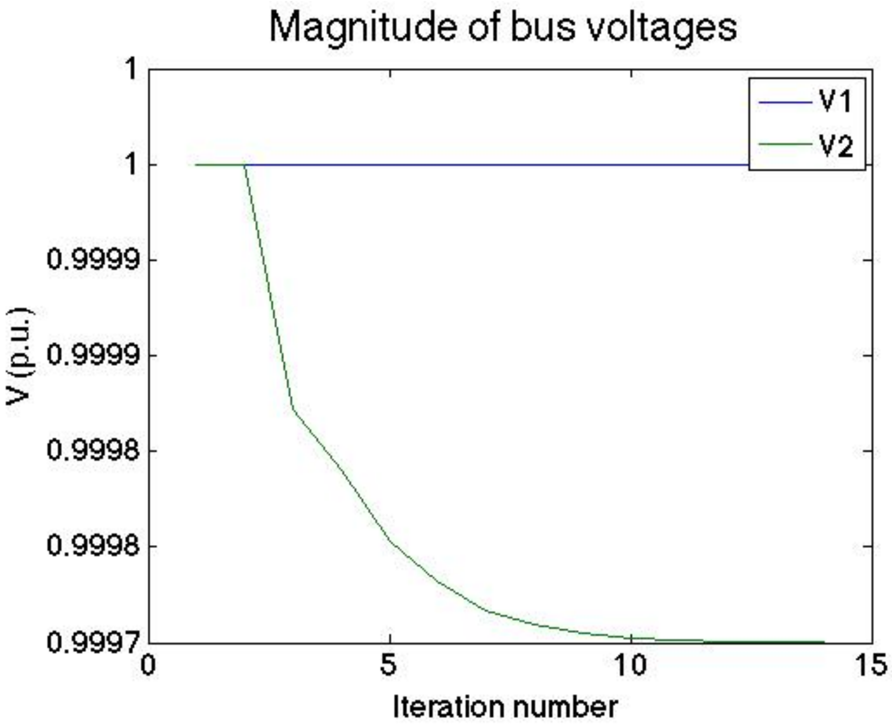
Given:
Z = 0.001 + 0.1j
Sload = 0.1 + 0.0001j

Expected Output of Voltage at Load Bus

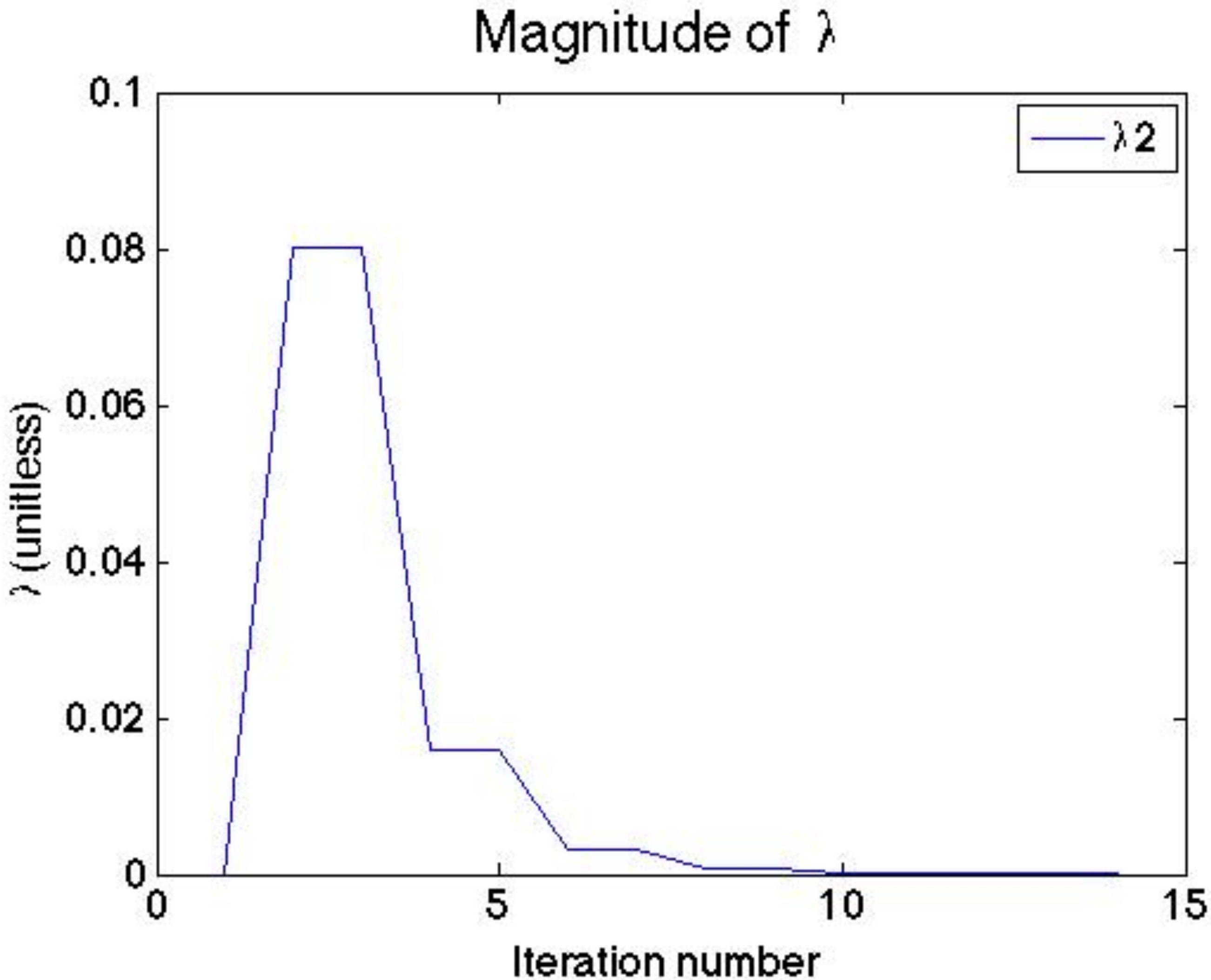
angle{V} per iteration



abs{V} per iteration



Expected Output of Lagrange Multiplier at Load Bus



Discussion and Conclusion

- Developed distributed algorithm that runs on lines and buses of an electric power network

- Show that distributed, localized logic can be developed and implemented in the NIST simulations platform

- Show how to conceptualize the algorithm to aid its deployment in the simulations platform

References

[1]Hsu, A. and Ilic, M. *Distributed newton method for computing real decoupled power flow in lossy electric energy networks*,North American Power Symposium (NAPS), 2012 , vol., no., pp. 1-7, 9-11 Sept. 2012.

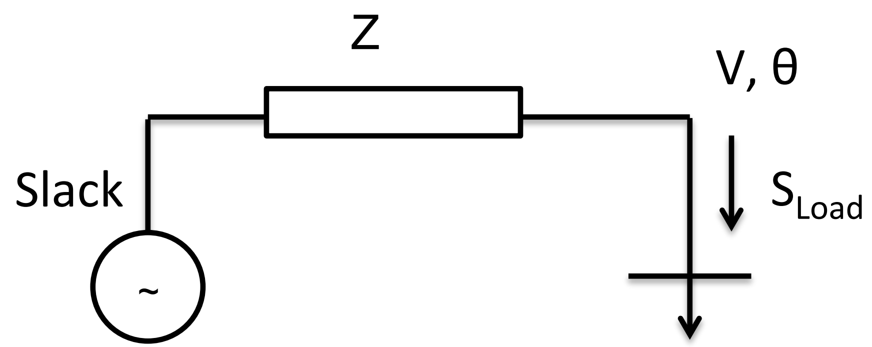
[2]Hsu, A. and Ilic, M. *Toward Distributed Contingency Screening Using Line Flow Calculators and Dynamic Line Rating Units (DLRs)* ,Hawaiian International Conference of System Sciences (HICSS). Maui, Hawaii, January 2012.

[3]Ilic, M. and Hsu, A. *General Method For Distributed Line Flow Computing With Local Communications In Meshed Electric Networks*, Pub. No. US 2013/0024168 A1, January 24, 2013 (status: allowed)

[4]Hsu, A., Ilic, M. *Ensuring Feasible Power Delivery Using An Optimization-based Power Flow Model,* Techcon 2013, Austin, Tx. Sept 2013

Infeasible Power Delivery

- Depending on the line parameters, a voltage solution may or may not exist (voltage instability)
- It can be solved in closed form for the 2 bus system, but not in multi-bus systems



2 bus system with generator and load

$$real\{V\} = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - 4(real\{Z * S_{Load}\} + imag\{Z * S_{Load}\}^2)}$$

$$imag\{V\} = imag\{Z * S_{Load}\}$$

Equations Slide 1

$$F = S_f - \frac{1}{2} Y^* (V^-)^* (2V_{rec} + V^-)$$

$$L = S_l - Y^* (V^-)^* (V^-)$$

$$\Delta\lambda = \frac{4}{5} N^{-1} \left(\sum_{neighbors} (dS_f + \frac{1}{2} S_l - dF + \frac{1}{2} L) \right) - b_{inj}$$

where N = # of connected lines to bus,

d = -1 if line orients towards bus, 1 if line orients away

Equations Slide 2

$$\Delta S_f = F + \lambda_{sen} - \lambda_{rec} + \Delta\lambda_{sen} - \Delta\lambda_{rec}$$

$$\Delta S_l = L + \frac{1}{2}(\lambda_{sen} + \lambda_{rec} + \Delta\lambda_{sen} + \Delta\lambda_{rec})$$

$$\Delta V^- = (H_{v2,1} - H_{v2,2} H_{v1,2}^{-1} H_{v1,1})^{-1} (V^- * - H_{v2,2} H_{v1,2}^{-1} V^-)$$

Equations Slide 3

$$H_{v1,1} = 2\frac{\partial F}{\partial V^-}\frac{\partial F^*}{\partial V^-} + 2\frac{\partial L}{\partial V^-}\frac{\partial L^*}{\partial V^-}$$

$$H_{v2,1} = \frac{\partial^2 F}{(\partial V^-)^2}F^* + \frac{\partial F^*}{\partial V^-}(\frac{\partial F^*}{\partial V^-})^* + \frac{\partial^2 F^*}{(\partial V^-)^2}F + \frac{\partial F}{\partial V^-}(\frac{\partial F}{\partial V^-})^*$$

$$+\frac{\partial^2 L}{(\partial V^-)^2}L^* + \frac{\partial L^*}{\partial V^-}(\frac{\partial L^*}{\partial V^-})^* + \frac{\partial^2 L^*}{(\partial V^-)^2}L + \frac{\partial L}{\partial V^-}(\frac{\partial L}{\partial V^-})^*$$

$$H_{v1,2} = H_{v2,1}^*$$

$$H_{v2,2} = H_{v1,1}^*$$