Efficient Dynamic Simulation in *PowerFactory*

Davide Fabozzi

Topics

- Principles of dynamic simulation in PowerFactory
- Performance results on large systems
- Speeding up simulations: a practical example

Principles of dynamic simulation in PowerFactory

Principles of dynamic simulation

- Algebraic and differential equations
- Algebraisation and integration step size
- Solution, convergence and accuracy
- Continuous-time and discrete-time equations
- Efficient user-defined models

Algebraic and differential equations

- Algebraic equations: $0 = g(x, y)$
  - Network currents/voltages (y) → infinitely fast

- Differential equations: $\frac{d}{dt} x = f(x, y)$
  - Machine/controller dynamics (x) → “slow” (w. r. t. algebraic equations)

- Differential equations must be “algebraised” in order to be solved:
  - With an integration formula (e.g. Trapezoidal Method, Backward Euler)

Algebraisation and integration step size

- Differential equation in the form:

$$\frac{d}{dt}x_t = f(x_t)$$

- Discretized in **integration step size** of length h and algebraised:

$$x_{t+h} = x_t + h\frac{\delta}{2}f(x_t) + h\frac{2-\delta}{2}f(x_{t+h})$$

- “**Damping factor**” δ may assume different values:

$x_{t+h} = x_t + \frac{h}{2}f(x_t) + \frac{h}{2}f(x_{t+h}) \rightarrow$ Trapezoidal Method ($\delta = 1$)

$x_{t+h} = x_t + hf(x_{t+h}) \rightarrow$ Backward Euler ($\delta = 0$)

Fixed step and automatic step size adaptation

- Step size tradeoff. The larger h:
  - Less steps to reach simulation horizon
  - At given step, more “distance” between predicted and correct solution

- Power systems are stiff systems: adaptive step for long term
  - Error bounded between **maximum** and **minimum prediction error***
  - Above maximum: h decreased immediately
  - Below minimum: h increased after **delay for step size increase**
  - Increase/decrease according to **speed factors** (usually ≤ 2)

* difference between prediction and correction of differential states

Solution, convergence and accuracy

- Algebraic and differential equations may be solved:
  - One after the other (partitioned method)
  - Together (simultaneous method) → **A-stable for all models**
  - Algebraic and **A-stable** models together, then **non-A-stable** models

- A-stable tradeoff. Setting models to A-stable:
  - Enlarges the size of the network Jacobian matrix
  - Avoids interface errors between the A-stable model and the network
  - Makes convergence better

- Step is declared “converged” as soon as desired accuracy is met

Convergence criteria and accuracy thresholds

- Convergence criteria may be checked:
  - On one model after the other (independent convergence)
  - Together (simultaneous convergence)

- Convergence criteria tradeoff. Checking all models simultaneously:
  - Reduces interface errors between models and network
  - Reduces interface errors between the models
  - May require some interface iterations between models and network (several complete solutions at a given step)

- Heuristics for **fast convergence check** can speedup the simulation

Solution and convergence tradeoffs

- Tradeoff of solution methods and convergence criteria

Accuracy: no network-dynamics causality, interface errors within tolerance

Speed: smaller matrices

	Simultaneous convergence (satisfy all equations at once)	Independent convergence (first network then dynamics)
Partitioned method (network and dynamic equations separately)	**Default options**	**Fast independent solution**
Simultaneous method (network and dynamic equations together)	**A-stable for all models**	

Accuracy: no interface errors

Speed: solve network then dynamics → no need for interface iterations

* **Speed** considerations refer to **fixed-step-size** *PowerFactory*

- Using the last available solution (instead of re-evaluating all models) for **fast computation of outputs** can speedup the simulation

Solution of linear systems

- Regardless of the specific algorithm used, getting the simulated trajectory ultimately needs solving many linear systems

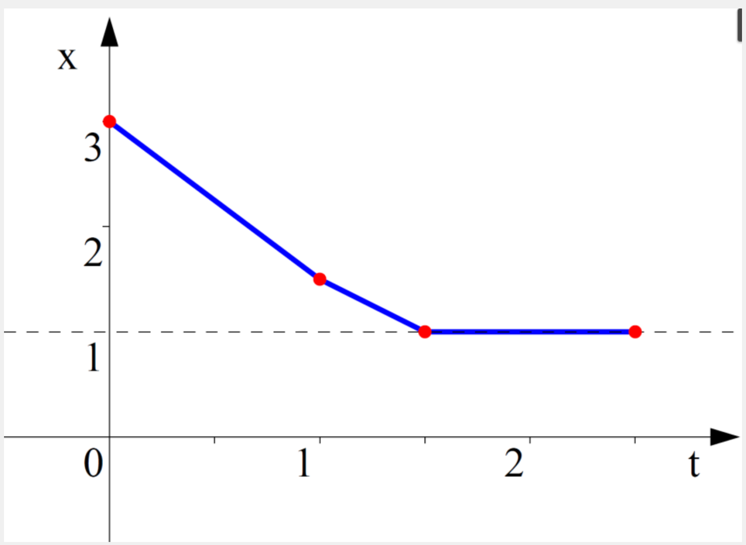
- State-of-the-art linear system solvers:
  - LU solver (**direct method**)
  - GMRES solver (**iterative method**)

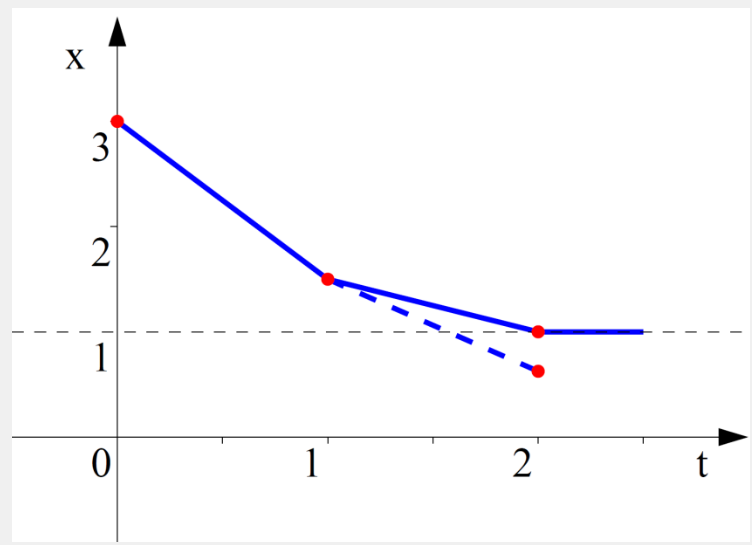
- **TIP**: the corresponding settings in PowerFactory are located in the **study case options**!

Continuous-time and discrete-time equations

- Discrete-time equations: $z_{t+} = H(x, y, z_{t-})$
  - responses of discrete controls (set-points, switches, modes…)
  - changes in continuous-time equations (**select**, **lim**, **limstate**…)

- Events may force simulation step to interrupt or to repeat itself





Discrete-time solution tradeoffs

- In case of repetition, **maximum number of event loops** may be reached without solution. Interruptions are expensive but avoid this

- The events occurring within a step may be applied simultaneously (**Apply DSL events directly**) or sequentially (starting with the first)

- “**Direct**” application is generally faster, with some exceptions, e.g.:
  - Model/system have events whose simultaneous execution is unfeasible
  - Model/system have no feasible transitions (for given conditions / step)

Efficient user-defined models

- User developed models should be as small and simple as possible to fit the purpose of the simulation. Standard models are optimised

- Modelling enhancements
  - Use one-shot DSL (**selfix**, **limfix**, **outfix**) for time-invariant expressions

- Solution enhancements
  - Use automatic DSL to C Interface converter to compile your models (requires C compiler, Microsoft Visual Studio 2012 recommended)

Enhanced DSL debugging possibilities

- The user can observe information on **DSL events and warnings**

```
--------------------------------------------------------------
'Grid\PicDrop.ElmDsl':
  control schedules internal event (min) at time 2,000000 to set transition of 'picdro(yi,1.,1.)' from 'WAIT-PICK' to 'ON'.
--------------------------------------------------------------
'Grid\PicDrop.ElmDsl':
Inconsistent argument in DSL function acos(x): argument x is greater than 1; x=1 will be assumed.
Argument x is the result of DSL function 'limstate(...)'.
```

- Primary issue for these options is model checking, not performance

- Consistent cases are generally faster than inconsistent cases (e.g. inconsistencies may bring large error thus small adaptive step size)

- Event information available for DSL and compiled C Interface models

Performance results on large systems

Results on System 1

- 1200 synchronous generators
- 2600 customised DSL models
- 24000 terminals (0.4÷700 kV)
- 30000 differential equations

- Generator tripping at 0 s, run up to 20 s, stable
- The step size is fixed and equal to 20 ms

PowerFactory version	Improved solver	One-shot DSL	Fast check	Fast output	Comp. models	Fast sol.	t_{CPU} (s)	Speed-up (wrt Base)
15.1.8							104	Base
15.2.4	✓						79	1.3
15.2.4	✓	✓					71	1.45
15.2.4	✓	✓	✓				63	1.65
15.2.4	✓	✓	✓	✓			37	2.8
15.2.4 (default)	✓	✓	✓	✓	✓		20	5.20
15.2.4	✓	✓	✓	✓	✓	✓	19	5.45

Results on System 2

- 200 synchronous generators
- 500 customised DSL models
- 1700 terminals (1÷500 kV)
- 4800 differential equations
- Short circuit at 1 s, cleared at 1.1 s, run up to 10 s, stable
- The step size is variable (2÷60 ms)

PowerFactory version	Improved solver	Fast check and output	Fast sol.	One-shot DSL	Comp. models	t_{CPU} (s)	Speed-up (wrt Base)
15.1.7						22	Base
15.2.1	✓					18	1.2
15.2.1 (default)	✓	✓				17	1.3
15.2.1	✓	✓	✓			11	2
15.2.1	✓	✓	✓	✓		10	2.2
15.2.1	✓	✓	✓	✓	✓	9	2.45

Results on System 3

- 160 synchronous generators
- 380 customised DSL models
- 570 terminals (0.4÷220 kV)
- 1600 differential equations
- Short circuit at 1 s, cleared at 1.12 s, run up to 15 s, stable
- The step size is fixed and equal to 10 ms

PowerFactory version	Improved solver	Fast output	Fast check	Fast sol.	One-shot DSL, comp. models	t_{CPU} (s)	Speed-up (wrt Base)
15.1.7						14	Base
15.2.1	✓					13	1.1
15.2.1	✓	✓				12	1.15
15.2.1 (default)	✓	✓	✓			9	1.55
15.2.1	✓	✓	✓	✓		4	3.5
15.2.1	✓	✓	✓	✓	✓	3	4.65

Live Demo on North Island System

- 150 synchronous generators
- 130 simple standard models
- 500 terminals (0.4÷220 kV)
- 1000 differential equations
- Short circuit at bus at 0 s, run up to 20 s, stable
- The step size is variable (10÷100 ms)
- Short circuit at line at 0 s, run up to 20 s, stable
- The step size is variable (10÷100 ms)
- Generator tripping at 0 s, run up to 500 s, stable
- The step size is variable (0.005÷5 s)

Speeding up simulations: a practical example

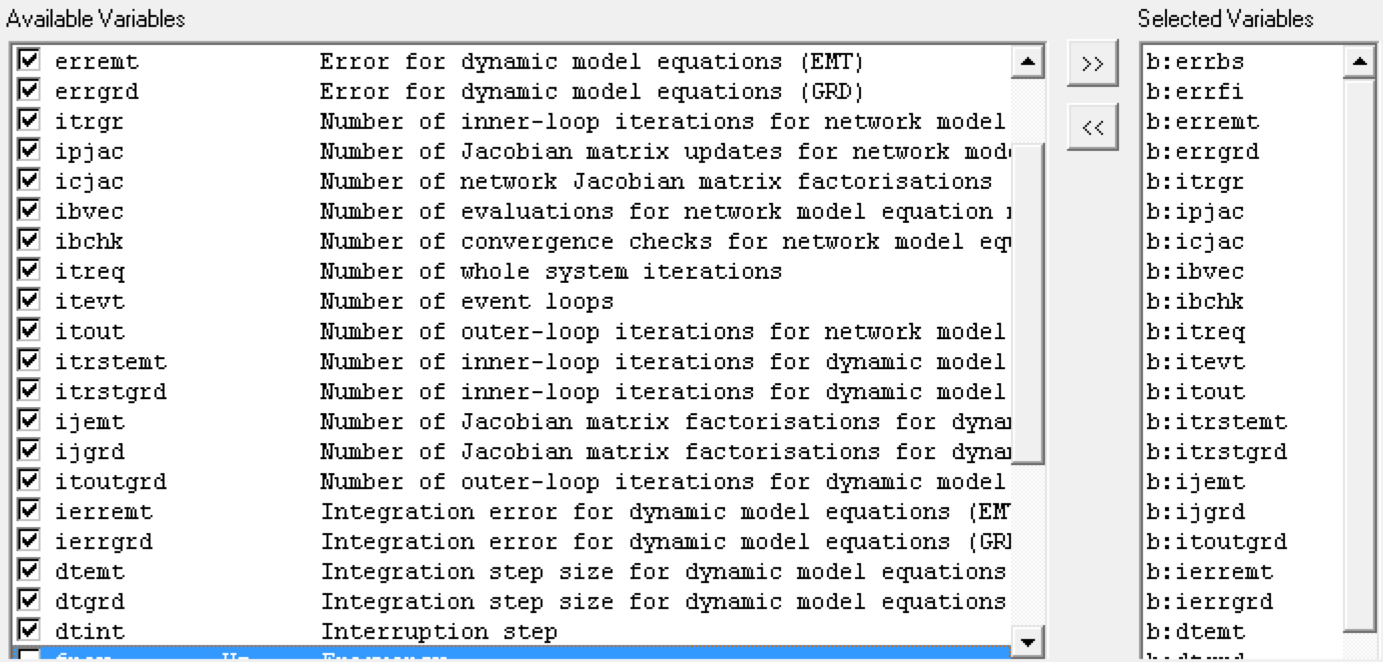
Once upon a time, System 2 was not as fast as it could be…

- 200 synchronous generators
- 500 customised DSL models
- 1700 terminals (1÷500 kV)
- 4800 differential equations
- Short circuit at 1 s, cleared at 1.1 s, run up to 10 s, stable
- The step size is variable (2÷60 ms)

PowerFactory version	Improved solver	Consistent model	Improved step strategy	All options Section 3.2	t_{CPU} (s)	Speed-up (wrt Base)
15.1.7					63	Base
15.2.1	✓				46	1.35
15.2.1	✓	✓			22	2.85
15.1.7		✓	✓		22	2.85
15.2.1	✓	✓	✓		18	3.5
15.2.1	✓	✓	✓	✓	9	7

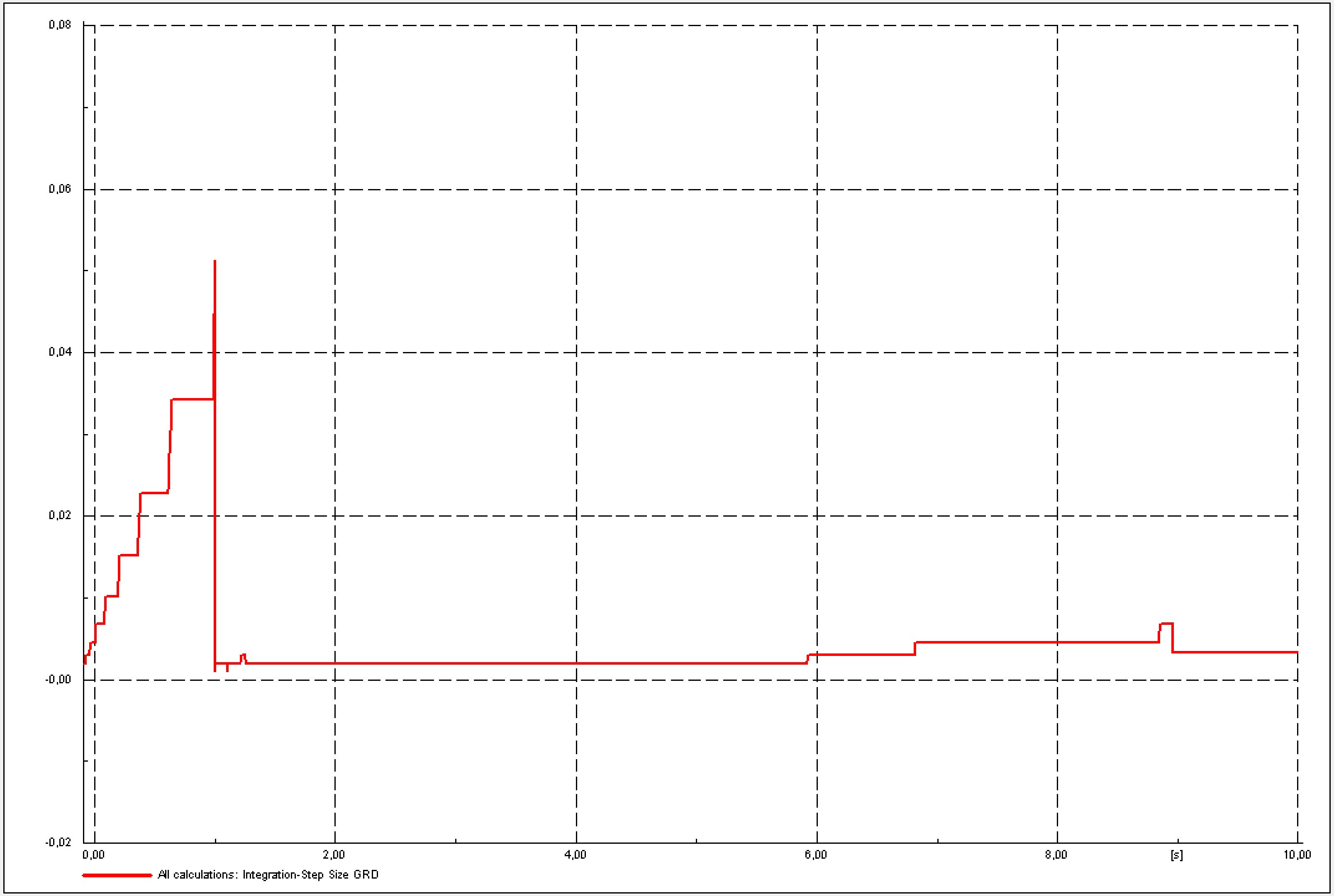
Enhanced simulation debugging information

- The user can observe useful information on the simulation regarding errors, step size, iteration and factorisation counters, …

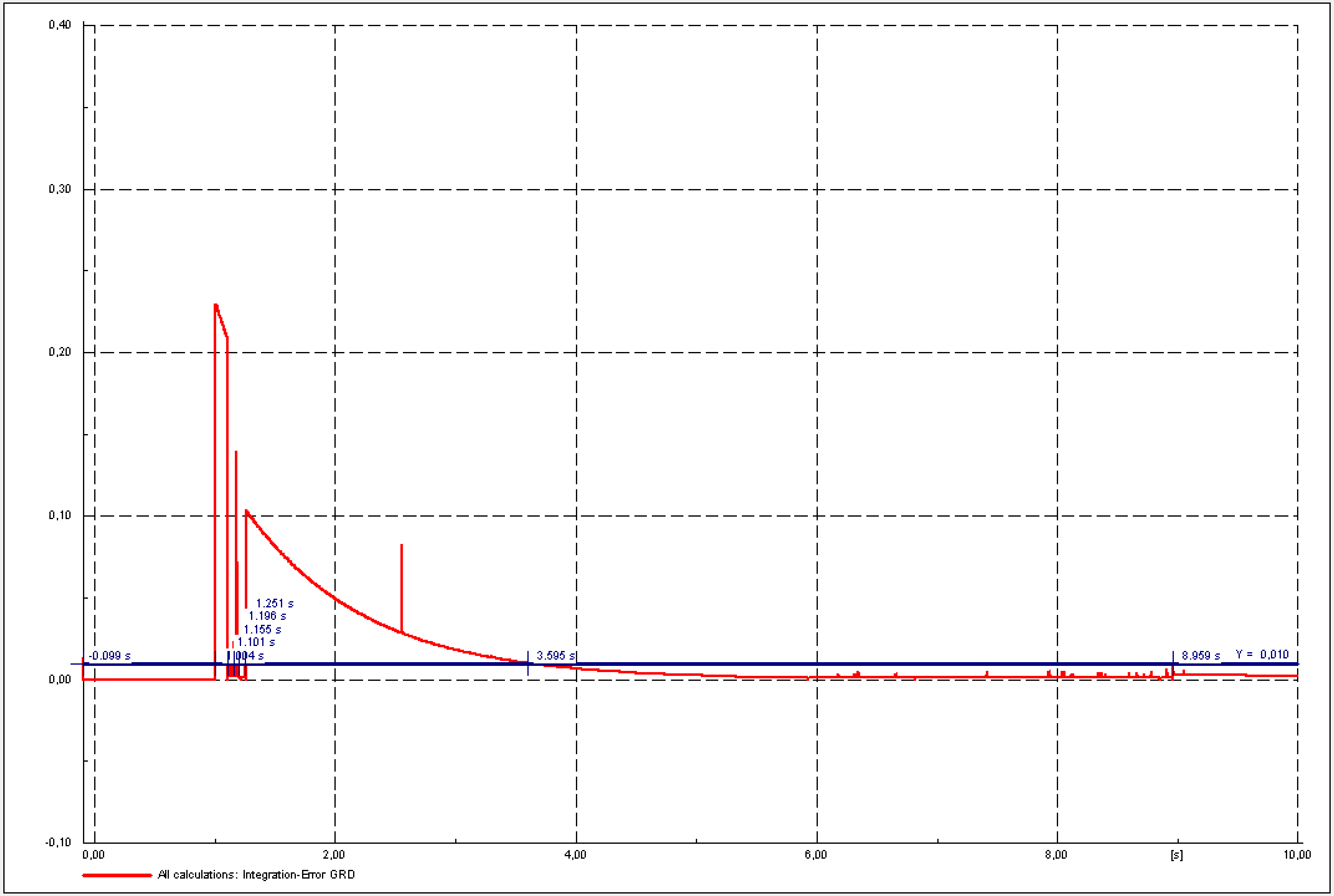


- **TIP**: the corresponding variables in PowerFactory are located in the **result file**!

In the initial PF15.1 simulation the step size was small…



… and the prediction error, used for step adaption, was big…



Imported to PF15.2, System 2 initializes with warnings…

- Several divisions by zero are detected, e.g.:

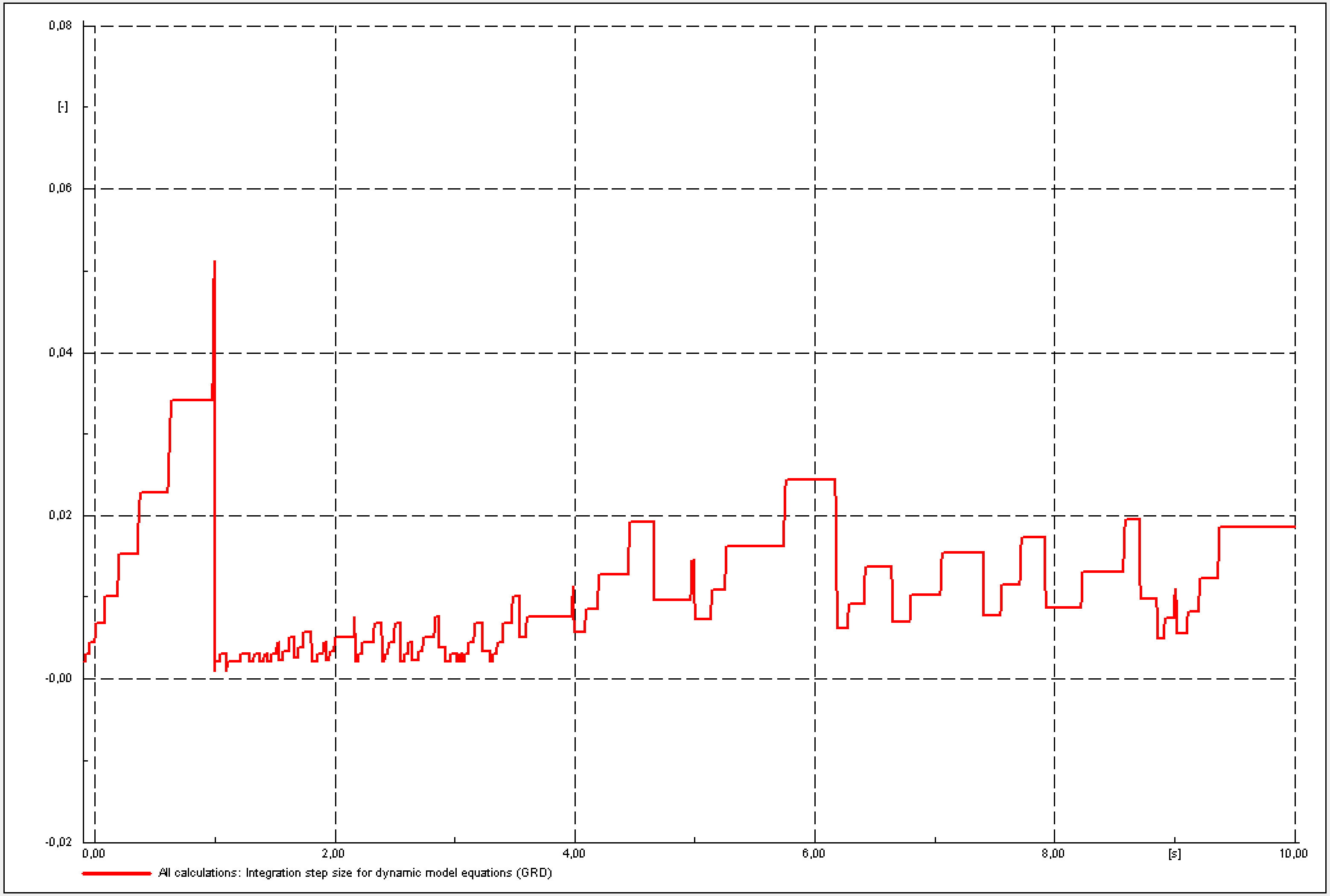
```
DIgSI/wrng - Inconsistent argument in DSL division by x: denominator argument x is zero; x=E-6 will be assumed.
DIgSI/wrng - Argument x is variable 'Ve'.
```

- In a model especially, the equation `x.=yi/T` issues a warning for T=0

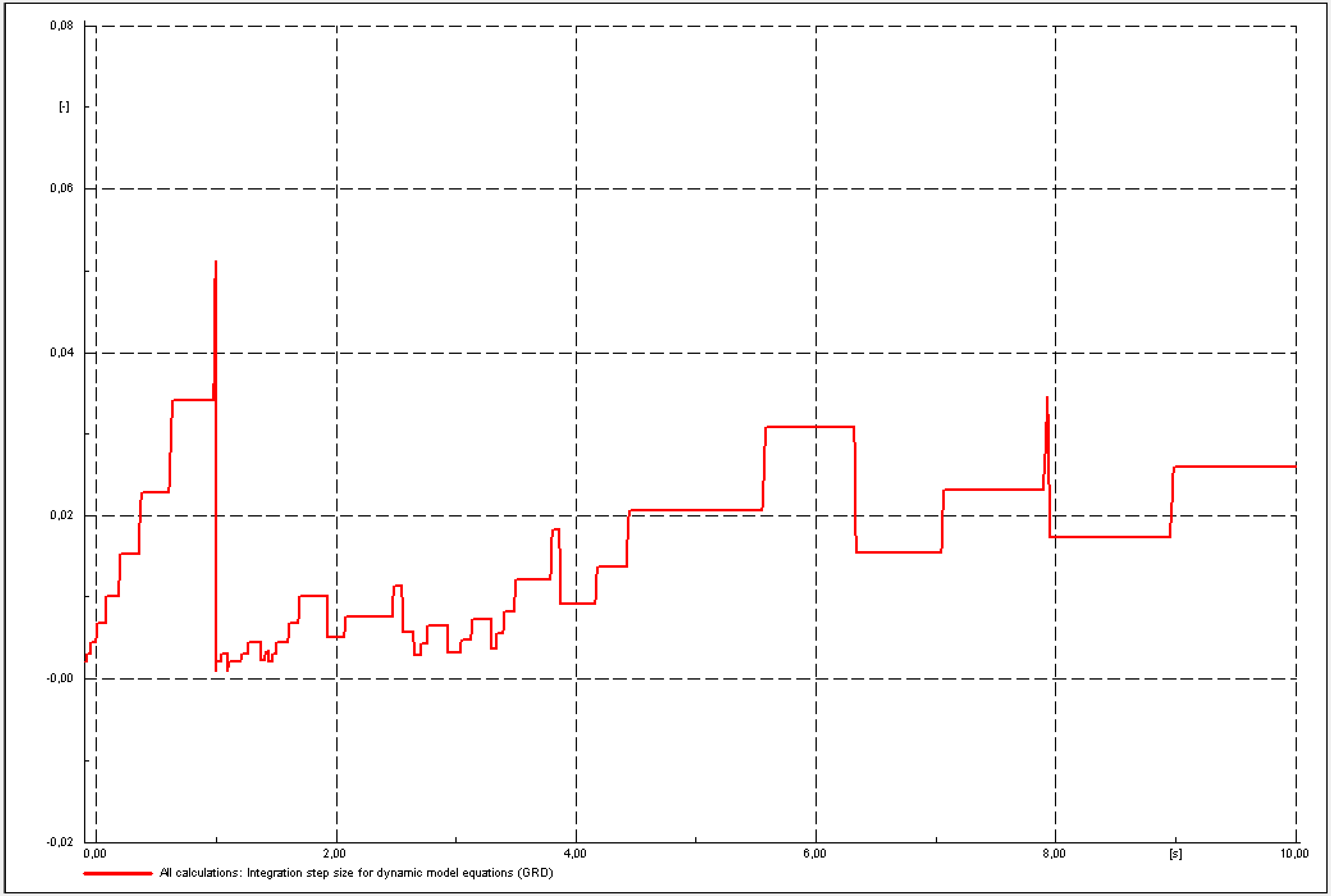
- This is the critical equation responsible for the large prediction error (a division by zero is interpreted by DSL as a very large number)

- All the issues are solved and no warnings are issued

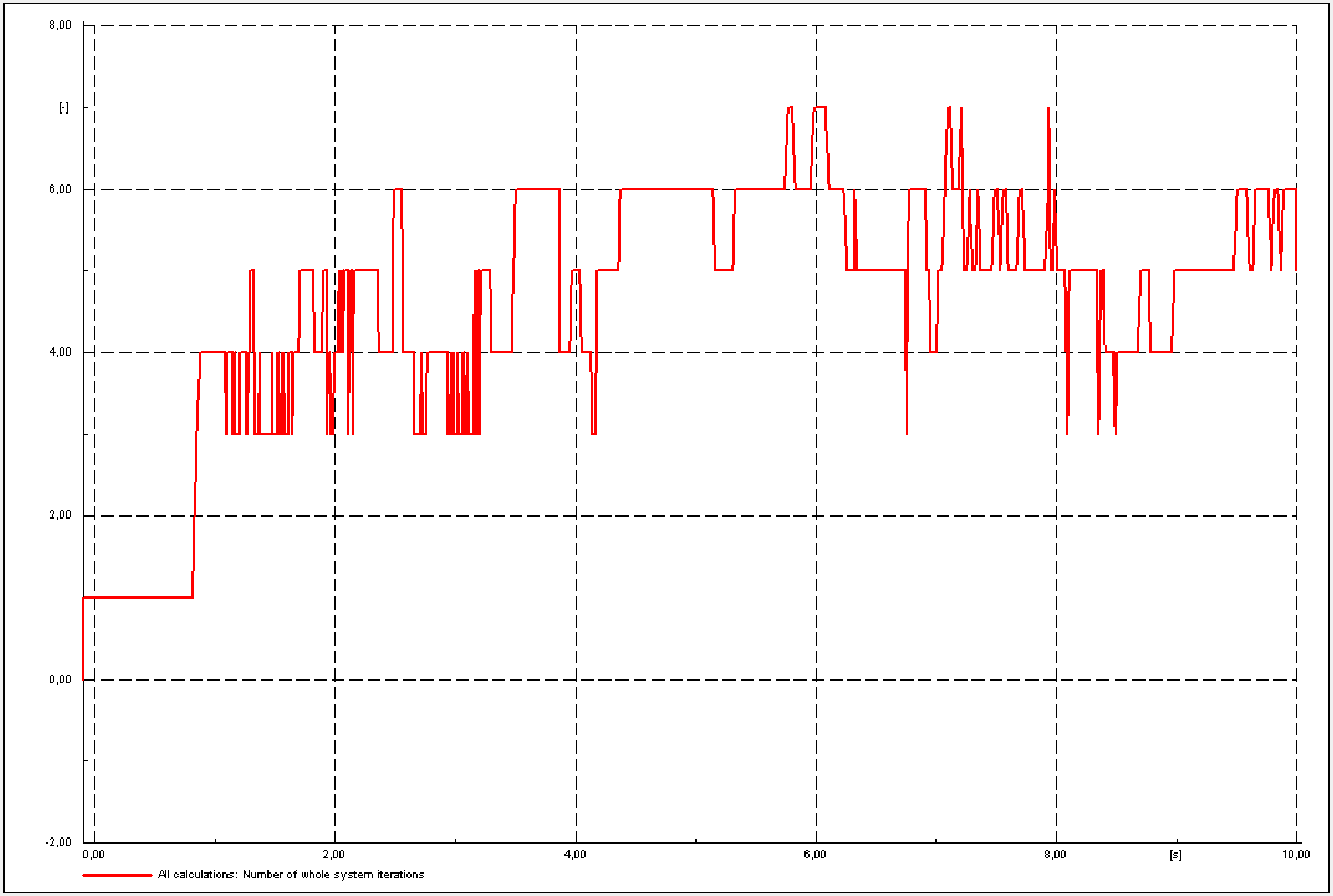
In the corrected PF15.2 model, the step size may increase…



… but it can be made better with less variations (larger band)



Several whole system (interface) iterations are performed…



Results on System 2

- Global speed-up (model/step issues and all PF15.2 enhancements)

PowerFactory version	Improved solver	Consistent model	Improved step strategy	All options Section 3.2	t_{CPU} (s)	Speed-up (wrt Base)
15.1.7					63	Base
15.2.1	✓				46	1.35
15.2.1	✓	✓			22	2.85
15.1.7		✓	✓		22	2.85
15.2.1	✓	✓	✓		18	3.5
15.2.1	✓	✓	✓	✓	9	7

- Speed-up due to PF15.2 performance enhancements

PowerFactory version	Improved solver	Fast check and output	Fast sol.	One-shot DSL	Comp. models	t_{CPU} (s)	Speed-up (wrt Base)
15.1.7						22	Base
15.2.1	✓					18	1.2
15.2.1 (default)	✓	✓				17	1.3
15.2.1	✓	✓	✓			11	2
15.2.1	✓	✓	✓	✓		10	2.2
15.2.1	✓	✓	✓	✓	✓	9	2.45

Take-home message

Take-home message

- PowerFactory 15.2 can make your simulations up to 5 times faster

- The rest is up to you. Please make sure to:
  - **Choose** algorithmic **options** that fit models and simulation purpose
  - **Observe** error messages, warnings and algorithmic **counters**
  - **Keep** own models as **small** and **simple** as possible and **compiled**

- More information on the examples shown in this presentation can be found in www.digsilent.com download area → PF15.2 → What's New → Simulation Performance Benchmark (for registered users)

When this is not enough…

- Please keep in mind that DIgSILENT support is staffed with simulation specialists that are always willing to help you:
  - Understanding the simulation
  - Detecting model/solver issues
  - Speeding up the simulation (or making sure it is as fast as possible ;-)

- Thank you for your time. Further comments may be directed to:

POWER SYSTEM ENGINEERING AND SOFTWARE

Dr. Davide Fabozzi
Application Development
DIgSILENT GmbH
Heinrich-Hertz-Straße 9
72810 Gomaringen / Germany

T: +49 (0)7072 9168-49
F: +49 (0)7072 9168-88
d.fabozzi@digsilent.de
www.digsilent.de

Managing Directors:
Dr. Martin E. Schmieg (chair)
Dr. Jochen Alber
Flavio Fernández
Christoph Schmid

Handelsregister Stuttgart
Reg-No. HRB 381759
VAT-No. DE 185636875