Modelling Three-Phase Power Network with a Multi-agent System

Miroslav Prýmek[1], Aleš Horák[1], and Tadeusz Sikora[2]

[1] Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic
{xprymek,hales}@fi.muni.cz
[2] Department of Electrical Power Engineering, FEECS,
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava - Poruba, Czech Republic
tadeusz.sikora@vsb.cz

Abstract. Multi-agent systems have already proven to be useful for modelling processes in distributed environments, such as an electricity distribution network. The autonomous design of each network element and its capabilities allows to adapt the model to any situation and the results can be used for monitoring as well as real time control.

One disadvantage of such models lies in a seeming inability to evaluate global analytical computations. In this paper, we show how such computations can be performed within a power network simulation system by means of purely local interactions and an iterative communication flow. The results are then compared to a standard analytical approach.

Keywords: power network simulation, power network modelling, multi-agent simulation.

1 Introduction

Static power network models are often used for *global analytical computations*[1] of dynamic processes between the network elements [1–3]. Multi-agent systems (MAS) offer new perspectives in computing complex network models. The standard analytical approach is suitable for optimizations, but requires usually a full recalculation after any (even slight) modifications to the network logic. To be able to gain full power of MAS power network modelling, the computation algorithm should not be based on global data and should not use any analytical technique that takes into account all network elements.

In the following text, we describe the algorithm that is based purely on *local interactions*, it is referred to as the "*local approach*" to power network modelling. This algorithm works only with neighbouring elements of the power network as "agents" in MAS (usually a *power line* agent and a *source/consumer* agent).

---

[1] In this perspective, "*global*" refers to the fact that the computations need to work with the network data as a whole, usually a (multi)graph.

J. O'Shea et al. (Eds.): KES-AMSTA 2011, LNAI 6682, pp. 130–137, 2011.

The algorithm is included and modelled in the MAS power network simulation system named Rice [4].

The algorithm was tested on a model of a real-world power network. The network steady-state was computed in two systems – *PAS Daisy* [5, 6], as an example of a stable and widely used system for power network modelling with the global analytical approach, and in Rice. The results of the modelled networks are presented further in this text.

2 Real Power Network Model for Power Network Steady-State Computation

For testing purposes of the Rice MAS the 10 kV electric distribution network (EDN) was chosen. Both the network parameters and load data were acquired from a real electric network. The network is fed from one 110 kV/10 kV transforming station named MART.

The network and the loads are typical for urban area. In the test system there are two big loads (Fnem, Energ), where the outgoing feeders from the supply node MART can be joined. Nevertheless in this test case the interconnection was not taken into account and the tie lines K5 and K10 are switched off.

As it is first design of MAS calculation of EDN steady state, only direct-current model was considered. So only resistance of lines was taken into account.

The line and node parameters are in Table 1. The power network scheme is depicted in Figure 1.

Table 1. Node and branch parameters and calculation results

Name	Load (kW)	Voltage (kV)
Mart	Source	10
BudN	220	9.8905
BytV	250	9.9743
CSAV	150	9.9441
Dom_D	200	9.6100
Energ	1100	9.9594
FNem	3000	9.7940
Menz	470	9.9602
Prad	260	9.8403
VTP1	400	9.9651

Name	Length (m)	Resistance (Ohm)	Current (A)
K1	811	0.1	74.70
K2	811	0.1	66.40
K3	811	0.1	49.80
K4	811	0.1	29.05
K5	811	0.1	0
K6	1169	0.15	211.65
K7	1169	0.15	203.35
K8	1169	0.15	190.90
K9	1169	0.15	178.45
K10	930	0.12	0
K12	2855	0.37	62.25

3 Analytical Computation of the Steady-State

To compare the results of steady-state MAS calculation in Rice the steady state of the test system was calculated also in program PAS Daisy Off-Line, which is

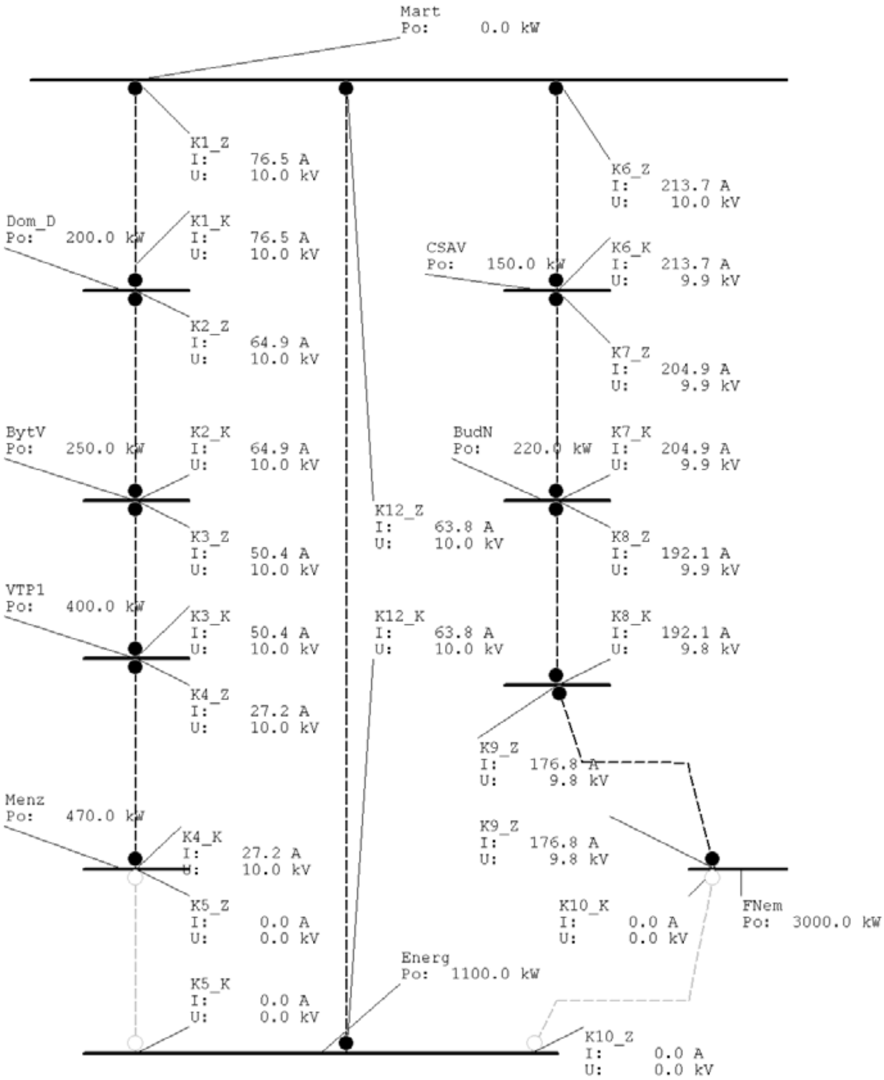

Fig. 1. Scheme of the evaluated power network

a part of the PAS Daisy system [5, 6]. This program is designed for calculating electric network modes, and it is commonly used by the utilities for distribution system operating and planning. The network parameters used for computation in PAS Daisy come from the model described in Section 2.

The analytical calculation is based on modified Newton-Raphson iteration method [7] for fast and reliable convergence of calculation. In the employed method it is no need to build and calculate Jacobian at each iteration step.

According to accuracy required, and network state, the calculation ends within 3 to 6 iterations.

The PAS Daisy program calculates with constant power loads, independent on voltage change (e.g. voltage drop).

4 MAS Computation of the Steady-State in Rice

As mentioned above, the multi-agent approach motivates developers to implement completely independent software components. The centrally-controlled functionalities are kept to the necessary minimum. Even the topology of the network is implemented as an *agreement* between agents. The main advantage of this approach is that the network topology can be changed dynamically according to the observations and decisions of the agents.

The agents which cause the first step to establish the topology, are the ones representing the power lines. For each agent there are two pieces of knowledge inserted into their knowledge base: the name of the agent representing the line's input and the name of the output agent. Immediately after the launch of the simulator, the line will send these agents a message telling "I am your input" and "I am your output." All the actions related to establishing a power flow channel will be taken as a reaction to the arrival of these messages as will be illustrated in the following paragraphs.

The modelled is 10 kV (nominal line-to-line voltage) three phase alternated current EDN. But in a matter of design simplicity, the network is considered ideally symmetric and thus only one phase is calculated. An other simplification is used, as the model is calculated as direct current power flow, so the lines have resistance R only, and the load power factor is equal to 1.

Rice computes the one-phase model first (line-to-ground voltage, 1-phase power) and then the values are transformed to 3-phase results, which are presented to users (line-to-line voltage, 3-phase power). In the following text, the described agent actions thus work with 1-phase values. The 3-phase values are calculated in consequence of the 1-phase ones.

The Source Node. The source node is an input point of the network. Its voltage is U_{in}. The immediate overall consumption of the network connected to this source is:

$$P = U \cdot I \tag{1}$$

The value of the maximal amount of electric power the agent can deliver to the network can be defined as a pre-set value. All power-increase requests will be denied if they exceed this limit.

The source node agent knowledge base contains the following items:

1. U – voltage (Volts)
2. I – current (Amperes)
3. P – load (Watts) – computed according to equation (1)

4. P_max – maximum load allowed
5. *overloaded* – true/false, meaning that the source is actually overloaded

To make all the knowledge pieces correct and make the agent cooperate with other agents in the network, we must define these behavioral reactive primitives:

1. an agent tells me that it is connecting to my output ⟶ subscribe for its I_demand knowledge
2. subscribed I_demand changes ⟶ change the current I accordingly
3. the current I changes ⟶ recalculate the value of the load P
4. the voltage U changes ⟶ recalculate the value of the load P
5. if the load $P > P\_max$ ⟶ assigns `true` to *overloaded*
6. if the load $P \leq P\_max$ ⟶ assigns `false` to *overloaded*

The Power Line. The power line agents initialize the network topology establishment. Each such agent automatically sends the only two unsolicited messages in the system thus interconnecting and starting the whole simulation. At each moment the line is connected to one source node and one consumer node at most but these connections can dynamically change throughout the simulation.

A power line agent has some features in common with the source node. Besides them, the power line has one specific piece of knowledge R, which is the *resistance* of the line, and two voltage indicators: the *input voltage* U and the *output voltage* $Uout$.

The line resistance causes a voltage drop on the line. The voltage drop is specified as:

$$\Delta U = R \cdot I \tag{2}$$

The output voltage is reduced by the voltage drop:

$$Uout = U - \Delta U \tag{3}$$

The behavior primitives of the line are as follows:

1. simulation start
   ⟶ send notifications to your input and output
   ⟶ subscribe for the voltage U and the current I of the input
   ⟶ subscribe for I_demand of the output
2. the voltage U has changed ⟶ recalculate the output voltage $Uout$ according to equation (3)
3. the current I has changed ⟶ recalculate the output voltage $Uout$ according to equation (3)

Details of the agent communication phases are discussed below.

The Consumer Node. The consumer agent represents one of the endpoints in the power distribution network. It initializes the flow of the electric power according to its current power consumption demands. This is implemented by setting the I_demand knowledge (which is automatically subscribed when the line connects as an input to this agent) according to the P_demand knowledge and equation (1).

The pieces of knowledge of the consumer agent are:

1. I – the input current granted from the Source Node through the network
2. U – the input voltage that propagates from the Source Node
3. *P_demand* – the actual power consumption requested by the Consumer Node
4. P – the actual power delivered by input (calculated using the equation 1)
5. *P_threshold* $= P -$ *P_demand* – the desired sensitivity of the computation. When P and *P_demand* differers less than *P_threshold* the Consumer Node demand is saturated.
6. *I_demand* – the current corresponding to *P_demand*

The consumer agent's behavioral primitives are:

1. generate *P_demand* according to the time of day and the given consumption table
2. the power consumption *P_demand* has changed ⟶ change the corresponding current *I_demand* according to equation (1)
3. I has changed ⟶ consider if the power demand is saturated
4. the consumption is not saturated ⟶ increase *I_demand*
5. the predefined consumption function

The main phases of the simulation are as follows – the first three phases represent the system initialization. The energy flow phase is started by setting *P_demand* knowledge of some agent to a non-zero value. The consumer node uses the network with the established topology to initiate the actual energy flow throughout the network. When *P_demand* is set to a non-zero value, *I_demand* is computed and dispatched to the subscriber (input line). The request then propagates throughout the whole network to the power source which decides if the request can be satisfied. If yes, it sets the I knowledge which then propagates through the network back to the consumer. All voltage drops on lines and the current splitting are computed automatically by the agents representing the appropriate network elements.

Throughout the whole process, the consumer agent simulates the power demand variations in time. All the power flow indicators get recomputed automatically. The Rice network power flow model implementation is based completely on local interactions between the agents and on-demand knowledge propagation. No information is communicated without being requested. Since each part of the simulation process is based on changes of agents' knowledge and the reactive nature of the agents, it is possible to manually change any of the involved indicators interactively during the simulation, including the network topology.

4.1 Comparison of the Two Steady-State Computation Approaches

A comparison between analytical calculation (PAS Daisy) and MAS calculation was performed. In this comparison, we have concentrated to the analysis of the results obtained by both methods. The size of the modelled network did not allow to make exact and meaningful measurements of the computational time of the methods – both computations were performed in fractions of seconds.

Table 2. Values and value differences measured during the comparison. Absolute error values are displayed in permille (per thousand) values ‰.

	PAS Daisy I (A)	Rice #1 I (A)	Rice #2 I (A)	Err #1 (abs.‰)	Err #2 (abs.‰)
K1	76.50	76.28	76.73	2.82	2.98
K2	64.90	64.73	64.98	2.56	1.29
K3	50.40	50.30	50.42	1.98	0.36
K4	27.20	27.16	27.24	1.65	1.60
K6	213.70	213.43	213.66	1.24	0.18
K9	176.80	176.83	176.84	0.20	0.23
K8	192.10	192.07	192.10	0.16	0.02
K12	63.80	63.77	63.77	0.52	0.52
K7	204.94	204.77	204.95	0.82	0.06

Table 3. Voltage results comparison (errors in permille ‰ values)

	Rice	PAS Daisy	Err (abs.‰)	Rice	PAS Daisy	Err (abs.‰)
	Voltage U (V)			Voltage drop ΔU (V)		
BudN	9890.79	9890.52	0.0268	109.21	109.48	2.42
BytV	9974.44	9974.29	0.0152	25.56	25.71	5.92
DomD	9986.17	9986.1	0.0072	13.83	13.90	5.20
CSAV	9944.26	9944.12	0.0139	55.74	55.88	2.49
Energ	9959.32	9959.36	0.0042	40.68	40.64	1.03
FNem	9794.45	9794.03	0.0428	205.55	205.97	2.04
Menz	9960.41	9960.17	0.0237	39.59	39.83	5.94
Prad	9840.63	9840.28	0.0353	159.37	159.72	2.18
VTP1	9965.33	9965.13	0.0198	34.67	34.87	5.66

The results of line current calculation are presented in Table 2. Two calculations in Rice were performed: Rice #1 with low sensitivity of consumer agents to the $P - P\_demand$ difference, and Rice #2 with higher sensitivity (smaller $P\_threshold$). The agents with higher sensitivity in test Rice #2 performed more iterations and their error value (*Err*) is mostly lower, and also closer to the analytical results. For the error calculation the analytical values (PAS Daisy) were taken as a reference. Higher sensitivity has shown generally better accuracy of the MAS calculation.

Table 3 shows Rice and PAS Daisy voltage differences in nodes. The voltage difference calculated as a proportion is very low even when expressed in permille (per thousand) values. This fact is caused by low voltage drop and thus small differences on relative scale between voltages calculated by Rice and PAS Daisy. We have thus displayed also the calculated voltage drops in Table 3. The PAS Daisy results were taken here as a reference. The difference varies from 1 to 6 ‰ in absolute permille values.

As a result of this comparison, we can tell that the average difference between MAS and analytical computations were less than 1 ‰ in current and about 2 ‰ in voltage drops (less than 0.02 ‰ in voltage values). We believe that in this respect the MAS results can be regarded as equivalent to the analytical ones while keeping the distributed manned of the overall computational model.

5 Conclusions

In this paper, we have described the local approach to computing power network steady-state by means of a model in a multi-agent system (MAS) named Rice. Using such local computations offers best results in the network flexibility and distributivity, making it an ideal tool not only for modelling, but also for monitoring and control.

To verify the algorithm output, the computation was modelled in two systems – PAS Daisy as a global approach system and Rice MAS as a local approach system. The results of these two computations were discussed and they prove that the iterative local approach provides practically the same values as the global analytical modelling, while keeping the flexibility of MAS.

Acknowledgments

This work has been partly supported by the Czech Science Foundation under the project 102/09/1842.

References

1. Hcbson, E.: Network constrained reactive power control using linear programming. IEEE Transactions on Power Apparatus and Systems PAS-99(3), 868–877 (1980)
2. Schweppe, F.C.: Power system static-state estimation, Part III: Implementation. IEEE Transactions on Power Apparatus and Systems, 130–135 (1970)
3. Krishnamoorthy, S., Chowdhury, M.H.: Investigation and a practical compact network model of thermal stress in integrated circuits. Integrated Computer-Aided Engineering 16(2), 131–140 (2009)
4. Prýmek, M., Horák, A.: Multi-agent Approach to Power Distribution Network Modelling. Integrated Computer-Aided Engineering 17(4), 291–304 (2010)
5. Střída, F., Stacho, B., Rusek, S.: Network steady-state modelling in the Bizon projektant program. In: 10^{th} International Scientific Conference Electric Power Engineering (EPE 2009), Ostrava, Czech Republic, VŠB TU Ostrava, pp. 186–189 (2009)
6. DAISY, s.r.o.: PAS DAISY Bizon Projektant (2010), `http://www.daisy.cz`
7. Kundur, P., Balu, N.J., Lauby, M.G.: Power system stability and control. McGraw-Hill Professional, New York (1994)