AN IMPROVED NETWORK SIMULATOR FOR EV / V2G STUDIES

Stephen BRODERICK
University of Southampton - UK
srb3g13@soton.ac.uk

Andrew CRUDEN
University of Southampton - UK
A.J.Cruden@soton.ac.uk

Suleiman SHARKH
University of Southampton - UK
S.M.Abu-Sharkh@soton.ac.uk

Nigel BESSANT
SSEPD - UK
nigel.bessant@sse.com

ABSTRACT

An 'Improved Network Simulator' (INSim) is described, offering EV and V2G aggregation on realistic networks.

INSim models distributed loads and supplies including V2G and various ESS on an arbitrary network solved by EPRI's OpenDSS. INSim is a work in progress and is to be accessible via an OpenSource platform.

INTRODUCTION

EV use in the UK is expected to grow in the near future. The growth brings new challenges and opportunities:

- LV is constrained by a built capacity of 2 kW average per home; mass EV charging loads will exceed this [1],
- EVs capable of V2G may be used as an Energy Storage System (ESS), potentially bringing system management, Renewable Energy Sources (RES) optimisation and financial benefit to the EV owner.

To analyse a DSO LV network in detail a simulation of likely scenarios is needed. However typical load-flow tools do not offer the simultaneous capability of:

- mass numbers of EVs modelled at an individual level, including EV movements and dynamic events
- with network load impacts on local feeders,
- with V2G aggregation working to contract, potentially aiding or congesting LV systems or effecting volt levels
- with dynamic DR style Feeder Demand Management (FDM) schemes [2], and specifically
- modelling the UK LV system.

Such a simulator must work from at least MV to LV with real-world networks and assets, with dynamics both predictable and situation-driven.

The primary author is undertaking an EngD and needs such a tool to address his Doctoral questions:

1. What are the benefits and impacts of V2G in the UK?
2. Given present UK LV networks, what are possible progressions to the benefits, while minimising impacts?

To explore these an "Improved Network Simulator" (INSim) is being developed; a program in python for PC. INSim strives to fulfil the capabilities listed above.

INSim gains the ability to consider complex real-world networks by using EPRI's proven distribution simulator, OpenDSS [3, 4]. To this is added a custom scheduler and dynamic-load layer for V2G aggregation. Although the focus here is LV, INSim is general. V2G EVs can represent ESS connected at any voltage, forming storage resources and interworking with RES models.

INSIM CAPABILITIES

Network Related

Hierarchical laying of networks by purpose or voltage with network cloning. A large LV system can be defined, cloned and independent copies placed as needed.

EV Modelling

Timed events (e.g. journeys), EV internal loads, losses and charge state. V2G charge / dispatch via intelligent Agents acting as Aggregators.

General Features

- Remote co-ordination of EV activity by N agents
- agents may be commanded by timed contract or signal (e.g. a network load or tariff change),
- timed sequences of events or environmentals (e.g. temperature) of N arbitrary values at 3 levels:

days of values, weeks of days and cycles of N weeks,

- repeatable, clone-able journeys,
- trips and power-fails,
- networks may be partitioned into named 'Zones',
- snapshot / rollback to aid 'what-if' analysis.

OPERATIONAL METHOD OF INSIM

INSim depends upon OpenDSS capabilities. OpenDSS can be used stand-alone or with other software via a 'COM interface', including control over this interface. Further, OpenDSS has a 'single-shot' mode, in which starting conditions are set and incremental changes applied. In this mode, OpenDSS can solve (determine) volts and currents for a complete network. The user may vary individual parameters then perform another solve.

INSim uses the single-shot method. OpenDSS is tasked with a starting state; INSim then determines load changes and sends these to OpenDSS for solving; this repeats.

INSim has several key elements:

- the concept of a 'solve tick', of set simulation period. Time is counted in solve ticks ('st') e.g. st 607;
- a steady-state system, only changed by:
  - scheduled events e.g. timed load changes and
  - dynamic loads / dispatch from EVs, each of which is

represented by a state-machine.

Note the user states events in 'simulation wall-time' i.e. as a clock on the wall would report, had it been in the simulation. Thus the user asserts 'depart EV at 07:30' (data output and reports also use wall-time).

INSim has a layered structure with high-level abstractions at the top. OpenDSS is the lowest stage.

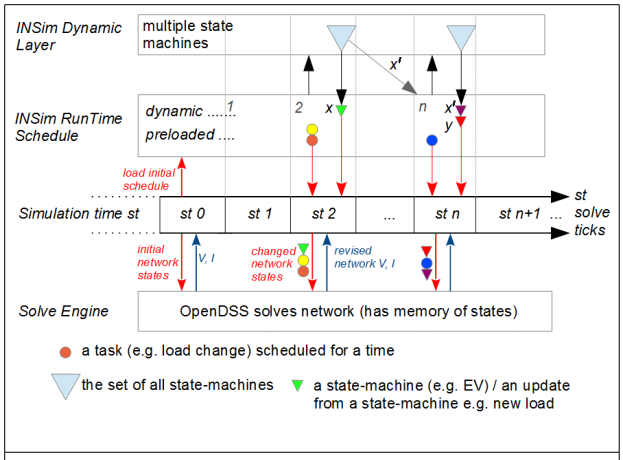

Figure 1: INSim software layers with simulation time left to right. Loads are updated and passed to OpenDSS

Figure 1 shows the system as a top-down hierarchy:

- the dynamic layer containing agents / state machines / rule-based systems. These determine any load changes based on situation e.g. an EV may check its SOC and may stop charging as SOC has reached a target value;
- the scheduler system, based about a list of events within each st period. The schedule can be pre-loaded with static items, or a pointer to a dynamic task (to pass a message to a specific state-machine),
- the progress of simulation time counted in st, and
- OpenDSS which performs network solves.

The core of INSim is the schedule (with an accompanying run-time scheduler). The schedule is a list of events which can be amended; INSim operation depends upon on 'on-the-fly' schedule updates.

The schedule can have two types of entries:

- preloaded events at set times (the coloured circles);
- dynamically assessed events (the inverted triangles).

Entries have a 'data packet' to pass to a recipient. Each entry is timed to occur in an st, counted from zero, which represents midnight 00:00 hrs on the first simulated day.

Sequence of Operation (simplified)

Setup

First the user needs to define an OpenDSS network. The user then adds events at known times and a set of EVs with their movements. Locations can be cited by common name e.g. “St. James' Car Park”.

INSim Startup / data load stage (occurs at st 0)

- Network maps, data etc. sent to OpenDSS as needed
- Static timing data loaded into the schedule
- Dynamic events and data-packets similarly loaded.

A trial solve is attempted; a successful solve becomes simulation start and the system is in the first steady state (the solution network V and I values are logged).

Process an st (loops till nothing more to process)

The scheduler advances to the next non-void st. Having found an occupied st, the scheduler:

- sequences all events 'by type' to ensure sensible function (e.g. power-loss precede EVs going on-charge)
- in the determined sequence:
  - copy preloaded events to OpenDSS
  - invoke any state-machines, obtain their data and pass to OpenDSS
- OpenDSS is told 'solve', forming a new system state
- with data capture as needed - and the cycle is complete.

Following the Figure 1 sequence from st 0:

- st 1 is passed over as it is empty
- st 2 holds two preloaded items and an instruction to update state-machine x. When x is called, it generates two items: a load update (green triangle) and a new event **x',** which is entered into the schedule.

Example: “x” is an EV at an EVSE. The state machine x (when called) finds it is time to 'Start Charging', sets the inverter to 'charge' so generating a local load. However there is a halting condition: charging must complete at some time; **x'** is calculated and scheduled. Note the new event is not 'stop charging', rather, it is 'Assess/Update Charging' (a power fail may limit SOC).

However item “x” might have been an agent, generating actions placed on other state-machines (agents or EVs).

Note that passing messages via the scheduler allows modelling of telecom delays (or even data losses). Further, as the scheduler accepts new events, cause / effect sequences may spontaneously arise i.e. a cascade of trips in fast succession.

WORKLOAD MINIMISATION

The design goal aims for 10,000+ active EVs and 100,000+ network locations with simulated timeframes of weeks to years (in tolerable processing time i.e. overnight); clearly CPU workload must be minimised.

INSim employs a steady-state discrete event model, with variable range partial steps. This asserts the system is in steady-state, perturbed only by scheduled events - giving a new steady state. Solves only consider new states.

This means that a typical EV (at a socket charging for hours) is only simulated at state-changes i.e. at charging start / end. This contrasts with traditional methods which

may investigate EV activity in every period. The majority of an EV charge cycle thus skipped. The range of this skip is driven by situation, hence variable range.

These methods mean workload is now a function of the number of state changes, not simulation duration. INSim assess only elements which change, when they change.

Another common simulation method is to quantise the calculated st. However, INSim does not quantise - 'partial st' are used; that is, st are not converted to integer.

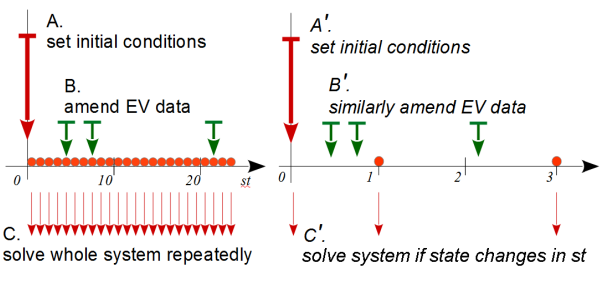

Figure 2: Traditional and INSim period solves. INSim does not need fine st for fidelity, events / st need not be coincident, solves occur if needed/voids (2) are skipped

Thus a “fast charge” (from st 205.160 to st 207.762) can be represented. Quantised st may have misrepresented the duration, causing error. Small st (perhaps micro-seconds) are an alternative; this limits error but increases system workload, lowering 'range of simulation'. With non-quantised st, small st are not required.

THE EV MODEL

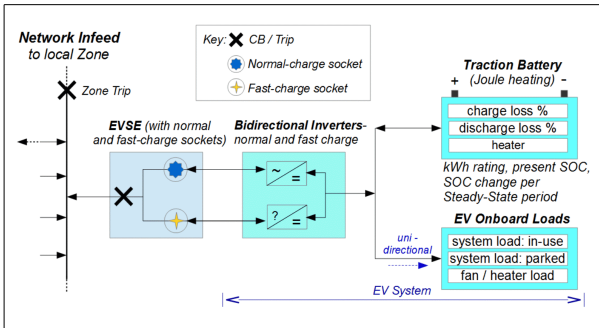

Figure 3: Zone network, EVSE and an EV with dual inverters and general battery components

The EV is modelled connected, in Fig. 3 above. Features include dual inverters for multiple charging modes / sockets, multi-level charging (fast-standard-slow-trickle), ability to cope with power losses and basic V1G and V2G capability (i.e. can accepts commands, report capacity for V2G tasks and perform dispatch).

SOC change calculations are by linear equation with a start-value, a rate of SOC change per st and st duration. Internal loads, inverter and battery losses are represented.

Journeys are timed 'Depart' and 'Arrive' events at named places, with per-hour driving erosion of battery SOC.

EVs record charge flow and by what authority (contract). EVs may refuse commands (a 'recusal': to stand down due to conflict of interest). Examples: Not at EVSE, no charge, conflict with driver needs. Recusals are logged.

ZONES / ZONE MANAGERS

Located at a PCC or feeder connection, Zones have geographic and network location, m be nested and can throw trips. Zones hold local data e.g. environmentals, EVSE availability, socket ***VI*** etc. A Zone plus an Agent form a 'Zone Manager', able to control local EVs.

AGENTS

Agents are chainable state-machines able to accept scheduled 'contracts of duty', signals (e.g. a load level) and may issue V2G commands. Agents mimic decision making to represent an interest, such as a strategic authority or DSO interests. Thus an Agent may be an Aggregator executing a commercial contract, or be a substation mounted Zone Manager issuing commands to local EVs, to maximise local capacity / manage EVs in sympathy with the network as well as to contract.

REPETATIVE (SCHEDULABLE) DATA

'Repetitive data' imply cyclic data structures. Data is sourced by the user in summary form as:

- named days of timed data items,
- named weeks containing days,
- repeating named long cycles containing weeks,
- 'special days'; single days occurring at set times.

Named days of timed values are created, e.g. for weekends and weekdays ('Day_1' and 'Day_2') containing sets of times and values (e.g. 00:00, 28.6; 09:30, 22.4; 10:40, 28.9 etc.). Weeks may be defined:

week_A = 5x Day_1, 2x Day_2
week_B = 1x Day_2, 4x Day_1, 2x Day_2

A long cycle lists weeks with scaling (for seasonality):

long_cycle_1 = 4x (week_A scale 1.1),
4x (week_B scale 0.95), 4x (week_A scale 0.9) etc.

The method can create timed: sources or loads, EV journeys, signals, network and environmental events. An example: cyclic insolation to mimic yearly PV, with special days of low light (a storm). By associating special days with Zones a weather system may 'track geographically' across the simulation.

FURTHER FEATURES

Python offers a digit range of 14 places. An st of two minutes allows simulation of activities 'hundreds of years'

apart, yet may see a trip cascade with milliseconds between events. This is feasible and would be simulated.

PRESENT DEVELOPMENT STATE

Prototyped elements include: Network layering and cloning, INSim / OpenDSS interaction and solving, the scheduler, a test EV able to: charge, dispatch, on-the-fly schedule updates; a data loader and log / event reporter.

Below is test output for "EV_L1". This EV experiences: Instantiation at st 0.0 (midnight), arrival / home charging at st 0.01 with stepped rates, departure, arrival etc.

	At st	B.SOC%	State-End	Steady State log description	Socket load
a	0.01	54.89	124.8293	ArriveAt: SOC_change_per_st: 0.313333	3,434.49
b	124.83	94.00	214.8300	Charging_Update: SOC_change_per_st: 0.033333	696.01
c	214.83	97.00		Charging_Update: SOC_change_per_st: 0.000333	62.84
d	252	97.00	0.0000	Depart: SOC_change_per_st: 0.000000	0.00
e	273	75.88	330.8191	ArriveAt: SOC_change_per_st: 0.313333	3,434.49
f	345	94.00		Zone_Trip: SOC_change_per_st: -0.009600	0.00
g	382.5	93.64	383.6489	Zone_UnTrip: SOC_change_per_st: 0.313333	3,434.49
	382.5	93.64	1462.5000	Zone_UnTrip: SOC_change_per_st: 0.000333	62.84
h	552	93.70	0.0000	Depart: SOC_change_per_st: 0.000000	0.00
i	573	72.58	641.3622	ArriveAt: SOC_change_per_st: 0.313333	3,434.49

Table 1: EV_L1 log: st timings, Battery SOC %, Steady State activity, SOC rate of change and socket load (W).

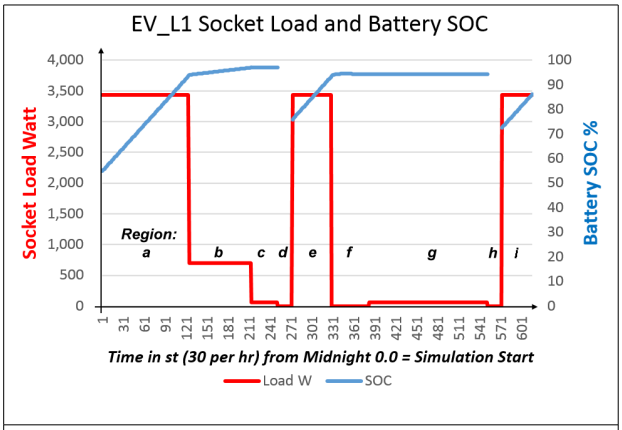

Figure 3: Socket load and SOC plot for EV_L1.

Note state-machine generated State-End st, scheduled 'on the fly' at ***b*** and ***c***. There is a power-outage; at ***f*** (st 345) a 'Zone Trip' occurs and the EV starts to loose SOC.

FUTURE DEVELOPMENT

To complete and test all components, followed by system integration then End-to-End tests vs. real-world data. The first simulation task will be to reproduce the University of Manchester's 'Esprit' FDM system [2].

RESEARCH BENEFITS OF INSIM

Primarily to quantify the impact of likely EV (and other load) growth on real-world LV networks. Following this, various LV FDM strategies will be investigated, noting for how long they may defer network re-enforcement.

The core of the work though involves V2G. The utility of V2G will be assessed over a variety of situations (and also the risk of loss, inherent in relying on an mobile and potentially unreliable resource). These will include:

- impacts / benefits of RES tariffs (a 'sunshine' signal)
- the benefits or not of local 'Zone Controller' Agents - dynamic V1G / V2G optimisation of feeders
- potential for V2G optimisation of local RES
- potential for profit though ESS and network services.

The load environment in which these are set will be taken from NGET's yearly Future Energy Scenarios [5] and the recent UK National Infrastructure Report [6].

SUMMARY

INSim offers a novel state-machine abstraction layer simulating EV and V2G agents, working with OpenDSS, a capable network solve engine. INSim may form and investigate dynamics on arbitrary realistic networks. Development progress is good yet many modules remain to be written. It is expected core components will be compete and validated by the end of 2016.

With thanks to University of Southampton's Centre for Doctoral Training and sponsors EPSRC and SSEPD.

REFERENCES

[1] Ofgem/EA Technology, 2012, "Smart Grid Forum Report: Assessing the Impact of Low Carbon Technologies on Great Britain's Power Distribution Networks", *Smart Grid Forum 2012*, p165-169 Available: https://www.ofgem.gov.uk/publications-and-updates/assessing-impact-low-carbon-technologies-great-britains-power-distribution-networks Accessed 2016-03-10

[2] E.SAUNDERS et al, 2015, "DIRECT CONTROL OF EV CHARGING ON FEEDERS WITH EV CLUSTERS", *23rd International Conference on Electricity Distribution*, CIRED, Paper 0635, 1-5

[3] R.C. Dugan, T. E. McDermott, 2011, "An Open Source Platform for Collaborating on Smart Grid Research", *IEEE Power and Energy Society General Meeting*, IEEE, vol-1, 1-7

[4] OpenDSS. Accessed 2016-03-11 at: http://smartgrid.epri.com/SimulationTool.aspx

[5] National Grid, "*2015 UK Future Energy Scenarios*", July 2015. Accessed 2016-03-11 at: http://www.nationalgridconnecting.com/2015-uk-future-energy-scenarios-published/

[6] National Infrastructure Commission, 2016, "*Smart power: A National Infrastructure Commission Report*". Accessed 2016-03-11 at: https://www.gov.uk/government/publications/smart-power-a-national-infrastructure-commission-report