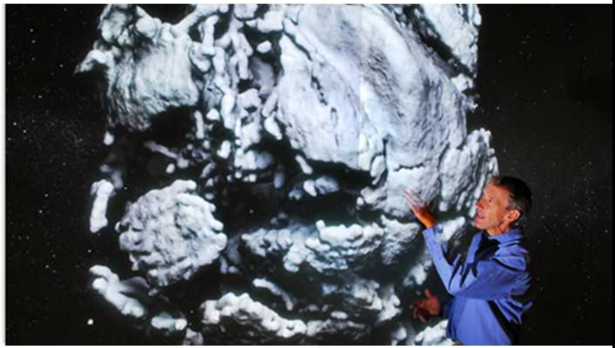
SAND2014-3553C

Exceptional service in the national interest

Sandia National Laboratories





Interoperability and Advanced Inverter Functionality Testing Update

29 April, 2014

Jay Johnson

Photovoltaic and Distributed Systems Integration, Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Project Description

		Year 1	Year 2	Year 3
Project Period		01-12-2012 ~ 30-11-2013	01-12-2013 ~ 30-11-2014	01-12-2014 ~ 30-11-2015
Total Budget (Unit: KRW 1000/US$)	KERI	KRW 373,700	KRW 300,000	KRW 300,000
		- determination of DG devices under test : PV and ESS - review of the existing test facilities - basic design of test bed and implementation - writing draft of interoperability test procedure	- construction of test bed - performing interoperability test on various DG devices - comparison of test results	- proposal of international standards -building mutual certification system on interoperability test regarding DG devices
	SNL	200,000 USD-	200,000 USD-	200,000 USD
		- Design and construct testbed - Establish utility to DER communication - Draft standardized test procedure and data acquisition protocol - Develop standardized analysis procedure and test reporting - Comparing Korean and U.S. results - Test PV inverters: 2 to 4 functions	- Amplification of PV inverter tests: 3-10 new inverter functions; residential to commercial scale - Revise interoperability protocol based on prior results - Comparing Korean and U.S. results - Design, construct, exercise testbed for ESS incorporation - Status report on program and results.	- Amplification of PV and ESS tests: complete set of inverter functions - Use experimental results to propose a test protocol standard (IEC) - Final report on program and results.

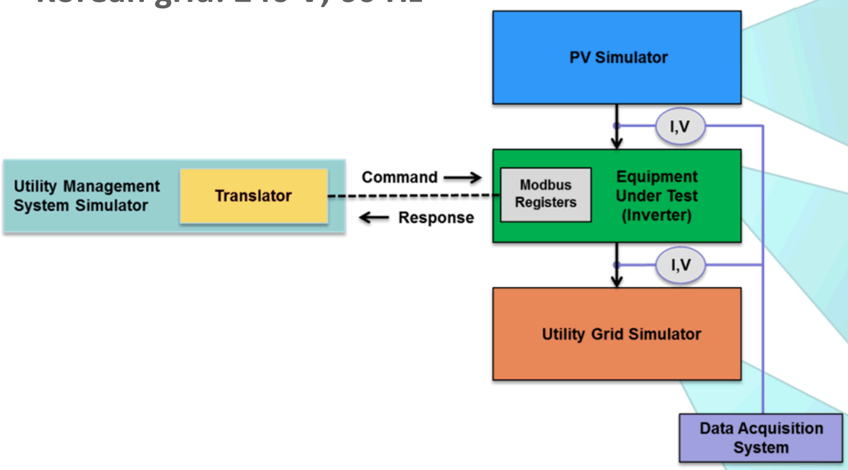
Inverter Interoperability Test beds at Sandia and KERI

Sandia National Laboratories

Both Sandia and KERI have designed interoperability testbeds. Differences in hardware and grid voltage and frequency demonstrate the capabilities of the testing protocols.

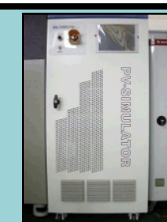
US grid: 120/240 V, 60 Hz

Korean grid: 240 V, 60 Hz



PV Simulator
Power: 200 kW
Voltage: 0-1000 V_{dc}/output
Current: 10 A/output

PV Simulator
Power: 10 kW
Voltage: 0-500 V_{dc}
Current: 50 A



Inverter: European Partner
Power: 3 kW
Communications: Modbus, RS485
- Available worldwide
- Inexpensive
- SunSpec compliant

Inverter: Korean Partner
Power: 3 kW
Communications: SEP 2.0

Grid Simulator
Power: 180 kVA
Voltage: 480 V 3ϕ
Current: 700 A/Phase

Grid Simulator
Power: 10 kVA
Voltage: 380 V_{L-L} 3ϕ, 4-wire

- **AIT is exercising the test protocols on a 5 kW Fronius inverter. They have also worked with a 680 kW Schneider Electric using an IEC 61850-90-7 gateway.**
- **RSE has a 5 kW Fronius inverter.**

SNL Status for IEC 61850-90-7 Function Tests

Sandia National Laboratories

Command	Function	Description	Type
INV1	Connect/Disconnect	Physically connect or disconnect from grid	Command
INV2	Adjust Max Generation Level	Set maximum generation level at Electrical Coupling Point	Command
INV3	Adjust Power Factor	Issues a power factor angle value	Command
INV4	Request Active Power	Request charging or discharging of the storage system	Request
INV5	PV/Storage Functions	Change the signal parameters for the storage system	Request
VV11	Volt-Var mode	Provide vars with no effect on watts	Set Parameter
VV12	Volt-Var mode	Provide maximum vars constrained by WMax	Set Parameter
VV13	Volt-Var mode	Establish fixed var settings	Set Parameter
VV14	Volt-Var mode	No var support	Set Parameter
FW21	Set maximum power output	Active power reduction due to high frequency	Set Parameter
FW22	Set maximum power output	Modify frequency-watts-delivered or watts-received curve	Set Parameter
TV31	Dynamic reactive power support	Provide var support at times of abnormally high or low voltage	Set Parameter
L/H VRT	Stay connected/disconnect settings	Set voltage ride-through or disconnect requirements	Set Parameter
WP41	Power factor settings	Set power factor in response to feed-in power	Set Parameter
WP42	Power factor settings	Modify power factor-watts curve	Set Parameter
VW51	Set output to smooth voltage	Voltage-watt curve of generator output	Set Parameter
VW52	Set output to smooth voltage	Voltage-watt curve of storage charge/discharge output	Set Parameter
TMP	Temperature mode behavior	Temperature-based curves	Set Parameter
PS	Signal mode behavior	Mode curves based on utility signal	Set Parameter
DS91	Modify DER Inverter Settings	Set default ramp rate, min./max. storage charge/disch. rate	Command
DS92	Event/History Logging	Request event logs	Command
DS93	Status Reporting	Request inverter status	Command
DS94	Time Synchronization	Set inverter time	Command

Implementation Successful (green box): INV1, INV2, INV3, VV12

Implementation Partly Successful (yellow box): FW21, DS91, DS93

Implementation Unsuccessful (red box): L/H VRT

Near-term Implementation (blue box): WP42, VW51

INV1 – Connect/Disconnect

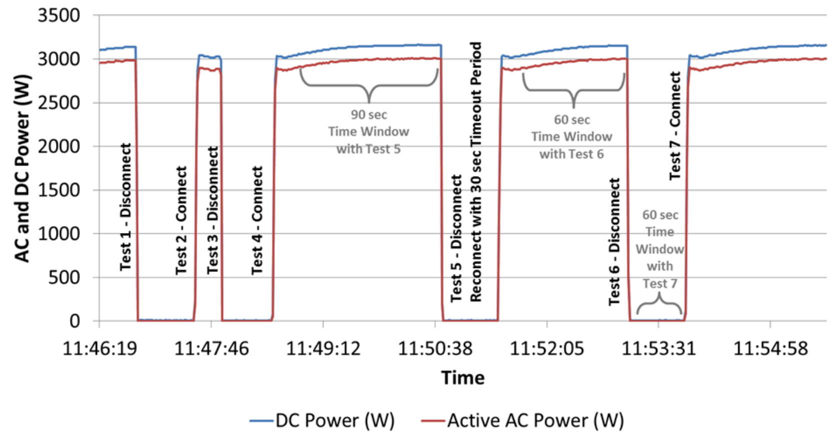
INV1 Test Protocol Sequence.

	Step	Task	Function	Notes
Communication	1	Request Status to EUT.	**DS93** (Status Reporting) & Direct Measurement (**DM**) of inverter output	Log time sent.
	2	UMS receives response to the DS93.command.		Log time received.
	3	Inverter output is measured and logged		Direct Measurement sensors record inverter output
	4	UMS issues a Disconnect /Connect Command to EUT.	**INV1** – Connect/ Disconnect	Log time sent. Command may include the following parameters: time window (optional) timeout period (optional) ramp rate (optional)
	4	EUT responds to the command.	**DS92 –** change in status is noted	Expected response message from EUT: Successful (DS92 status change logged and DM monitored output) Rejected (includes reason)
Electrical Behavior	5	Verify command was successfully executed (DS92 and DM).	**DM –** EUT output is recorded & logged	Monitor electrical output of EUT to determine if EUT connects/ disconnects and at what time. Measure voltage, current, power factor Record time
	6	Repeat test with varying parameters (see Table A1-3). Each test should be repeated at least once as needed.	–	
Analysis	7	Characterize EUT's response.	**DS92; DM**	Determine how command was executed.

INV1 Test Matrix.

Test Number	EUT Initial Operating State	Connect/Disconnect Command	Time Window (seconds)	Timeout Period (seconds)
Test 1	>50% rated power, unity power factor	Disconnect 1	Default (e.g., 0)	Default (e.g., 0)
Test 2	Inverter off	Connect 1	Default (e.g., 0)	Default (e.g., 0)
Test 3	>50% rated power, unity power factor	Disconnect 2	0	Default (e.g., 0)
Test 4	Inverter off	Connect 2	0	Default (e.g., 0)
Test 5	>50% rated power, unity power factor	Disconnect 3	90 seconds	30
Test 6	>50% rated power, unity power factor	Disconnect 4	60 seconds	0 (No Timeout)
Test 7	Inverter off	Connect 4	60 seconds	0 (No Timeout)

Connect/Disconnect (INV1) Test Results at Sandia



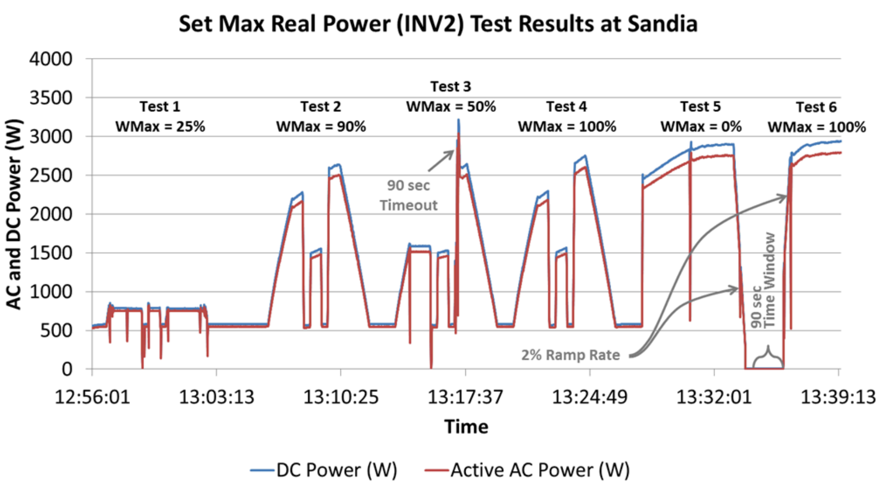
INV3: Set Constant PF – Electrical Characterization Results

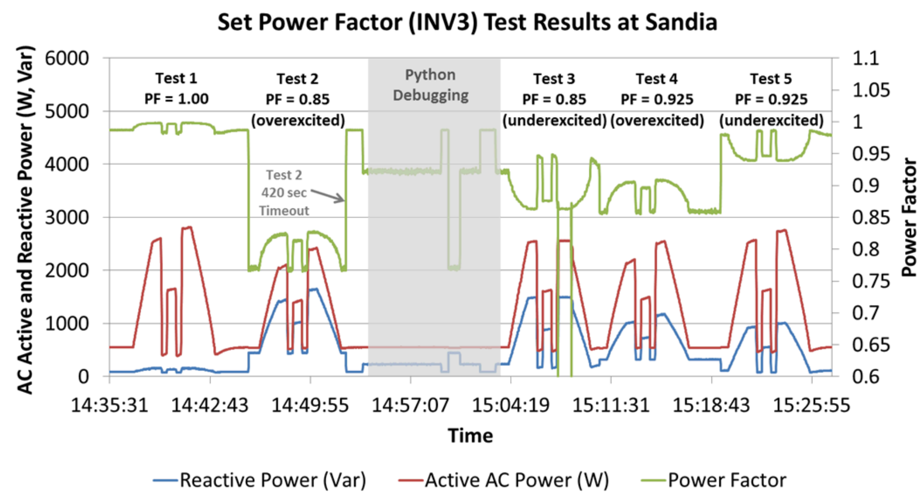
New INV2 Test Matrix.

Test	WMax (% nameplate)	Ramp Rate (% nameplate watts/sec)	Time Window (sec)	Timeout Period (sec)	Input PV Power Curve
Test 1	25	0	0	0	Figure A2- 1
Test 2	90	0	0	0	Figure A2- 1
Test 3	50	20	0	90	Figure A2- 1
Test 4	100	0	0	0	Figure A2- 1
Test 5	0	2	0	0	Constant at 100%
Test 6	100	2	90	0	Constant at 100%

New INV3 Test Matrix.

Power Factor (INV3)	Ramp Rate (%/sec)	Timeout Period (sec)	Time Delay (sec)
1.00 (default)	0	0	0
MaxPFAval (overexcited)	0	60	420
MaxPFAval (underexcited)	0	90	0
0.5+MaxPFAval/2 (overexcited)	10	0	0
0.5+MaxPFAval/2 (underexcited)	0	0	480

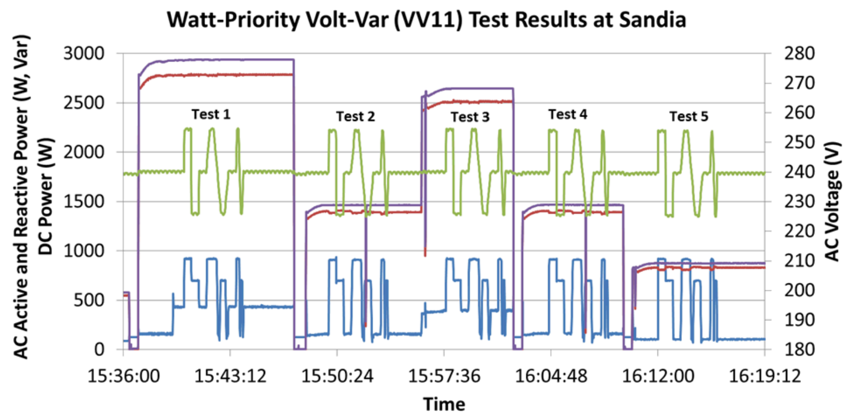


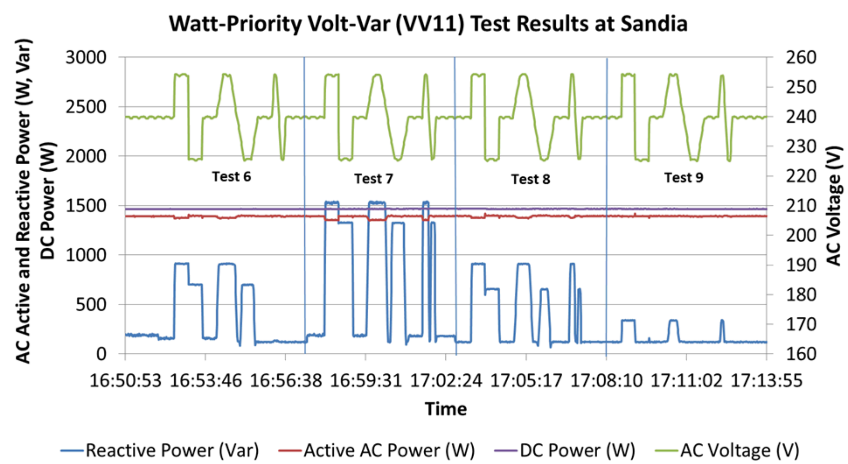


VV11 – Watt-priority volt-var

New VV11 Test Matrix.

Test	EUT Initial Operating State	Volt/Var Initiation	Volt/Var [V,Q] Array				Requested Ramp Time (% VArAval/s)	Time Window (seconds)	Timeout Period (seconds)
1	100% rated power, unity power factor	Binary, 1	V1	97	Q1	50	0	0	0
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
2	50% rated power, unity power factor	Binary, 1	V1	97	Q1	50	0	0	0
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
3	90% rated power, unity power factor	Binary, 1	V1	97	Q1	50	10	0	0
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
4	50% rated power, unity power factor	Binary, 1	V1	97	Q1	50	0	0	0
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
5	30% rated power, unity power factor	Binary, 1	V1	97	Q1	50	0	0	300
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
6	50% rated power, unity power factor	Binary, 1	V1	97	Q1	50	10	0	180
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-50			
7	50% rated power, unity power factor	Binary, 1	V1	97	Q1	100	0	0	0
			V2	99	Q2	0			
			V3	101	Q3	0			
			V4	103	Q4	-100			
8	50% rated power, unity power factor	Binary, 1	V1	95	Q1	50	0	120	0
			V2	98	Q2	0			
			V3	102	Q3	0			
			V4	105	Q4	-50			
9	50% rated power, unity power factor	Binary, 1	V1	80	Q1	100	0	0	0
			V2	95	Q2	0			
			V3	105	Q3	0			
			V4	120	Q4	-100			





SunSpec Alliance Specifications

- SunSpec has released a number of specifications format for Modbus registers on SunSpec-compliant devices:
  - Common Elements v1.4
  - Inverter Models v1.1
  - String Combiner Model v1.1.1
  - Inverter Controls DRAFT 10
  - ….

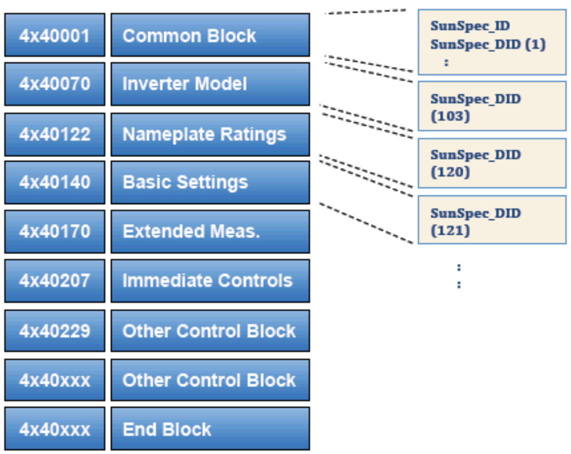


Figure 1: Chained Inverter Control Blocks

Automating Certification Tests

- Using the SunSpec Modbus mapping and python scripting tool, the interoperability test procedure can be quickly executed on any SunSpec-compliant device.

INV1 Main Program

```python
if __name__ == "__main__":

    prompt_resp = raw_input('Do you want to run the script with prompts for connect/disconnect (y/n)?')

    print 'Talking to the inverter... hold on a sec'
    f = client.SunSpecClientDevice(client.RTU, 1, 'COM3')

    print f.models
    print f.controls

    for model in f.models:
        print model, '\n\n'
        print f[model]

    # Test 1
    #parameters: time_window, timeout_period, test_num, prompt_resp
    disconnect(0, 0, 1, prompt_resp)
    time.sleep(5)
    verify_disconnect(50)

    # Test 2
    #parameters: time_window, timeout_period, test_num, prompt_resp
    connect(0, 0, 2, prompt_resp)
    time.sleep(25)
    verify_connect(50)

    # Test 3
    #parameters: time_window, timeout_period, test_num, prompt_resp
    disconnect(0, 0, 3, prompt_resp)
    time.sleep(5)
    verify_disconnect(50)

    # Test 4
    #parameters: time_window, timeout_period, test_num, prompt_resp
    connect(0, 0, 4, prompt_resp)
    time.sleep(25)
    verify_connect(50)

    # Test 5
    #parameters: time_window, timeout_period, test_num, prompt_resp
    disconnect(90, 30, 5, prompt_resp)
    # timeout is 30 seconds
    still_on = True
    count = 0
    print 'Randomization window in use. Wait up to 90 seconds for disconnect'
    while still_on == True:
        time.sleep(1)
        count += 1
        f.controls.read()
        if f.controls.Conn == 0:
            still_on = False
            print 'Inverter disconnected after %d seconds' % (count)
            print 'Please wait 30 seconds for the revert time to reconnect the inverter'
    time.sleep(55)
    verify_connect(50)

    # Test 6
    #parameters: time_window, timeout_period, test_num, prompt_resp
    disconnect(60, 0, 6, prompt_resp)
    still_on = True
    count = 0
    print 'Randomization window in use. Wait up to 60 seconds for disconnect'
    while still_on == True:
        time.sleep(1)
        count += 1
        f.controls.read()
        if f.controls.Conn == 0:
            still_on = False
            print 'Inverter disconnected after %d seconds' % (count)
    time.sleep(5)
    verify_disconnect(50)

    # Test 7
    #parameters: time_window, timeout_period, test_num, prompt_resp
    connect(60, 0, 7, prompt_resp)
    still_off = True
    count = 0
    print 'Randomization window in use. Wait up to 60 seconds for connect'
    while still_off == True:
        time.sleep(1)
        count += 1
        f.controls.read()
        if f.controls.Conn == 1:
            still_off = False
            print 'Inverter is in the process of connecting after %d seconds' % (count)
    time.sleep(25)
    verify_connect(50)
```

Parameters from Sandia Test Protocols INV1 Test Matrix

Main Program Runs through the Test Procedure

```python
# Test 1
#parameters: time_window, timeout_period, test_num, prompt_resp
disconnect(0, 0, 1, prompt_resp)
time.sleep(5)
verify_disconnect(50)

# Test 2
#parameters: time_window, timeout_period, test_num, prompt_resp
connect(0, 0, 2, prompt_resp)
time.sleep(25)
verify_connect(50)
```

Disconnect Function

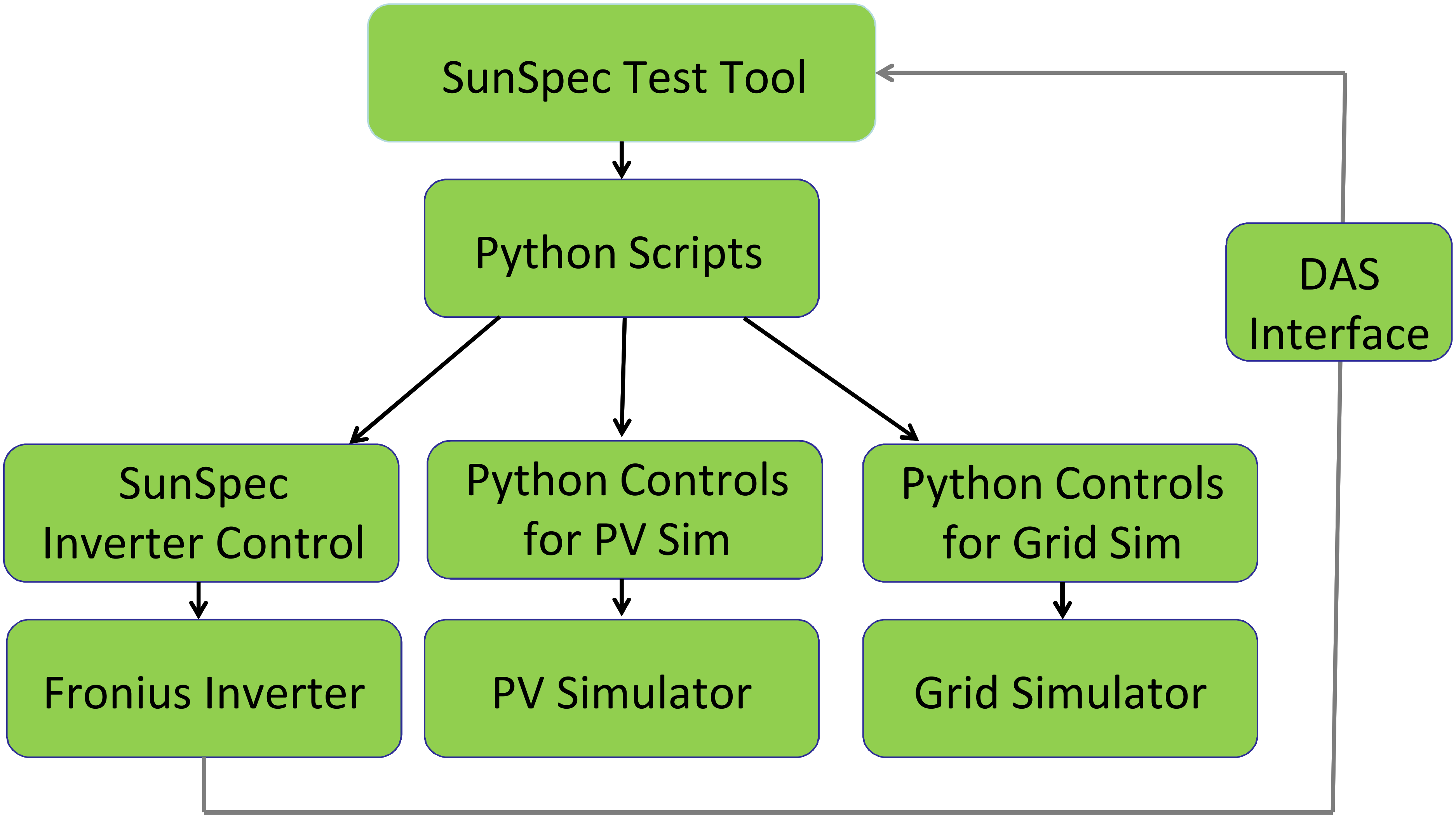
```python
def disconnect(time_window, timeout_period, test_num, prompt_resp):
    #Put EUT in connected state - Initialize test
    f.controls.read()
    if f.controls.Conn == 0:
        f.controls.Conn = 1
        f.controls.write()
        time.sleep(25)

    # Request status from EUT and display connection status
    f.controls.read()
    con_stat = f.controls.Conn
    print 'Current connection status is ', con_stat

    #Inverter output is measured and logged
    f.inverter.read()
    print 'Current power output is ', f.inverter.W

    #UMS issues a disconnect command
    if prompt_resp == 'y':
        prompt("\nExecute disconnect? (Test number is %s)" % str(test_num), prompt_resp)
    else:
        print "\nExecuting disconnect (Test number is %s)" % str(test_num)
    f.controls.Conn_WinTms = time_window
    f.controls.Conn_RvtTms = timeout_period
    f.controls.write()
    f.controls.Conn = 0
    f.controls.write()
```

Sandia Control Architecture



Future work: advanced ESS test protocols

- Sandia and the SIRFN group would like to start shifting toward testing DER functionality on more than just PV inverters.
- Some IEC TR 61850-90-7 functions are specifically designed for energy storage systems. SIRFN will need battery or other controllable ESS in order to evaluate them.
  - INV4 – Request Active Power from Storage
  - INV5 –Signal for Charge/Discharge Action
  - FW22 - limits active power generation and charging in the case of a storage system
  - VW52 - limit energy storage charging rates in response to low grid voltage.
- Sandia is partnering with a SunSpec-compliant power conditioning system designer to start advanced ESS functionality testing with their 3-port design this summer.

Advanced Inverter Collaborations

- Sandia and the Korea Electrotechnology Research Institute (KERI) are comparing test results from PV inverters and ESS.
- Within the SIRFN group
  - SNL, NREL, AIT, RSE are testing residential inverters from the same manufacturer with the Sandia Test Protocols, in order to:
    - exchange technical information on the design and operation of the advanced DER test beds and software for executing interoperability testing
    - provide refinements to the test procedures and test parameters
    - compare results for different DER sizes and designs on grids with different frequencies and voltages
  - SNL, RSE, and AIST will begin testing ESS with the test protocols this year.
  - Sig Gonzalez (Sandia) visited NREL in May 2013 to begin advanced inverter function testing with a domestic company's 500 kW inverter.
- SIRFN Collaborations with other groups
  - SIRFN is updating NIST SGIP (Smart Grid Interoperability Panel) on implementation of test protocols for advanced inverter functions.
  - Collaborating with the California Public Utilities Commission (CPUC) to determine which functions will be required for California's new Rule 21.
  - UL was granted legal permission to use the Sandia Test Protocols to update UL 1741. Similar conversations have started with the IEC Smart Grid Strategic Group.
  - PV manufacturers, CA utilities, SunSpec Alliance, EPRI and Sandia are working on the CSI grant to create translators and create a CA-specific set of test protocols.
  - SNL and SunSpec/Loggerware are creating the inverter control test tool for automating the protocols.

Sandia and AIT collaboration

- Sandia and AIT submitted an abstract regarding advanced inverter testing to EU PVSEC (Amsterdam, 22-26 Sept.)
- The paper:
  - Discusses experience exercising the Sandia Test Protocols and modifications to the test procedures/test matrices (parameters) based on the experiments.
  - Compares the different laboratory test beds (hardware and software).
  - Methods for automating the test protocols
    - Sandia is using python and SunSpec controllers
    - AIT is using Matlab controllers
  - Comparison of results for four (or more) functions
    - Currently INV1, INV2, INV3, VV11

California Solar Initiative (CSI) Proposal

- Sandia partnered with EPRI, SMA, Fronius, SCE, SMUD, and SunSpec on the "Standard Communication Interface and Certification Test Program" to develop communications specifications for the utility-to-DER interchange over Modbus, SEP 2.0, and Zigbee gateways.
- Sandia portion of the grant will:
  - Create test protocols for the certification/conformance of CA Rule 21 inverter control and interoperability requirements.
  - Address cybersecurity concerns by establishing the underlying rules for the utility-to-DER interoperability.