Field of View Test Bench

Table of Contents

1.0 Purpose

The purpose of the Field of View Test Bench is to provide the designer with information about what the driver and vehicle commander can see through the periscopes.

2.0 Procedures

2.1 Installation

Initial installations will be provided with the installation of the CyPhy tool suite. Future version may be packaged as a standalone or combined package for test benches.

2.2 Tool

The Field of View Test Bench in GME allows the user to interface with the Field of View Tool which is a software application that accepts a design, analyzes it and returns a list of vision parameters for the driver and vehicle commander.

Test Bench: Field of View

Requirements tested

- **Horizon percentage:** Percentage of the horizon viewable to the crew
- **Max Uplook:** The highest point visible in front of the vehicle.
- **Closest Visible Point:** the closest point viewable by the crew in front of and behind the vehicle.

Theory of Operation

The system (design) is assembled into a 3D CAD representation with the customization / generation of parameterized components. The data is analyzed to determine if the crew can interface with the periscopes, and through the periscopes what is visible.

Test Bench Structure

The test bench contains a system under test that is assembled and analyzed for the transportability parameters.

Step 1

In the GME Browser, insert a new Test Bench subfolder (“Field of View”) within the “Testing” Test Bench folder.

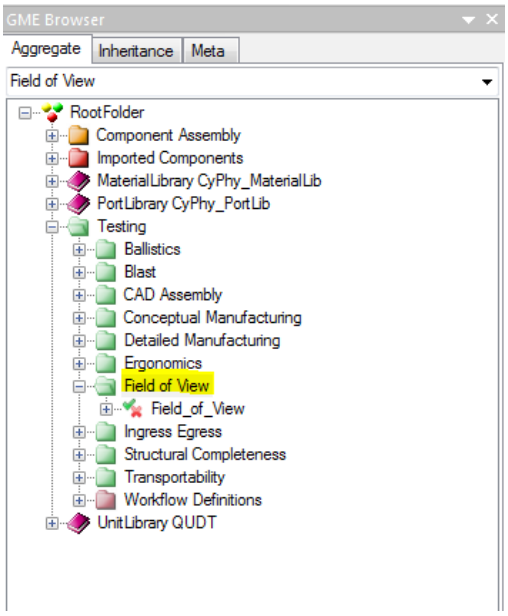


Figure 1: Insert Field of View subfolder into testing

An assembly now needs to be added to the test bench. In the “Field of View” test bench Copy/Paste…As Reference the assembly to be tested.

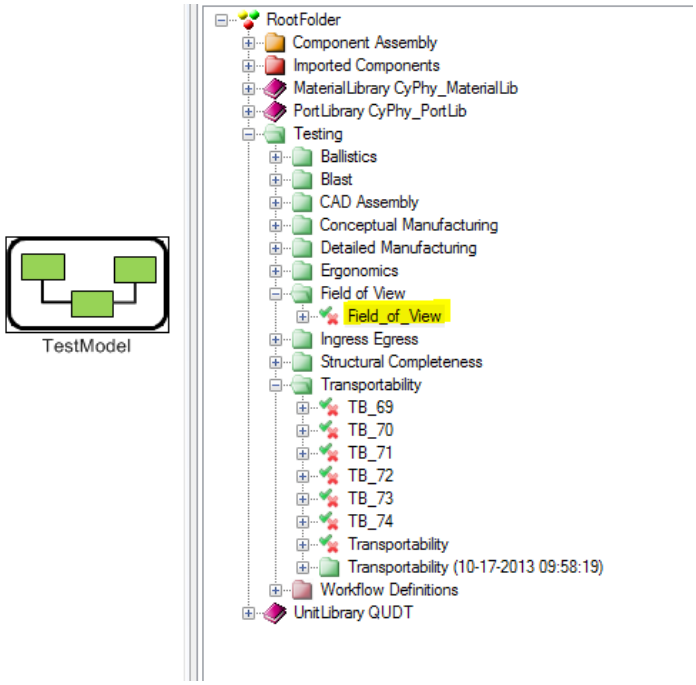


Figure 2: Copy/Paste...As Reference

Step 2

In GME, within the Workflow Definition subfolder create a new workflow model named “FOV”.

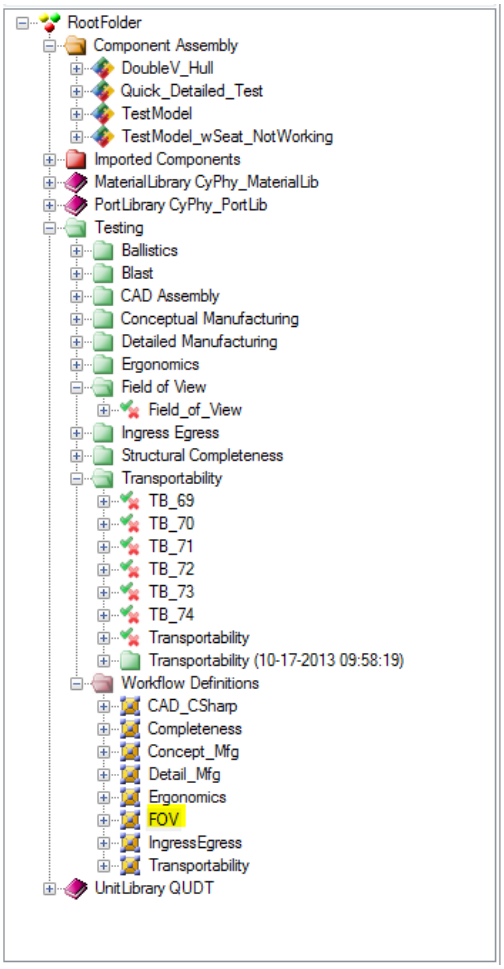


Figure 3: Workflow definition

Open the “FOV” Workflow Model and drag a “Task” element into the workspace. Select “CyPhyCADAnalysis” as the interpreter from the window that pops up.

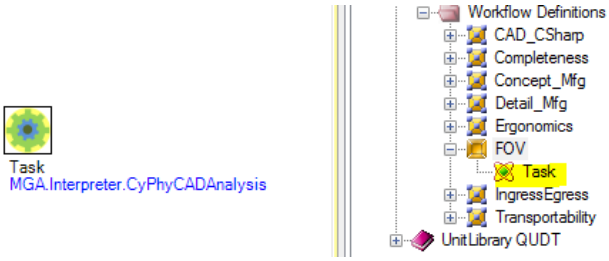


Figure 4: Add a task to the FOV Workflow

Double click the newly created task and select “field_of_view” as the analysis tool. Set the Workflow Parameters as shown in Figure 5.

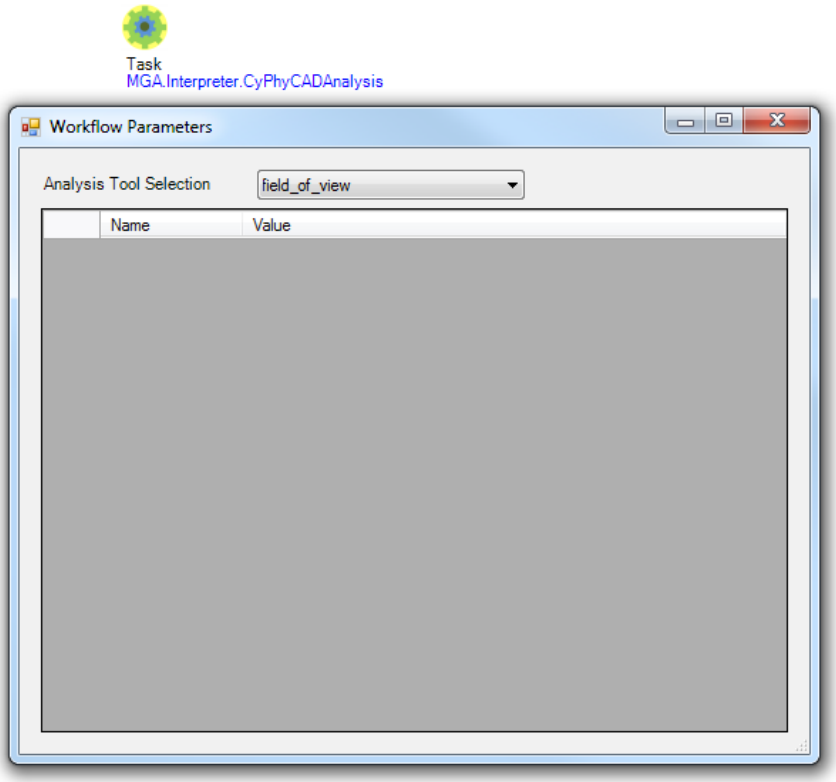


Figure 5: Setting Workflow parameters for the new Task

Open the “Field of View” test bench drag and drop the “FOV” workflow definition and 12 metrics. Metrics can be dragged and dropped from the Part Browser.

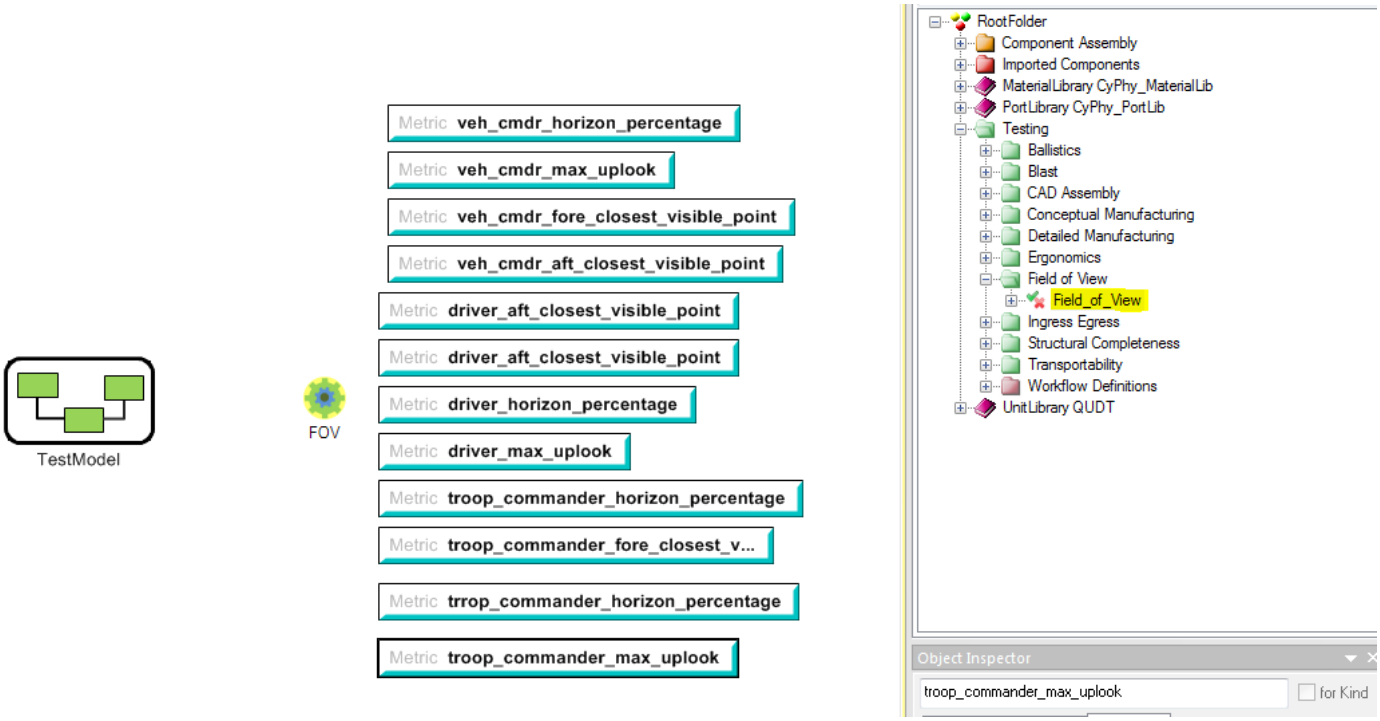


Figure 6: Field of View Test Bench

Step 3

To exercise the test bench, run the Master Interpreter (highlighted in task bar). Check the “Post to META Job Manager” and “Use Project Manifest” boxes as well as the “AP203_E2_Single_File” and “AP203_E2_Separate_Part_Files” as shown in Figure 7. Also point to the Creo file location for the “CAD Auxiliary Directory:” field.

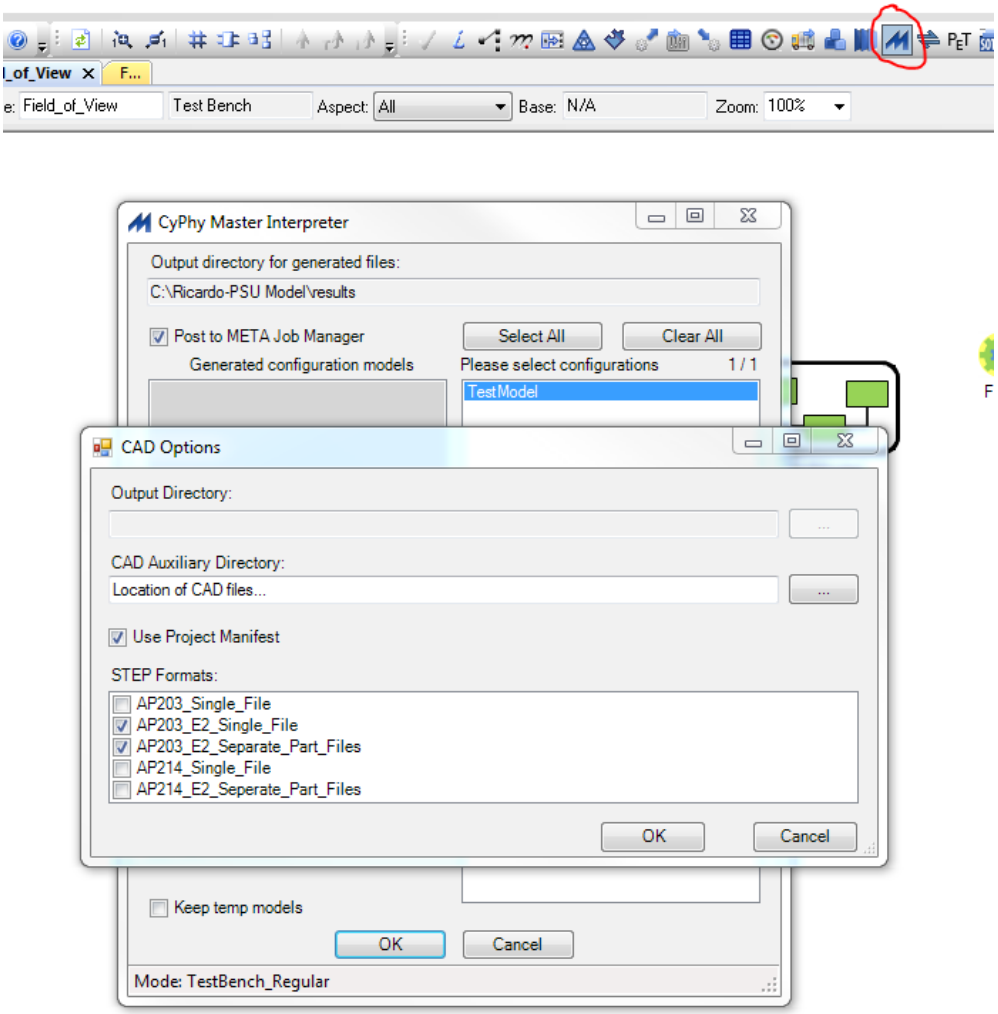


Figure 7: Running the Field of View test bench

The test bench will create a results folder and then run. To access the results folder right click the job in Job Manager and choose “Show in explorer”.

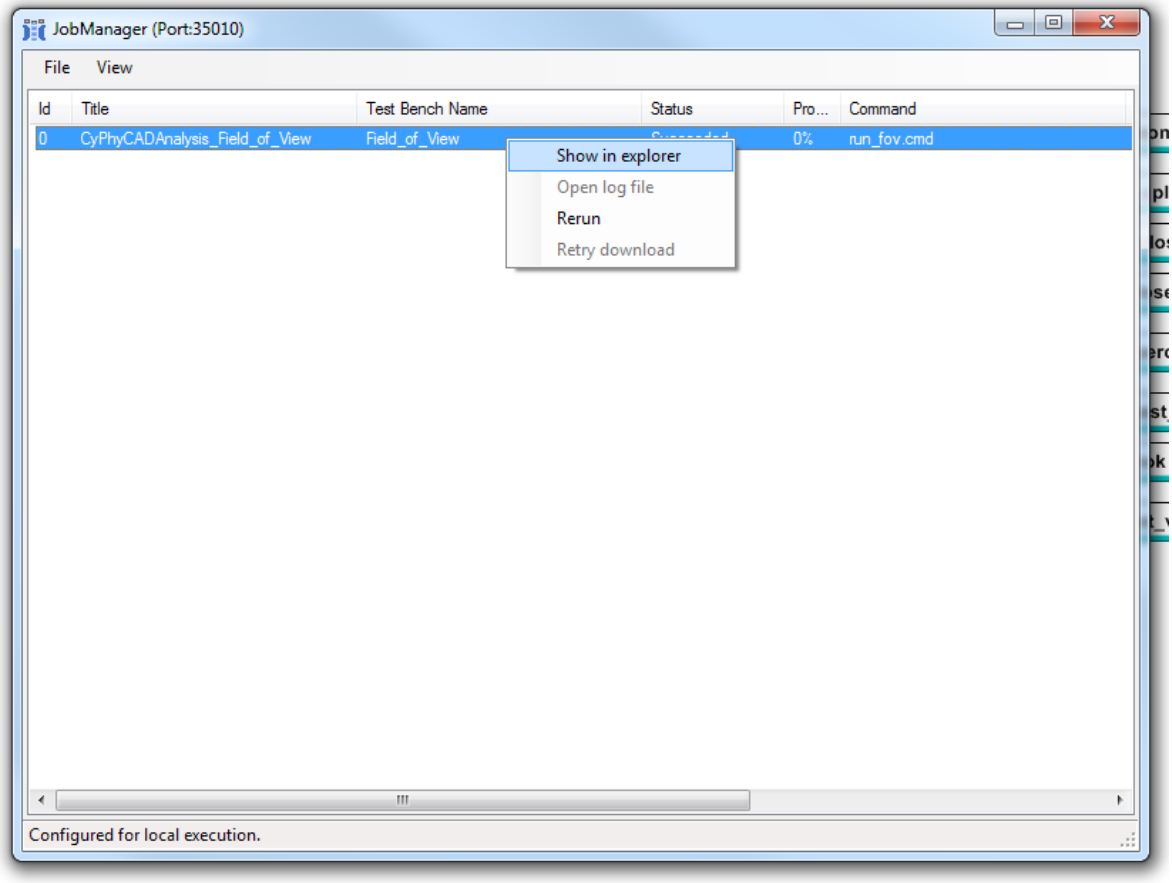


Figure 8: Accessing the test bench result folder

Description

Field of View analysis, finds the position of the crew and periscopes, assigns the periscopes to individual crew members, and then determines what is visible through the device for that crew member.

The System Under Test is assembled in CREO and then each component making up the system is saved as an individual step file. Data about procurement for each component is also gathered. This information is packaged and analyzed by the FOV Tool as a post-processing step.

Results are returned in two files. “summary.testresults.json” is a summary of the test bench status and results. “output.json” is just the results that the test bench calculated.

Metrics

Role	Test Bench #	Metric	Description
Vehicle Commander	23	veh_cmdr_horizon_percentage	The percentage of the horizon that can be seen by the vehicle commander.
	25	veh_cmdr_max_up_look	The highest point in front of the vehicle the vehicle commander can see.
	24.1	veh_cmdr_fore_closest_point	The closest point in front of the vehicle that the vehicle commander can see.
	24.2	veh_cmdr_aft_closest_point	The closes point behind the vehicle that the vehicle commander can see.
Driver	19	driver_horizon_percentage	The percentage of the horizon that can be seen by the driver.
	22	driver_max_up_look	The highest point in front of the vehicle the driver can see.
	20.1	driver_fore_closest_point	The closest point in front of the vehicle that the driver can see.
	20.2	driver_aft_closest_point	The closest point behind the vehicle that the driver can see.
Troop Commander	28	troop_commander_horizon_percentage	The percentage of the horizon that can be seen by the troop commander.
	30	troop_commander_max_up_look	The highest point in front of the vehicle the troop commander can see.
	29.1	troop_commander_fore_closest_point	The closest point in front of the vehicle that the troop commander can see.

31	troop_commander_aft_closest_point	The closes point behind the vehicle that the troop commander can see.

Table 1: FoV Metrics by vehicle role

Required Connection to System Under Test

NONE

Outputs

Text

The output of the test bench is in two files: “summary.testresults.json” and “output.json”.


```
"AnalysisStatus": "OK",
"TestBench": "Field_of_View",
"DesignName": "Field_of_View",
"Metrics": [
    {
        "Name": "driver_aft_closest_visible_point",
        "DisplayedName": null,
        "GMEID": "id-0067-00035793",
        "Value": "2000000.0",
        "ID": "70108124-7c33-4ce4-88af-b26da6449a91",
        "Unit": ""
    },
    {
        "Name": "driver_fore_closest_visible_point",
        "DisplayedName": null,
        "GMEID": "id-0067-00035792",
        "Value": "21.4502215385",
        "ID": "96dcc2db-37bf-48b7-bf30-6e121e98b748",
        "Unit": ""
    },
    {
        "Name": "veh_cmdr_aft_closest_visible_point",
        "DisplayedName": null,
        "GMEID": "id-0067-00035795",
        "Value": "2000000.0",
        "ID": "2508e49d-686c-456a-b2b5-0146e6bcb58c",
        "Unit": ""
    },
    {
        "Name": "driver_horizon_percentage",
        "DisplayedName": null,
        "GMEID": "id-0067-00035794",
        "Value": "0.638888888889",
        "ID": "e2d7bbe7-272c-4cf8-9859-de59c6515398",
        "Unit": ""
```


Figure 9: Summary Results sample


```
{
    "veh_cmdr_horizon_percentage": 0.5805555555555538,
    "veh_cmdr_max_uplook": 17.2607421875,
    "veh_cmdr_aft_closest_visible_point": 2000000.0,
    "driver_aft_closest_visible_point": 2000000.0,
    "driver_horizon_percentage": 0.5791666666666649,
    "veh_cmdr_fore_closest_visible_point": 48.231333494186401,
    "driver_fore_closest_visible_point": 48.218294978141785,
    "driver_max_uplook": 17.2607421875
}
```

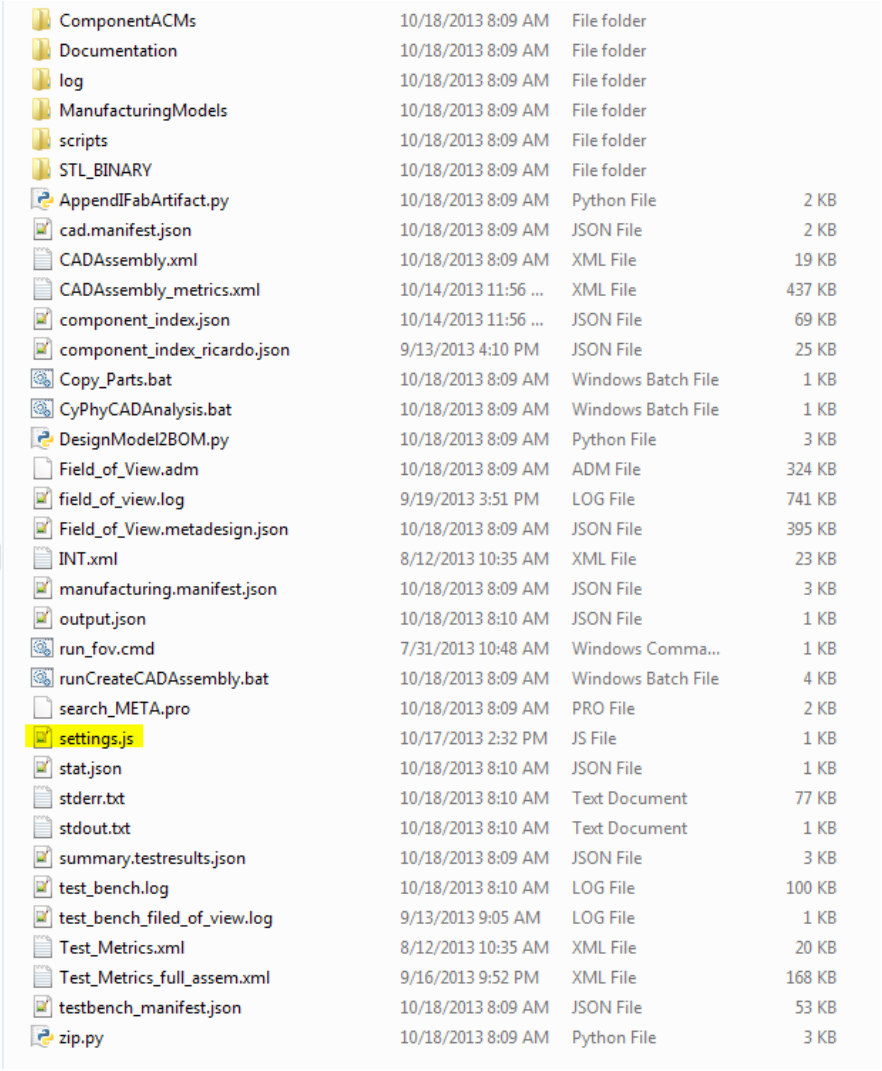

Figure 10: Field of View output.json

Visualization

A visual representation of the Field of View is also available.

First, Mayavi needs to be installed to run with your python distribution. Mayavi can be found at http://docs.enthought.com/mayavi/mayavi/installation.html.

Second, “show_3d” in the “settings.js” file needs to be changed to “true”. The “settings.js” file can be found in the results folder created when the test bench was run (see Figure 9).



ComponentACMs	10/18/2013 8:09 AM	File folder	
Documentation	10/18/2013 8:09 AM	File folder	
log	10/18/2013 8:09 AM	File folder	
ManufacturingModels	10/18/2013 8:09 AM	File folder	
scripts	10/18/2013 8:09 AM	File folder	
STL_BINARY	10/18/2013 8:09 AM	File folder	
AppendlFabArtifact.py	10/18/2013 8:09 AM	Python File	2 KB
cad.manifest.json	10/18/2013 8:09 AM	JSON File	2 KB
CADAssembly.xml	10/18/2013 8:09 AM	XML File	19 KB
CADAssembly_metrics.xml	10/14/2013 11:56 ...	XML File	437 KB
component_index.json	10/14/2013 11:56 ...	JSON File	69 KB
component_index_ricardo.json	9/13/2013 4:10 PM	JSON File	25 KB
Copy_Parts.bat	10/18/2013 8:09 AM	Windows Batch File	1 KB
CyPhyCADAnalysis.bat	10/18/2013 8:09 AM	Windows Batch File	1 KB
DesignModel2BOM.py	10/18/2013 8:09 AM	Python File	3 KB
Field_of_View.adm	10/18/2013 8:09 AM	ADM File	324 KB
field_of_view.log	9/19/2013 3:51 PM	LOG File	741 KB
Field_of_View.metadesign.json	10/18/2013 8:09 AM	JSON File	395 KB
INT.xml	8/12/2013 10:35 AM	XML File	23 KB
manufacturing.manifest.json	10/18/2013 8:09 AM	JSON File	3 KB
output.json	10/18/2013 8:10 AM	JSON File	1 KB
run_fov.cmd	7/31/2013 10:48 AM	Windows Comma...	1 KB
runCreateCADAssembly.bat	10/18/2013 8:09 AM	Windows Batch File	4 KB
search_META.pro	10/18/2013 8:09 AM	PRO File	2 KB
settings.js	10/17/2013 2:32 PM	JS File	1 KB
stat.json	10/18/2013 8:10 AM	JSON File	1 KB
stderr.txt	10/18/2013 8:10 AM	Text Document	77 KB
stdout.txt	10/18/2013 8:10 AM	Text Document	1 KB
summary.testresults.json	10/18/2013 8:09 AM	JSON File	3 KB
test_bench.log	10/18/2013 8:10 AM	LOG File	100 KB
test_bench_filed_of_view.log	9/13/2013 9:05 AM	LOG File	1 KB
Test_Metrics.xml	8/12/2013 10:35 AM	XML File	20 KB
Test_Metrics_full_assem.xml	9/16/2013 9:52 PM	XML File	168 KB
testbench_manifest.json	10/18/2013 8:09 AM	JSON File	53 KB
zip.py	10/18/2013 8:09 AM	Python File	3 KB

Figure 11: Location of "settings.js" in the results folder

```
{
    "metrics_file" : "CADAssembly_metrics.xml",
    "path_to_instance_xmls" : "ComponentACMs",
    "path_to_instance_stls" : "STL_BINARY",
    "instance_file": "component_index.json",
    "output_json_file" : "output.json",
    "show_3d" : true
}
```

Figure 12: Change "show_3d" to "true"

After editing the “settings.js” file, re-run the FOV Test Bench from the Job Manager.

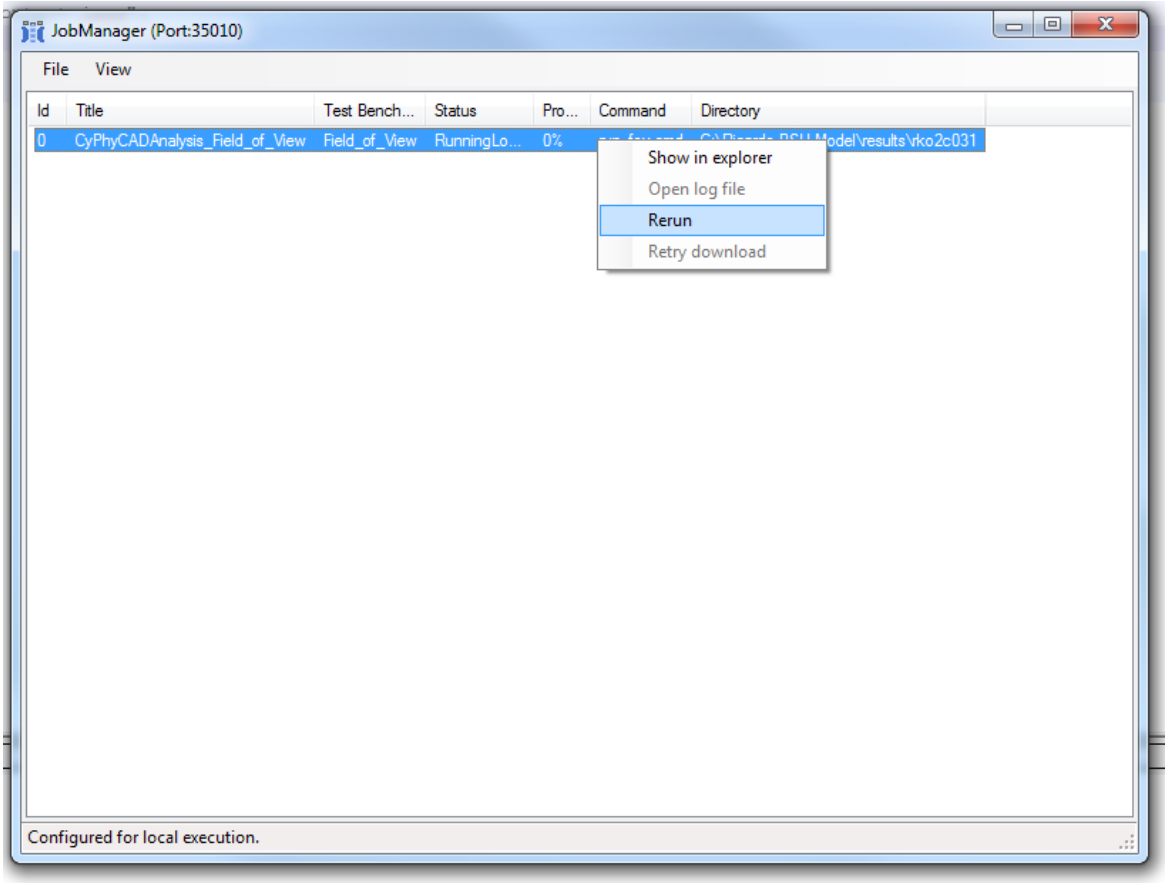


Figure 13: Re-running the FOV Test Bench

When the FOV Test Bench has completed successfully, a new window will be show with a 3D visualization of the design and the field of view.