Summary of projects

By Gaurav Rajendra Bokil (Email-ID)

I joined Altair Engineering in India in February, 2019 and worked till September, 2020. As a part of MotionView and MotionSolve team, I was involved in developing, testing and validation of MotionView multibody dynamics software. My main contributions are summarized below:

Python API

Testing

MotionView software has a Python API that can be used to create complex models, custom tools and leverage python modules to extend model capabilities. I was involved in testing of the API to suggest improvement and check for robustness. I built several scripts to automate testing and achieve maximum code coverage. I performed these tests by:

- comparing the response of application from Python commands with that from GUI
- creating small Python scripts to validate the combination of functionalities,
- negative testing and
- suggesting new features to improve user experience

Creating tutorials

After testing the API, I created an example library to showcase the benefits of using Python in MotionView. The idea was to create simple MBD models using Python, but it would not be interesting enough for the user. So, I decided to formulate a tutorial library starting from a very basic mechanism model and increasing the complexity of the same model in the next tutorial.

By collaborating with teams in the USA and Greece, I arranged the example tutorials in the following way:

- creating a simple parametric gear-pair model using Python script
- creating multiple variants of the model and simulating them
- using Python to perform signal processing on results using NumPy
- creating an input sample space for 3 parameters in the model using PyDOE
- performing a Design of Experiments (DOE) and exporting data to a CSV,
- using this data to create a predictive model using Scikit-learn

Export MotionView model to Python

The goal was to develop a script that will parse the MBD model to export a python file. The code was fundamentally divided into:

- Reader to parse all types of entities and properties in the model,
- Parser to understand, filter and organize the collected information,
- Writer to write the commands in a Python file using information from Parser

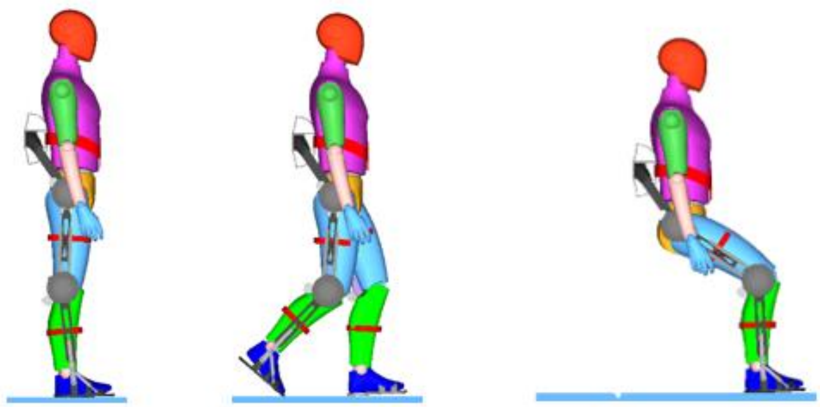
This enabled the team members and external users to take advantage of Python and improve model building, manipulating and post-processing.

Model development and validation

Human body model

The MotionSolve application team in France developed a human body model and I was asked to validate it. I tested the UI and provided usability improvement feedback to the developers.

After testing the human body tool, I also created a working simulation model to demonstrate the usage of the tool and MotionView as a whole. I thought it would be best to use as many functionalities as possible and display multidisciplinary capabilities of the software.



(images of a similar model to protect confidentiality)

So, I designed a simple “chairless-chair exoskeleton supported by spring-dampers and attached to the human body. The aim of the chairless-chair was to support the user’s back in sitting position, reduce the weight on the knees and not resist the user while walking. I used the MOCAP database from Carnegie Mellon university to get walking and sitting motion capture data in AMC file format. In order to use this data, I wrote a translator script in Python to convert the AMC angle versus time data to a MotionView curve data, which was then used to provide walking and sitting motion to joints in the human body model.

The devised problem statement was to optimize the chairless-chair assembly using MotionSolve. By utilizing the optimization wizard in MotionView, I set the objective to minimize torque on knee joints with design variables as the nonlinear spring-damper properties in the exoskeleton and chose Finite Differencing for design sensitivity analysis (DSA).
After optimization, the torque on the user's knees in sitting position was reduced by 97% due to the chairless-chair. The spring-damper properties were smoothened using HyperView and force-displacement graph was generated to validate if the spring-damper properties were sensible to manufacture.

Excavator validation with test data

The motive of this project was to simulate “pit-test” procedures of excavator using HyperWorks products and compare simulation results versus test data. I had to achieve maximum accuracy between results and data so as to prove the capabilities of MotionSolve and thus help the business team in acquiring customers in the heavy engineering industry.

Using the CAD model, I built a multibody model in MotionView. The process of modeling and validation with test results was as follows:

- Create a rigid body model with multiple bodies and connections,
- Simulate the model to check the low-fidelity model,
- Use Hypermesh to mesh Boom, Arm, Bucket and Mainframe,
- Create a CMS flexbody using the meshed components,
- Integrate the flexible bodies with rigid parts and create new connections,
- Perform static simulation to check the high-fidelity model,
- Create loadcases for max pull, side hit, jack up and digging test scenario,
- Simulate the loadcases and extract results at multiple locations on the excavator,
- Compare the results with strain gauge data at those locations.

Overall, *86% correlation* was achieved between MBD simulation results and test data. I also created a PowerPoint presentation to explain the modelling and validation process.

(image of a similar model to protect confidentiality)

After validation, I performed co-simulation with EDEM to model the interaction of the excavator with rocks and simulate picking and dumping operation. This simulation was used for marketing purposes to exhibit co-simulation capabilities of MotionSolve with EDEM.