PyMotorCAD-API cheat sheet

Version: 0.1 (stable)

Launch Motor-CAD using PyMotorCAD

Launch a Motor-CAD instance locally and then exit:

```
import ansys.motorcad.core as pymotorcad

mcad = pymotorcad.MotorCAD()
mcad.load_from_file(r"path/motor_cad.mot")
# Load predefined template
mcad.load_template(template_name)
# saving file to working_folder
mcad.save_to_file(
    os.path.join(working_folder, mcad_name + ".mot")
)
# Exit Motor-CAD
mcad.quit()
```

Geometry properties

Specify the motor geometry:

```
mcad.set_variable("Slot_Number", 24)
mcad.set_variable("Tooth_Width", 6)
mcad.set_variable("Magnet_Thickness", 4.5)
mcad.set_variable("Pole_Number", 4)
```

Set winding patterns:

```
mcad.set_variable("MagWindingType", 1)
set_variable("MagTurnsConductor", 12)
```

Material assignment

Set stater and rotor lamination materials:

```
mcad.set_component_material("Stator Lam (Back Iron)",
    "M250-35A")
mcad.set_component_material("Rotor Lam (Back Iron)", "
    M250-35A")
```

Model motor

Set building commands:

```
# Build options
mcad.set_variable("ModelType_MotorLAB", 1)
mcad.set_variable("SatModelPoints_MotorLAB", 0)
mcad.set_variable("LossModel_Lab", 0)
mcad.set_variable("ModelBuildSpeed_MotorLAB", 10000)
mcad.set_variable("MaxModelCurrent_MotorLAB", 480)
mcad.set_variable("BuildSatModel_MotorLAB", True)

# Set lab context and build model
mcad.set_motorlab_context()
mcad.clear_model_build_lab()
mcad.build_model_lab()
```

E-magnetic performance curves in LAB

Set lab operating mode:

```
mcad.set_variable("OperatingMode_Lab", 0)
# setting magnetic calculation options
mcad.set_variable("EmagneticCalcType_Lab", 0)
mcad.set_variable("SpeedMax_MotorLAB", 10000)
mcad.set_variable("Speedinc_MotorLAB", 250)
mcad.set_variable("SpeedMin_MotorLAB", 500)
mcad.set_variable("Imax_MotorLAB", 480)
```

Calculate the E-Magnetic performance:

```
mcad.calculate_magnetic_lab()
```

Operating point calculation

Set the operating point variables:

```
mcad.set_variable("OpPointSpec_MotorLAB", 1)
mcad.set_variable("StatorCurrentDemand_Lab", 480)
mcad.set_variable("SpeedDemand_MotorLAB", 4000)
mcad.set_variable("LabThermalCoupling", 0)
mcad.set_variable("LabMagneticCoupling", 0)
```

Calculate the operating point:

```
mcad.calculate_operating_point_lab()
```

Electromagnetic calculations in Emag

Perform electromagnetic calculations:

```
# Set the torque calculation options
points_per_cycle = 60
number_cycles = 1
mcad.set_variable("TorquePointsPerCycle",
    points_per_cycle)
mcad.set_variable("TorqueNumberCycles", number_cycles)
# Disable all performance tests except the ones for
    transient torque
mcad.set_variable("BackEMFCalculation", False)
mcad.set_variable("CoggingTorqueCalculation", False)
# Enable transient torque
mcad.set_variable("TorqueCalculation", True)
# Run Emag performance tests
mcad.do_magnetic_calculation()
```

Results

```
# Get the transient torque waveform in Emag
for n in range(points_per_cycle):
    (x, y) = mcad.get_magnetic_graph_point("TorqueVW",
        n)
    rotor_position.append(x)
    torque_.append(y)
# Calculate line voltage
line_voltage = mcad.get_variable("PeakLineLineVoltage"
    )
# Retrieve lab performance curves
data = io.loadmat(
    os.path.join(
        working_folder, mcad_name, "Lab", "
            MotorLAB_elecdata.mat"
    )
)
speed = data["Speed"]
shaft_torque = data["Shaft_Torque"]
shaft_power = data["Shaft_Power"]
```

PyMotorCAD scripting in MATLAB

The Python version that MATLAB is using has the *ansys.motorcad.core* package installed. PyMotorCAD is available to use in MATLAB.
Import the *ansys.motorcad.core* Python package for use in MATLAB:

```
pymotorcad = py.importlib.import_module("ansys.
    motorcad.core")
```

Backwards compatibility

Use ActiveX scripts to connect to Motor-CAD:

```
import win32com.client

mcad = win32com.client.Dispatch("MotorCAD.
    AppAutomation")
```

Use PyMotorCAD to connect to Motor-CAD:

```
import ansys.motorcad.core as pymotorcad

mcad = pymotorcad.MotorCADCompatibility()
```

References from PyMotorCAD documentation

- Getting started
- Examples
- API reference