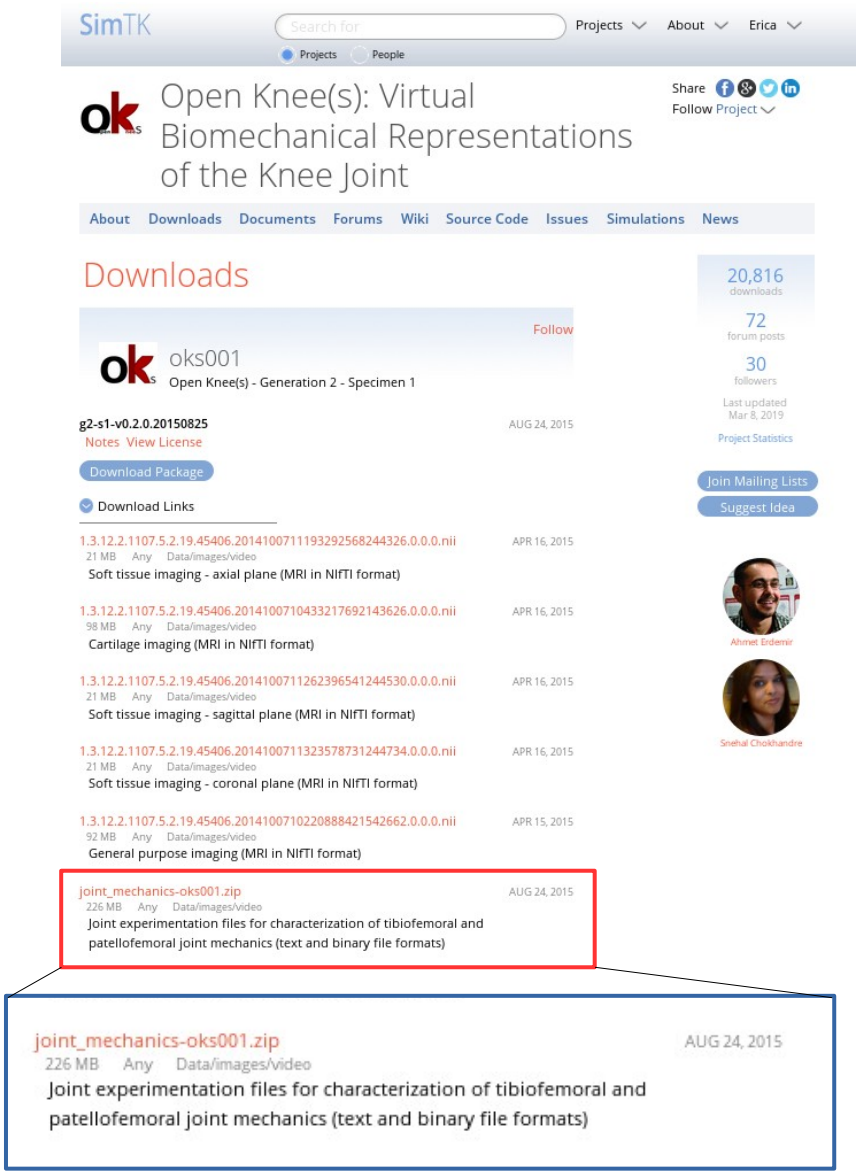
OpenKnee Data Navigation - Registration

Rici Neumann
April 23, 2019

Download Data

- Joint mechanics
- Registration marker STLs
- Setup knee directory
  - geometry-oks0XX: contains bone STLs built from segmentation (in repository)
  - joint_mechanics-oks003: contains all joint mechanics data (downloads section of SimTk)



Setup for registration

- Make sure there is a file with digitizing order within the main Configuration folder of the tibiofemoral joint testing directory (ReadMe_Notes.txt)
- Create registration XML
  - Root directory of knee data
  - Left or Right?
  - Bone STLs
  - Sphere STLs

```
MRI Spheres

Tibia:
Sphere 1 = Lateral
Sphere 2 = Medial
Sphere 3 = Posterior

Femur:
Sphere 1 = Lateral
Sphere 2 = Medial
Sphere 3 = Posterior
```

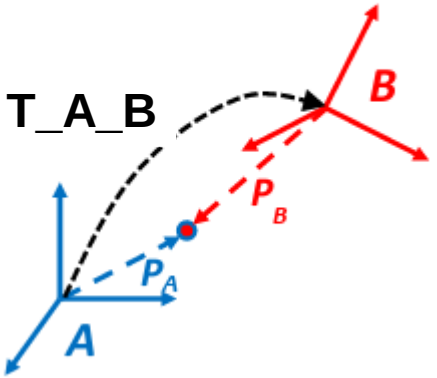
```
<Registration_Input>
    <Knee_directory>/home/morrile2/Documents/OpenKnee/oks001</Knee_directory>
    <Knee_side>Right</Knee_side>
    <Bones>
        <Femur>
            <file>oks001_MRG_FMB_CPL_RAW_02.stl</file>
        </Femur>
        <Tibia>
            <file>oks001_MRC_TBB_CPL_RAW_02.stl</file>
        </Tibia>
        <Fibula>
            <file></file>
        </Fibula>
        <Patella>
            <file>oks001_MRG_PTB_CPL_RAW_02.stl</file>
        </Patella>
    </Bones>
    <Registration>
        <FMR>
            <Posterior>oks001_MRG_FMR-P_RAW_02.stl</Posterior>
            <Lateral>oks001_MRG_FMR-L_RAW_02.stl</Lateral>
            <Medial>oks001_MRG_FMR-M_RAW_02.stl</Medial>
        </FMR>
        <TBR>
            <Posterior>oks001_MRG_TBR-P_RAW_02.stl</Posterior>
            <Lateral>oks001_MRG_TBR-L_RAW_02.stl</Lateral>
            <Medial>oks001_MRG_TBR-M_RAW_02.stl</Medial>
        </TBR>
        <PTR>
            <Superior>oks001_MRG_PTR-S_RAW_01.stl</Superior>
            <Lateral>oks001_MRG_PTR-L_RAW_01.stl</Lateral>
            <Medial>oks001_MRG_PTR-M_RAW_01.stl</Medial>
        </PTR>
    </Registration>
</Registration_Input>
```

Coordinate Systems

$$P_B = (T\_A\_B)^{-1} \cdot P_A$$

Transformation of points from A CS to B CS – T_A_B.

Transform B coordinates in the A coordinates.



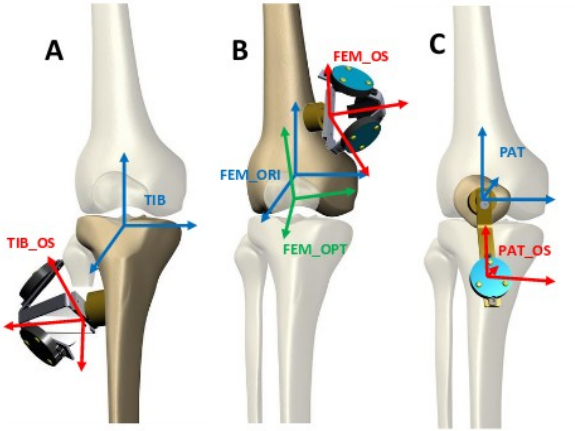


Figure 3: Coordinate Systems for A) tibia, B) femur and C) patella

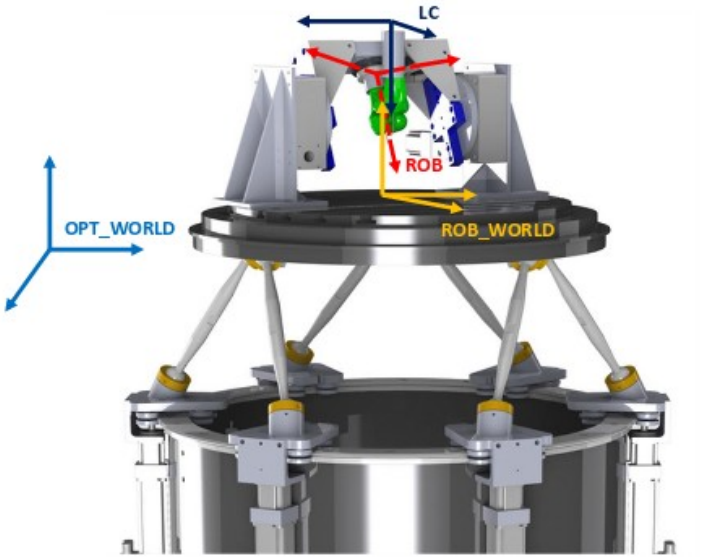


Figure 1: Coordinate Systems for Tibiofemoral testing

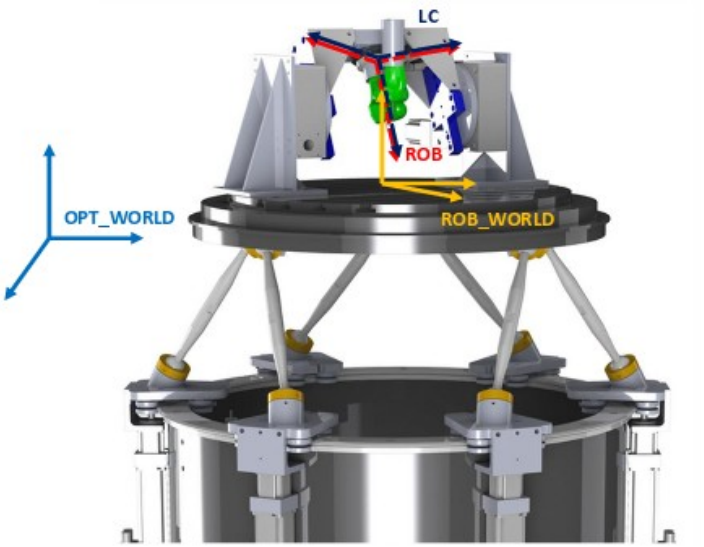


Figure 2: Coordinate Systems for Patellofemoral Testing (Note: During patellofemoral testing, the tibia is mounted to the robot and the femur is mounted to the frame. Also, there are two physical load cells attached to the stage that are not displayed in the image)

https://simtk.org/plugins/moinmoin/openknee/Infrastructure/ExperimentationMechanics?action=AttachFile&do=view&target=2013CB-031-001.A_State+Configuration+File+Explanations.pdf

Registration Script

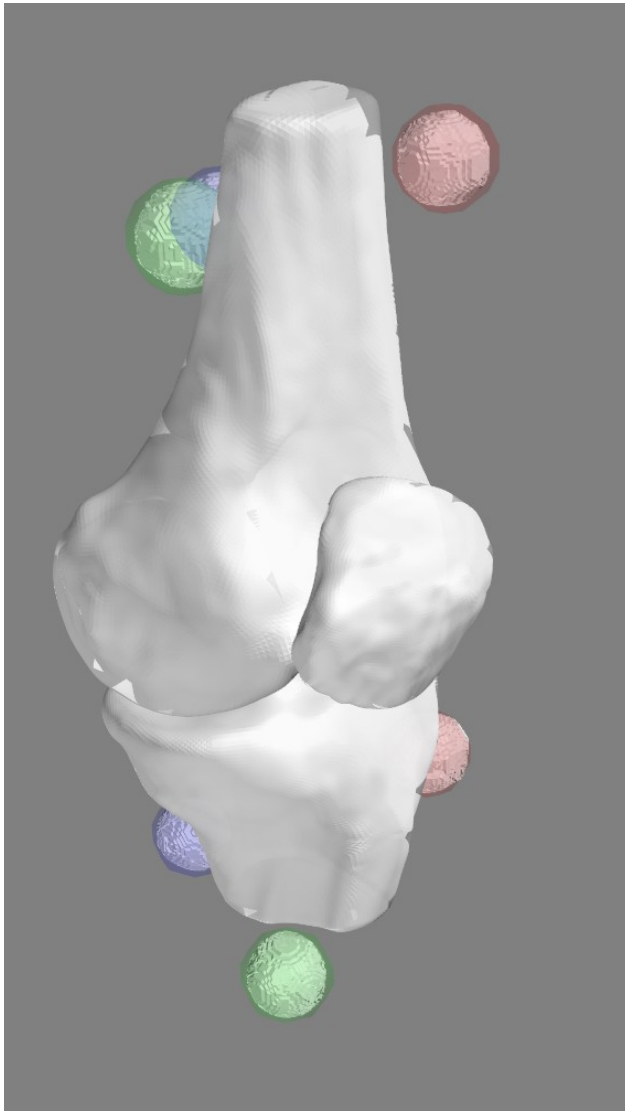
- Change directory to desired OpenKnee specimen
- Run convert_to_image_CS.py <oks0XX_registration_0x.xml> <1>
  - Inputs:
    - Registration XML
    - Boolean: 0 – no visualization, 1 – visualization
  - Outputs:
    - Transformation matrices saved within .npz file for each trial with processed TDMS data.

What is the script doing?

- **Performing patella, tibia, and femur registration using digitized points and segmented spheres captured in MRI.**

Tibiofemoral Registration

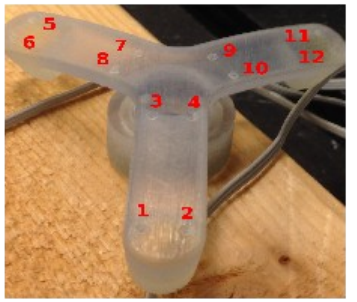
- Completed for each individual TF joint trial
- Transformations directly from the State.cfg file
  - T_TIBOS_TIB
  - T_FEMOS_FEMORI
  - Offset_FEMORI_TIB
- Extract sphere centers
  - Tibia and Femur: 3 spheres per bone
  - Two coordinate systems to relate experiment to image/model coordinate systems
    - Digitized during experimentation
      - Points are collected in the WORLD CS, but position sensor data is also collected for the relevant bone. Transform these points into FEMOS or TIBOS.
      - Fit sphere to digitized points and find the center
    - STLs from MRI segmentation
      - Fit sphere to STL surface to get center (image coordinate system)
  - Singular value decomposition (SVD) to find T_I_TIBOS and T_I_FEMOS

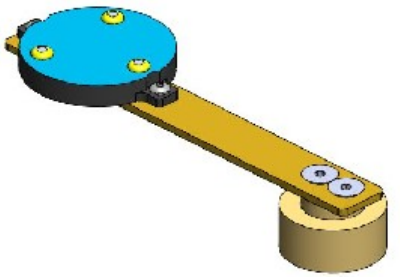


Patella Registration

- Extract divot points that were digitized on the registration marker assembly (State.cfg file of patella registration folder)
  - Points converted to REFOPT CS
  - T_REFOPT_WORLD
- Extract CAD points (CAD_PT_DIMENSIONS.txt)
  - Divots
  - Spheres
- Extract sphere centers (STLs in MRI)
- T_REFOPT_PATOS (Registration marker relative placement.tdms)
  - T_WORLD_PATOS from patella sensor
  - T_WORLD_REFOPT from reference sensor
- T_REFOPT_CAD: SVD of divots in CAD and REFOPT CS
- T_I_CAD: SVD of sphere centers from CAD and MRI
- T_I_PATOS: built from the transformation matrices above.
  - Note: This registration is only performed once for all trials!

Registration marker assembly

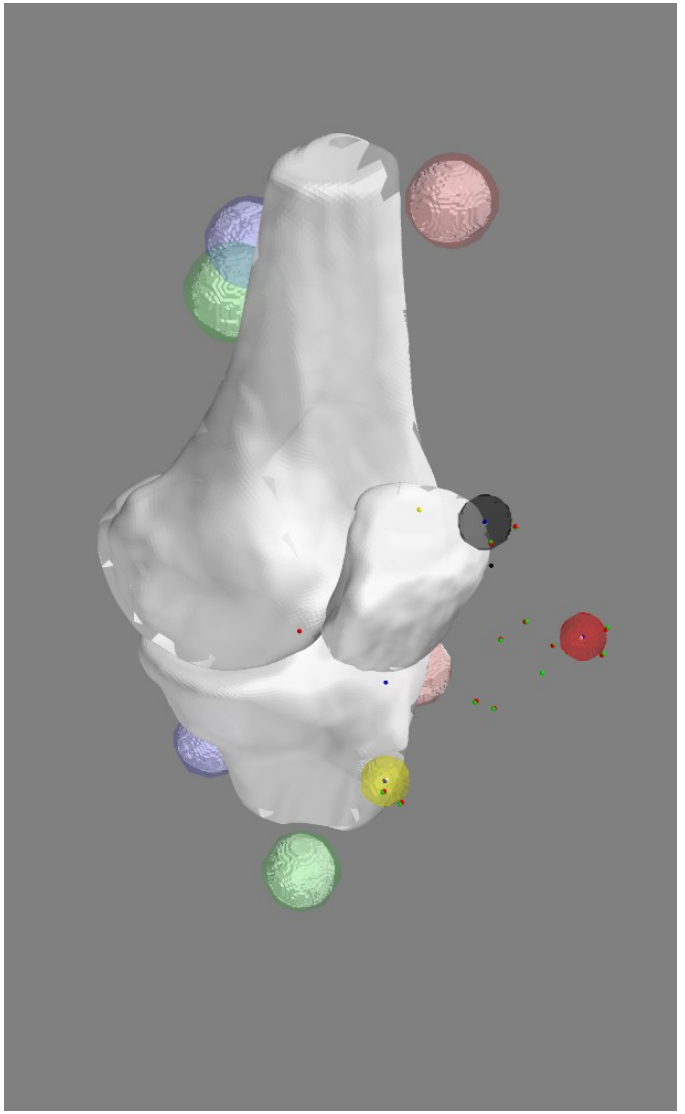




Patella Optotrak sensor

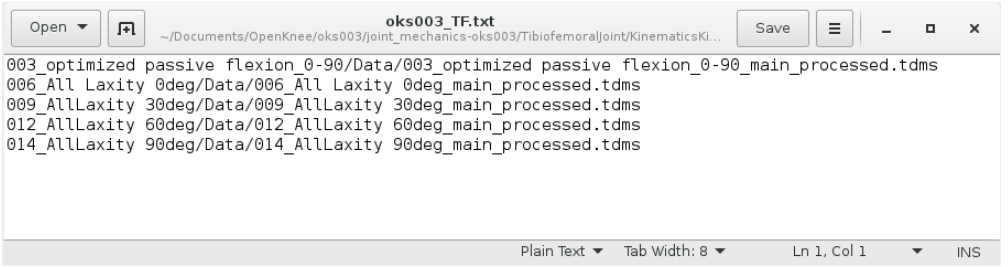
Patellofemoral Registration

- Completed for each individual PF joint trial
- Transformations directly from the State.cfg file
  - T_TIBOS_TIB
  - T_FEMOS_FEMORI
  - T_PATOS_PAT
  - Offset_FEMORI_TIB
  - Offset_FEMORI_PAT
- Extract sphere centers (same process as TF joint trials)
  - T_I_TIBOS and T_I_FEMOS



Re-sampling Data Script

- After registration script has been run.
- Run resampling_plotting.py <oks0XX_registration_0x.xml> <tdms_list.txt>
  - Inputs:
    - Registration XML
    - Text file of tdms files that you would like to re-sample (text file needs to be within the general test folder, i.e. TibiofemoralJoint or PatellofemoralJoint)



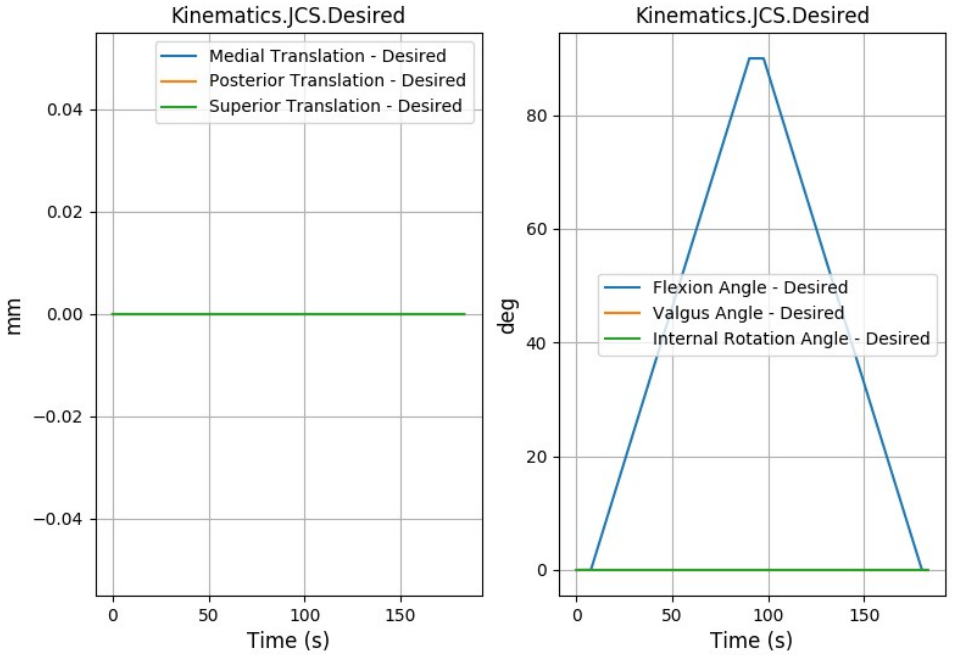
  - Outputs:
    - CSV file of all channels within the TDMS file at the re-sampled locations
    - Video animation of the test
    - Graphs of the JCS kinematics and loads (both raw and re-sampled)

What is the script doing?

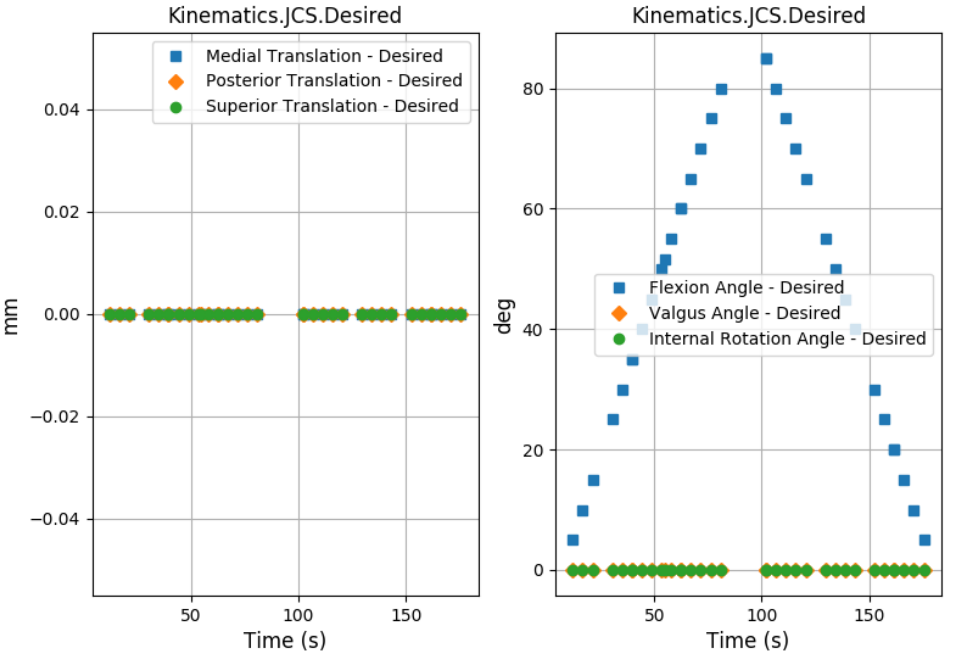
- **Sampling the data into a more usable format for comparison to model results and/or boundary condition input.**

Animation – Passive flexion oks003

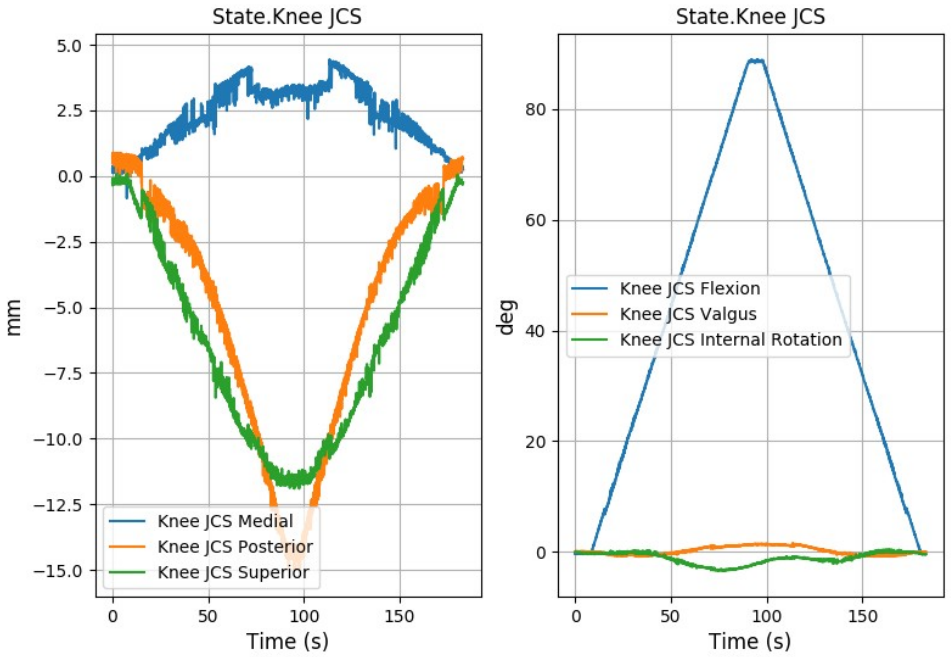
003_optimized passive flexion_0-90_main_processed



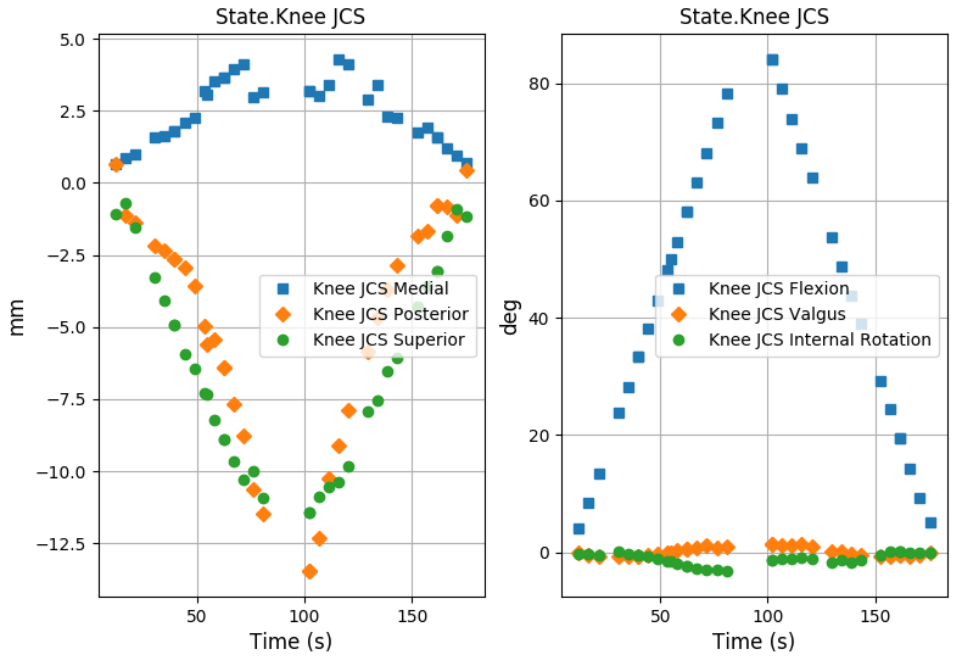
003_optimized passive flexion_0-90_main_processed_extracted

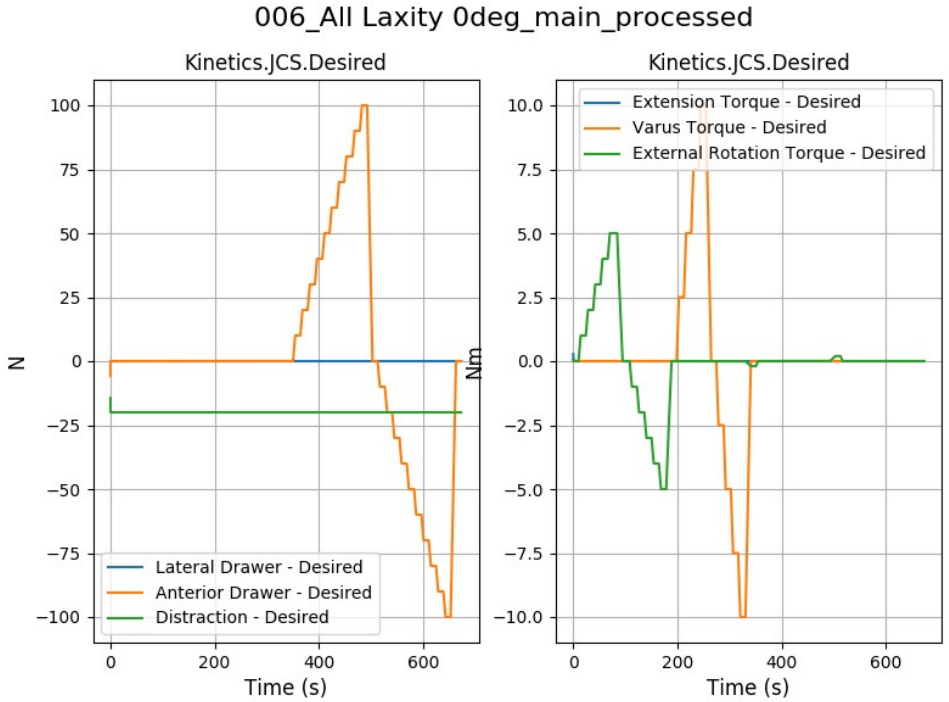


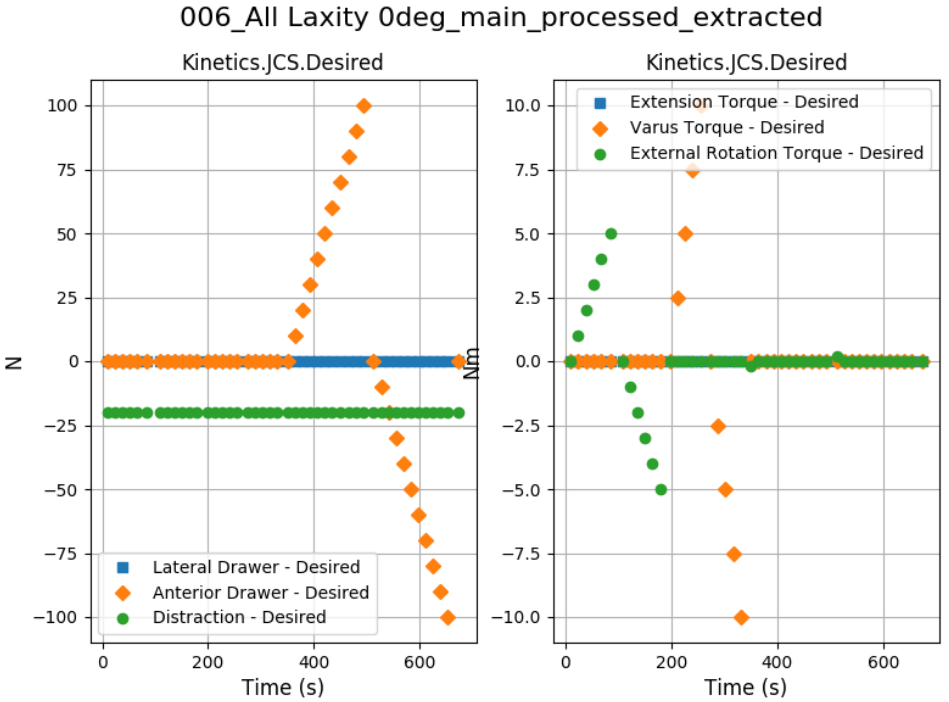
003_optimized passive flexion_0-90_main_processed

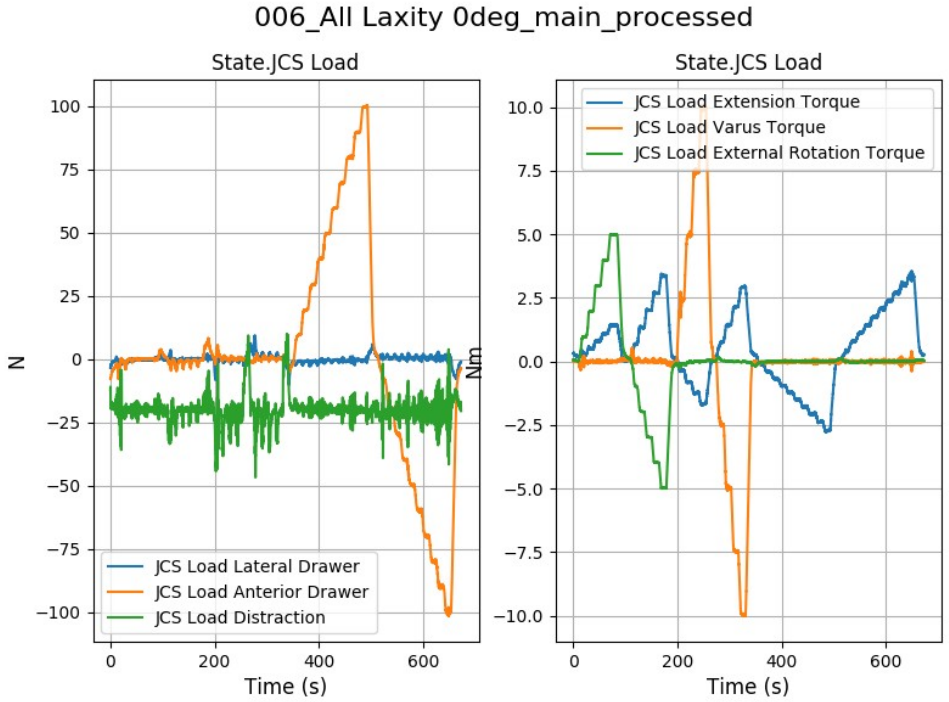


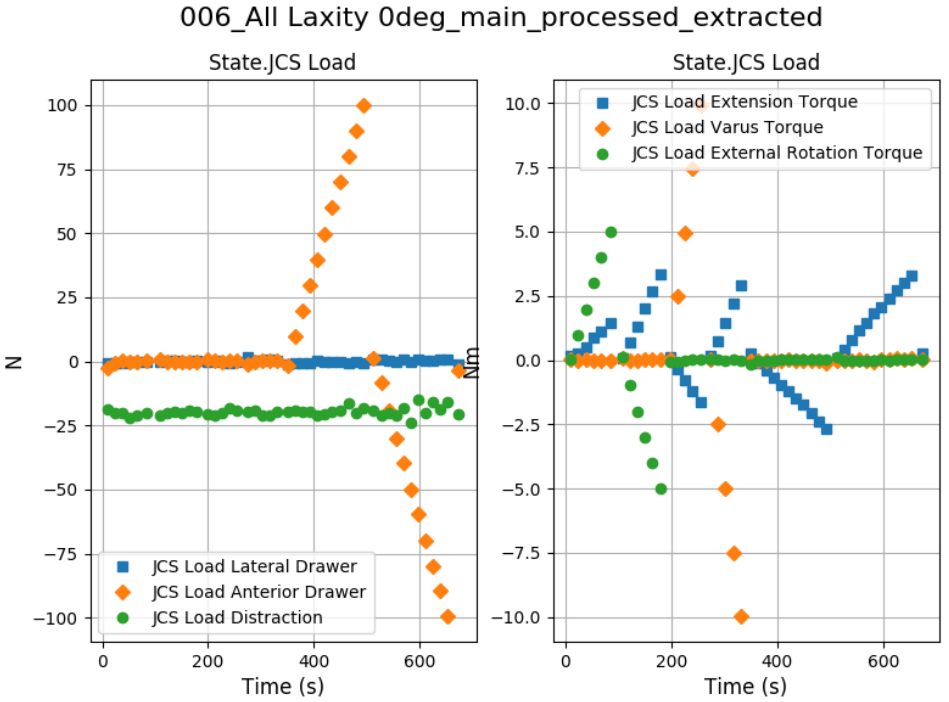
003_optimized passive flexion_0-90_main_processed_extracted




006_All Laxity 0deg_main_processed
Kinetics.JCS.Desired
Kinetics.JCS.Desired
Lateral Drawer - Desired
Anterior Drawer - Desired
Distraction - Desired
Extension Torque - Desired
Varus Torque - Desired
External Rotation Torque - Desired
N
Nm
Time (s)
Time (s)


006_All Laxity 0deg_main_processed_extracted
Kinetics.JCS.Desired
Kinetics.JCS.Desired
Lateral Drawer - Desired
Anterior Drawer - Desired
Distraction - Desired
Extension Torque - Desired
Varus Torque - Desired
External Rotation Torque - Desired
N
Nm
Time (s)
Time (s)


006_All Laxity 0deg_main_processed
State.JCS Load
State.JCS Load
JCS Load Lateral Drawer
JCS Load Anterior Drawer
JCS Load Distraction
JCS Load Extension Torque
JCS Load Varus Torque
JCS Load External Rotation Torque
N
Nm
Time (s)
Time (s)


006_All Laxity 0deg_main_processed_extracted
State.JCS Load
State.JCS Load
JCS Load Lateral Drawer
JCS Load Anterior Drawer
JCS Load Distraction
JCS Load Extension Torque
JCS Load Varus Torque
JCS Load External Rotation Torque
N
Nm
Time (s)
Time (s)