Advancing Knee Joint Modeling towards clinical use

NEW ANYBODY FEATURE: SCALABLE MOVING-AXIS TIBIOFEMORAL JOINT

Contents lists available at ScienceDirect

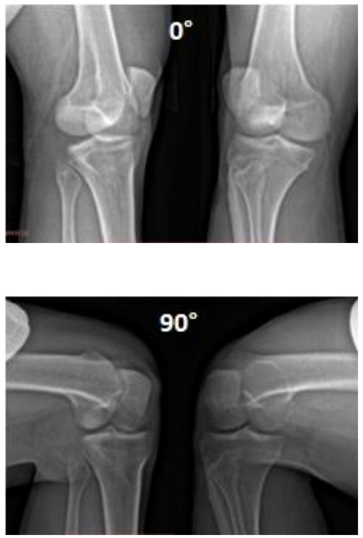
Journal of Biomechanics

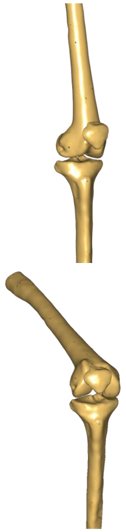
journal homepage: www.elsevier.com/locate/jbiomech
www.JBiomech.com

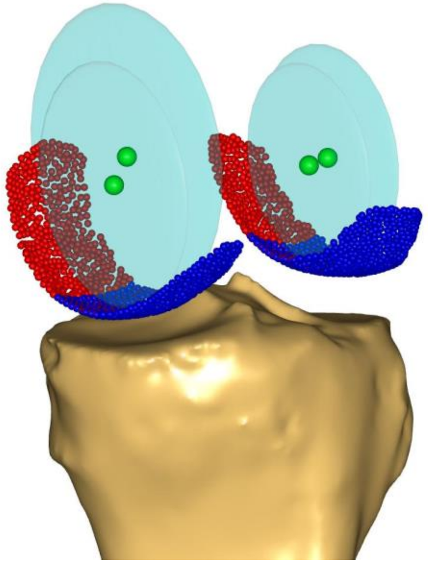
Development and validation of a subject-specific moving-axis tibiofemoral joint model using MRI and EOS imaging during a quasi-static lunge

C.M. Dzialo [a,*], P.H. Pedersen [b], C.W. Simonsen [c], K.K. Jensen [c], M. de Zee [d], M.S. Andersen [a]

[a] Department of Materials and Production, Aalborg University, Fibigerstræde 16, DK-9220 Aalborg, Denmark
[b] Department of Orthopedic Surgery, Aalborg University Hospital, Hobrovej 18-22, DK-9000 Aalborg, Denmark
[c] Department of Radiology, Aalborg University Hospital, Hobrovej 18-22, DK-9000 Aalborg, Denmark
[d] Department of Health Science and Technology, Aalborg University, Fredrik Bajers Vej 7D, DK-9220 Aalborg, Denmark







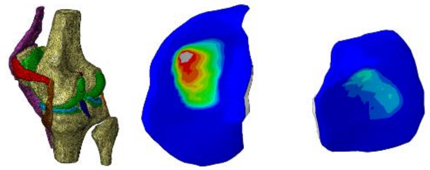
Interested in learning more? See publication ↑

Introduction

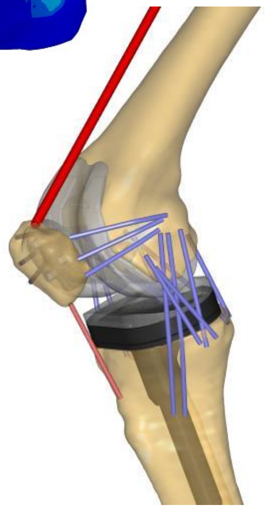
Why model the tibiofemoral (TF) joint?

To employ computational modeling to study:

- Pathologies (ex. knee osteoarthritis)
- Healthy biomechanics (pre/post surgery)
- Sports performance

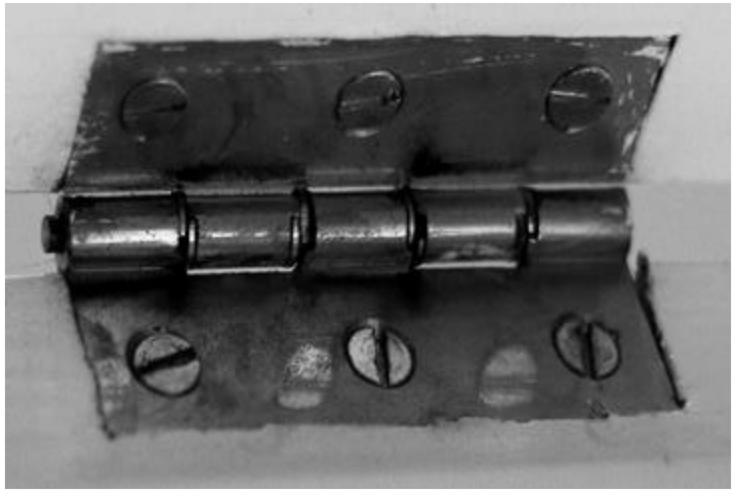


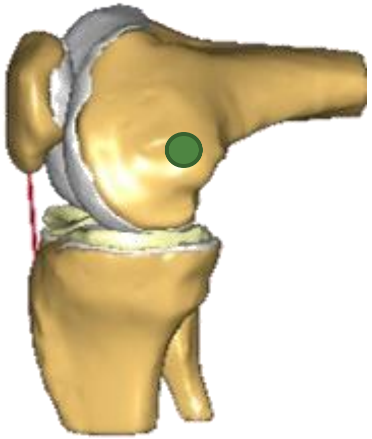
Halonen et al 2018



Existing Knee Models

- **Generic hinge**

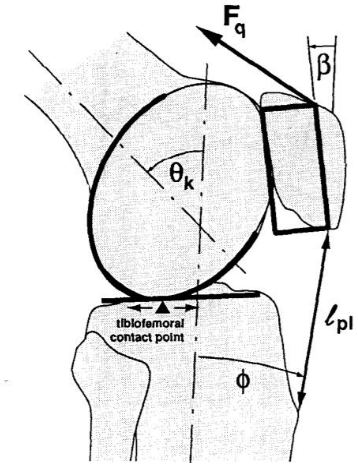




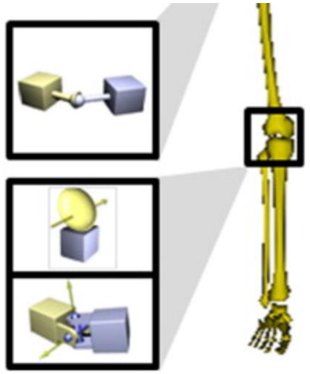
Based on cadaver landmarks
(Epicondyles)

Existing Knee Models

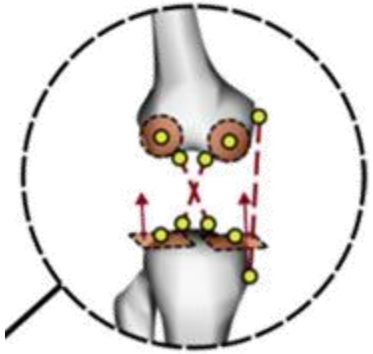
- **Generic hinge**
- **Simple generic:** 4-bar linkage, coupling constraints, sphere-on-plane, etc….



Delp et al 1990



Donnelly et al 2011



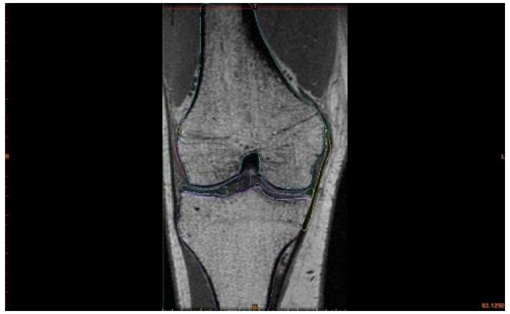
Habachi et al 2015

Existing Knee Models

Simple

Complex

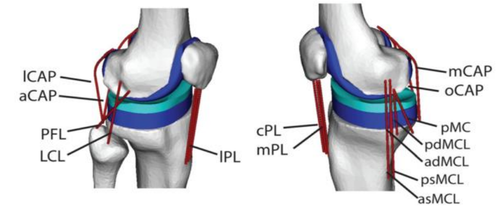
- **Generic hinge**
- **Simple generic:** 4-bar linkage, coupling constraints, sphere-on-plane, etc.…
- **Simple subject-specific (SS):** same as above with SS inputs



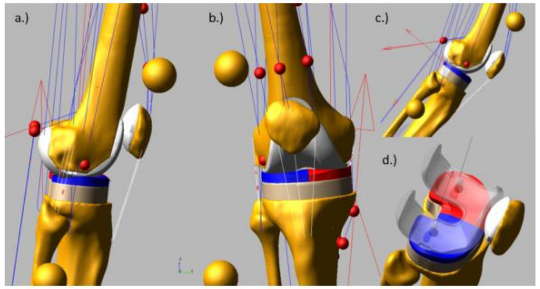
Existing Knee Models

- **Generic hinge**
- **Simple generic:** 4-bar linkage, coupling constraints, sphere-on-plane, etc.…
- **Simple subject-specific (SS):** same as above with SS inputs
- **Multi-body** models with contacts

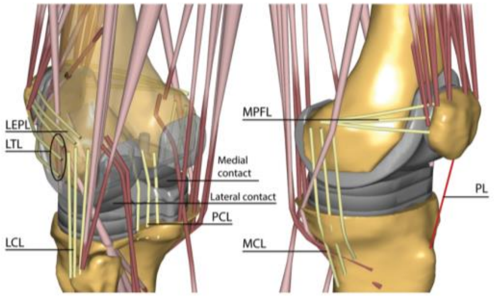
Thelen et al. 2014



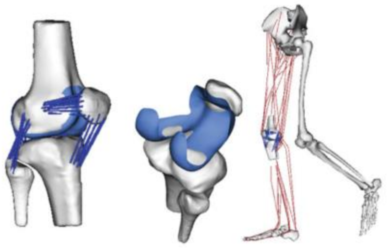
Guess et al. 2014



Marra et al. 2015



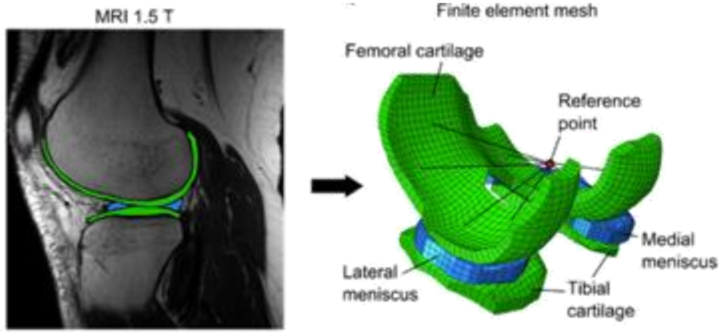
Smith et al. 2016



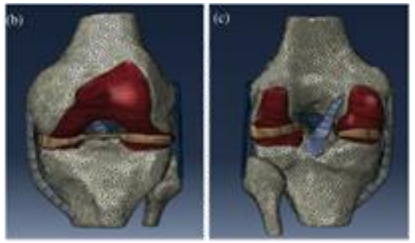
Existing Knee Models

- **Generic hinge**
- **Simple generic:** 4-bar linkage, coupling constraints, sphere-on-plane, etc.…
- **Simple subject-specific (SS):** same as above with SS inputs
- **Multi-body** models with contacts
- **Finite Element Analysis** (FEA) models

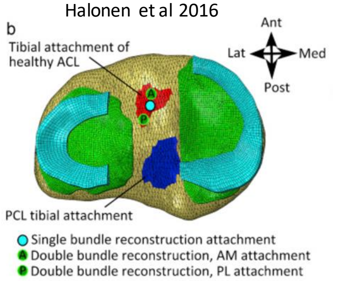
Halonen et al 2013c



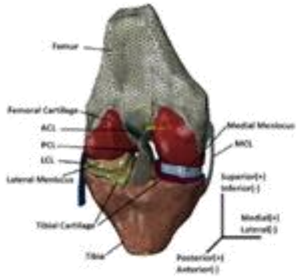
Mootanah et al 2014



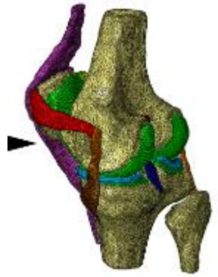
Halonen et al 2016



Beidokhti et al 2016



Halonen et al 2018



Existing Knee Models

Simple

Complex

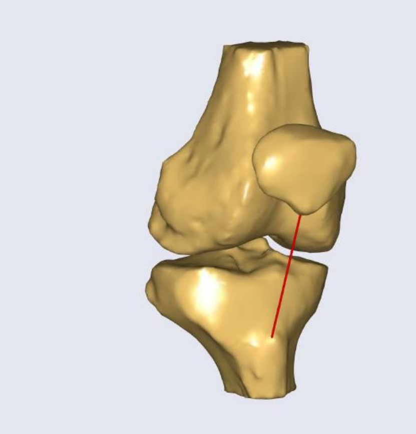
- **Generic hinge**
- **Simple generic:** 4-bar linkage, coupling constraints, sphere-on-plane, etc.…
- **Simple subject-specific (SS):** same as above with SS inputs
- **Multi-body** models with contacts
- **Finite Element Analysis** (FEA) models

Motivation

Clinics would need something in between

Simple ⟷ **Complex**

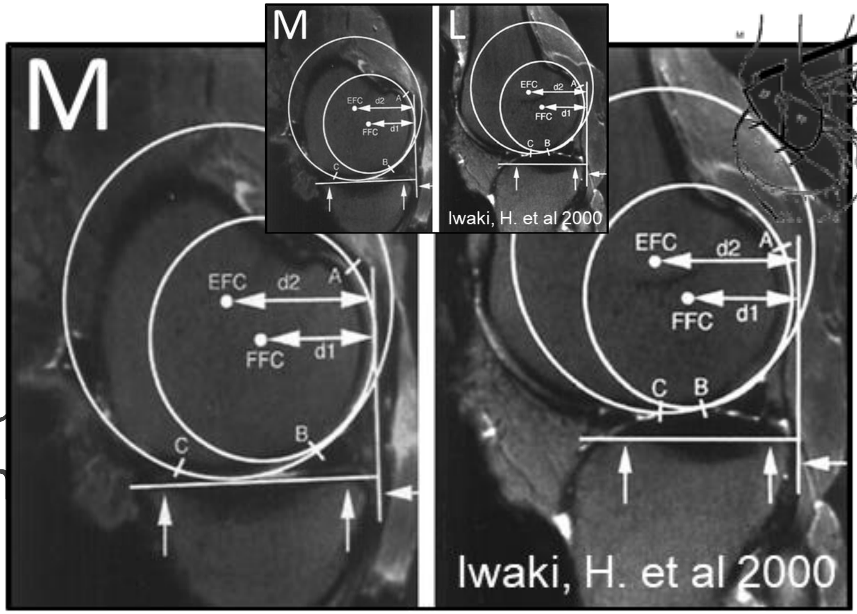
- **Realistic joint motion**
- **Computational fast**

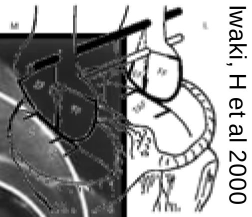


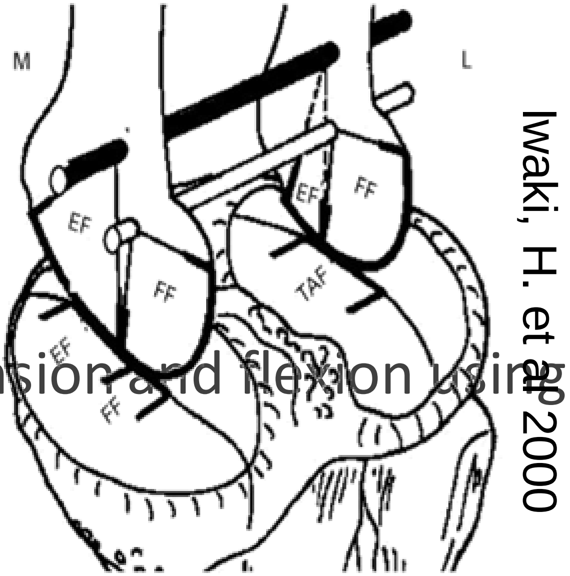
Development and validation of moving-axis knee models

Motivation

- Tibiofemoral (TF) axis shifts as contact surface changes (Iwaki 2000)







- Find su… extension and flexion using EOS Im…
- Develop moving-axis TF joint to capture secondary joint kinematics

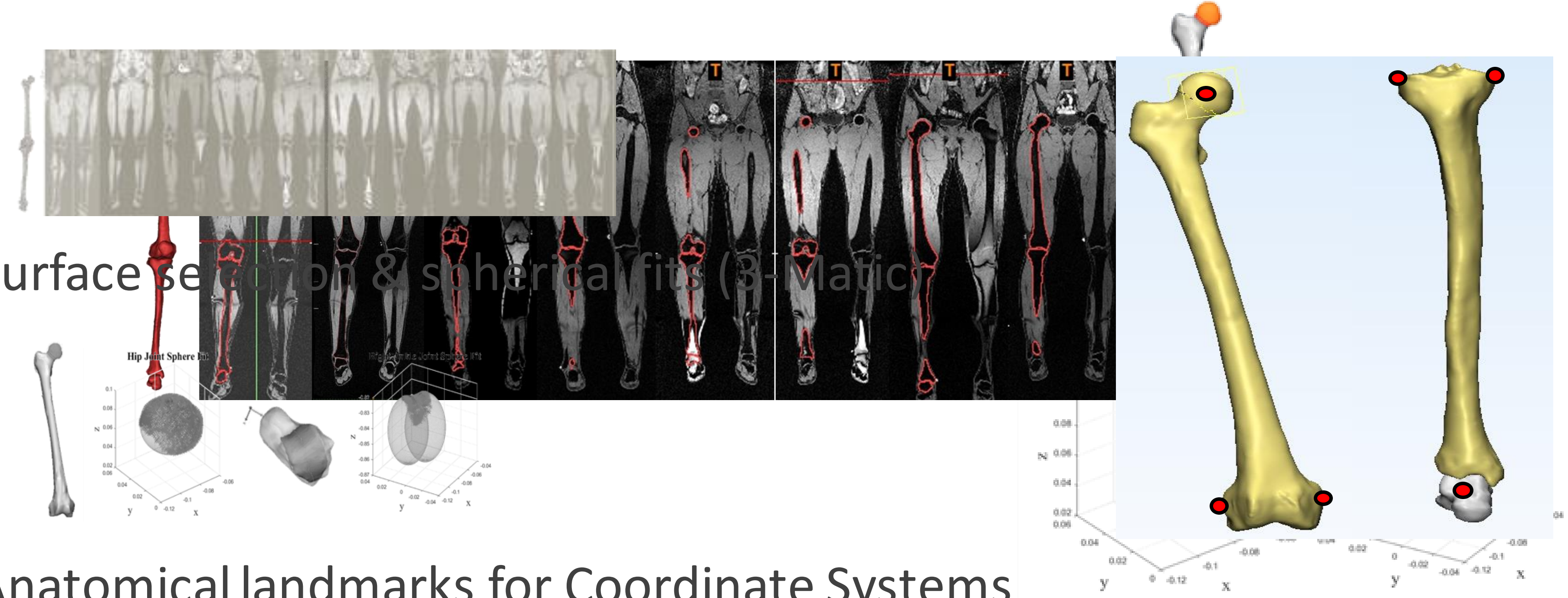
Data Collection

- 10 healthy male subjects
- MRI Lower Limbs (GE 1.5T)
    - COR T1W-LAVA-XV-IDEAL 1.6 mm
    - ~ 12 minutes
- EOS scans
    - at 0, 20, 45, 60, and 90
    - knee flexion

Model Geometry

- Segmentation (Mimics)

- Surface selection & spherical fits (3-Matic)



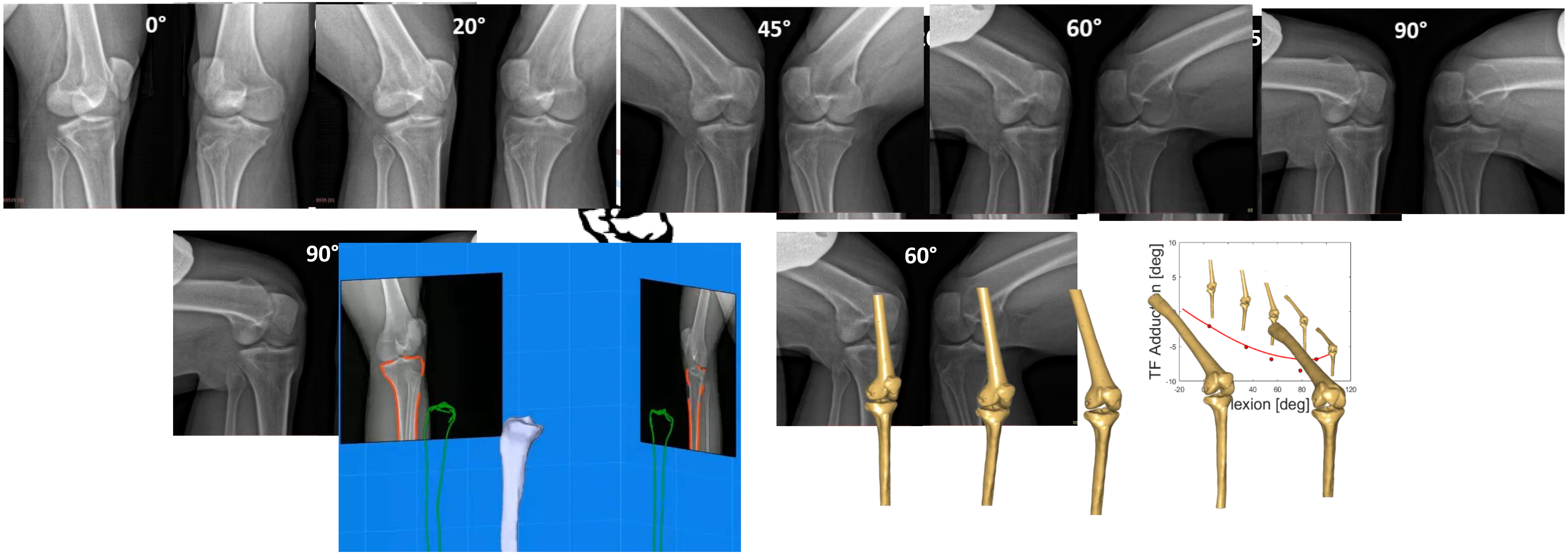
- Anatomical landmarks for Coordinate Systems

EOS Imaging & Reconstruction

Bone positions were found using custom software

Model calibration

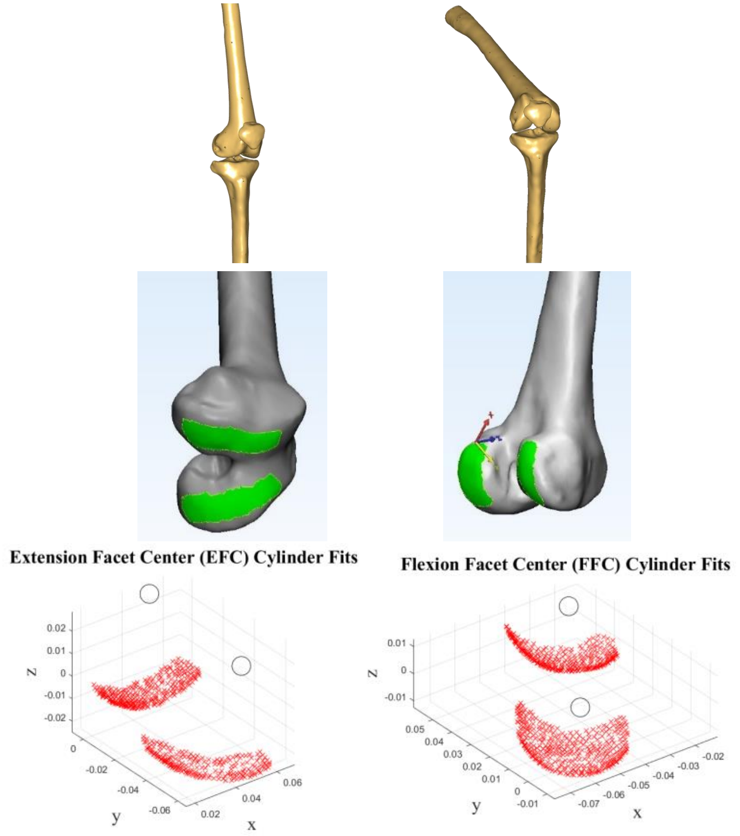
Model validation



Model Development

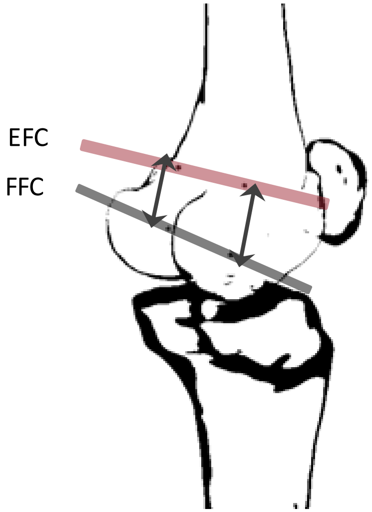
- Export EOS bone positions (0 & 90)
- Surface Selection (3-Matic)
- Knee Joint Definition (MATLAB)

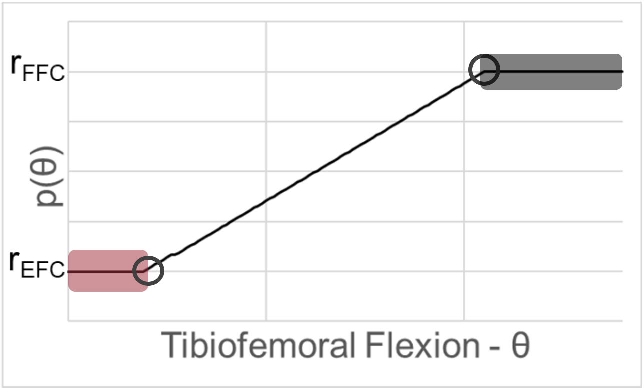
→ using Cylinder Fits

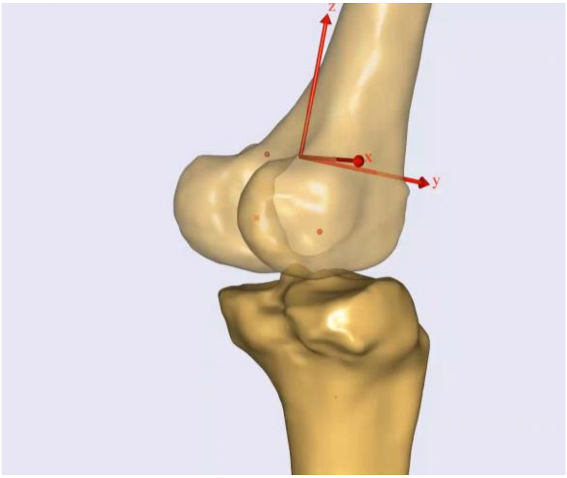


Model Development

Linear Combination Functions (AMS)

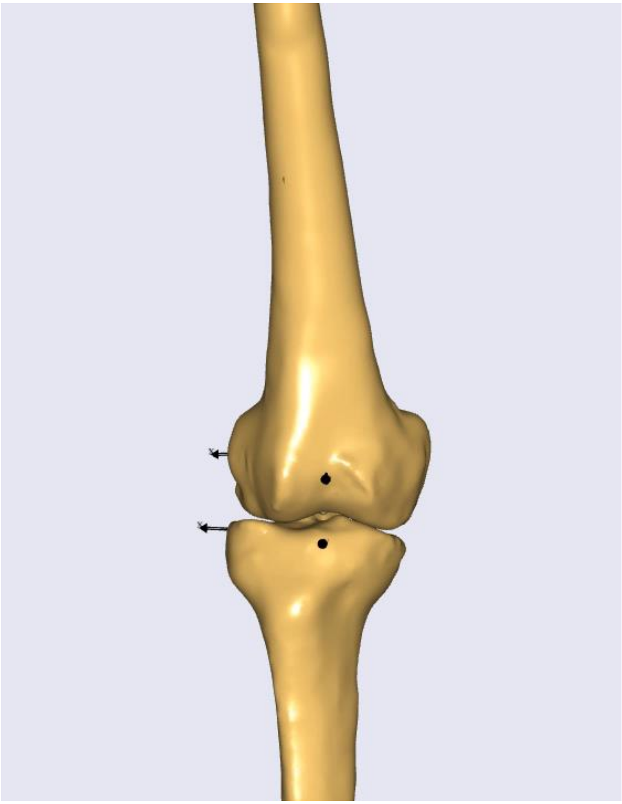


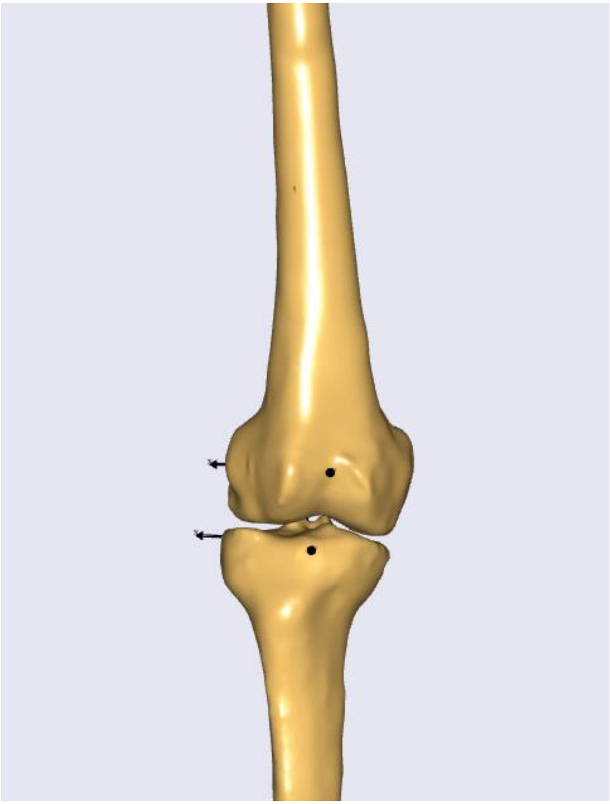




$$p(\theta) = \begin{cases} r_{EFC} & \text{"0"}_{EOS} > \theta \\ r_{EFC}\left(1 - \frac{\theta}{\text{"90"}_{EOS}}\right) + r_{FFC}\left(\frac{\theta}{\text{"90"}_{EOS}}\right) & \text{"0"}_{EOS} \le \theta \le \text{"90"}_{EOS} \\ r_{FFC} & \theta > \text{"90"}_{EOS} \end{cases}$$

Results





Subject-specific bone geometry

TF axis through epicondyles

Results

	Coefficient of Determination (R^2)	
	Hinge	Moving Axis (3)
Dislocation	0.26	0.31
Drawer	0.21	0.71
Distraction	0.70	0.91
Adduction	0.08	0.79
External Rot.	0.27	0.67

Increase →

- Examined model predictive capabilities
- Moving-axis can better estimate secondary kinematics compared to hinge

Discussion & Conclusions

Limitations

- Linear relationship may not represent realistic joint movements
- Validation for only one movement type, cannot generalize…

Future Work

→ Dynamic imaging to investigate polynomial relationships

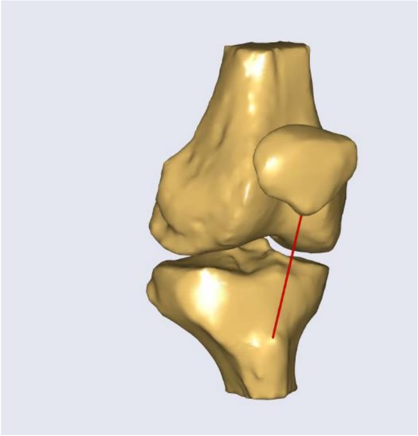
→ Examine various activities of daily living

Discussion & Conclusions

Clinics would need something in between

Developed & validated moving-axis knee joint

- Computationally efficient (once established)
- More realistic secondary kinematics compared to hinge
- Subject-specific geometries

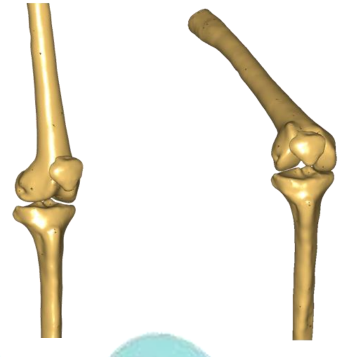


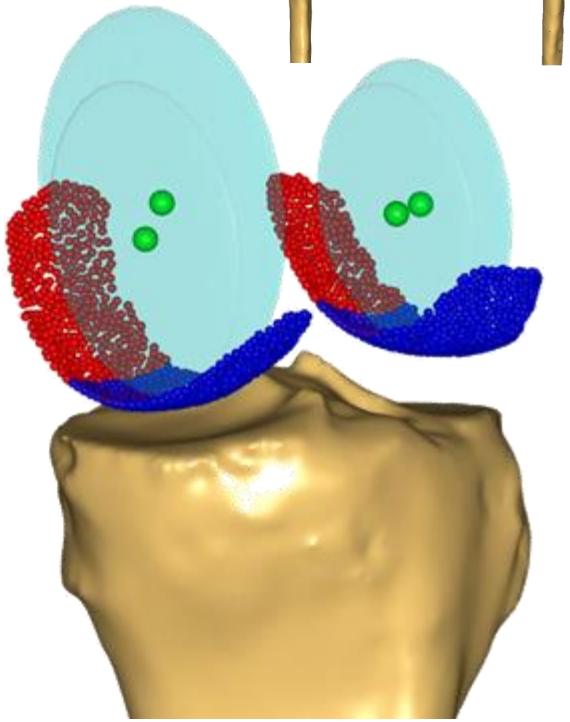
- How can we speed up the development process?

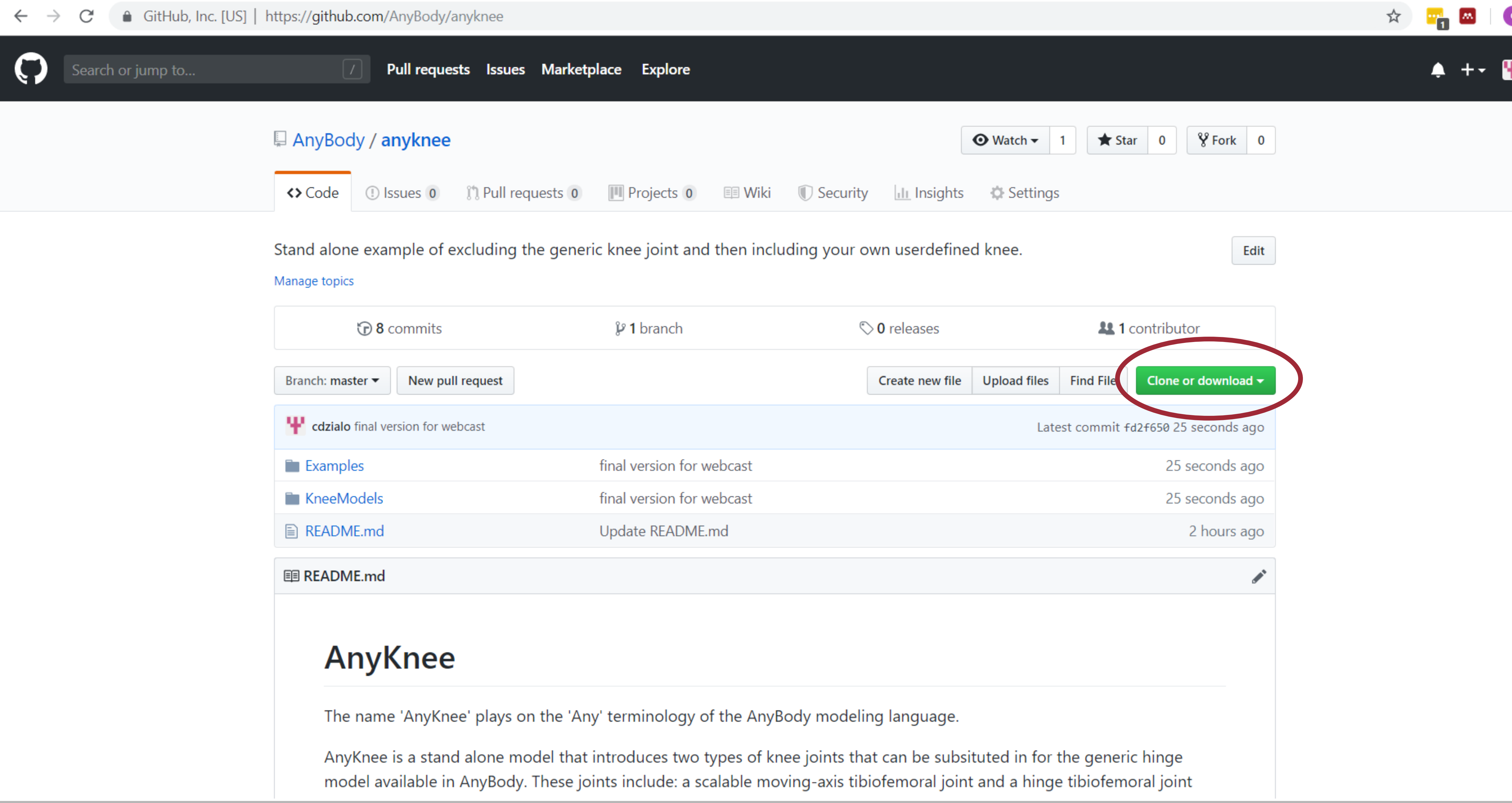
- What if users do not have access to bi-planar imaging?

AnyBody Modeling System

- Scalable moving-axis knee joint
  - tibiofemoral ☑
  - patellofemoral ☐
- Initially released as stand-alone model
  https://github.com/AnyBody/anyknee



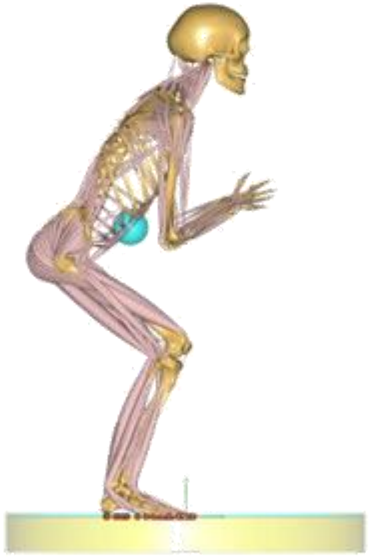



GitHub, Inc. [US] | https://github.com/AnyBody/anyknee
Search or jump to...
Pull requests
Issues
Marketplace
Explore
AnyBody / anyknee
Watch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
Stand alone example of excluding the generic knee joint and then including your own userdefined knee.
Edit
Manage topics
8 commits
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
cdzialo final version for webcast
Latest commit fd2f650 25 seconds ago
Examples
final version for webcast
25 seconds ago
KneeModels
final version for webcast
25 seconds ago
README.md
Update README.md
2 hours ago
README.md
AnyKnee
The name 'AnyKnee' plays on the 'Any' terminology of the AnyBody modeling language.
AnyKnee is a stand alone model that introduces two types of knee joints that can be subsituted in for the generic hinge model available in AnyBody. These joints include: a scalable moving-axis tibiofemoral joint and a hinge tibiofemoral joint

AnyBody Modeling System

- Easy to incorporate into existing models
- Insert code before either:
    - `#include HumanModel.any`
    - `#include "<ANYMOCAP_MODEL>“`
- Exclude right, left, or both knee joints

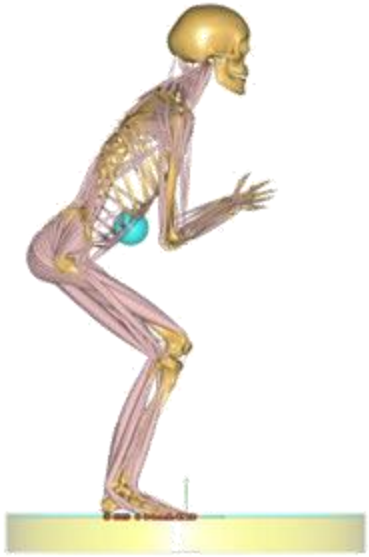
```
// Exclude right & left knee
#define BM_JOINT_TYPE_KNEE_RIGHT _JOINT_TYPE_USERDEFINED_
#define BM_JOINT_TYPE_KNEE_LEFT _JOINT_TYPE_USERDEFINED_

// Add in new knee joint
#include "../../KneeModels/MovingAxisKnee/AddOnKnee.any“
```



AnyBody Modeling System

- **NOTE:** This stand alone moving-axis model is ready to use
- What is present next is **how it was built**,

  i.e. what's happening behind the scenes
- **Aim:** inspire users to generate new models



User-defined joint requirements

```
#define BM_JOINT_TYPE_KNEE_RIGHT _JOINT_TYPE_USERDEFINED_
#define BM_JOINT_TYPE_KNEE_LEFT _JOINT_TYPE_USERDEFINED_
 sRel=(.EpicondylusFemorisMedialis.sRel+
        .EpicondylusFemorisLateralis.sRel)/2.0;;
 ARel=RotMat(sRel,
               .EpicondylusFemorisLateralis.sRel,
```

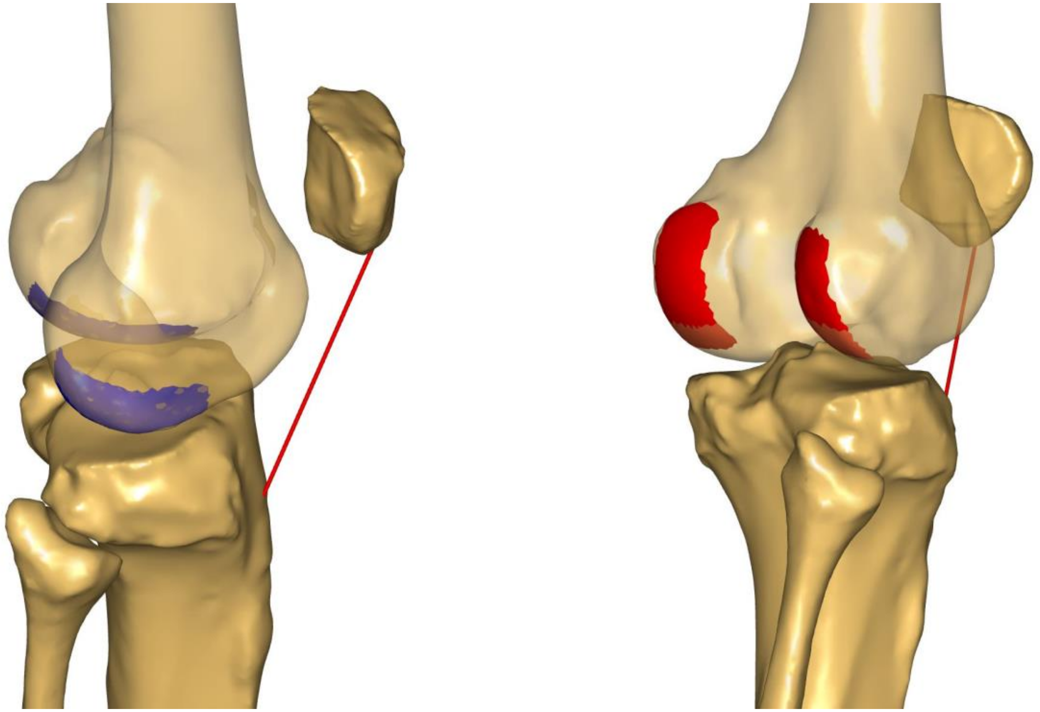
- **Knee.Constraint.Reactions** based on user-defined drivers

```
 AnyKinMeasureOrg Knee = {
     AnyKinEq Constraints = {
       AnyKinMeasure &InvisibleFemurMedial_Driver = .InvisibleFemurMedialEFCDriver_LinComb;
       AnyKinMeasure &InvisibleFemurLateral_Driver  = .InvisibleFemurLateralEFCDriver_LinComb;
       AnyKinMeasure &InvisibleFemurZRot = .InvisibleFemurZRot_org;
       AnyKinMeasure &InvisibleTibiaMedial_Driver = .InvisibleTibiaMedialEFCDriver_LinComb;
       AnyKinMeasure &InvisibleTibiaLateral_Driver = .InvisibleTibiaLateralEFCDriver_LinComb;
       AnyKinMeasure &InvisibleTibiaZRot = .InvisibleTibiaZRot_org;
     };
};
```

- **KneeJoint** nodes
  - *ModelSegmentParameter.any*
  - estimate thigh & shank lengths
- **sRelUnscaled** vector
  - *Seg.any*
  - for ARel of HipJoint

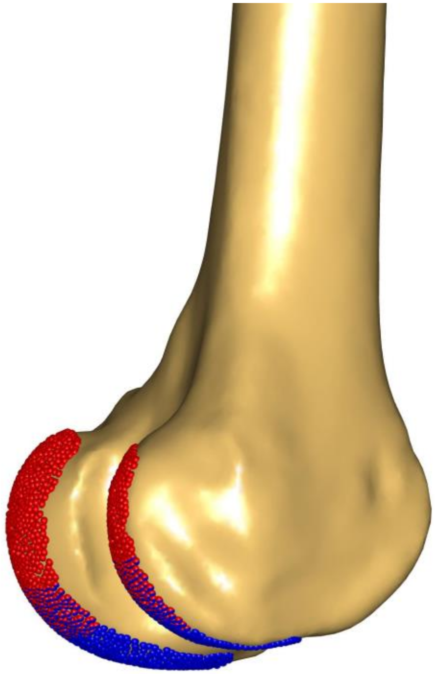
Code adjusted for readability

Model Development



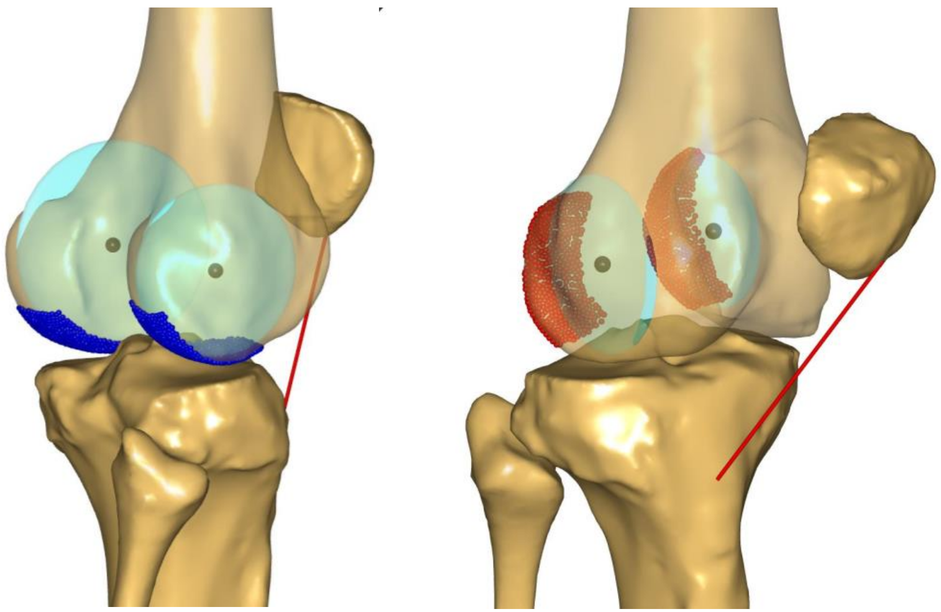
Contact Surface selections imported

Twente Lower Extremity Model 2.0 (cadaver dataset)

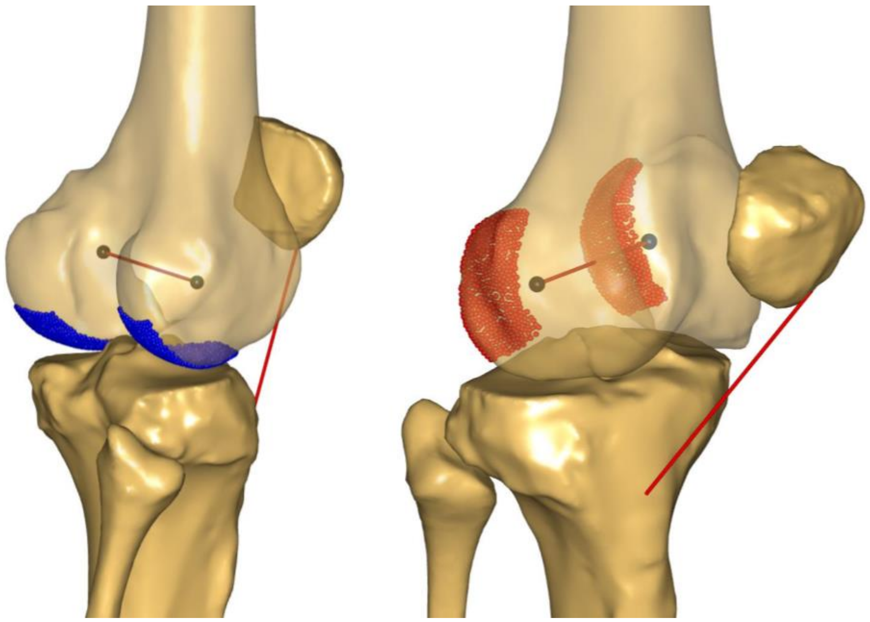


GetVertices function to obtain points

Model Development



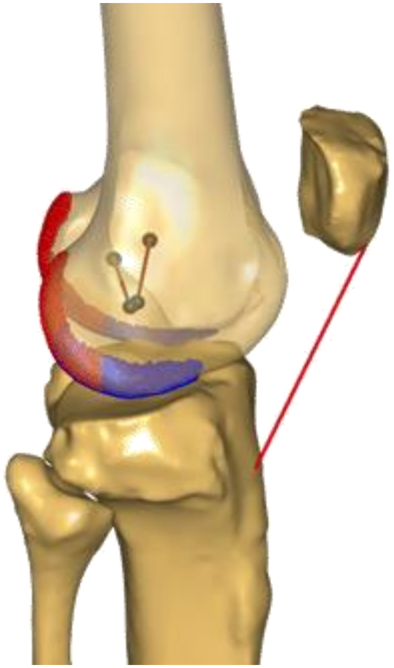
AnySurfCylinderFit applied to point clouds



Resulting centers define EFC & FFC axes

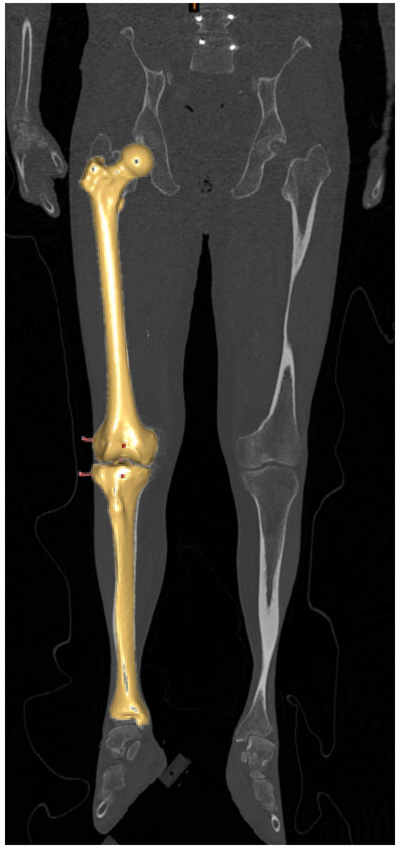
Model Development

- Position of tibia with respect to femur is required to run moving-axis function



0° kinematics derived from:

- TLEM 2.0 CT scan



90° kinematics derived from:

- B. Yue et al. 2011
- W. Qi et al. 2013

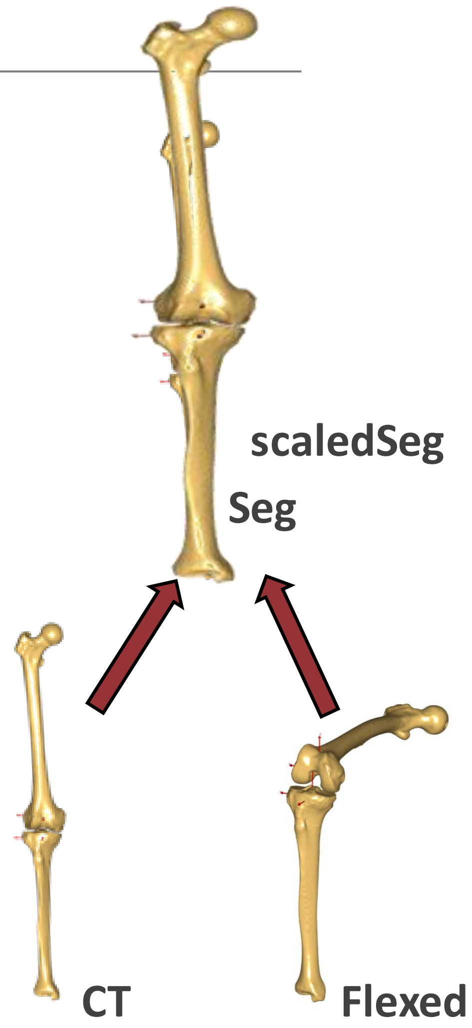
Model Development

- **SubjectSpecificScaling.any**

- AnyFunTransform3DLin2: ‘***Flexed2Seg***’ & ‘*CT2Seg*’
- AnyFunTransform3DIdentity: ‘***Flexed2scaledSeg***’ & ‘*CT2scaledSeg*’

```
//transformation from flexed bone position ref. frame to unscaled segmental ref. frame
 AnyFunTransform3DLin2 Flexed2Seg = {
   Points0 = STL_Vertices(Flexed_Bone)*AMirroring;
   Points1 = STL_Vertices(Seg_Bone)*AMirroring;
   Mode = VTK_LANDMARK_RIGIDBODY;
 };
```

```
//transformation from flexed bone position to scaled segmental ref. frame
 AnyFunTransform3DIdentity Flexed2scaledSeg = {
   PreTransforms = {&.Flexed2Seg, &..Scale};
 };
```

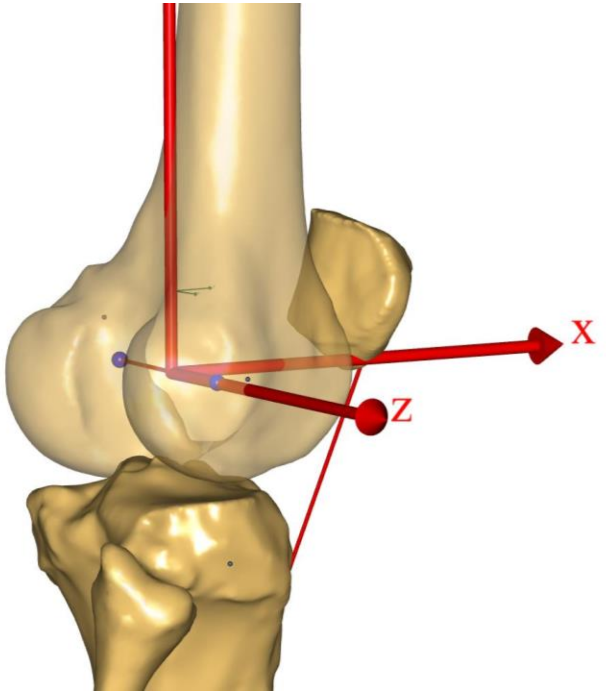


Code adjusted for readability

Model Development

- **InvisibleSegments.any**
- Invisible femur and tibia (*IF & IT*) needed to drive motion of rigid bodies

```
 AnySeg InvisibleFemur_TF = {

  Mass = 0;
  Jii = {0,0,0};
  r0 = Thigh.r0+MA_AnatomicalFrame.rOffset * Thigh.Axes0';
  Axes0 = Thigh.Axes0 * MA_AnatomicalFrame.AOffset;

  AnyRefNode MA_AnatomicalFrame = {
    sRel = (Thigh.Medial_TF_EFC + Thigh.Lateral_TF_EFC)/2
    ARel = RotMat(sRel,Thigh.Lateral_TF_EFC, Thigh.HipJointCenter)
  };
};

AnySeg InvisibleTibia = {[…]};
```



Code adjusted for readability

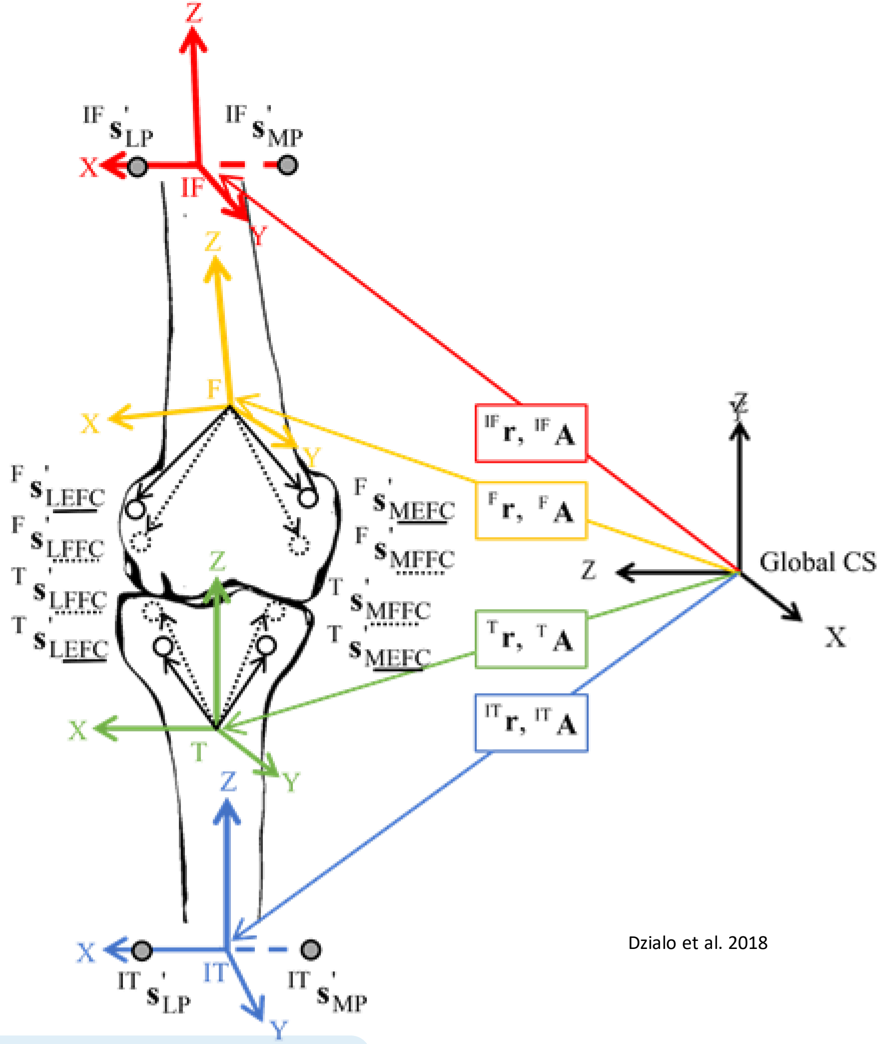
Model Development

- **InvisibleSegments.any**
- Invisible femur and tibia (*IF & IT*) needed to drive motion of rigid bodies

solid color: transformation matrix between GCS → (*F, T, IF, & IT*) CS

solid black : position vector of EFC points in (*F & T*) CS

dotted black: position vector of FFC points in (*F & T*) CS



Dzialo et al. 2018

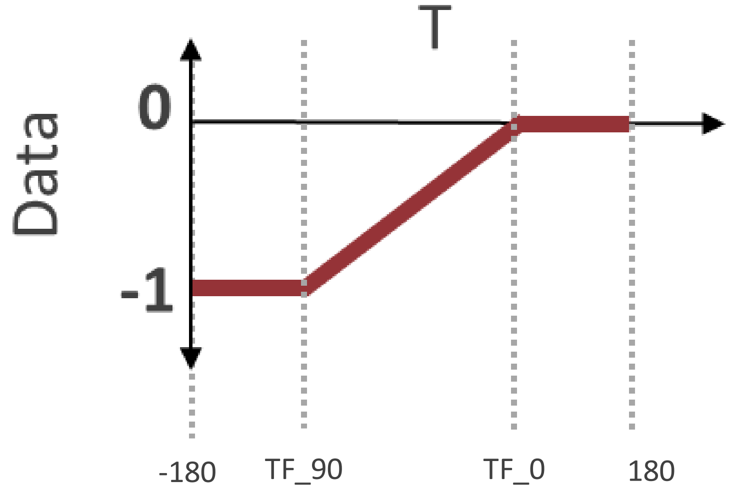
For more details see: https://doi.org/10.1016/j.jbiomech.2018.02.032

Model Development

- InvisibleDriversandJoints.any
    - Invisible Revolute Joint (IF & IT)
    - **Invisible femur drivers (2 AnyKinMeasureLinComb, 1 AnyKinRotational)**
    - Invisible tibia drivers (2 AnyKinMeasureLinComb, 1 AnyKinRotational)
- AnyKinMeasureFunComb1 Creates new measures by applying a parameter function to the input measures (flexion)

```
AnyFunInterpol MovingAxisCutOffFunction = {
   Type = PiecewiseLinear ;
   T = {-180.0, TF_FE_90, TF_FE_0, 180.0};
   Data = {{-1.0, -1.0, 0.0, 0.0}};
 };
```

```
AnyKinMeasureFunComb1 CutOffKneeFlexion = {
  Functions = {&.MovingAxisCutOffFunction};

  AnyKinMeasureLinComb knee_flexion = {
    AnyKinMeasure &knee_angles = Tibiofemoral_Measures.JCS_rotation;
    Coef   = {1, 0, 0};  // {z,x,y} order
    OutDim = 1;
  };
};
```

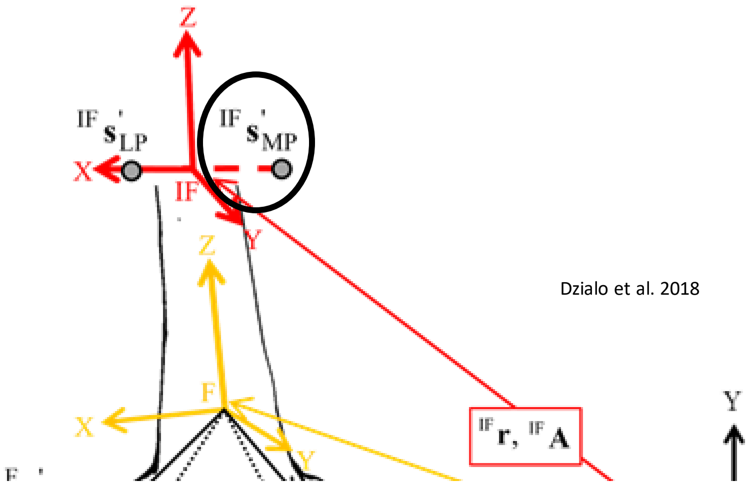


Code adjusted for readability

Constraint Equations

- Constrain **medial** point of the invisible segments to a position in between the EFC and FFC (**xyz**)

$$v = \frac{\theta_{\mathrm{TF}}}{\theta^{90}_{\mathrm{EOS}} - \theta^{0}_{\mathrm{EOS}}} - \frac{\theta^{0}_{\mathrm{EOS}}}{\theta^{90}_{\mathrm{EOS}} - \theta^{0}_{\mathrm{EOS}}}$$



Dzialo et al. 2018

$$\boldsymbol{\Phi}(\mathbf{q}) = \begin{bmatrix} {}_{(\mathrm{F})}\mathbf{f}_{\mathrm{MP}} - {}_{(\mathrm{IF})}\mathbf{r} - {}_{(\mathrm{IF})}\mathrm{A}_{(\mathrm{IF})}\mathbf{s}'_{\mathrm{MP}} \\ {}_{(\mathrm{T})}\mathbf{f}_{\mathrm{MP}} - {}_{(\mathrm{IT})}\mathbf{r} - {}_{(\mathrm{IT})}\mathrm{A}_{(\mathrm{IT})}\mathbf{s}'_{\mathrm{MP}} \\ ({}_{(\mathrm{IF})}\mathrm{A}^{\mathrm{T}}({}_{(\mathrm{F})}\mathbf{f}_{\mathrm{LP}} - {}_{(\mathrm{IF})}\mathbf{r} - {}_{(\mathrm{IF})}\mathrm{A}_{(\mathrm{IF})}\mathbf{s}'_{\mathrm{LP}}))_{xy} \\ ({}_{(\mathrm{IT})}\mathrm{A}^{\mathrm{T}}({}_{(\mathrm{T})}\mathbf{f}_{\mathrm{LP}} - {}_{(\mathrm{IT})}\mathbf{r} - {}_{(\mathrm{IT})}\mathrm{A}_{(\mathrm{IT})}\mathbf{s}'_{\mathrm{LP}}))_{xy} \\ {}_{(\mathrm{IF})}\theta_z \\ {}_{(\mathrm{IT})}\theta_z \\ {}_{(\mathrm{IF})}\mathbf{r} - {}_{(\mathrm{IT})}\mathbf{r} \\ {}_{(\mathrm{IF})}\mathbf{a}_z^T\,{}_{(\mathrm{IT})}\mathbf{a}_y \\ {}_{(\mathrm{IF})}\mathbf{a}_z^T\,{}_{(\mathrm{IT})}\mathbf{a}_x \\ {}_{(\mathrm{F})}\mathbf{p}^T\,{}_{(\mathrm{F})}\mathbf{p} - 1 \\ {}_{(\mathrm{IF})}\mathbf{p}^T\,{}_{(\mathrm{IF})}\mathbf{p} - 1 \\ {}_{(\mathrm{T})}\mathbf{p}^T\,{}_{(\mathrm{T})}\mathbf{p} - 1 \\ {}_{(\mathrm{IT})}\mathbf{p}^T\,{}_{(\mathrm{IT})}\mathbf{p} - 1 \end{bmatrix} = \mathbf{0}$$

where

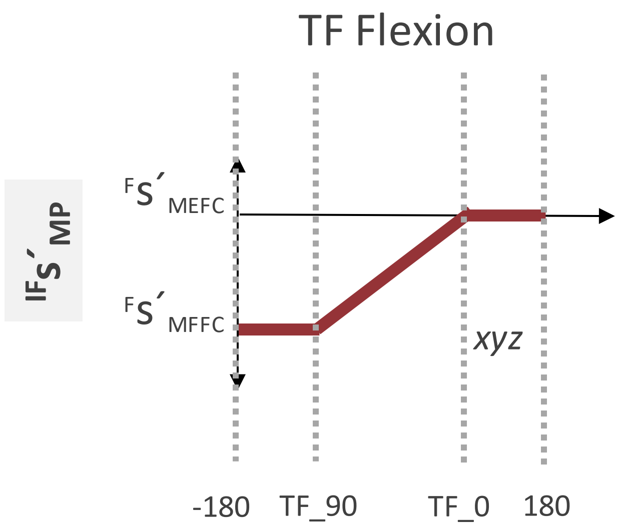
$${}_{(\mathrm{F})}\mathbf{f}_{\mathrm{MP}} = \begin{cases} {}_{(\mathrm{T})}\mathbf{r}_{\mathrm{MEFC}} & \theta_{\mathrm{TF}} < \theta^{0}_{\mathrm{EOS}} \\ {}_{(\mathrm{T})}\mathbf{r}_{\mathrm{MEFC}}(1-v) + {}_{(\mathrm{T})}\mathbf{r}_{\mathrm{MFFC}}v & \theta^{0}_{\mathrm{EOS}} \leqslant \theta_{\mathrm{TF}} \leqslant \theta^{90}_{\mathrm{EOS}} \\ {}_{(\mathrm{T})}\mathbf{r}_{\mathrm{MFFC}} & \theta^{90}_{\mathrm{EOS}} < \theta_{\mathrm{TF}} \end{cases}$$

For more details see: https://doi.org/10.1016/j.jbiomech.2018.02.032

Model Development

- Constrain **medial** point of the invisible segments to a position in between the EFC and FFC (**xyz**)

```
AnyKinMeasureLinComb InvFemur_Medial_Driver = {

      Coef = {{1.0,0.0,0.0,1.0/(1.0)*(Medial_TF_FFC.sRel[0]-Medial_TF_EFC.sRel[0])},
              {0.0,1.0,0.0,1.0/(1.0)*(Medial_TF_FFC.sRel[1]-Medial_TF_EFC.sRel[1])},
              {0.0,0.0,1.0,1.0/(1.0)*(Medial_TF_FFC.sRel[2]-Medial_TF_EFC.sRel[2])}};


      AnyKinLinear Medial_EFC_Trans = {

        AnyRefFrame &ref1 = Thigh.Medial_TF_EFC;
        AnyRefFrame &ref2 = InvisibleFemur_TF.Medial_TF_EFC;

      };

      AnyKinMeasure &KneeFlexion = CutOffKneeFlexion;
};
```

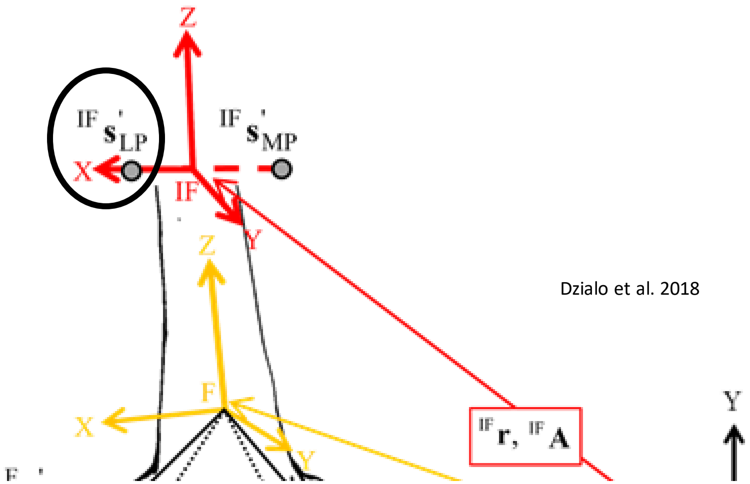


Code adjusted for readability

Constraint Equations

- Constrain **lateral** point of the invisible segments to a position in between the EFC and FFC (**xy**)

$$v = \frac{\theta_{TF}}{\theta^{90}_{EOS} - \theta^{0}_{EOS}} - \frac{\theta^{0}_{EOS}}{\theta^{90}_{EOS} - \theta^{0}_{EOS}}$$



Dzialo et al. 2018

$$\mathbf{\Phi}(\mathbf{q}) = \begin{bmatrix} {}_{(F)}\mathbf{f}_{MP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}\mathrm{A}_{(IF)}\mathbf{s}'_{MP} \\ {}_{(T)}\mathbf{f}_{MP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}\mathrm{A}_{(IT)}\mathbf{s}'_{MP} \\ \left({}_{(IF)}\mathrm{A}^{\mathrm{T}}\left({}_{(F)}\mathbf{f}_{LP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}\mathrm{A}_{(IF)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ \left({}_{(IT)}\mathrm{A}^{\mathrm{T}}\left({}_{(T)}\mathbf{f}_{LP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}\mathrm{A}_{(IT)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ {}_{(IF)}\theta_z \\ {}_{(IT)}\theta_z \\ {}_{(IF)}\mathbf{r} - {}_{(IT)}\mathbf{r} \\ {}_{(IF)}\mathbf{a}_z^T\,{}_{(IT)}\mathbf{a}_y \\ {}_{(IF)}\mathbf{a}_z^T\,{}_{(IT)}\mathbf{a}_x \\ {}_{(F)}\mathbf{p}^T\,{}_{(F)}\mathbf{p} - 1 \\ {}_{(IF)}\mathbf{p}^T\,{}_{(IF)}\mathbf{p} - 1 \\ {}_{(T)}\mathbf{p}^T\,{}_{(T)}\mathbf{p} - 1 \\ {}_{(IT)}\mathbf{p}^T\,{}_{(IT)}\mathbf{p} - 1 \end{bmatrix} = \mathbf{0}$$

where

$${}_{(F)}\mathbf{f}_{MP} = \begin{cases} {}_{(T)}\mathbf{r}_{MEFC} & \theta_{TF} < \theta^{0}_{EOS} \\ {}_{(T)}\mathbf{r}_{MEFC}(1 - v) + {}_{(T)}\mathbf{r}_{MFFC}\,v & \theta^{0}_{EOS} \leqslant \theta_{TF} \leqslant \theta^{90}_{EOS} \\ {}_{(T)}\mathbf{r}_{MFFC} & \theta^{90}_{EOS} < \theta_{TF} \end{cases}$$

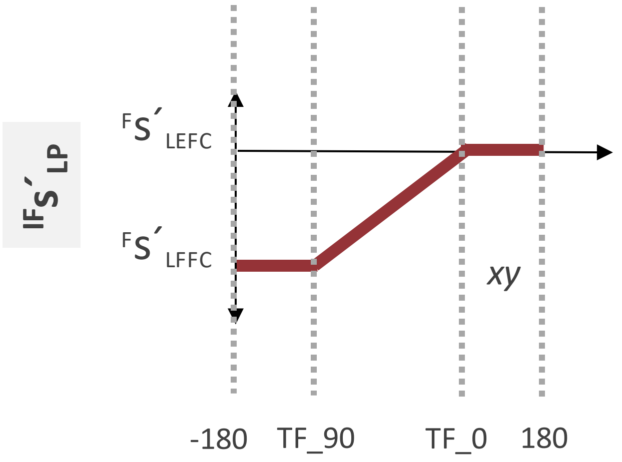
For more details see: https://doi.org/10.1016/j.jbiomech.2018.02.032

Model Development

- Constrain **lateral** point of the invisible segments to a position in between the EFC and FFC (**xy**)

```
AnyKinMeasureLinComb InvFemur_Lateral_Driver = {

      Coef = {{1.0,0.0,1.0/(1.0)*(Lateral_TF_FFC.sRel[0]-Lateral_TF_EFC.sRel[0])},
              {0.0,1.0,1.0/(1.0)*(Lateral_TF_FFC.sRel[1]-Lateral_TF_EFC.sRel[1])}};

      AnyKinMeasureOrg org = {
        AnyKinLinear FemurTrans = {
          AnyRefFrame &ref1 = Thigh.Lateral_TF_EFC;
          AnyRefFrame &ref2 = InvisibleFemur_TF.Lateral_TF_EFC;
        };
        MeasureOrganizer = {0,1};
      };
      AnyKinMeasure &KneeFlexion = CutOffKneeFlexion;
      OutDim = 2;
    };
```

TF Flexion



Code adjusted for readability

Constraint Equations

- Ensure 0-rotation about z axis of invisible relative to anatomical

```
AnyKinMeasureOrg InvisibleFemur_flexion= {
      AnyKinRotational rot = {
        Type = RotAxesAngles;
        Axis1 = z;
        Axis2 = x;
        Axis3 = y;
        AnyRefFrame &ref1 = Thigh;
        AnyRefFrame &ref2 = InvisibleFemur_TF;
      };
      MeasureOrganizer = {0};
    };
```

$$\boldsymbol{\Phi}(\mathbf{q}) = \begin{bmatrix} {}_{(F)}\mathbf{f}_{MP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}A_{(IF)}\mathbf{s}'_{MP} \\ {}_{(T)}\mathbf{f}_{MP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}A_{(IT)}\mathbf{s}'_{MP} \\ \left({}_{(IF)}A^{T}\left({}_{(F)}\mathbf{f}_{LP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}A_{(IF)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ \left({}_{(IT)}A^{T}\left({}_{(T)}\mathbf{f}_{LP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}A_{(IT)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ {}_{(IF)}\theta_z \\ {}_{(IT)}\theta_z \\ {}_{(IF)}\mathbf{r} - {}_{(IT)}\mathbf{r} \\ {}_{(IF)}\mathbf{a}_z^{T}\,{}_{(IT)}\mathbf{a}_y \\ {}_{(IF)}\mathbf{a}_z^{T}\,{}_{(IT)}\mathbf{a}_x \\ {}_{(F)}\mathbf{p}^{T}\,{}_{(F)}\mathbf{p} - 1 \\ {}_{(IF)}\mathbf{p}^{T}\,{}_{(IF)}\mathbf{p} - 1 \\ {}_{(T)}\mathbf{p}^{T}\,{}_{(T)}\mathbf{p} - 1 \\ {}_{(IT)}\mathbf{p}^{T}\,{}_{(IT)}\mathbf{p} - 1 \end{bmatrix} = \mathbf{0}$$

where

$${}_{(F)}\mathbf{f}_{MP} = \begin{cases} {}_{(T)}\mathbf{r}_{MEFC} & \theta_{TF} < \theta^{0}_{EOS} \\ {}_{(T)}\mathbf{r}_{MEFC}(1-\nu) + {}_{(T)}\mathbf{r}_{MFFC}\nu & \theta^{0}_{EOS} \leqslant \theta_{TF} \leqslant \theta^{90}_{EOS} \\ {}_{(T)}\mathbf{r}_{MFFC} & \theta^{90}_{EOS} < \theta_{TF} \end{cases}$$

For more details see: https://doi.org/10.1016/j.jbiomech.2018.02.032

Constraint Equations

- Enforce revolute joint between invisible segments
- Euler parameters have unity length

```
AnyRevoluteJoint Inv_TF = {
   Axis = z;
   AnyRefFrame &InvFemur = InvisibleFemur_TF;
   AnyRefFrame &InvTibia = InvisibleTibia;
 };
```

$$\mathbf{\Phi}(\mathbf{q}) = \begin{bmatrix} {}_{(F)}\mathbf{f}_{MP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}\mathrm{A}_{(IF)}\mathbf{s}'_{MP} \\ {}_{(T)}\mathbf{f}_{MP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}\mathrm{A}_{(IT)}\mathbf{s}'_{MP} \\ \left({}_{(IF)}\mathrm{A}^{\mathrm{T}}\left({}_{(F)}\mathbf{f}_{LP} - {}_{(IF)}\mathbf{r} - {}_{(IF)}\mathrm{A}_{(IF)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ \left({}_{(IT)}\mathrm{A}^{\mathrm{T}}\left({}_{(T)}\mathbf{f}_{LP} - {}_{(IT)}\mathbf{r} - {}_{(IT)}\mathrm{A}_{(IT)}\mathbf{s}'_{LP}\right)\right)_{xy} \\ {}_{(IF)}\theta_z \\ {}_{(IT)}\theta_z \\ {}_{(IF)}\mathbf{r} - {}_{(IT)}\mathbf{r} \\ {}_{(IF)}\mathbf{a}_z^T\,{}_{(IT)}\mathbf{a}_y \\ {}_{(IF)}\mathbf{a}_z^T\,{}_{(IT)}\mathbf{a}_x \\ {}_{(F)}\mathbf{p}^T\,{}_{(F)}\mathbf{p} - 1 \\ {}_{(IF)}\mathbf{p}^T\,{}_{(IF)}\mathbf{p} - 1 \\ {}_{(T)}\mathbf{p}^T\,{}_{(T)}\mathbf{p} - 1 \\ {}_{(IT)}\mathbf{p}^T\,{}_{(IT)}\mathbf{p} - 1 \end{bmatrix} = \mathbf{0}$$

where

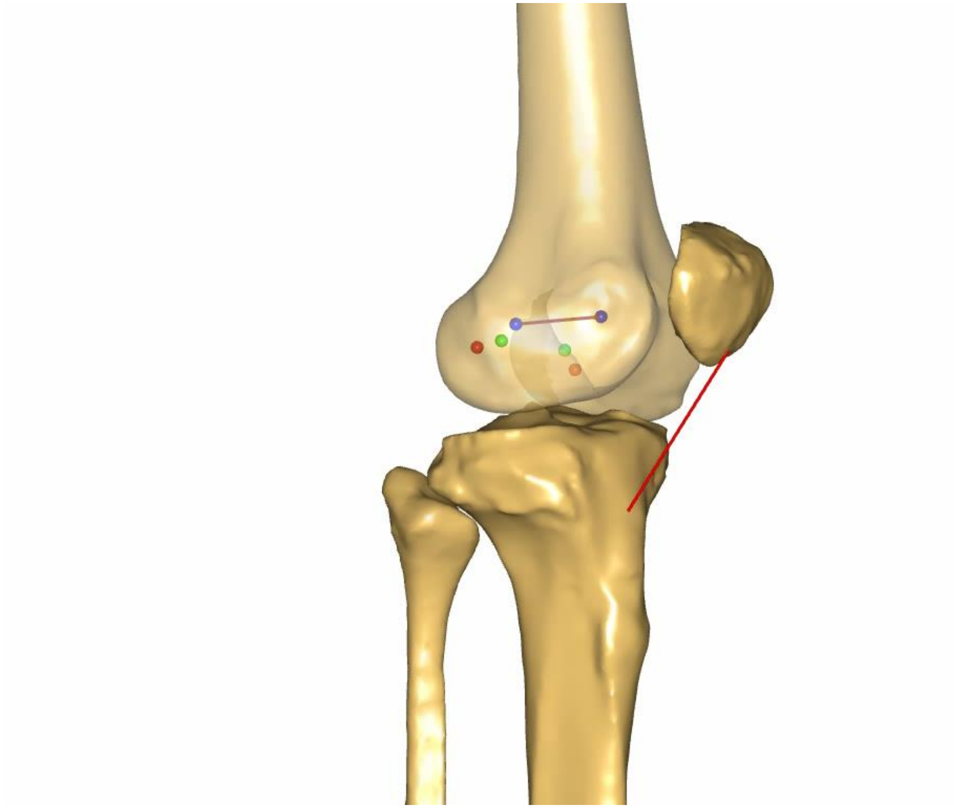
$${}_{(F)}\mathbf{f}_{MP} = \begin{cases} {}_{(T)}\mathbf{r}_{MEFC} & \theta_{TF} < \theta^0_{EOS} \\ {}_{(T)}\mathbf{r}_{MEFC}(1-v) + {}_{(T)}\mathbf{r}_{MFFC}v & \theta^0_{EOS} \leqslant \theta_{TF} \leqslant \theta^{90}_{EOS} \\ {}_{(T)}\mathbf{r}_{MFFC} & \theta^{90}_{EOS} < \theta_{TF} \end{cases}$$

For more details see: https://doi.org/10.1016/j.jbiomech.2018.02.032

AnyBody Modeling System

Concept of Personalization levels:

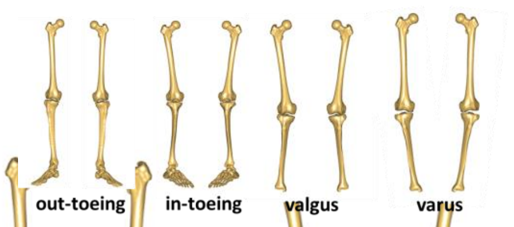
- Generic scalable moving-axis ✔
- What if subject has malalignment?
- What if user has subject-specific STL bone positions?

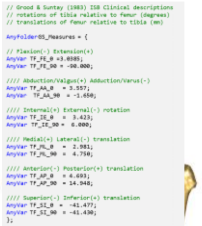


What can you expect next…?

- User-adjustments

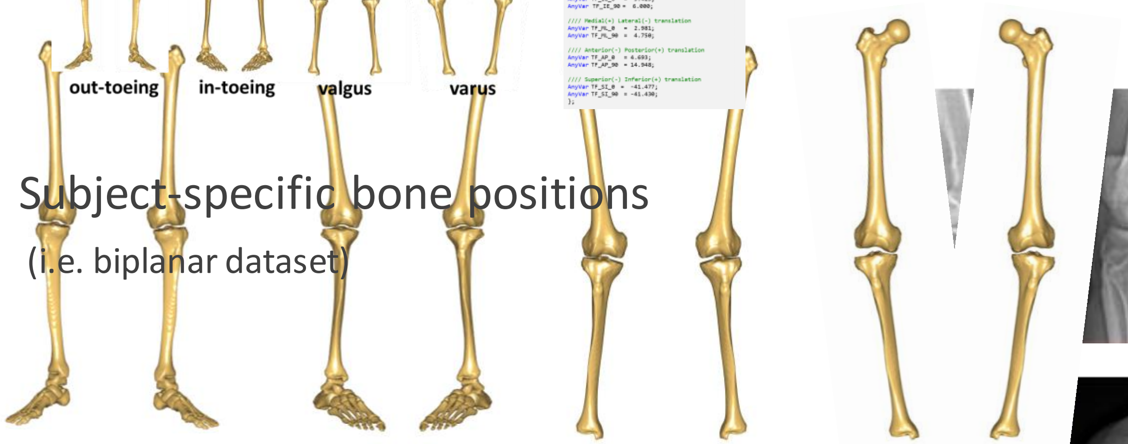
  (known varus/valgus, in-toeing/out-toeing)



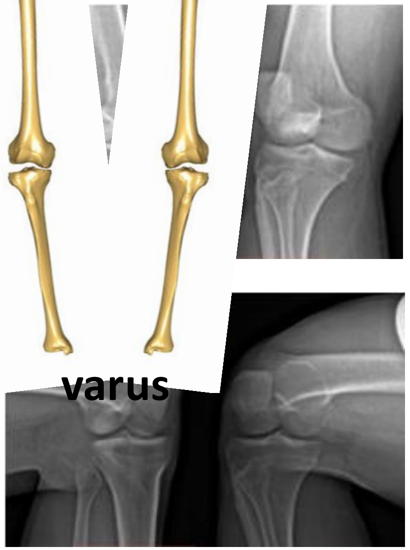


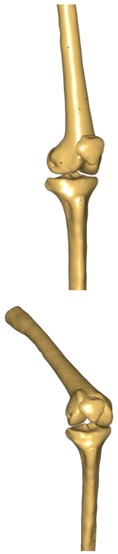
- Subject-specific bone positions

  (i.e. biplanar dataset)



out-toeing in-toeing valgus varus



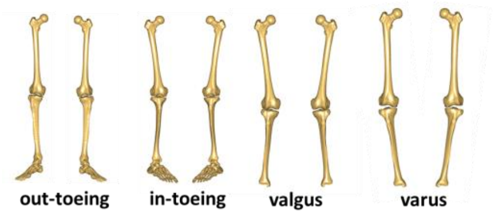


```
// Grood & Suntay (1983) ISB Clinical descriptions
// rotations of tibia relative to femur (degrees)
// translations of femur relative to tibia (mm)

AnyFolder GS_Measures  = {

// Flexion(-) Extension(+)
AnyVar TF_FE_0 =3.0385;
AnyVar TF_FE_90 = -90.000;

//// Abduction/Valgus(+) Adduction/Varus(-)
AnyVar TF_AA_0    = 3.557;
AnyVar  TF_AA_90   = -1.650;

//// Internal(+) External(-) rotation
AnyVar TF_IE_0    =  3.423;
AnyVar  TF_IE_90 =  6.000;

//// Medial(+) Lateral(-) translation
AnyVar TF_ML_0    =  2.981;
AnyVar TF_ML_90   =  4.750;

//// Anterior(-) Posterior(+) translation
AnyVar TF_AP_0    = 4.693;
AnyVar TF_AP_90   = 14.948;

//// Superior(-) Inferior(+) translation
AnyVar TF_SI_0   =  -41.477;
AnyVar TF_SI_90   = -41.430;
};
```

What can you expect next...?

- User-adjustments

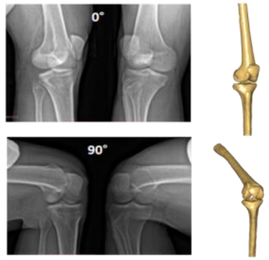
  (known varus/valgus, in-toeing/out-toeing)



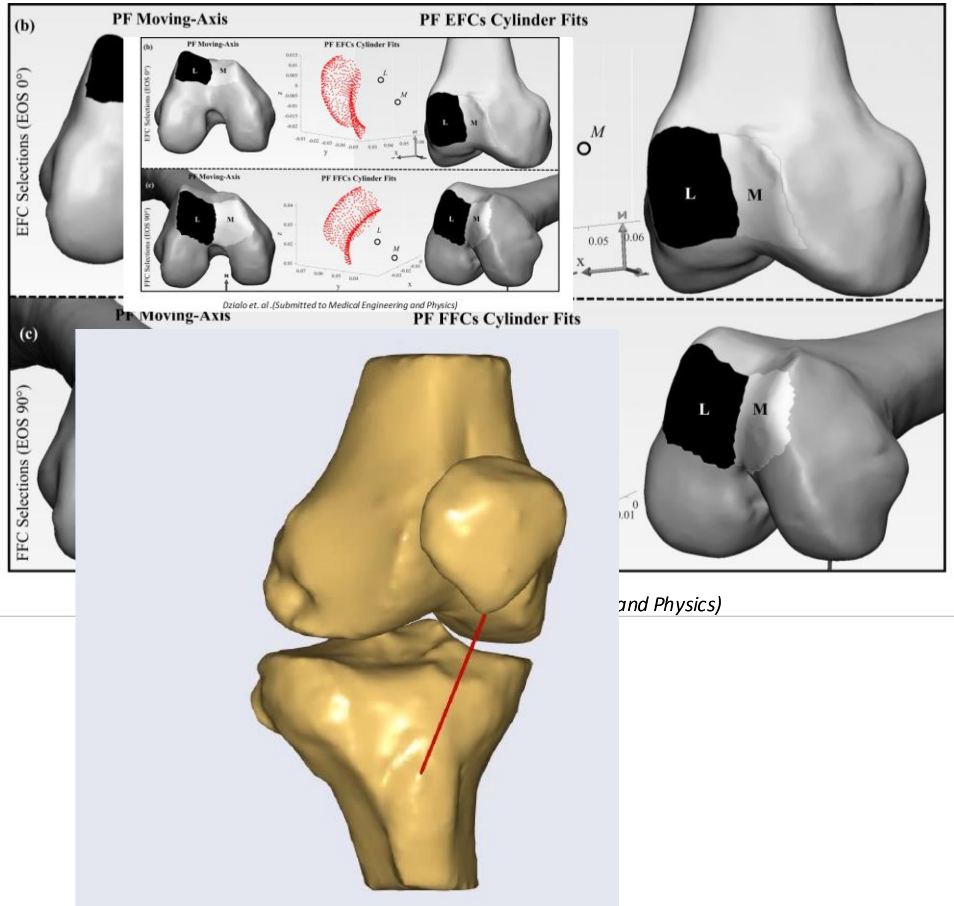

```
// Grood & Suntay (1983) ISB Clinical descriptions
// rotations of tibia relative to femur (degrees)
// translations of femur relative to tibia (mm)

AnyFolder OS_Measures = {

// Flexion(-) Extension(+)
AnyVar TF_FE_0 =3.0385;
AnyVar TF_FE_90 = -90.000;

//// Abduction/Valgus(+) Adduction/Varus(-)
AnyVar TF_AA_0   = 3.557;
AnyVar  TF_AA_90  = -1.650;

//// Internal(+) External(-) rotation
AnyVar TF_IE_0   =  3.423;
AnyVar TF_IE_90 =  6.000;

//// Medial(+) Lateral(-) translation
AnyVar TF_ML_0   =  2.981;
AnyVar TF_ML_90  =  4.750;

//// Anterior(-) Posterior(+) translation
AnyVar TF_AP_0   = 4.693;
AnyVar TF_AP_90  = 14.948;

//// Superior(-) Inferior(+) translation
AnyVar TF_SI_0  =  -41.477;
AnyVar TF_SI_90  = -41.430;
};
```

- Subject-specific bone positions

  (i.e. biplanar dataset)



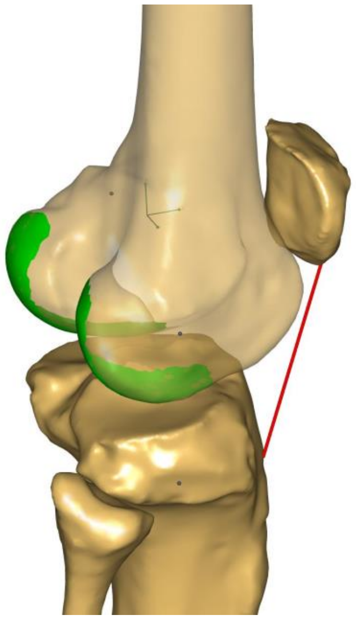
- Patellofemoral moving-axis joint



and Physics)

Adding your own knee joint….

```
// Exclude right and left knee joints
#define BM_JOINT_TYPE_KNEE_RIGHT _JOINT_TYPE_USERDEFINED_
#define BM_JOINT_TYPE_KNEE_LEFT _JOINT_TYPE_USERDEFINED_

// Add in new knee joint
#include "../../KneeModels/YourOwnKnee/AddOnKnee.any"
//#include "../../KneeModels/MovingAxisKnee/AddOnKnee.any"
```



- Wants to create new hinge model based off surface fitting

Name
- MovingAxisKnee
- New folder

SurfaceFitHinge

Name
- STLs
- AddOnKnee.any
- Femur_FitCylinders.any
- Joints_Drivers.any
- SubjectSpecificScaling.any
- Tibia_AddOn.any

- Lateral_TF.stl
- Medial_TF.stl
- Lateral_TF.anysurf3
- Medial_TF.anysurf3

Adding your own knee joint….

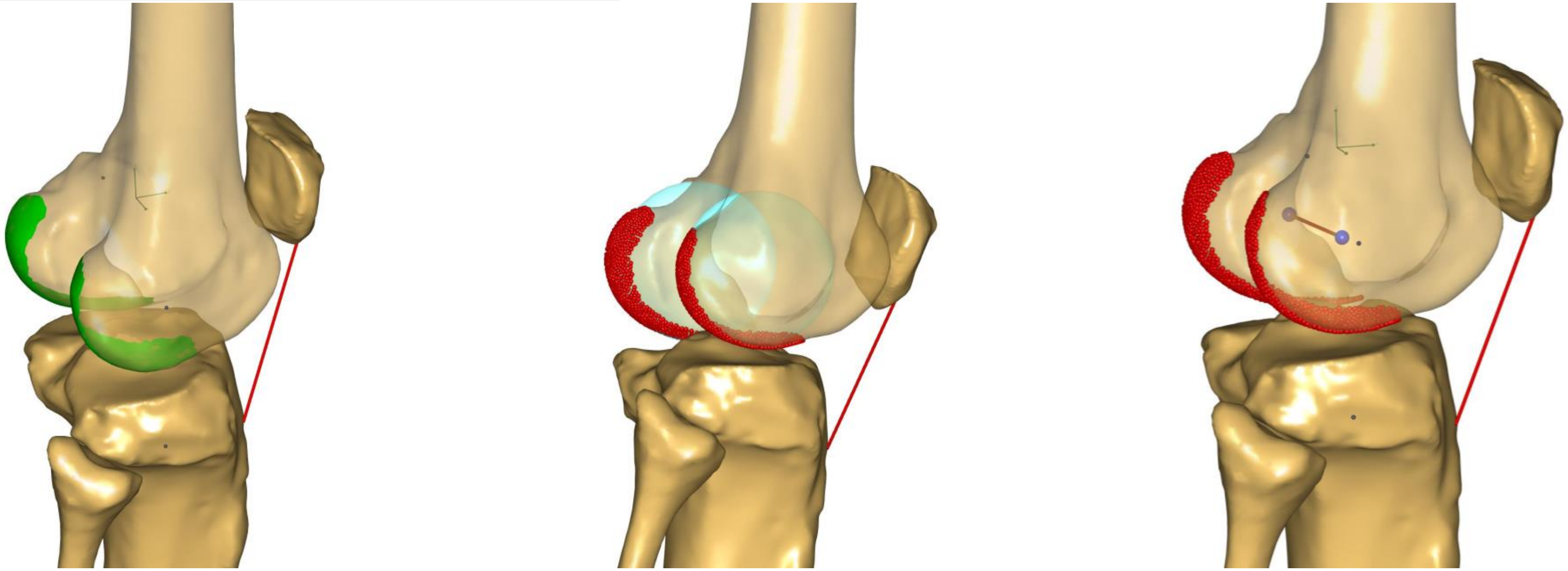
```
// Add in new knee joint
#include "../../KneeModels/SurfaceFitHinge/AddOnKnee.any"
```

- **KneeJoint nodes (Thigh & Shank)**
- **Thigh.KneeJoint.sRelUnscaled**
- **Jnts.Knee.Constraint.Reaction**

```
#include "SubjectSpecificScaling.any"
#include "Femur_FitCylinders.any"
#include "TibiaAddOn.any"
#include "Joints_Drivers.any"
```

Adding your own knee joint….

```
#include "SubjectSpecificScaling.any"
#include "Femur_FitCylinders.any"
```



```
AnyRefNode KneeJoint = {
  sRel = (.Lateral_TF.sRel + .Medial_TF.sRel)/2.0;
  ARel = RotMat(sRel, .Lateral_TF.sRel, .HipJoint.sRel)*RotMat(...Sign*pi/2,y);

AnyVec3 sRelUnscaled = ((.StdPar.EpicondylusFemorisLateralis)+(.StdPar.EpicondylusFemorisMedialis))/2;
};
```

Adding your own knee joint....

```
#include "TibiaAddOn.any"
```

- Define Lateral_TF, Medial_TF, and HipJoint nodes
    - Using Thigh.CT_Data (defined in Femur_FitCylinders.any)
    - CT2scaledSeg transformation (defined in SubjectSpecificScaling.any)

```
AnyRefNode KneeJoint = {
  sRel = (.Lateral_TF.sRel + .Medial_TF.sRel)/2.0;
  ARel = RotMat(sRel, .Lateral_TF.sRel, .HipJoint.sRel)*RotMat(...Sign*pi/2,y);
};
```

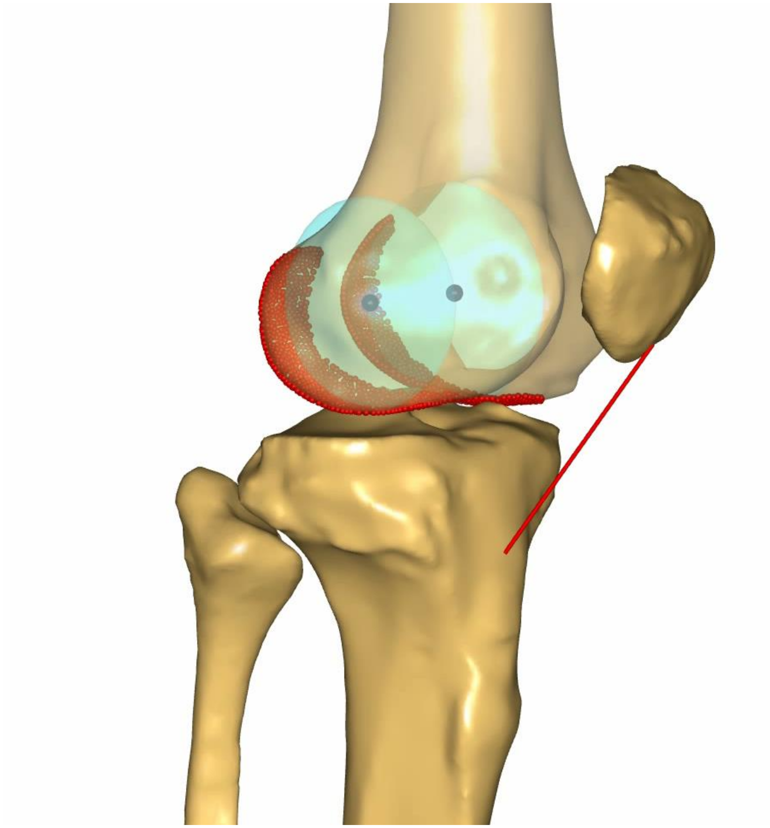
Code adjusted for readability

Adding your own knee joint....

```
#include "Joints_Drivers.any"
```

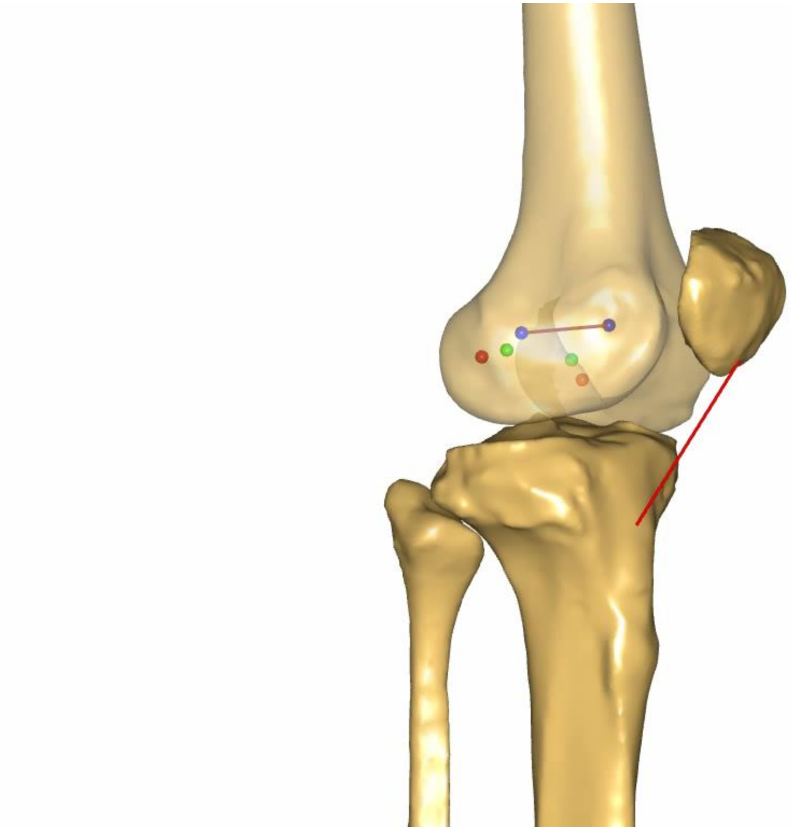
```
AnyRevoluteJoint Knee = {
    Axis = z;
    AnyRefFrame &Thigh = Thigh.KneeJoint;
    AnyRefFrame &Shank = Shank.KneeJoint;
  };
```

Re-establishes Knee.Constraint.Reactions

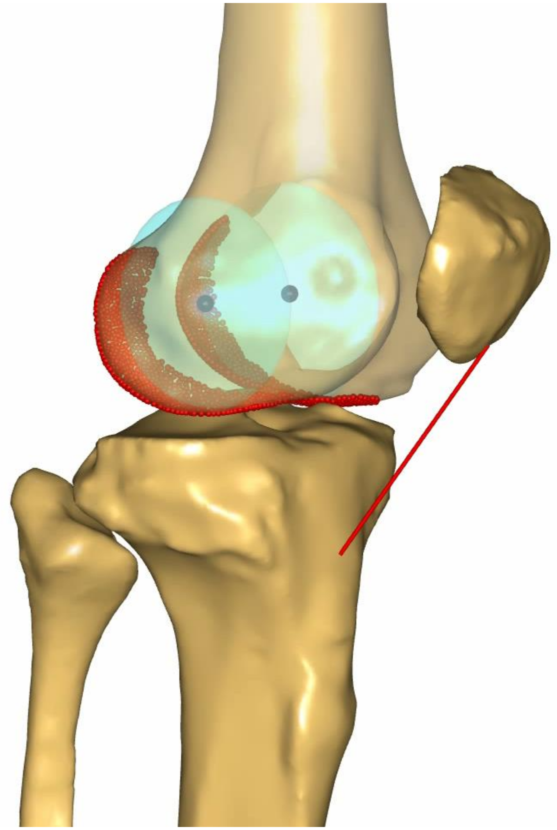


Try out AnyKnee...

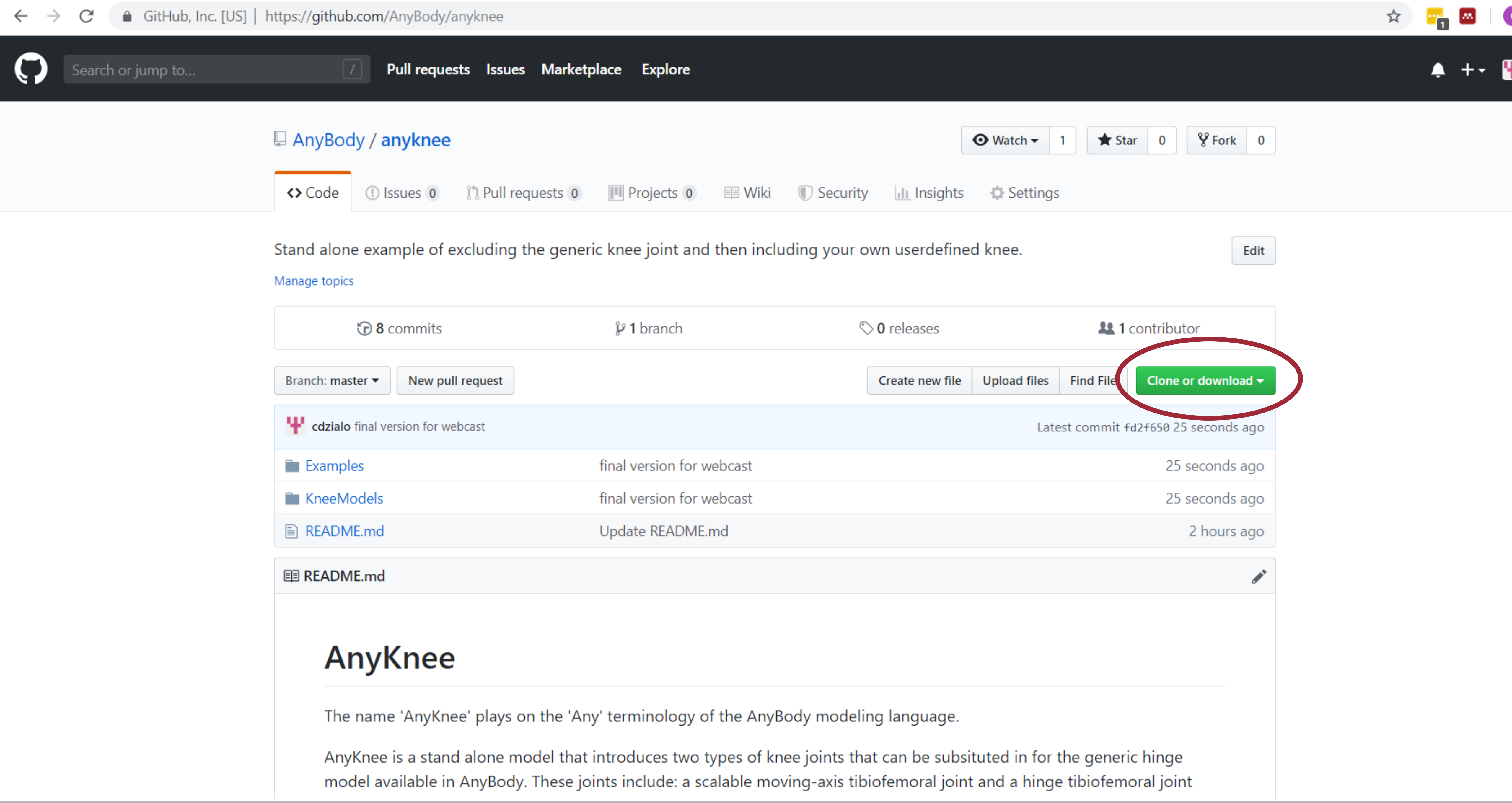
https://github.com/AnyBody/anyknee



Moving-axis



Surface Fit Hinge


GitHub, Inc. [US] | https://github.com/AnyBody/anyknee
Search or jump to...
Pull requests
Issues
Marketplace
Explore
AnyBody / anyknee
Watch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
Stand alone example of excluding the generic knee joint and then including your own userdefined knee.
Edit
Manage topics
8 commits
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
cdzialo final version for webcast
Latest commit fd2f650 25 seconds ago
Examples
final version for webcast
25 seconds ago
KneeModels
final version for webcast
25 seconds ago
README.md
Update README.md
2 hours ago
README.md
AnyKnee
The name 'AnyKnee' plays on the 'Any' terminology of the AnyBody modeling language.
AnyKnee is a stand alone model that introduces two types of knee joints that can be subsituted in for the generic hinge
model available in AnyBody. These joints include: a scalable moving-axis tibiofemoral joint and a hinge tibiofemoral joint