Manual SimTK optcntrlmuscle (v2.1)

Friedl De Groote, Maarten Afschrift, Tom Van Wouwe, Antoine Falisse

07/11/2018

Contents

1 Release notes

1.1 Release 2.1

- CasADi was added as an alternative to GPOPS-II and ADiGator (see section 2 for details).
- The reserve actuators (RActivation) were unscaled in the output of the main functions.
- The time derivatives of the muscle contraction dynamics states, i.e. normalized muscle fiber velocities or derivatives of normalized tendon forces, were added to the cost function with a small weighting factor to prevent spiky outputs.
- The tendon stiffness was added as an optional user parameter (see section 9.3 for example).

2 Overview

The provided MATLAB code solves the muscle redundancy problem using direct collocation as described in *De Groote F, Kinney AL, Rao AV, Fregly BJ. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. Annals of Biomedical Engineering (2016).* `http://link.springer.com/article/10.1007%2Fs10439-016-1591-9`.

From v2.1, CasADi can be used as an alternative to GPOPS-II and ADiGator. CasADi is an open-source tool for nonlinear optimization and algorithmic differentiation (`https://web.casadi.org/`). Results using CasADi and GPOPS-II are very similar (differences can be attributed to the different direct collocation formulations and scaling). We used CasADi's Opti stack, which is a collection of CasADi helper classes that provides a close correspondence between mathematical NLP notation and computer code (`https://web.casadi.org/docs/#document-opti`). CasADi is actively maintained and developed, and has an active forum (`https://groups.google.com/forum/#!forum/casadi-users`).

From v1.1, an implicit formulation of activation dynamics can be used to solve the muscle redundancy problem. Additionally, by using the activation dynamics model proposed by Raasch et al. (1997), we could introduce a nonlinear change of variables to exactly impose activation dynamics in a continuously differentiable form, omitting the need for a smooth approximation such as described in De Groote et al. (2016). A result of this change of variables is that muscle excitations are not directly accessible during the optimization. Therefore, we replaced muscle excitations by muscle activations in the objective function. This implicit formulation is described in *De Groote F, Pipeleers G, Jonkers I, Demeulenaere B, Patten C, Swevers J, De Schutter J. A physiology based inverse dynamic analysis of human gait: potential and perspectives F. Computer Methods in Biomechanics and Biomedical Engineering (2009).* `http://www.tandfonline.com/doi/full/10.1080/10255840902788587`. Results from both formulations are very similar (differences can be attributed to the slightly different activation dynamics models and cost functions). However, the formulation with implicit activation dynamics (De Groote et al., (2009)) is computationally faster. This can mainly be explained by the omission of a tanh function in the constraint definition, whose evaluation is computationally expensive when solving the NLP.

Any questions? Please contact us: friedl.degroote@kuleuven.be, maarten.afschrift@kuleuven.be, tom.vanwouwe@kuleuven.be, and antoine.falisse@kuleuven.be.

3 Installation Instruction

Add the main folder and subfolder to your MATLAB path

```
addpath(genpath('C/......./SimTK_optcntrlmuscle'))).
```

Several software packages are needed to run the program

- The OpenSim MATLAB interface is used to generate the inputs to the optimal control problem based on a scaled OpenSim model and the solution of inverse kinematics (providing the solution of inverse dynamics is optional). To this aim, install OpenSim and set up the OpenSim MATLAB interface (OpenSim: https://simtk.org/frs/?group_id=91, OpenSim API: http://simtk-confluence.stanford.edu:8080/display/OpenSim/Scripting+with+Matlab).
- Using GPOPS
  - GPOPS-II is used to solve the optimal control problem using direct collocation (http://www.gpops2.com/). A one-time 30-day trial license is avaiable for all users who register.
  - ADiGator is used for automatic differentiation (https://sourceforge.net/projects/adigator/).
- Using CasADi
  - CasADi is used for nonlinear optimization and algorithmic differentiation (https://web.casadi.org/).

4 Main Function

SolveMuscleRedundancy is the main function of this program and is used to solve the muscle redundancy problem. There are eight variants of this function that differ based on the chosen optimization package (GPOPS or CasADi), the formulation of the contraction dynamics (normalized tendon force or normalized muscle fiber length as a state), and the formulation of the activation dynamics (explicit or implicit).

4.1 Using GPOPS

4.1.1 With explicit activation dynamics formulation (De Groote et al. (2016))

- SolveMuscleRedundancy_FtildeState_GPOPS uses the normalized tendon force as a state
- SolveMuscleRedundancy_lMtildeState_GPOPS uses the normalized muscle fiber length as a state

4.1.2 With implicit activation dynamics formulation (De Groote et al. (2009))

- SolveMuscleRedundancy_FtildeState_actdyn_GPOPS uses the normalized tendon force as a state
- SolveMuscleRedundancy_lMtildeState_actdyn_GPOPS uses the normalized muscle fiber length as a state

4.2 Using CasADi

4.2.1 With explicit activation dynamics formulation (De Groote et al. (2016))

- SolveMuscleRedundancy_FtildeState_CasADi uses the normalized tendon force as a state
- SolveMuscleRedundancy_lMtildeState_CasADi uses the normalized muscle fiber length as a state

4.2.2 With implicit activation dynamics formulation (De Groote et al. (2009))

- SolveMuscleRedundancy_FtildeState_actdyn_CasADi uses the normalized tendon force as a state
- SolveMuscleRedundancy_lMtildeState_actdyn_CasADi uses the normalized muscle fiber length as a state

4.3 Input Arguments

Required input arguments for SolveMuscleRedundancy

1. **model_path**: directory and filename of the scaled OpenSim model (.osim file). The code should work with any OpenSim model with valid muscle-tendon parameters for which OpenSim's Inverse Dynamics and Muscle Analysis Tools generate reliable results. Note that only the muscle-tendon parameters and not the muscle model specified in the osim-file are used (for details see Muscle model).
2. **IK_path**: directory and filename of the inverse kinematics solution (.mot file).
3. **ID_path**: directory and filename of the inverse dynamics solution (.sto file). If left empty, the inverse dynamics solution will be computed from the external loads (see Optional input arguments).
4. **time**: 1 x 2 MATLAB array with the initial and final time of the analysis in seconds. Initial and final states influence the optimal controls over a period of about 50 ms at the beginning and end of the time interval over which the optimal control problem is solved. Since in practice the initial and final states are generally unknown, problems should be solved for a time interval containing five additional data points (considering a 100Hz sampling frequency) at the beginning and end of the motion cycle. Those additional data points should not be considered in further analyses. The user should thus not be surprised to observe unrealistically high muscle activation at the beginning of the motion (more details in companion paper).
5. **Out_path**: directory where you want to store the results from the muscle analysis.
6. **Misc**: miscellaneous input arguments
   - *DofNames_Input* is a cell array specifying for which degrees of freedom you want to solve the muscle redundancy problem. Typically the muscle redundancy problem is solved for one leg at a time (there are no muscles spanning both legs).
   - *MuscleNames_Input* is a cell array that specifies the muscles to be included when solving the muscle redundancy problem. All muscles that actuate (i.e. have a moment arm with respect to) the degrees of freedom specified in *DofNames_Input* will be selected by default if this array is left empty.

Optional input arguments for SolveMuscleRedundancy

1. **Misc.***Loads_path*: directory and filename of the external loads (.xml file). The program will use the OpenSim libraries to solve the inverse dynamics problem when the required input argument ID_path is empty and Misc.*Loads_path* points to an external loads file.

2. **Misc.***ID_ResultsPath*: directory where the inverse dynamics results will be saved when the input argument *ID_path* is left empty.

3. **Misc.***f_cutoff_ID*: cutoff frequency for the butterworth recursive low pass filter applied to the inverse dynamics data (default is 6 Hz).

4. **Misc.***f_order_ID*: order of the butterworth recursive low pass filter applied to the inverse dynamics data (default is 6).

5. **Misc.***f_cutoff_LMT*: cutoff frequency for the butterworth recursive low pass filter applied to the muscle tendon lengths from the muscle analysis (default is 6 Hz).

6. **Misc.***f_order_LMT*: order of the butterworth recursive low pass filter applied to the muscle tendon lengths from the muscle analysis (default is 6).

7. **Misc.***f_cutoff_dM*: cutoff frequency for the butterworth recursive low pass filter applied to the muscle moment arms from the muscle analysis (default is 6 Hz).

8. **Misc.***f_order_dM*: order of the butterworth recursive low pass filter applied to the muscle moment arms from the muscle analysis (default is 6).

9. **Misc.***f_cutoff_IK*: cutoff frequency for the butterworth recursive low pass filter applied to the inverse kinematics data (default is 6 Hz) when performing the muscle analysis to compute muscle-tendon lengths and moment arms.

10. **Misc.***f_order_IK*: order of the butterworth recursive low pass filter applied to the inverse kinematics data (default is 6).

11. **Misc.***Mesh_Frequency*: number of mesh interval per second (default is 100, but a denser mesh might be required to obtain the desired accuracy especially for faster motions).

12. **Misc.***Atendon*: vector with tendon stiffness for the selected muscles. The order should correspond to *MuscleNames_Input*. The default value is 35 and a lower value corresponds to a more compliant tendon. The default value will be used when left empty. An example is provided in section 9.3 to set a different stiffness to the Achilles tendon.

4.4 Output arguments

4.4.1 Using GPOPS

1. Time: time vector.

2. MExcitation: optimal muscle excitation (matrix dimension: number of collocation points x number of muscles).

3. MActivation: optimal muscle activation (matrix dimension: number of collocation points x number of muscles).

4. RActivation: activation of the reserve actuators (matrix dimension: number of collocation points x number of degrees of freedom).

5. TForcetilde: normalized tendon force (matrix dimension: number of collocation points x number of muscles).

6. TForce: tendon force (matrix dimension: number of collocation points x number of muscles).

7. lMtilde: normalized muscle fiber length (matrix dimension: number of collocation points x number of muscles).

8. lM: muscle fiber length (matrix dimension: number of collocation points x number of muscles) .

9. MuscleNames: cell array that contains the names of the selected muscles (matrix dimension: number of muscles).

10. OptInfo: output structure created by GPOPS-II.

11. DatStore: data structure with input information for the optimal control problem.

4.4.2 Using CasADi

CasADi uses piecewise-constant controls in the mesh intervals. We therefore distinguish between collocation points (for the states) and mesh points (for the states and controls) in the output arguments.

1. Time: time vector
   (a) Time.meshPoints: time at mesh points
   (b) Time.collocationPoints: time at collocation points

2. MExcitation.meshPoints: muscle excitation (matrix dimension: number of mesh points points x number of muscles).

3. MActivation: muscle activation
   (a) MActivation.meshPoints: muscle activation at mesh points (matrix dimension: number of mesh points x number of muscles).
   (b) MActivation.collocationPoints: muscle activation at collocation points (matrix dimension: number of collocation points x number of muscles).

4. RActivation.meshPoints: activation of the reserve actuators (matrix dimension: number of mesh points x number of degrees of freedom).

5. TForcetilde: normalized tendon force
   (a) TForcetilde.meshPoints: normalized tendon force at mesh points (matrix dimension: number of mesh points x number of muscles).
   (b) TForcetilde.collocationPoints: normalized tendon force at collocation points (matrix dimension: number of collocation points x number of muscles).

6. TForce: tendon force

   (a) TForce.meshPoints: tendon force at mesh points (matrix dimension: number of mesh points x number of muscles).
   (b) TForce.collocationPoints: tendon force at collocation points (matrix dimension: number of collocation points x number of muscles).

7. lMtilde: normalized muscle fiber length
   (a) lMtilde.meshPoints: normalized muscle fiber length at mesh points (matrix dimension: number of mesh points x number of muscles).
   (b) lMtilde.collocationPoints: normalized muscle fiber length at collocation points (matrix dimension: number of collocation points x number of muscles).

8. lM: muscle fiber length
   (a) lM.meshPoints: muscle fiber length at mesh points (matrix dimension: number of mesh points x number of muscles).
   (b) lM.collocationPoints: muscle fiber length at collocation points (matrix dimension: number of collocation points x number of muscles).

9. MuscleNames: cell array that contains the names of the selected muscles (matrix dimension: number of muscles).

10. OptInfo: output structure with settings used in CasADi.

11. DatStore: data structure with input information for the optimal control problem.

5 GPOPS-II

5.1 Setup

The GPOPS-II setup is accessible through the function SolveMuscleRedundancy_< ... >_GPOPS.m under GPOPS setup. The user is referred to the GPOPS-II user guide for setup options. A higher accuracy can be reached by adjusting, for instance, the number of mesh intervals. This however comes at the expense of the computational time. 100 mesh intervals per second are used by default.

5.2 Output

The output variable OptInfo contains all information related to the optimal control problem solution. Convergence to an optimal solution is reached when output.result.nlpinfo is flagged 0 ("EXIT: Optimal solution found" in the command window of MATLAB). The mesh accuracy can be assessed with output.result.maxerrors. Cost functional, control, state (and costate) can be accessed in output.result.solution.phase.

To recall, the user should consider extending the time interval by 50-100 ms at the beginning and end of the motion to limit the influence of the unknown initial and final state on the solution. Results from those additional periods should not be considered realistic and will typically result in high muscle activation.

6 CasADi

6.1 Setup

The CasADi setup is accessible through the function SolveMuscleRedundancy_< ... >_CasADi.m under CasADi setup. The user is referred to the CasADi user guide for setup options.

6.2 Output

The output variable OptInfo contains information related to the optimal control problem settings. As with GPOPS, the user should consider extending the time interval by 50-100 ms at the beginning and end of the motion to limit the influence of the unknown initial and final state on the solution. Results from those additional periods should not be considered realistic and will typically result in high muscle activation.

7 Muscle model

The musculotendon properties are fully described in the supplementary materials of the aforementioned publication. Importantly, only the tendon slack length, optimal muscle fiber length, maximal isometric muscle force, optimal pennation angle and maximal muscle fiber contraction velocity are extracted from the referred OpenSim model. Other properties are defined in the code and can be changed if desired. By default, the activation and deactivation time constants are 15 and 60 ms respectively (see tau_act and tau_deact in SolveMuscleRedundancy_< *state* >.m).

8 Tips and Tricks

1. We advise users to use the formulations with normalized tendon force as a state. They appear to have improved convergence properties as compared to formulations with normalized muscle fiber length as a state. This might be due to the more linear relationship between muscle activation and tendon force than between muscle activation and muscle length.

2. We advise users to use the formulations with implicit activation dynamics as they are in most cases computationally more efficient.

3. If you observe spiky output variables (e.g. muscle excitations), try increasing the mesh frequency. This might improve the results although it might also increase the computational time.

9 Examples

Four examples are provided in the folder examples.

9.1 Walking example De Groote et al. 2016

```
clear all;close all;clc

%% Choose optimization framework
% framework = 'GPOPS';
framework = 'CasADi';

```

```
%% Choose contraction dynamics formulation
% formulation_contdyn = 'lMtildeState';
formulation_contdyn = 'FtildeState';

%% Choose activation dynamics formulation
% formulation_actdyn = 'DeGroote2016';
formulation_actdyn = 'DeGroote2009';

%% Example
% Add main folder and subfolder to matlab path (installation)
filepath=which('Walking_DeGrooteetal2016.m');
[DirExample_Walking,¬,¬]=fileparts(filepath);
[DirExample,¬]=fileparts(DirExample_Walking);
[MainDir,¬]=fileparts(DirExample);
addpath(genpath(MainDir));

% Needed Input Arguments
IK_path=fullfile(MainDir,'Examples','Walking_DeGrooteetal2016','WalkingData','Walking_IK.mot');
ID_path=fullfile(MainDir,'Examples','Walking_DeGrooteetal2016','WalkingData','Walking_ID.sto');
model_path=fullfile(MainDir,'Examples','Walking_DeGrooteetal2016','WalkingData','subject1.osim');
time=[0.516 1.95]; % Right stance phase (+50ms beginning and end of time interval, more ... details see manual and publication)
Out_path=fullfile(MainDir,'Examples','Walking_DeGrooteetal2016','Results');

Misc.DofNames_Input={'ankle_angle_r','knee_angle_r','hip_flexion_r','hip_rotation_r','hip_adduction_r'};
Misc.MuscleNames_Input={}; % Selects all muscles for input DOFs when empty

% Optional Input Arguments
Misc.f_cutoff_ID = 6;     % cutoff frequency filtering ID
Misc.f_order_ID = 4;      % order frequency filtering ID
Misc.f_cutoff_lMT = 6;    % cutoff frequency filtering lMT
Misc.f_order_lMT = 4;     % order frequency filtering lMT
Misc.f_cutoff_dM= 6;      % cutoff frequency filtering MA
Misc.f_order_dM = 4;      % order frequency filtering MA
Misc.f_cutoff_IK= 6;      % cutoff frequency filtering IK
Misc.f_order_IK = 4;      % order frequency filtering IK

%% Solve the problem
switch framework
    case 'GPOPS'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
    case 'CasADi'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
```

```
                            lMtilde_actdyn , lM_actdyn , MuscleNames_actdyn , ...
                            OptInfo_actdyn , DatStore_actdyn]= ...
                            SolveMuscleRedundancy_lMtildeState_actdyn_CasADi( ...
                            model_path , IK_path , ID_path , time , Out_path , Misc);
                    case 'FtildeState'
                        [Time_actdyn , MExcitation_actdyn , MActivation_actdyn , ...
                            RActivation_actdyn , TForcetilde_actdyn , TForce_actdyn , ...
                            lMtilde_actdyn , lM_actdyn , MuscleNames_actdyn , ...
                            OptInfo_actdyn , DatStore_actdyn]= ...
                            SolveMuscleRedundancy_FtildeState_actdyn_CasADi( ...
                            model_path , IK_path , ID_path , time , Out_path , Misc);
                end
        end
end
```

9.2 Running example De Groote et al. 2016

```
clear all;close all;clc

%% Choose optimization framework
% framework = 'GPOPS';
framework = 'CasADi';

%% Choose contraction dynamics formulation
% formulation_contdyn = 'lMtildeState';
formulation_contdyn = 'FtildeState';

%% Choose activation dynamics formulation
% formulation_actdyn = 'DeGroote2016';
formulation_actdyn = 'DeGroote2009';

%% Example
% Add main folder and subfolder to matlab path (installation)
filepath=which('Running_DeGrooteetal2016.m');
[DirExample_Running,¬,¬]=fileparts(filepath);
[DirExample,¬]=fileparts(DirExample_Running);
[MainDir,¬]=fileparts(DirExample);
addpath(genpath(MainDir));

% Needed Input Arguments
IK_path=fullfile(MainDir,'Examples','Running_DeGrooteetal2016','RunningData','Running_IK.mot');
ID_path=fullfile(MainDir,'Examples','Running_DeGrooteetal2016','RunningData','Running_ID.sto');
model_path=fullfile(MainDir,'Examples','Running_DeGrooteetal2016','RunningData','subject1.osim');
time=[0.05 0.98]; % Right stance phase (+50ms beginning and end of time interval, more details ...
    see manual and publication)
Out_path=fullfile(MainDir,'Examples','Running_DeGrooteetal2016','Results');

Misc.DofNames_Input={'ankle_angle_r','knee_angle_r','hip_flexion_r','hip_adduction_r','hip_rotation_r'};
Misc.MuscleNames_Input={}; % Selects all muscles for input DOFs when empty

% Optional Input Arguments
Misc.f_cutoff_ID = 10;    % cutoff frequency filtering ID
Misc.f_order_ID = 5;      % order frequency filtering ID
Misc.f_cutoff_lMT = 10;   % cutoff frequency filtering lMT
Misc.f_order_lMT = 5;     % order frequency filtering lMT
Misc.f_cutoff_dM= 10;     % cutoff frequency filtering MA
Misc.f_order_dM = 5;      % order frequency filtering MA
Misc.f_cutoff_IK= 10;     % cutoff frequency filtering IK
Misc.f_order_IK = 5;      % order frequency filtering IK

%% Solve the problem
switch framework
    case 'GPOPS'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde, ...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]= ...
                            SolveMuscleRedundancy_lMtildeState_GPOPS( ...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde, ...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]= ...
                            SolveMuscleRedundancy_FtildeState_GPOPS( ...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn , MExcitation_actdyn , MActivation_actdyn , ...
                            RActivation_actdyn , TForcetilde_actdyn , TForce_actdyn , ...
                            lMtilde_actdyn , lM_actdyn , MuscleNames_actdyn , ...
                            OptInfo_actdyn , DatStore_actdyn]= ...
                            SolveMuscleRedundancy_lMtildeState_actdyn_GPOPS( ...
                            model_path , IK_path , ID_path , time , Out_path , Misc);
                    case 'FtildeState'
```

```
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
    case 'CasADi'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
end
```

9.3 OpenSim installation example Gait10dof18m

```
clear all;close all;clc

%% Choose optimization framework
% framework = 'GPOPS';
framework = 'CasADi';

%% Choose contraction dynamics formulation
% formulation_contdyn = 'lMtildeState';
formulation_contdyn = 'FtildeState';

%% Choose activation dynamics formulation
% formulation_actdyn = 'DeGroote2016';
formulation_actdyn = 'DeGroote2009';

%% Example
% Add main folder and subfolder to matlab path (installation)
filepath=which('Example_Gait10dof18m.m');
[DirExample,¬,¬]=fileparts(filepath);
[DirExample2,¬,¬]=fileparts(DirExample);
[MainDir,¬]=fileparts(DirExample2);
addpath(genpath(MainDir));

% Needed Input Arguments
Datapath='C:\OpenSim 3.3\Models\Gait10dof18musc\OutputReference';
IK_path=fullfile(Datapath,'IK','subject01_walk_IK.mot');
ID_path=[]; % Compute ID from the external loads
model_path=fullfile(Datapath,'subject01.osim');
time=[0.7 1.4]; % Part of the right stance phase
Out_path=fullfile(MainDir,'Examples','OpenSimInstallation_Gait10dof18m','Results');

Misc.DofNames_Input={'ankle_angle_r','knee_angle_r','hip_flexion_r'};
Misc.Loads_path=fullfile(Datapath,'ExperimentalData','subject01_walk_grf.xml');
Misc.ID_ResultsPath=fullfile(Datapath,'ID','inversedynamics.sto');

% Optional Input Arguments
% Here is an example of how to adjust the Achilles tendon stiffness.
% We first add the input argument MuscleNames_Input with ALL muscles
% that actuate the degrees of freedom listed in DofNames_Input.
Misc.MuscleNames_Input={'hamstrings_r','bifemsh_r','glut_max_r',...
    'iliopsoas_r','rect_fem_r','vasti_r','gastroc_r','soleus_r',...
    'tib_ant_r'};
% We then change the compliance of the Achilles tendon by changing the
```

```
% parameter Atendon of the gastrocnemius and the soleus. The default
% value is 35 and a lower value will result in a more compliant tendon.
Misc.Atendon=[35,35,35,35,35,35,15,15,35];

%% Solve the problem
switch framework
    case 'GPOPS'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
    case 'CasADi'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
end
```

9.4 OpenSim installation example Gait23dof54m

```
clear all;close all;clc

%% Choose optimization framework
% framework = 'GPOPS';
framework = 'CasADi';

%% Choose contraction dynamics formulation
% formulation_contdyn = 'lMtildeState';
formulation_contdyn = 'FtildeState';

```

```
%% Choose activation dynamics formulation
% formulation_actdyn = 'DeGroote2016';
formulation_actdyn = 'DeGroote2009';

%% Example
% add main folder and subfolder to matlab path (installation)
filepath=which('Example_Gait23dof54m.m');
[DirExample,¬,¬]=fileparts(filepath);
[DirExample2,¬,¬]=fileparts(DirExample);
[MainDir,¬]=fileparts(DirExample2);
addpath(genpath(MainDir));

% Needed Input Arguments
Datapath='C:\OpenSim 3.3\Models\Gait2354_Simbody\OutputReference';
IK_path=fullfile(Datapath,'subject01_walk1_ik.mot');
ID_path=fullfile(Datapath,'ResultsInverseDynamics','inverse_dynamics.sto');
model_path=fullfile(Datapath,'subject01_scaledOnly.osim');
time=[0.7 1.4]; % Part of the right stance phase
Out_path=fullfile(MainDir,'Examples','OpenSimInstallation_Gait23dof54m','Results');

Misc.DofNames_Input={'ankle_angle_r','knee_angle_r','hip_flexion_r'};
Misc.MuscleNames_Input={}; % Selects all muscles for input DOFs when empty

%% Solve the problem
switch framework
    case 'GPOPS'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_GPOPS(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
    case 'CasADi'
        switch formulation_actdyn
            case 'DeGroote2016'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_lMtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time,MExcitation,MActivation,RActivation,TForcetilde,...
                            TForce,lMtilde,lM,MuscleNames,OptInfo,DatStore]=...
                            SolveMuscleRedundancy_FtildeState_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end

            case 'DeGroote2009'
                switch formulation_contdyn
                    case 'lMtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_lMtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                    case 'FtildeState'
                        [Time_actdyn,MExcitation_actdyn,MActivation_actdyn,...
                            RActivation_actdyn,TForcetilde_actdyn,TForce_actdyn,...
                            lMtilde_actdyn,lM_actdyn,MuscleNames_actdyn,...
                            OptInfo_actdyn,DatStore_actdyn]=...
                            SolveMuscleRedundancy_FtildeState_actdyn_CasADi(...
                            model_path,IK_path,ID_path,time,Out_path,Misc);
                end
        end
end
```