An OpenSim Plugin for EMG-Constrained Static Optimization with Alternative Objective Functions

Alexander MacIntosh, M.Sc.

McMaster University

License

The plugin is linked to the following open access publications; please cite them if using it for your research:

MacIntosh A.R., Keir, P.J. (2016) An open-source model and solution method to predict co-contraction in the finger. *Submitted* to Computer Methods in Biomechanics and Biomedical Engineering.

MacIntosh, Alexander. An open-source model and solution method to predict co-contraction in the index finger. Diss. 2014. (MacSphere link to Thesis)

SimTK project: https://simtk.org/home/projects/finger

Overview

The plugin and analysis were completed with OpenSim 3.3 and Visual Studio 2013 on a 64-bit operating system.

Two changes were made to the OpenSim Static Optimization Analysis Tool to allow for:

1) Experimental EMG boundary constraints.
    a. Frame by frame EMG activations are used as the initial guess in Static Optimization, and
    b. Maximum and Minimum muscle activation are set by Experimental EMG ± [EMG_control_range]

2) Alternative objective functions.
    a. A detailed discussion regarding each objective function can be found in the thesis: "An open-source musculoskeletal model and EMG-constrained static optimization solution method to predict co-contraction in the index finger" Link to Thesis
    b. Briefly, three objective functions have been tested:

a. *Sum of Activation*- the standard method used in OpenSim

$$J = \sum_{m=1}^{N} (a_m)^p$$

where, J is the objective function, N is the number of muscles in the model, a_m is the activation of muscle *m* at a discrete time point, and *p* is the activation exponent.

b. *Shift parameter*- Forster et al. (2004)

$$J = \sum_{m=1}^{N} (a_m - a_s)^p$$

where, J is the objective function, N is the number of muscles in the model, a_m is the activation of muscle m at a discrete time point, p is the activation exponent (squared), and as is the shift parameter that changes each muscles contribution to the objective function.

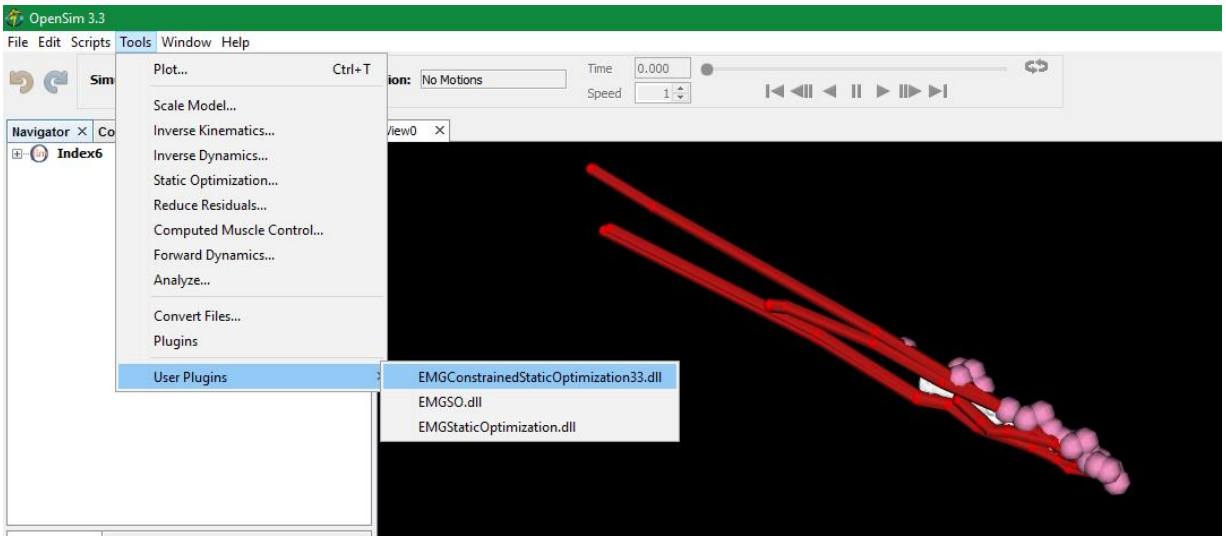
c. *Entropy Assisted*- Jiang and Mirka (2007)

$$J = (1\text{-}W) \sum_{m=1}^{N} (a_m)^p + W \sum_{m=1}^{N} a_m \log a_m$$

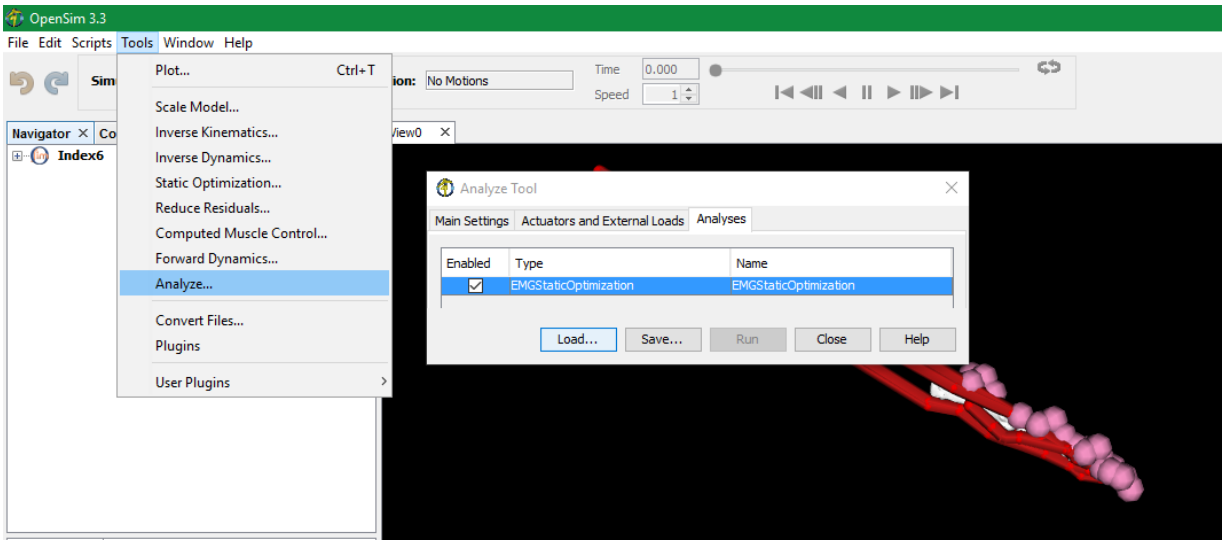
where, J is the objective function, N is the number of muscles in the model, a_m is the activation of muscle m at a discrete time point, p is the activation exponent (squared), and W is the co-contraction weight factor.

Analysis Plugin Description

After loading the plugin from: Tools >> User Plugins >> EMGConstrainedStaticOptimization33.dll



the EMGConstrainedStaticOptimization33 tool can be accessed through the Analyze Tool menu.



The following variables have been added to the Static Optimization Analysis Tool:

1. EMG_file
    a. The name of the '.sto' file containing experimental EMG data. If a file name is provided, the EMG for each actuator in the file will be used.
2. EMG_control_range
    a. The range of activation around the EMG signal that the optimizer is permitted to use. Example: if normalized EMG Activation at t_1 = 0.25 and EMG_control_range = 0.1, then the permitted range of the optimizer is 0.15 - 0.35.
3. Objective_function

    a. The objective function used by the optimizer
        i. 0 - Sum of Activation
        ii. 1 - Shift parameter
        iii. 2 - Entropy Assisted
4. Objective_actuator_names
    a. Names of the actuators on which to perform the objective function. The key word 'All' indicates that the objective function should be performed for all bodies.
    b. “All” places all muscles in the model in the order they appear in the model file.
    c. “None” for list of muscle names makes the scalar array fill with default for every entry.
5. Default_objective_actuator_scalar
    a. Relevant for objective functions 1 and 2.
    b. Default scalar value per actuator. This is used by objective functions 1 and 2
        i. In objective function 1: this value becomes the shift parameter for every muscle
        ii. In objective function 2: this value becomes the weight factor (w) for every muscle
    c. The default will be used for any actuator not specified in the Objective_actuator_names list
6. Objective_actuator_scalars
    a. Relevant for objective functions 1 and 2.
    b. In objective function 1: this value becomes the shift parameter for each muscle in the array Objective_actuator_names (in the corresponding position of the array)
    c. In objective function 2: this value becomes the weight factor (w) for each muscle in the array Objective_actuator_names (in the correspond in position of the array)

The output files are the same as in the OpenSim Native Static Optimization Analysis.

Acknowledgements

The development of this plugin was in partial completion of an M.Sc. thesis by Alexander MacIntosh. Programmatic modifications were made with the assistance of Tyler Szepesi, who's contribution is greatly appreciated.

References

Forster, E., Simon, U., Augat, P., & Claes, L. (2004). Extension of a state-of-the-art optimization criterion to predict co-contraction. Journal of Biomechanics, 37(4), 577-581.

Jiang, Z., & Mirka, G. A. (2007). Application of an Entropy-Assisted Optimization Model in Prediction of Agonist and Antagonist Muscle Forces. Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting.

For full details please see: MacIntosh, Alexander. An open-source model and solution method to predict co-contraction in the index finger. Diss. 2014. (MacSphere link to Thesis)