Tutorial 8 – Synergy Driven Treatment Optimization

The Treatment Optimization toolset generates predictive simulations of a patient's post-treatment movement function by optimizing specified treatment design parameters. The toolset consists of three tools designed using a "theme and variation" approach, where the tools are intended to be used in a specific order, with each tool serving a distinct purpose. Each tool uses the GPOPS-II direct collocation optimal control software for MATLAB and maintains a consistent structure for data inputs, problem design, cost function terms, constraint terms, and outputs, with variations.

This second section of the treatment optimization tutorial will cover the process of running a "Synergy Driven" Tracking Optimization (TO), Verification Optimization (VO), and Design Optimization (DO) in which joints are actuated by muscles that are controlled by synergies. It will work through the design of settings files for all tools, along with analyzing the outputs of each tool with iterative problem-solving methods to generate good solutions.

Torque control of the right hip, knee, and ankle joints is replaced with synergy control using a reduced set of 9 leg muscles (bflh_r bfsh_r gasmed_r glmax2_r iliacus_r recfem_r soleus_r tibant_r vasmed_r) providing one representative uniarticular and biarticular muscle for each joint, where the strength of each muscle has been increased significantly to account for other muscles that have been eliminated as well as the increased strength required to perform a soccer kick. To avoid expensive computations of muscle geometry, surrogate muscle geometry from tutorial 6 is used.

Initial solutions for muscle activations and associated synergy controls (synergy activations + synergy vectors) were pre-generated using the Neural Control Model Personalization (NCP) Tool within the Model Personalization toolset. The NCP optimization found a set of four muscle synergies that minimized two cost function terms: 1) tracking errors for hip, knee, and ankle joint inverse dynamics moments, and 2) were muscle activations since no EMG data were available. Thus, the final muscle synergy solution closely reproduced the right leg joint moments found by inverse dynamics.

Section 1: Tracking Optimization

The Tracking Optimization (TO) tool uses a personalized model to produce a dynamically consistent movement simulation that closely reproduces all available experimental motion data, including joint motions, joint moments, ground reaction forces and moments, and muscle activations. To achieve a dynamically consistent motion, the tool spreads out matching errors between the different experimental quantities based on user-specified maximum allowable errors.

The tool accepts a post-JMP OpenSim model (.osim file) and personalized NMSM Pipeline model (.osimx file) along with experimental IK motions, ID loads, ground reactions, muscle–tendon lengths and velocities, muscle moment arms, and, if using synergy controls, NCP results for the trial of interest.

This section of the tutorial will be using synergy controls with no external forces, so the inputs we will use are a post-JMP OpenSim model, experimental IK motions, ID loads, and an .osimx file created by NCP.

Before running Tracking Optimization:

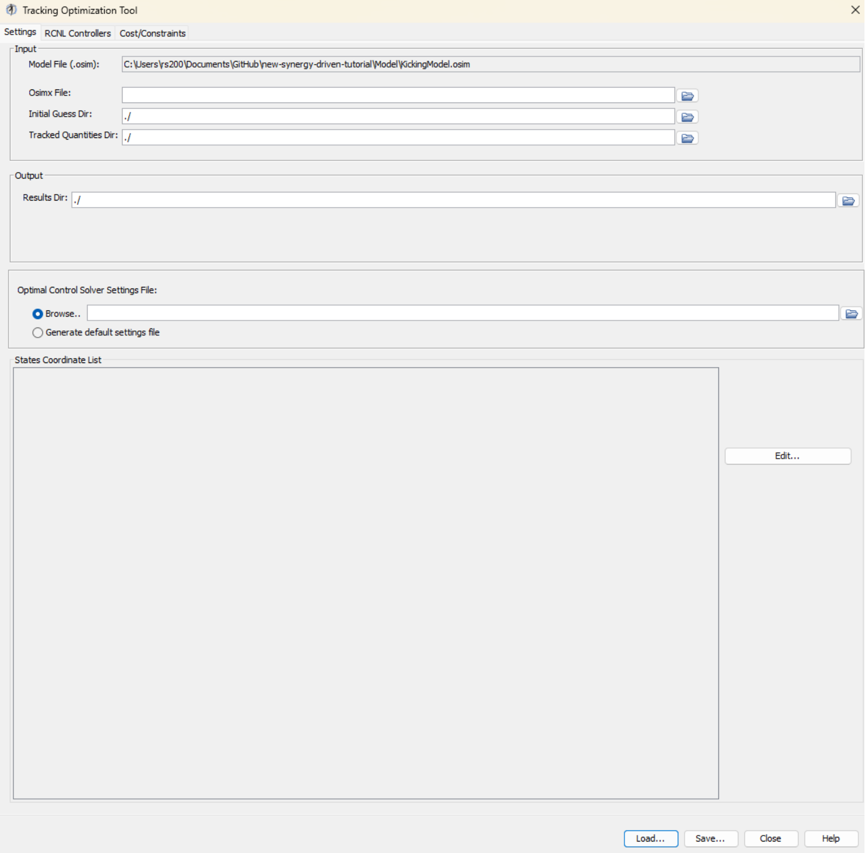
Unlike Torque Driven TO, Synergy Driven TO uses separate directories for its initial guess and tracked quantities. Synergy Driven TO uses an NCP results directory (**ncpResults**) as its initial guess. This is so the TO has an initial guess for synergy controls. When synergy controls are used, the Treatment Optimization run can use either fixed synergy vectors as found by the NCP tool or variable synergy vectors that area allowed to deviate away from the initial guesses provided by the NCP tool. Furthermore, during the solution process, the synergy vectors can be left as either unconstrained or constrained to have their sum or magnitude equal a user-specified value.

1. In your tutorial directory, open **PlotNCPResults.m** and click run.
2. This will create plots for:
    a. Muscle activations generated by NCP. Note that this NCP run minimized muscle activations instead of tracking experimental data. These muscle activations will be tracked in the TO run.

b. Joint moment matching for the NCP run. These moments are not used by the TO run.
c. Time-varying synergy commands created by NCP. These will serve as the initial guess for TO.
d. Corresponding time-invariant synergy vector weights created by NCP. You have the option to change these during TO, or keep them constant. To ensure that the solution is unique, the vector weights in NCP are normalized so that the max weight for each synergy equals 1. This normalization method can be changed in TO if desired.

Creating a Tracking Optimization settings file:

1. Activate the NMSM GUI in OpenSim by navigating to *Tools>User Plugins*, and click **rcnlPlugin.dll**
2. With **KickingModel.osim** selected in the OpenSim GUI, go to *Tools>Treatment Optimization >Tracking Optimization*
   a. The following window should be opened:



Model File (.osim): C:\Users\rs200\Documents\GitHub\new-synergy-driven-tutorial\Model\KickingModel.osim

Osimx File:

Initial Guess Dir: ./

Tracked Quantities Dir: ./

Results Dir: ./

Optimal Control Solver Settings File:

● Browse..

○ Generate default settings file

3. Load your old torque driven Tracking Optimization settings file from step 7, **TorqueDrivenTO.xml.** If you skipped that part of the tutorial, use the TO settings file **Step-7-Torque-Driven-Treatment-Optimization\CompletedSettingsFiles\TorqueDrivenTO.xml.**
4. Set the *Osimx file* to by **ncpResults\KickingModel_ncp.osimx**.
5. Set *the initial guess directory* to be **ncpResults**
   a. This is the directory that the TO will parse the synergy initial guess from.

6. Set the *tracked quantities directory* to be **preprocessed**
7. Set the *results directory* to be **SynergyTOResults**
8. Set the *optimal control solver settings file* to be **gpopsSettings.xml**
9. Under *states coordinate list*, select (**hip_flexion_r, knee_angle_r, ankle_angle_r**)
10. Click to the *RCNL Controllers* tab at the top.
11. Under *RCNL Synergy Controller*, add (**hip_flexion_r, hip_adduction_r, hip_rotation_r, knee_angle_r, ankle_angle_r, subtalar_angle_r**) to the coordinate list.
    a. Note that not all of these coordinates are in the states. The coordinate list in the synergy controller dictates the coordinates that are used to fit the surrogate model. Therefore, we need to include all of the right leg coordinates to accurately model the muscle kinematics.
12. Do not check *optimize synergy vectors*
13. Set the *surrogate model data directory* to be **surrogateData**.
14. **Clear the coordinates out of *RCNL Torque Controller.***
15. Click to the *Cost/Constraints* tab at the top.
    a. Because we loaded a previous torque driven settings file, we only need to add one cost term to include muscle activation tracking.
16. **Cost terms:**
    a. Add a new cost term:
        i. Name: **Muscle activation tracking**
        ii. Cost term type: **muscle_activation_tracking**
        iii. Component list: (**bflh_r bfsh_r gasmed_r glmax2_r iliacus_r recfem_r soleus_r tibant_r vasmed_r**)
        iv. Max allowable error: **0.1**
17. **Constraint terms:**
    a. We are adding no new constraint terms.
18. Save this settings file as **SynergyTOSettings.xml**
19. Open **SynergyTOSettings.xml** in a text editor and explore it.

20. Inside your <RCNLSynergyController> field, copy and paste the lines:

```
<load_surrogate_model>true</load_surrogate_model>
<optimize_synergy_vectors>false</optimize_synergy_vectors>
<synergy_vector_normalization_method>sum</synergy_vector_normalization_method>
<synergy_vector_normalization_value>1</synergy_vector_normalization_value>
<maximum_allowable_synergy_activation>2</maximum_allowable_synergy_activation>
```

    a. These lines will load a pre-fitted surrogate model to save on computation time.

Running Tracking Optimization:

1. Open MATLAB and open **runTOTool.m** in your tutorial directory.
2. Open the project file (**Project.prj** inside your installation of nmsm-core.)
3. Run the MATLAB file **runTOTool.m**

Post TO analysis:

1. The script will create 6 plots for you:
    a. <u>Joint Angles:</u> Joint angles for all model coordinates (including prescribed) as output by the TO run.
    b. <u>Joint Velocities:</u> Joint velocities for all states coordinates.
    c. <u>Joint Loads:</u> Joint loads for all model coordinates.
    d. <u>Synergy Controls:</u> Synergy controls for all synergy sets used in the optimization
    e. <u>Synergy Weights:</u> Corresponding synergy weights for the synergy commands.
    f. <u>Muscle Activations:</u> Muscle activations as created by the synergy controls.
2. These plots are a valuable way to analyze the results of the TO run. RMSE values between the tracked data and the TO results are reported for every plot where applicable.
3. It is also valuable to visualize the motion in the OpenSim GUI
    a. With the model selected in OpenSim, load the newly created IK motion in your TO results directory. Ensure the motion looks as close to the experimental motion as possible.
4. Getting a good TO run is very hard and often requires additional iteration after the first attempt. It is recommended to add/remove cost terms if you believe the problem would

benefit, or change max allowable errors for cost terms to “nudge” the solution in a desired direction.

Alternative TO Formulations

The code that you copied into your `<RCNLSynergyController>` allow you to change your synergy vector normalization parameters. The supported normalization methods are “sum”, “magnitude”, and “none”. “sum” imposes that each synergy vector must sum to the designated value. “magnitude” imposes that each synergy vector must have a desired magnitude. “none” does not re-normalize input synergy vectors at the start of treatment optimization. You can explore different normalization methods and study how the synergy vectors change as a result.

Section 2: Verification Optimization

The Synergy Driven VO settings file is very similar to the Torque Driven VO settings file with the only difference being that we are now tracking synergy controls instead of torque controls.

Creating a Verification Optimization settings file:

1. Activate the NMSM GUI in OpenSim by navigating to *Tools>User Plugins*, and click **rcnlPlugin.dll**
2. With **KickingModel.osim** selected in the OpenSim GUI, go to *Tools>Treatment Optimization >Verification Optimization*
3. Load your old torque driven Tracking Optimization settings file from step 7, **TorqueDrivenVO.xml.** If you skipped that part of the tutorial, use the TO settings file **Step-7-Torque-Driven-Treatment-Optimization\CompletedSettingsFiles\TorqueDrivenVO.xml.**
4. Set the *Osimx file* to by **ncpResults\KickingModel_ncp.osimx**.
5. Set *the initial guess directory* to be **SynergyTOResults**
6. Set the *tracked quantities directory* to be **SynergyTOResults**
7. Set the *results directory* to be **SynergyVOResults**
8. Set the *optimal control solver settings file* to be **gpopsSettings.xml**
9. Under *states coordinate list*, select (**hip_flexion_r, knee_angle_r, ankle_angle_r**)
10. Click to the *RCNL Controllers* tab at the top.
11. Under *RCNL Synergy Controller*, add (**hip_flexion_r, hip_flexion_r, hip_rotation_r, knee_angle_r, ankle_angle_r, subtalar_angle_r**) to the coordinate list.
12. Set the *surrogate model data directory* to be **surrogateData**.
13. Clear the coordinates out of *RCNL Torque Controller.*
14. Click to the *Cost/Constraints* tab at the top.
    a. Because we loaded a previous torque driven settings file, we only need to change the controller tracking term. All of the constraint terms can stay the same.
15. **Cost terms:**
    a. Remove the existing controller tracking term. That term tracks torque controls, which we no longer have.

b. Add a new cost term:

   i. Name: **Controller tracking**

   ii. Cost term type: **controller_tracking**

   iii. Component list: (**RightLeg_1 RightLeg_2 RightLeg_3 RightLeg_4**)

   iv. Max allowable error: **1**

16. **Constraint terms:**

    a. We are adding no new constraint terms.

17. Save this settings file as **SynergyVOSettings.xml**

18. Open **SynergyVOSettings.xml** in a text editor of your choice and explore it

19. Inside your `<RCNLSynergyController>` field, copy and paste the lines:

```
<load_surrogate_model>true</load_surrogate_model>
<optimize_synergy_vectors>false</optimize_synergy_vectors>
<synergy_vector_normalization_method>sum</synergy_vector_normalization_method>
<synergy_vector_normalization_value>1</synergy_vector_normalization_value>
<maximum_allowable_synergy_activation>2</maximum_allowable_synergy_activation>
```

Running Verification Optimization:

1. Open MATLAB and open **runVOTool.m** in your tutorial directory.
2. Open the project file (**Project.prj** inside your installation of nmsm-core.)
3. Run the MATLAB file **runVOTool.m**.

Post VO Analysis:

1. If everything was done correctly, this optimization should converge quickly.
2. The same plots as in TO will be generated automatically, and the red and blue lines should be near identical.

Section 3: Design Optimization

With VO completed, we are now satisfied with our solution and can move forward with designing a better kick motion just like we did with the Torque Driven problem. As with the above sections, the problem formulation will be very similar to how we did the Torque Driven DO runs.

Creating a Design Optimization settings file:

1. Create a copy of **SynergyVOSettings.xml** and name it **SynergyDOSettingsTemp.xml**
    a. **Note**: You need to save this settings file as **SynergyDOSettingsTemp** because there is currently a bug with the GUI that may prevent you from opening a settings file and saving to a settings file with the same name. This way, we can save our final settings file as **SynergyDOSettings.xml** later.
2. Open **SynergyDOSettingsTemp.xml** in a text editor and at the top and bottom of the document, change `<VerificationOptimizationTool>` to `<DesignOptimizationTool>` and save the file.
3. Activate the NMSM GUI in OpenSim by navigating to *Tools>User Plugins*, and click **rcnlPlugin.dll**
4. With **KickingModel.osim** selected in the OpenSim GUI, go to *Tools>Treatment Optimization >Design Optimization*
5. Set *the initial guess directory* to be **SynergyVOResults**
6. Set the *tracked quantities directory* to be **SynergyVOResults**
7. Set the *results directory* to be **SynergyDOResults**.
8. Because we loaded a previous synergy driven VO settings file, we don't need to add any extra cost or constraint terms here. We will add a user defined term, but that happens outside of the GUI.
9. Save this settings file as **SynergyDOSettings.xml**

10. Inside your <RCNLCostTermSet>, copy and paste:

```
<RCNLCostTerm name="User defined">
        <type>user_defined</type>
        <function_name>footSpeedCost</function_name>
        <cost_term_type>discrete</cost_term_type>
        <is_enabled>true</is_enabled>
        <marker_name>R_Toe</marker_name>
        <target_speed>14.35</target_speed>
</RCNLCostTerm>
```

Running Design Optimization:

1. Open MATLAB and open **runDOTool.m** in your tutorial directory.
2. Open the project file (**Project.prj** inside your installation of nmsm-core.)
3. Run the MATLAB section labelled **Run Synergy DO**.

Post DO Analysis:

At the bottom of the **runDOTool.m**, there are two plotting function calls. The first plotting function plots just the synergy driven DO results compared to the VO results. The second plotting function plots the synergy driven DO results compared to the torque driven DO results from step 7. These results should look very similar to each other.

Alternative DO Formulations:

If desired, you may run NCP again with a different number of synergies, and re-run Synergy Driven TO, VO, and DO with this new NCP results directory.

To do this, open **NCPSettings.xml** and edit the `<num_synergies>` field in `<RCNLSynergySet>`. Next, open **RunNCP.m** and run the script. This will overwrite what you have in **ncpResults**, and you can then modify and re-run the Treatment Optimization settings files you created.