Tutorial 8 – Synergy Driven Treatment Optimization

Tutorial Developer: Robert Salati, Rice Computational Neuromechanics Lab, Rice University

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:

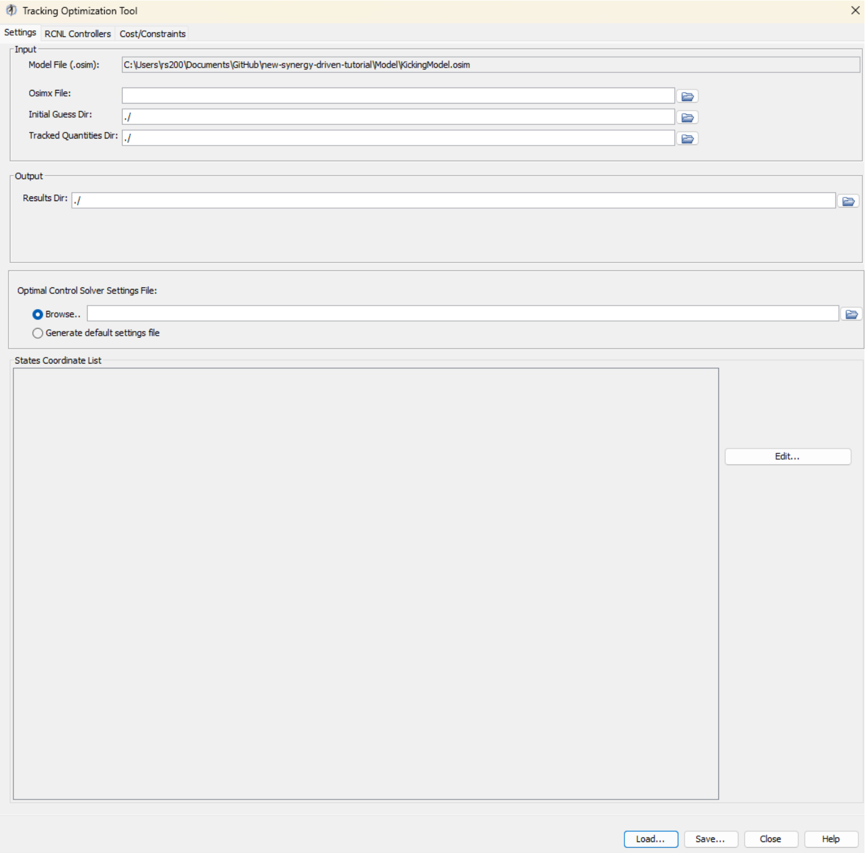
Unlike Torque Driven TO, Synergy Driven TO uses separate directories for its initial guess and tracked quantities. Synergy Driven TO uses an NCP results directory (**ncpResults**) as its initial guess. This is so the TO has an initial guess for synergy controls. When synergy controls are used, the Treatment 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:



Tracking Optimization Tool

Input

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

Osimx File:

Initial Guess Dir: ./

Tracked Quantities Dir: ./

Output

Results Dir: ./

Optimal Control Solver Settings File:

● Browse..

○ Generate default settings file

States Coordinate List

3. Set the *Osimx file* to by **ncpResults\KickingModel_ncp.osimx**.
4. Set *the initial guess directory* to be **ncpResults**
    a. This is the directory that the TO will parse the synergy initial guess from.
5. Set the *tracked quantities directory* to be **preprocessed**
6. Set the *results directory* to be **SynergyTOResultsV1**
7. Set the *optimal control solver settings file* to be **gpopsSettings.xml**
8. Under *states coordinate list*, select (**hip_flexion_r, knee_angle_r, ankle_angle_r**)
9. Click to the *RCNL Controllers* tab at the top.

10. 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.
11. Do not check *optimize synergy vectors*
12. Set the *surrogate model data directory* to be **surrogateData**.
13. Click to the *Cost/Constraints* tab at the top.
    a. We will start by using the same set of cost terms and constraints as we had in our Torque Driven TO in tutorial 7 with some changes to max allowable errors, and add an additional cost term to track muscle activations.
14. **Cost terms:**
    a. Add a new cost term:
        i. Name: **Coordinate tracking**
        ii. Cost term type: **generalized_coordinate_tracking**
        iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
        iv. Max allowable error: **0.0873**
    b. Add a new cost term:
        i. Name: **Speed tracking**
        ii. Cost term type: **generalized_speed_tracking**
        iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
        iv. Max allowable error: **0.873**
    c. Add a new cost term:
        i. Name: **Load tracking**
        ii. Cost term type: **inverse_dynamics_load_tracking**
        iii. Component list: (**hip_flexion_r_moment knee_angle_r_moment ankle_angle_r_moment**)
        iv. Max allowable error: **50**
    d. 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**
15. **Constraint terms:**
    a. Add a new constraint term:
        i. Name: **Kinetic consistency**
        ii. Constraint term type: **kinetic_consistency**

iii. Component list: (**hip_flexion_r_moment knee_angle_r_moment ankle_angle_r_moment**)
iv. Max error: **0.1**
v. Min error: **-0.1**

16. Save this settings file as **SynergyTOSettingsV1.xml**
17. Open the settings file in a text editor of your choice and explore it.
18. Change the `<trial_name>` field to be **drive_kick1**
19. Underneath the `<trial_name>` field, copy and paste the lines:

```
<load_surrogate_model>true</load_surrogate_model>
<save_surrogate_model>false</save_surrogate_model>
```

   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 section labelled **Run Synergy TO V1**

Post TO analysis:

1. The script will create 6 plots for you:
   a. Joint Angles: Joint angles for all model coordinates (including prescribed) as output by the TO run.
   b. Joint Velocities: Joint velocities for all states coordinates.
   c. Joint Loads: Joint loads for all model coordinates.
   d. Synergy Controls: Synergy controls for all synergy sets used in the optimization
   e. Synergy Weights: Corresponding synergy weights for the synergy commands.
   f. Muscle Activations: 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.

Optimize synergy vector weights:

1. Open **SynergyTOSettingsV1.xml** in a text editor.
2. Inside your `<RCNLSynergyController>`, copy and paste:

```
<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>
```

3. Create four copies of **SynergyTOSettingsV1.xml** and rename them **SynergyTOSettingsV2.xml, SynergyTOSettingsV3.xml, SynergyTOSettingsV4.xml, SynergyTOSettingsV5.xml**.
3. Configure your settings files as follows:
4. **SynergyTOSettingsV2.xml:**
    a. `<results_directory>` **SynergyTOResultsV2**
    b. `<optimize_synergy_vectors>` **false**
    c. `<synergy_vector_normalization_method>` **sum**
    d. `<synergy_vector_normalization_value>` **1**
    e. `<maximum_allowable_synergy_activation>` **10**
5. **SynergyTOSettingsV3.xml:**
    a. `<results_directory>` **SynergyTOResultsV3**
    b. `<optimize_synergy_vectors>` **false**
    c. `<synergy_vector_normalization_method>` **sum**
    d. `<synergy_vector_normalization_value>` **1**
    e. `<maximum_allowable_synergy_activation>` **2**
6. **SynergyTOSettingsV4.xml:**
    a. `<results_directory>` **SynergyTOResultsV4**
    b. `<optimize_synergy_vectors>` **true**
    c. `<synergy_vector_normalization_method>` **sum**
    d. `<synergy_vector_normalization_value>` **1**
    e. `<maximum_allowable_synergy_activation>` **2**
7. **SynergyTOSettingsV5.xml:**
    a. `<results_directory>` **SynergyTOResultsV5**
    b. `<optimize_synergy_vectors>` **true**
    c. `<synergy_vector_normalization_method>` **magnitude**
    d. `<synergy_vector_normalization_value>` **1**
    e. `<maximum_allowable_synergy_activation>` **2**
8. Run the corresponding sections in MATLAB.

Post TO Analysis:

1. While all runs produced good tracking quality, SynergyTOSettingsV3.xml mostly had the lowest RMSE of all the runs, therefore we will use that as our TO solution moving forwards.

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.

Just like with the Torque Driven tutorial, we will do two separate VO runs. A “sanity check” run, and a “dry run” run.

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. Set the *Osimx file* to by **ncpResults\KickingModel_ncp.osimx**.
4. Set *the initial guess directory* to be **SynergyTOResultsV3**
5. Set the *tracked quantities directory* to be **SynergyTOResultsV3**
6. Set the *results directory* to be **SynergyVOResultsV1**
7. Set the *optimal control solver settings file* to be **gpopsSettings.xml**
8. Under *states coordinate list*, select (**hip_flexion_r, knee_angle_r, ankle_angle_r**)
9. Click to the *RCNL Controllers* tab at the top.
10. 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.
11. Set the *surrogate model data directory* to be **surrogateData**.
12. Click to the *Cost/Constraints* tab at the top.
13. **Cost terms:**
    a. 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**
14. **Constraint terms:**
    a. Add a new constraint term:
        i. Name: **Kinetic consistency**
        ii. Constraint term type: **kinetic_consistency**
        iii. Component list: (**hip_flexion_r_moment knee_angle_r_moment ankle_angle_r_moment**)
        iv. Max error: **0.1**
        v. Min error: **-0.1**
15. Save this settings file as **SynergyVOSettingsV1.xml**
16. Open **SynergyVOSettings.xml** in a text editor of your choice

17. Change the `<trial_name>` field to be **drive_kick1**
18. Underneath the `<trial_name>` field, copy and paste the lines:

```
<load_surrogate_model>true</load_surrogate_model>
<save_surrogate_model>false</save_surrogate_model>
```

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

```
<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 section labeled **Run Synergy VO V1**.

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.

Adding new constraints into VO:

The next step is to add constraints on top of our previous VO run to ensure that the constraints are consistent with your problem formulation. These settings files will be nearly identical to the Torque Driven VO settings files, so you can create a copy of **SynergyVOSettingsV1.xml** and copy and paste your `<RCNLConstraintTermSet>` from the **Torque Driven DO V3** settings file into this settings file to save time. Remember to change your `<results_directory>` to **SynergyVOResultsV2.**

If copy and pasting, skip to bullet point 6.

If doing this through the GUI:

1. Load **SynergyVOSettingsV1.xml** into the VO NMSM OpenSim GUI.
2. Set `<results_directory>` as **SynergyVOResultsV2**
3. Click to the *Cost/Constraints tab* at the top.
4. **Constraint terms:**
    a. Add a new constraint term:
        i. Name: **Initial coordinate position deviation**
        ii. Constraint term type: **initial_generalized_coordinate_deviation**
        iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
        iv. Max error: **0.0175**
        v. Min error: **-0.0175**
    b. Add a new constraint term:
        i. Name: **Initial coordinate speed deviation**
        ii. Constraint term type: **initial_generalized_speed_deviation**
        iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
        iv. Max error: **0.175**
        v. Min error: **-0.175**
    c. Add a new constraint term:
        i. Name: **Final toe marker position deviation**
        ii. Constraint term type: **final_marker_position_deviation**
        iii. Component list: (**R_Toe**)
        iv. Max error: **0.001**
        v. Min error: **-0.001**
    d. Add a new constraint term:
        i. Name: **Final calcaneus orientation deviation**
        ii. Constraint term type: **final_body_orientation_deviation**
        iii. Component list: (**calcn_r**)
        iv. Max error: **0.0175**
        v. Min error: **-0.0175**

e. Add a new constraint term:
   i. Name: **Toe marker final velocity**
   ii. Constraint term type: **final_marker_velocity_value**
   iii. Component list: (**R_Toe**)
   iv. Max error: **13.2**
   v. Min error: **13.1**

f. Add a new constraint term:
   i. Name: **Limit ankle position**
   ii. Constraint term type: **generalized_coordinate_value**
   iii. Component list: (**ankle_angle_r**)
   iv. Max error: **0.0475**
   v. Min error: **-0.3187**

g. Add a new constraint term:
   i. Name: **Limit ankle velocity**
   ii. Constraint term type: **generalized_speed_deviation**
   iii. Component list: (**ankle_angle_r**)
   iv. Max error: **0.175**
   v. Min error: **-0.175**

5. Save this settings file as **SynergyVOSettingsV2.xml**
6. Open **SynergyVOSettingsV2.xml** in a text editor of your choice.
7. Inside your `<RCNLConstraintTerm>` for **Toe marker final velocity,** change `<axes>` to only include **X**, and ensure that `<max_error>` is **13.2** and `<min_error>` is **13.1**
8. Run the MATLAB section labelled **Run Synergy Driven VO V2**.
9. To compare both VO results to the previous TO results, run the MATLAB section labelled **Compare Results.** Both solutions should be near identical to the TO results.

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. 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 >Design Optimization*
3. Set the *Osimx file* to by **ncpResults\KickingModel_ncp.osimx**.
4. Set *the initial guess directory* to be **SynergyVOResultsV2**
5. Set the *tracked quantities directory* to be **SynergyVOResultsV2**
6. Set the *results directory* to be **SynergyDOResultsV1**
7. Set the *optimal control solver settings file* to be **gpopsSettings.xml**
8. Under *states coordinate list*, select (**hip_flexion_r, knee_angle_r, ankle_angle_r**)
9. Click to the *RCNL Controllers* tab at the top.
10. 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.
11. Set the *surrogate model data directory* to be **surrogateData**.
12. Click to the *Cost/Constraints* tab at the top.
    a. The below steps are using the same cost terms and constraints as in **SynergyVOSettingsV2.xml.** To save time, you may save the settings file right now as **SynergyDOSettingsV1.xml** and copy the `<RCNLCostTermSet>` and `<RCNLConstraintTermSet>` from **SynergyVOSettingsV2.xml** over to **SynergyDOSettingsV1.xml.**
    b. If copy and pasting, skip to bullet point 16.
13. **Cost terms:**
    a. 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**

14. **Constraint terms:**
   a. Add a new constraint term:
      i. Name: **Kinetic consistency**
      ii. Constraint term type: **kinetic_consistency**
      iii. Component list: (**hip_flexion_r_moment knee_angle_r_moment ankle_angle_r_moment**)
      iv. Max error: **0.1**
      v. Min error: **-0.1**
   b. Add a new constraint term:
      i. Name: **Initial coordinate position deviation**
      ii. Constraint term type: **initial_generalized_coordinate_deviation**
      iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
      iv. Max error: **0.0175**
      v. Min error: **-0.0175**
   c. Add a new constraint term:
      i. Name: **Initial coordinate speed deviation**
      ii. Constraint term type: **initial_generalized_speed_deviation**
      iii. Component list: (**hip_flexion_r knee_angle_r ankle_angle_r**)
      iv. Max error: **0.175**
      v. Min error: **-0.175**
   d. Add a new constraint term:
      i. Name: **Final toe marker position deviation**
      ii. Constraint term type: **final_marker_position_deviation**
      iii. Component list: (**R_Toe**)
      iv. Max error: **0.001**
      v. Min error: **-0.001**
   e. Add a new constraint term:
      i. Name: **Final calcaneus orientation deviation**
      ii. Constraint term type: **final_body_orientation_deviation**
      iii. Component list: (**calcn_r**)
      iv. Max error: **0.0175**
      v. Min error: **-0.0175**
   f. Add a new constraint term:
      i. Name: **Toe marker final velocity**
      ii. Constraint term type: **final_marker_velocity_value**
      iii. Component list: (**R_Toe**)
      iv. Max error: **14.4**
      v. Min error: **14.3**
   g. Add a new constraint term:
      i. Name: **Limit ankle position**

        ii. Constraint term type: **generalized_coordinate_value**
        iii. Component list: (**ankle_angle_r**)
        iv. Max error: **0.0475**
        v. Min error: **-0.3187**
    h. Add a new constraint term:
        i. Name: **Limit ankle velocity**
        ii. Constraint term type: **generalized_speed_deviation**
        iii. Component list: (**ankle_angle_r**)
        iv. Max error: **0.175**
        v. Min error: **-0.175**

15. Save this settings file as **SynergyDOSettingsV1.xml**
16. Change the `<trial_name>` field to be **drive_kick1**
17. Underneath the `<trial_name>` field, copy and paste:

```
<load_surrogate_model>true</load_surrogate_model>
<save_surrogate_model>false</save_surrogate_model>
```

18. Underneath the `<trial_name>` field, copy and paste:

```
<final_time_range>0.26 0.36</final_time_range>
```

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

```
<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>
```

20. Inside your `<RCNLConstraintTerm>` for **Toe marker final velocity,** change `<axes>` to only include **X,** and change the `<max_error>` to **14.4** and `<min_error>` to **14.3** if not already done**.**

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 V1**.

User Designed Cost Functions:

To compare to our Torque Driven solutions from the previous tutorial, we will run a Synergy Driven DO with the same user defined cost function as before.

1. Create a copy of **SynergyDOSettingsV1.xml** and name it **SynergyDOSettingsV2.xml**
2. Open **SynergyDOSettingsV2.xml** in a text editor of your choice
3. Change <results_directory> to **SynergyDOResultsV2**
4. 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>
```

5. Inside your <RCNLConstraintTerm> for **Toe marker final velocity,** set <is_enabled> to **false**
6. Run the MATLAB section labelled **Run Synergy DO V2**.

Post DO Analysis:

At the bottom of the **runDOTool.m**, there are two sections for plotting results. The first section **Compare Synergy Driven Results** compares the two Synergy Driven DO results we did. These solutions should be very similar to each other. The second section **Compare to Torque Driven Results** compares the Synergy Driven to the Torque Driven results (both using user defined cost functions). These results will look slightly different because of the different controllers used. However, both results should follow the same general trends.

Experiment with different numbers of synergies:

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.