Derivative-Free Optimization Algorithm Implementation User Manual

Klevis Aliaj

1 Installation

The following installation steps have been verified on a fresh installation of Ubuntu 16.04.

1.1 Prepare Robot Operating System

1. Install the Robot Operation System (ROS) Kinetic release
   (a) Follow instructions at `http://wiki.ros.org/kinetic/Installation/Ubuntu`.
   (b) The 'Desktop Install' option will suffice
2. Create a catkin workspace
   (a) Follow instructions at `http://wiki.ros.org/catkin/Tutorials/create_a_workspace`.
   (b) Although the instructions above say to run `source /opt/ros/melodic/setup.bash` since the Kinetic release of ROS is utilized the following command should be run `source /opt/ros/kinetic/setup.bash`.

1.2 Install Dependencies

1. Clone or download the Derivative-Free Optimization Algorithm repository to `~/catkin_ws/src`.
2. Install library dependencies
   (a) Copy the installScripts directory from the repository to a path of your choosing and within a terminal window switch to the newly copied directory.
      i. `cp -R ~/catkin_ws/src/OptRepo/installScripts ~/myDir`
      ii. `cd ~/myDir/installScripts`
   (b) Install NLopt library: `./installNLOPT.sh`
   (c) Install Pagmo2 library: `./installPagmo2.sh`

1.3 Build the Derivative-Free Optimization Algorithm Catkin Package

1. Switch to the catkin workspace: `cd ~/catkin_ws`
2. Overlay the catkin workspace on top of the current environment: `source devel/setup.bash`
3. Build the catkin workspace: `catkin_make`

2 Running the Derivative-Free Optimization Algorithm

The optimization algorithm package is comprised of 5 executables and 1 shared library. The 5 executables have been created to meet various optimization needs although it is very likely that users will need to create new executables based on the shared library to fit their specific needs. All executables depend on 2 configuration files, whose paths are specified as command line arguments to the executable. The first parameter file to be specified contains robot specifications and generic optimization algorithm parameters. These parameters are described in Optimization Algorithm Generic Parameters File. The second parameter file to be specified contains specification on the input to the optimization algorithm, such as the path to the trajectory file, the tool frame to utilize, etc. The parameters in this configuration file are executable specific, so they are described Task Descriptions.

Once the catkin workspace has been built as described in Installation the shared library will reside in `~/catkin_ws/devel/lib` while the executables will reside in `~/catkin_ws/devel/lib/OptRepo`. The example below demonstrates how to run the Single Trajectory Multiple Frames Solver task but the other task have the same exact syntax.

1. Switch to the executable directory: `cd ~/catkin_ws/devel/lib/OptRepo`
2. Run the task specifying the full path to both parameter files:
   `./singleTraj_MultFrames_solver /full/path/generic_params.xml /full/path/input_params.xml`

3 Optimization Algorithm Generic Parameters File

- **URDF_File** - File path to the Unified Robotics Description Format (URDF) file for the robot. (e.g. `D:\Optimization_Algorithm_Data\m20iag.urdf`).
- **EndEffectorName** - The name of the link within the URDF file for the robot representing the end-effector of the robot (e.g. tool0).
- **BaseName** - The name of the link within the URDF file for the robot representing the base of the robot (e.g. base).
- **VelocityLimits** - A 6 component vector specifiying the velocity limits of each of the six joints of the robot - specified in rad/second.
- **Tolerance** - The joint limit, joint velocity limit, and total positional accuracy tolerance. Typically very small, e.g. 0.001.
- **RotTolerance** - A trajectory maintains its kinetic properties if it is rotated about the gravitational axis (in the formulation of this optimization problem, the z-axis). This parameter specifies the tolerance of the actual axis of rotation for the trajectory from the gravitational axis. Expressed as the dot product of the actual axis of rotation with the x and y axes. Typically very small, e.g. 0.001.
- **XTol** - A stopping criterion for the optimization algorithm. If every optimization parameter changes by less than its absolute value multiplied by this parameter, then the optimization algorithm is stopped. Typically very small, e.g. 0.001.
- **MaxEval** - The maximum number of iterations that the optimization algorithm will run.
- **Algorithm** - The NLopt algorithm to utilize for solving the optimization problem. Only the COBYLA algorithm has been tested. If another algorithm is chosen, it will need to be a derivative-free numerical optimization method.
- **WorkspaceLowerCorner** - The optimization problem allows for the specification of workspace bounds in terms of a rectangular prism. The bounds are imposed upon the position of the robot end-effector. The units will be based on the units utilized in the URDF file. This parameter specifies the lower left-hand corner of the rectangular prism.

- **WorkspaceUpperCorner** - The optimization problem allows for the specification of workspace bounds in terms of a rectangular prism. The bounds are imposed upon the position of the robot end-effector. The units will be based on the units utilized in the URDF file. This parameter specifies the upper right-hand corner of the rectangular prism.

4 Task Listings

1. Single Trajectory Multiple Frames Solver
2. Single Trajectory Multiple Seeds Solver
3. Batch Solver
4. Variable Seed Batch Solver
5. Multiple Frames Folder Solver

5 Task Descriptions

Single Trajectory Multiple Frames Solver

Description: This task optimizes a single trajectory specified in **TrajectoryFile** and outputs a joint space trajectory for each tool frame specified in **Toolframes** in the path specified by **OutputDirectory**. Even though multiple seeds may be specified in **SeedsFile** these are not optimized individually. Pagmo2, the underlying optimization library, utilizes the first few iterations of the optimization algorithm to determine a seed that it deems is best suited for the problem and continues optimization using that seed.
Executable File: `singleTraj_MultFrames_Solver`
Input Parameters:

- **TrajectoryFile** - File path to a frames trajectory file. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`).
- **SeedsFile** - File path to `.csv` file containing initial guesses for the robot joint angles for the first timepoint of the trajectory. (e.g. `D:\Optimization_Algorithm_Data\Seeds\jointCombosH.csv`).
- **Toolframes** - File path to `.csv` file containing tool frames to utilize for the optimization algorithm (e.g. `D:\Optimization_Algorithm_Data\Toolframes\toolframes30.csv`).
- **OutputDirectory** - Path to a directory where solutions generated by the optimization algorithm should be output (e.g. `D:\OptAlgo_Output`).

Single Trajectory Multiple Seeds Solver

Description: This task optimizes a single trajectory specified in **TrajectoryFile**. Each seed specified in **SeedsFile** is optimized individually and will result in a separate solution. This task only optimizes the first tool frame specified in **Toolframes**.
Executable File: `singleTraj_MultFrames_Solver`
Input Parameters:

- **TrajectoryFile** - File path to a frames trajectory file. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`).
- **SeedsFile** - File path to `.csv` file containing initial guesses for the robot joint angles for the first timepoint of the trajectory. (e.g. `D:\Optimization_Algorithm_Data\Seeds\jointCombosH.csv`).
- **Toolframes** - File path to `.csv` file containing tool frames to utilize for the optimization algorithm (e.g. `D:\Optimization_Algorithm_Data\Toolframes\toolframes30.csv`).
- **OutputDirectory** - Path to a directory where solutions generated by the optimization algorithm should be output (e.g. `D:\OptAlgo_Output`).

Batch Solver

Description: This task is similar to the Single Trajectory Multiple Seeds Solver task but it runs over an entire folder containing multiple subjects. It optimizes all trajectories for all subjects in the directory specified by **TrajectoryFolder** for each individual seed specified in **SeedsFile**. The joint space trajectories for each seed are output in the same folder as the trial files. This task only optimizes the first tool frame specified in **Toolframes**.
Executable File: `batch_solver`
Input Parameters:

- **TrajectoryFolder** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **SeedsFile** - File path to `.csv` file containing initial guesses for the robot joint angles for the first timepoint of the trajectory. (e.g. `D:\Optimization_Algorithm_Data\Seeds\jointCombosH.csv`).
- **Toolframes** - File path to `.csv` file containing tool frames to utilize for the optimization algorithm (e.g. `D:\Optimization_Algorithm_Data\Toolframes\toolframes30.csv`).

Variable Seed Batch Solver

Description: This task is similar to the Batch Solver task but the seeds utilized for optimization are defined on a per-trial basis. Similar to Batch Solver, it optimizes all trajectories for all subjects in the directory specified by **TrajectoryFolder**. However, the seeds utilized for optimizing a particular trial are specified in a file ending in `seedJoint.txt`. For example, to specify the seeds to utilize for optimizing the frames trajectory file, `UEK_015_JJ_free_01.smoothFrames.txt`, a seeds file with a filename of `UEK_015_JJ_free_01.seedJoint.txt` is created in the same directory as the frames trajectory file. Even though multiple seeds may be specified in the seeds file these are not optimized individually. Pagmo2, the underlying optimization library, utilizes the first few iterations of the optimization algorithm to determine a seed that it deems is best suited for the problem and continues optimization using that seed. This task only optimizes the first tool frame specified in **Toolframes**.
Executable File: `batch_solver_varseed`
Input Parameters:

- **TrajectoryFolder** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **Toolframes** - File path to `.csv` file containing tool frames to utilize for the optimization algorithm (e.g. `D:\Optimization_Algorithm_Data\Toolframes\toolframes30.csv`).

Multiple Frames Folder Solver

Description: This task optimizes all frame trajectories (files ending in `smoothFrames.txt`) found in **TrajectoryFolder** for all tool frames specified in **Toolframes**. Joint space trajectories are output for each individual toolframe in the directory specified by **OutputDirectory**. Even though multiple seeds may be specified in the seeds file these are not optimized individually. Pagmo2, the underlying optimization library, utilizes the first few iterations of the optimization algorithm to determine a seed that it deems is best suited for the problem and continues optimization using that seed.
Executable File: `folderSolver_MultFrames`
Input Parameters:

- **TrajectoryFolder** - Directory path pointing to a folder containing frames trajectories (ending in `smoothFrames.txt`) to be optimized.
- **SeedsFile** - File path to `.csv` file containing initial guesses for the robot joint angles for the first timepoint of the trajectory. (e.g. `D:\Optimization_Algorithm_Data\Seeds\jointCombosH.csv`).

- **Toolframes** - File path to `.csv` file containing tool frames to utilize for the optimization algorithm (e.g. `D:\Optimization_Algorithm_Data\Toolframes\toolframes30.csv`).
- **OutputDirectory** - Path to a directory where solutions generated by the optimization algorithm should be output (e.g. `D:\OptAlgo_Output`).