Mathematical definition of the multi-objective optimization problem

# Optimization Problem

The optimization problem is to find a set of controllers that simultaneously minimizes all the objective functions for all scenarios while obeying the constraints. The formulation of the optimization is given in the following:

For a given set of scenarios, :

where, is the set of feasible controllers and is the vector of objective (cost) functions to be minimized. In this formulation, each component of is associated with one scenario. Namely, the component of is obtained by applying the controller for the scenario. The rest of this document is organized as follows; first the scenario-based objective function is defined. Next, the constraints are explained. Finally, the decision variables are described.

# Objective Functions

Let's assume we have n objective functions that we want to optimize. We can represent them as follows:

Minimize:

Where R is a scalar, the magnitude of the radius vector:

(t)represents the desired position and (t) represents the actual position.

Moreover, an example of a desired motion profile is:

where ai and bi are given scalars.

is the allowed time for the ith scenario.

# Constraints

The constraints are needed in order to maintain the stability of the system and achieve the goal of optimizing the objective functions. They also perform as a measure of the accuracy of the controller.

The first constraint, **,** aims to ensure that the obtained error is less than , which is a predefined required error. In the formulation, is the maximal position error obtained by the controller , in all scenarios. It is calculated as the maximal (Euclid) distance between the desired and the actual position of the end-effector.

The second constraint, **,** aims to ensure that the obtained velocity error is less than , which is a predefined required velocity error. In the formulation, is the maximal velocity error obtained by the controller , in all scenarios. It is calculated as the maximal (Euclid) distance between the desired and the actual velocity of the end-effector.

The third constraint, **,** aims to ensure that the obtained acceleration error is less than , which is a predefined required acceleration error. In the formulation, is the maximal acceleration error obtained by the controller , in all scenarios. It is calculated as the maximal (Euclid) distance between the desired and the actual acceleration of the end-effector.

The forth constraint, aims to ensure that the obtained torque of the first joint is less than , which is a predefined required torque. In the formulation, is the maximal torque of the first joint obtained by the controller , in all scenarios.

The fifth constraint, aims to ensure that the obtained torque of the second joint is less than , which is a predefined required torque. In the formulation, is the maximal torque of the second joint obtained by the controller , in all scenarios. It's calculation is described in the "Dynamic Model for a Planar Robotic Arm with Two Rotational degrees of Freedom" document.

Therefor,

**τ**is the vector of joint torques.

and are the actual and desired positions regarding scenario

and are the actual and desired velocities regarding scenario

and are the actual and desired accelerations regarding scenario

# Decision Space

In this study, the search space is defined as all feasible fixed topologies feed-forward fully connected neural networks with a single hidden layer.

Each network in the decision space is represented as set of two weight vectors

.

The first dimension of is the same as the number of hidden layers and the second dimension of is the same as the number of elements in the input layer. The first dimension of is the same as the number of elements in the output layer and the second dimension of is the same as the number of hidden layers.