

Constrained Optimization of Trajectory

Introduction

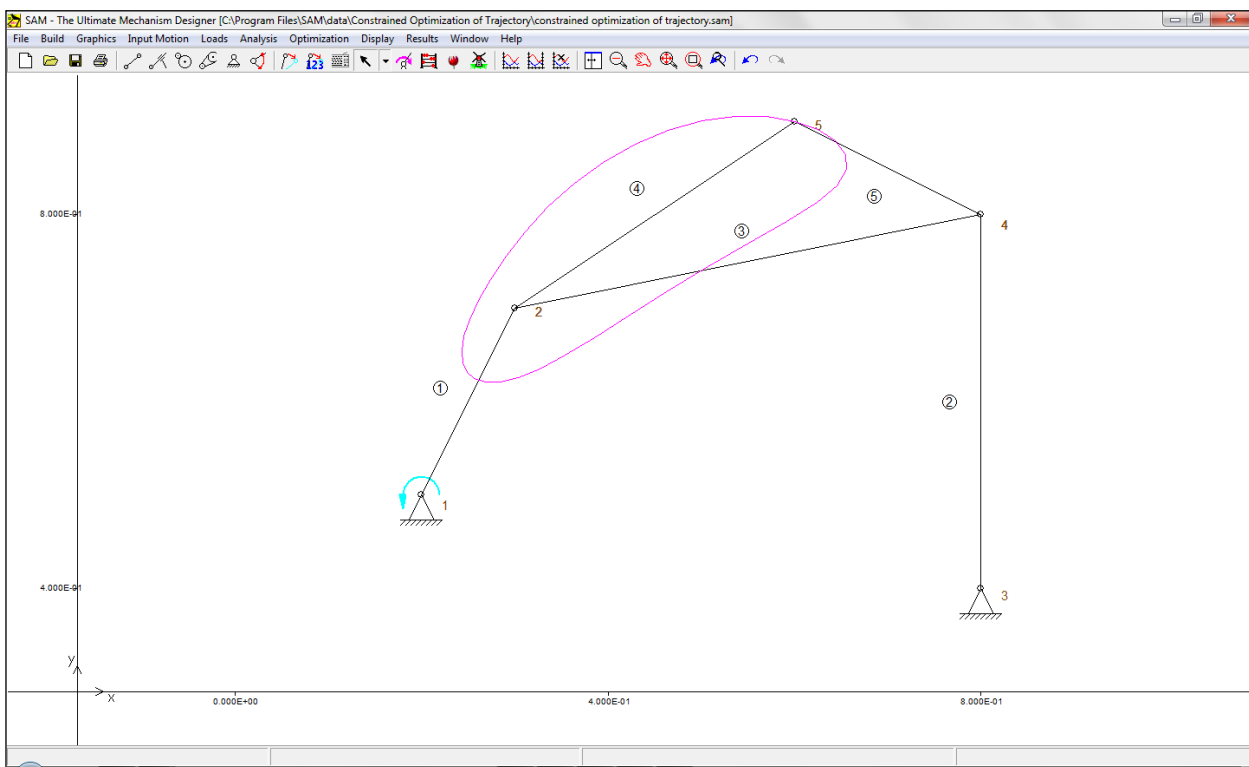
The optimization module of SAM offers single-function multi-parameter optimization based on a mix of evolutionary algorithms and Simplex techniques. By defining a penalty function it is possible to add extra constraints.

In this example, the trajectory/path of a specific node should match as much as possible the desired reference trajectory. As a quality number to judge a solution candidate, the RMS value of the difference between actual and reference path is calculated. Each of the nodes of the mechanism can be varied within certain specified limits. However, there is also an additional constraint that the length of one of the elements should be within a certain range.

The example is based on the default 4-bar mechanism that is generated by the command sequence FILE / WIZARD / 4 BAR MECHANISM and by accepting the default values.

Reference Mechanism

In the default situation the mechanism and the path of node 5 looks as follows

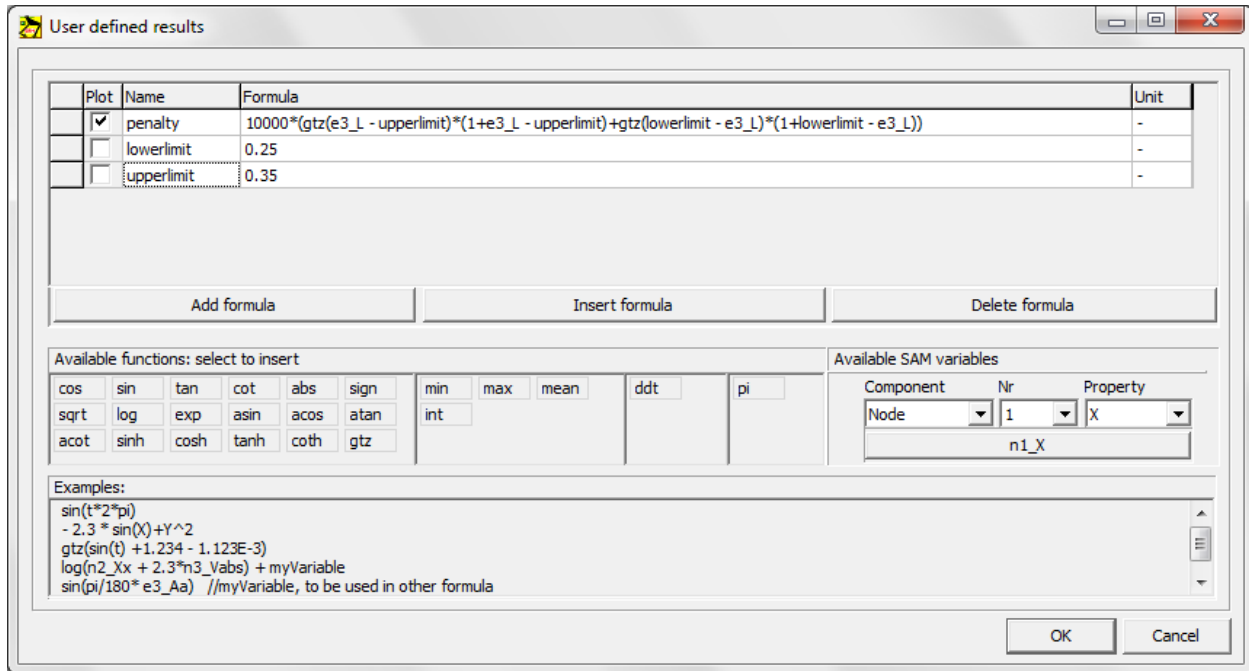


Optimization

The next step one needs to define the optimization target and the penalty function that should incorporate the constraints.

Penalty Function

In order to incorporate the constraint, that the length of element 3 should be in the range of 0.25-0.35m (the default length was 0.51m) a user defined variable needs to be created via RESULTS / USER DEFINED (in this case it is named *penalty*, but one can choose any name).

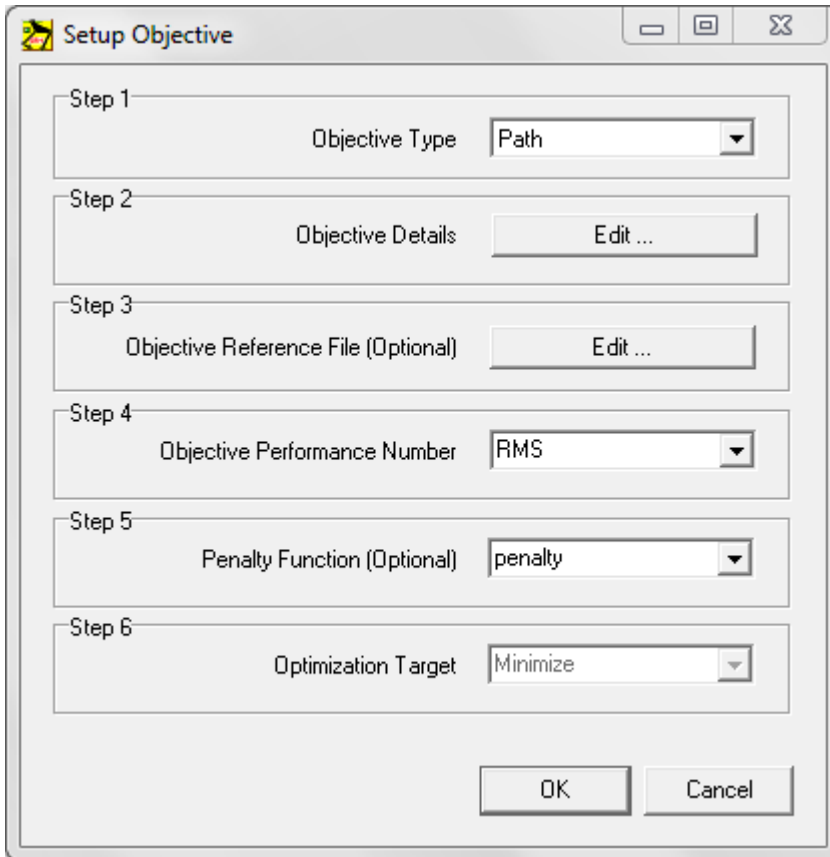


The constraint that the length of element 3 should be in the range of 0.25-0.35 is translated into a penalty that is added to the cost function during the optimization process. The penalty is based on the Heaviside or Greater Than Zero function GTZ(x), which yields an output of 1 if $x \geq 0$ and an output of 0 if $x < 0$. In order to achieve a larger penalty for larger deviations from the target and thereby “help” the optimization routine, the outcome is multiplied by the actual deviation plus 1. Finally a multiplication factor of 10000 is applied based on engineering judgment. The outcome of the penalty function should be large enough to significantly impact the overall cost function but not too large to lead to numerical problems.

A very sound engineering approach is to perform an unconstrained optimization first in order to see the number of magnitude of the optimized cost function without any penalty. That gives a good estimate of the size of the penalty that should be applied as soon as the constrained is violated. Also does is give an indication of the performance that can be achieved without the constraint.

Optimization Objective

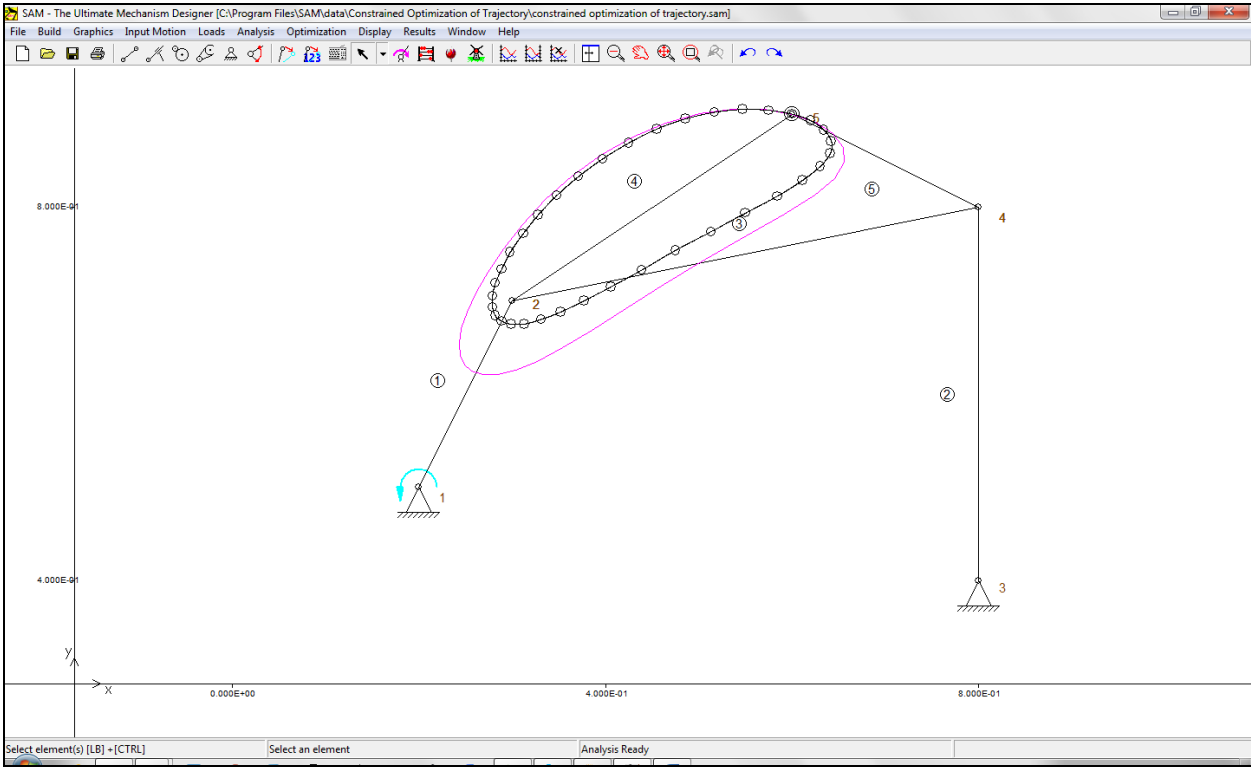
The settings of the optimization are shown in the following screenshot. The trajectory/path of node 5 should match as much as possible the desired reference trajectory. As a quality number to judge a solution candidate, the RMS value of the difference between actual and reference path is calculated. Each of the nodes of the mechanism can be varied within certain specified limits. In order to incorporate the constraint, a penalty variable is selected that is added to the quality number of a solution candidate during the optimization process.



Reference Trajectory

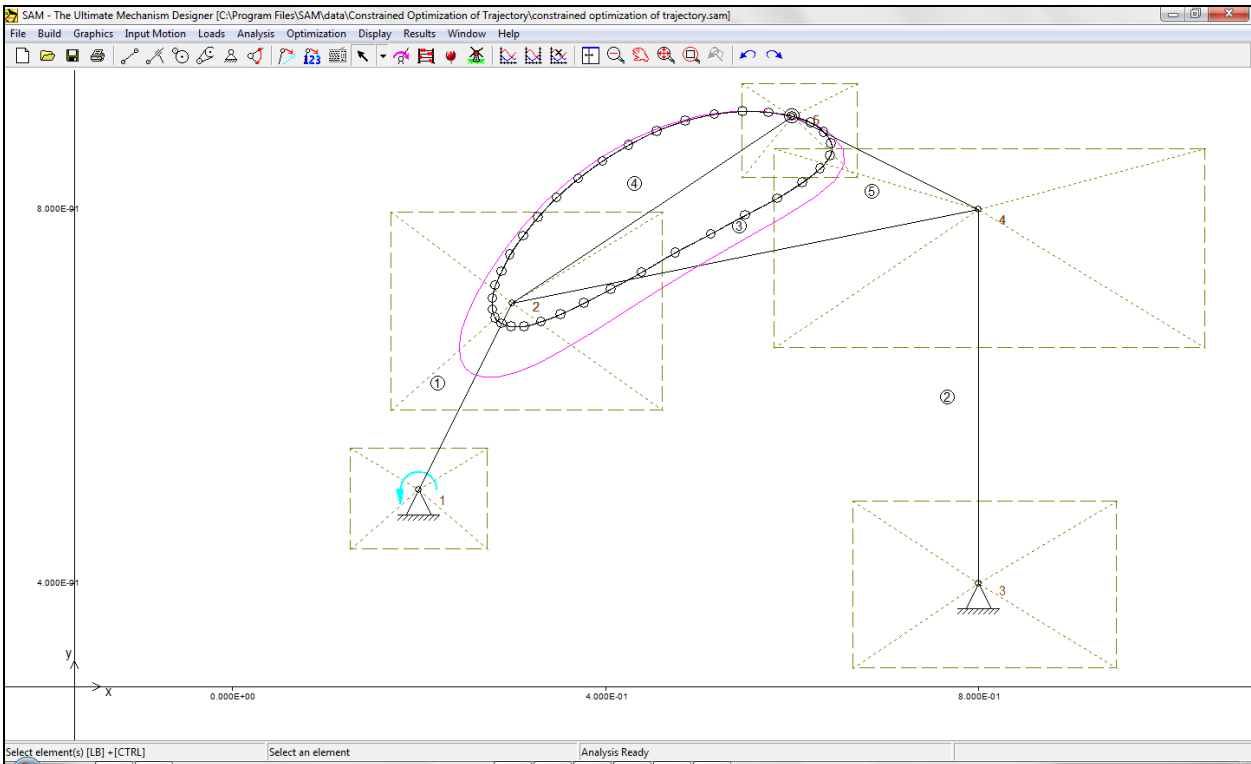
The reference file is a simple ANSI text file with x and y coordinates in **SI-units** (so meters in this case) In order to generate a closed reference trajectory, the first and last entry should be identical. The reference data used in this example are:

0.600	0.900	0.289	0.734	0.475	0.754
0.575	0.904	0.282	0.719	0.513	0.774
0.547	0.905	0.279	0.705	0.550	0.794
0.517	0.902	0.279	0.693	0.584	0.812
0.486	0.895	0.282	0.684	0.611	0.829
0.455	0.884	0.289	0.678	0.630	0.844
0.425	0.869	0.299	0.675	0.641	0.858
0.397	0.852	0.313	0.675	0.642	0.871
0.371	0.833	0.331	0.680	0.634	0.883
0.348	0.813	0.352	0.688	0.620	0.893
0.328	0.792	0.377	0.700	0.600	0.900
0.312	0.772	0.406	0.715		
0.298	0.752	0.439	0.733		
...		...			



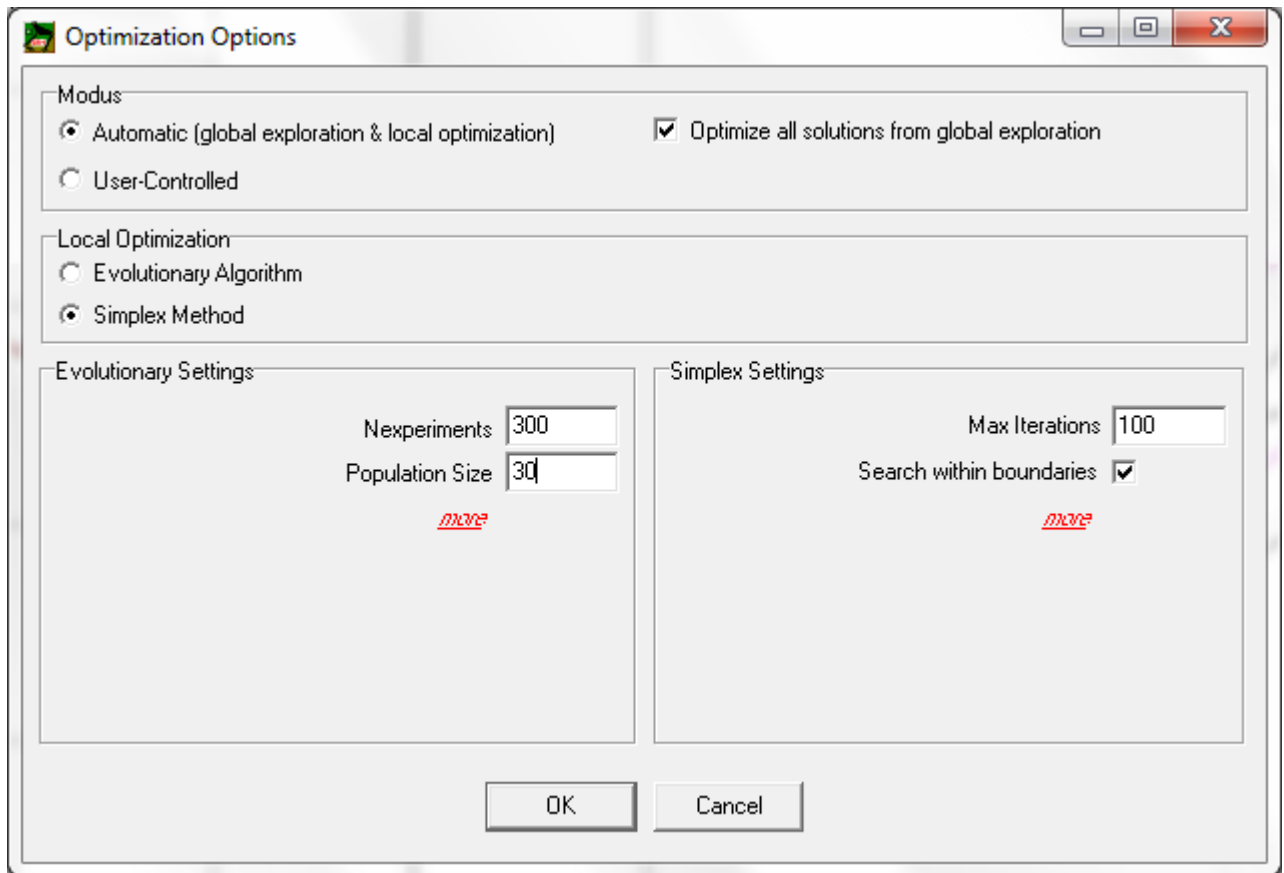
Optimization Parameter Range

Each of the nodes of the mechanism can be varied within the indicated borders.



Optimization Options

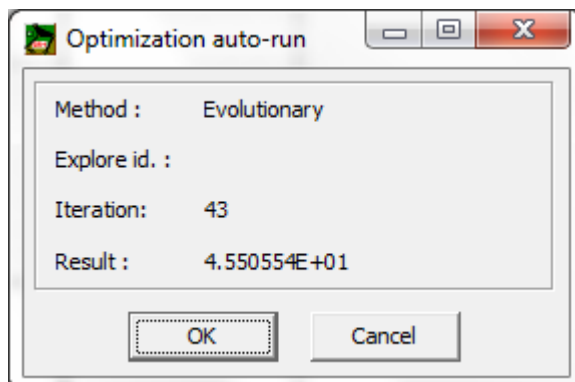
For the sake of simplicity, the automatic modus was chosen. In this modus a global exploration is performed based on evolutionary algorithms using the settings as shown. The best results of that global approach are each then further optimized automatically - in this case using the Simplex method as indicated.



Optimization Run

Global Exploration

The status window shows the process of the global exploration.



Method: The method for global exploration is always the Evolutionary Algorithm.
Explore id: empty (has only meaning when doing the local optimization).
Iteration: Current iteration step in the evolutionary process
Result: Current value of the experiment.

Modus

- Choosing **AUTOMATIC** is the easiest way to do an optimization.
- The checkbox “Optimize all solutions from global exploration” indicates that after the global exploration of the solution space (always done via evolutionary algorithm), ALL best results of the exploration are further improved via a local optimization. After the global exploration a pool of best results exists and the size of that pool of results is equal to the Population Size defined in the Evolutionary Settings. So in this case 300 evolutionary (or genetic) experiments are done based on a population size of 30 “members”. At the end a pool of 30 best members or solutions exist. If the checkbox is checked, all these 30 best results are further optimized via local optimization. If the checkbox is NOT checked, only the best solution found in the global exploration is further improved via local optimization.

Local Optimization

- In general using the Simplex Method for the local optimization yields good results.

Evolutionary Setting

- The evolutionary algorithm starts with filling a pool or population of results via a random process (Population Size indicates the size of the population). This population is then improved via an evolutionary (nature inspired genetic) process. The number of genetic experiments is indicated by Nexperiments.

Simplex Setting

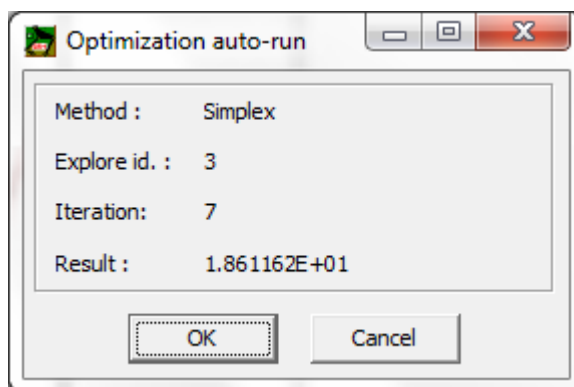
The main setting for the Simplex Method is the max. number of iterations. The default value is generally OK but based on the progress that is monitored online one can decide that a smaller number is sufficient. On the other hand, if one observes significant progress just before reaching the max. number of iterations it is very likely that continuing the process yields better results and it makes sense to increase this number.

In principle the boundaries of the optimization space should always be respected. However, sometimes it is very interesting – and one may be triggered to find new solutions – to allow the algorithm to disrespect these boundaries. This is the reason why the user has a choice.

If the process takes too long one can decide to do less experiments, choose a smaller population size (though it should always be at least 20-30 to give the evolutionary algorithm a chance), do less iterations in the local optimization process or decide NOT to further optimize all results from the global search but to only post-process the best result of the global search.

Local Optimization

In this case all 30 results of the global exploration are improved via a local optimization process. The status window shows:

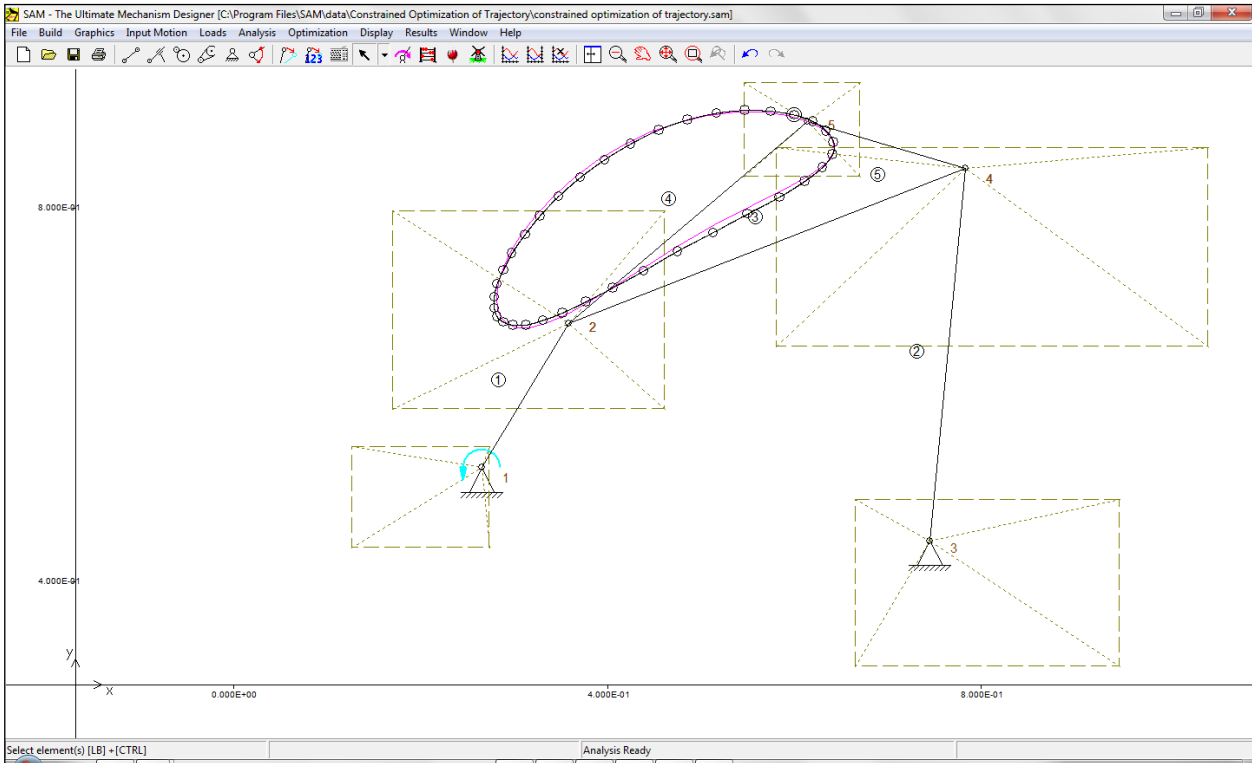


Method: chosen method for local optimization
Explore id: ID of the solution of the global exploration that is currently locally optimized.
Iteration: Current iteration step in the local optimization process
Result: Current value of the optimization goal.

Optimized Mechanism – without constraint/penalty

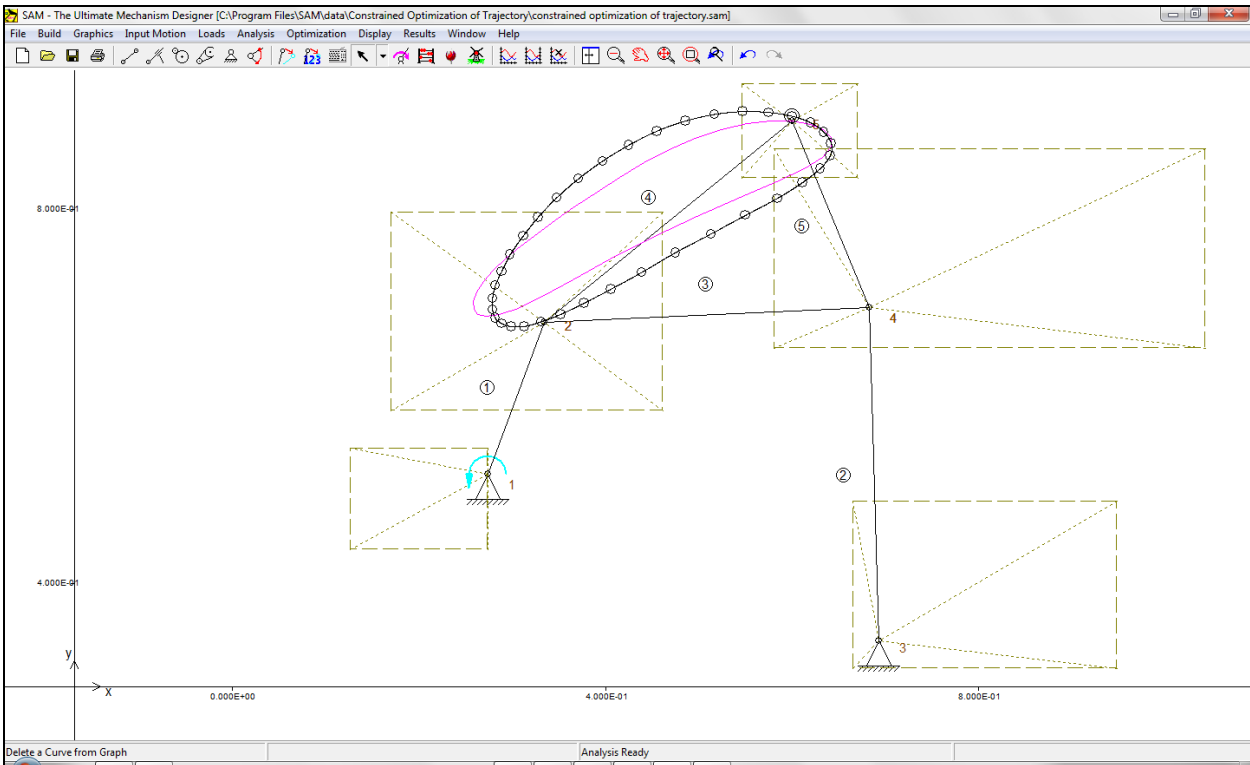
As indicated before, it is always a good practice to start the optimization WITHOUT the constraint/penalty because it helps to better understand the optimization problem and also the most ideal solution one can find, including the amount of violation of the constraint.

If the penalty function (or constraint) is switched OFF, the following almost perfect result is found after optimization. In this case the optimized length of element 3 is equal to 0.455m.

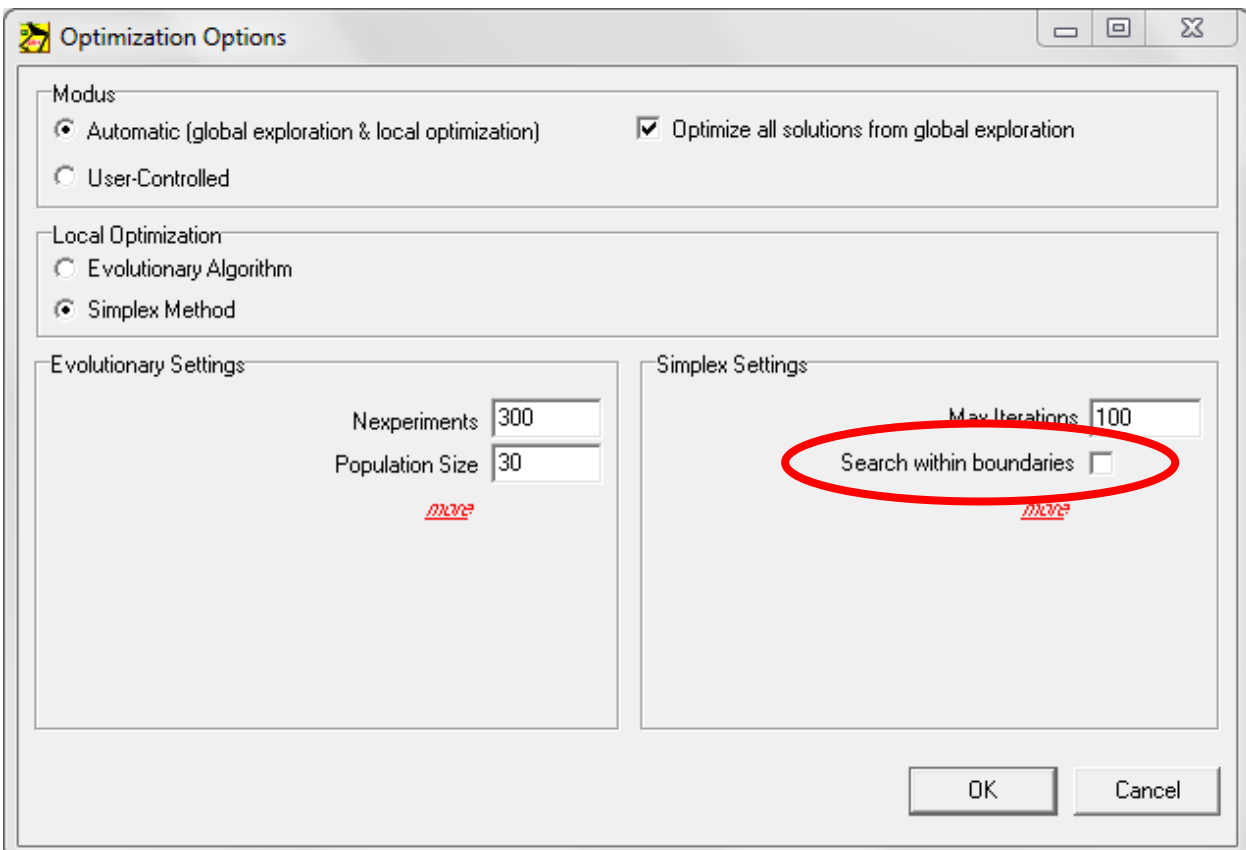


Optimized Mechanism – with constraint/penalty

If the penalty function is switched ON, which limits the length of element 3 to the range of 0.25-0.35m, the solution space is clearly smaller and the best result of the optimization is shown below.

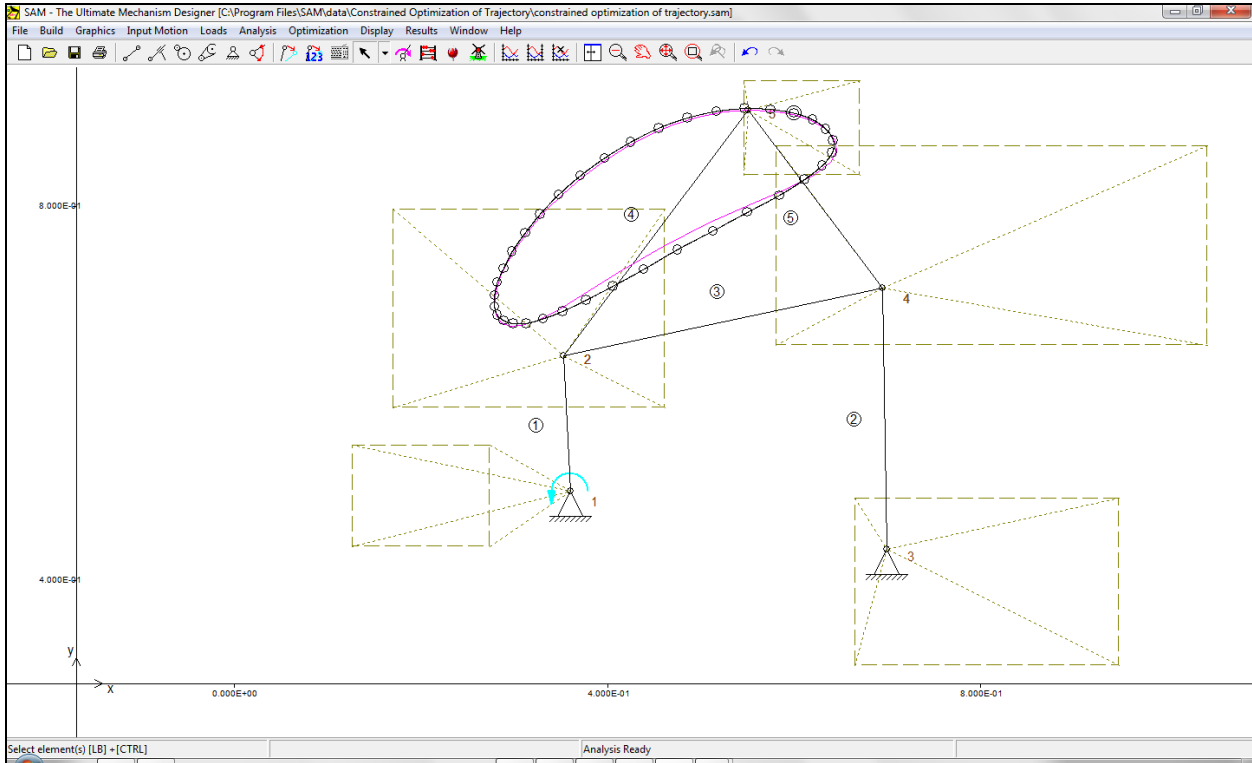


As one can see, node 1 is on the edge of the parameter rectangle, which suggests that better solutions might be possible if that limit is relaxed. The easiest way to relax all parameter space limits, without effecting the constraint defined by the penalty value, is to modify the Simplex settings (see below)



Optimized Mechanism – with constraint/penalty & relaxed design parameter boundaries

If the penalty function is switched ON, but the limits of the parameter space during the Simplex optimization are switched OFF (relaxed design parameter boundaries), one can find again a good match between actual and target trajectory, as is shown below. The only parameter no longer within the predefined regions is the x-coordinate of node 1. Whether this is a feasible option depends on the design challenge. This example is purely intended to illustrate the use of relaxing the parameter space, since it might help the designer in finding a good solution to his multi-dimensional design challenge.



Conclusion

A good engineering approach to finding a good solution to a multi-dimensional design challenge that also involves constraints is to use the following three-step approach:

1. Unconstraint optimization (to find the best case solution in the absence of constraints)
2. Constraint optimization (to find the best case solution to the constraint optimization problem)
3. Constraint optimization with relaxed parameter boundaries (to increase the solution space)

This is demonstrated in this example in which the trajectory/path node 5 should match as much as possible the desired reference trajectory. As a quality number to judge a solution candidate, the RMS value of the difference between actual and reference path is calculated. Each of the nodes of the mechanism can be varied within certain specified limits. As an additional constraint the length of element 3 should be within a range of 0.25-0.35m. It is demonstrated that the automated optimization procedure provided by SAM helps in tackling this design challenge.