Heron's Synthesis Engine Applied to Linkage Design
The philosophy of WATT Software

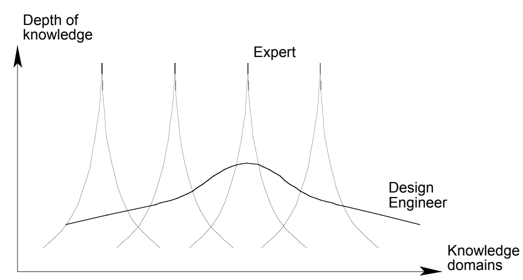
By:
Harry Draijer (Heron Technologies), h.draijer@heron-technologies.com
Frans Kokkeler (University Twente), f.kokkeler@ctw.utwente.nl

Abstract: This paper discusses the need for synthesis tools in more fields than mechanism design. Described are the principles to create more general synthesis tools and the application of WATT Mechanism Design Tool to the ASME Design Mechanism Challenge.

Introduction

Engineering Design is the task of creating a product, machine or device to solve a problem. The design task is never clear and a good solution often asks for integration of multidisciplinary solutions. The responsible designer becomes an integrator and the question is how many experts to throw at the problem. More people manning a project is not a guarantee to meet deadlines or stay within budget. A high quality design is never the sum of partial solutions. For true integration, insight in all areas of expertise is needed. A Design Engineer will never in his life be an expert in all fields.

To achieve more insight, the Engineer of today needs synthesis tools, to help him find solutions and explore alternatives in all sub-domains he needs, within project time constraints. In many knowledge domains the methods and theory have been and are being developed by the experts, but only very little is available for instant use.



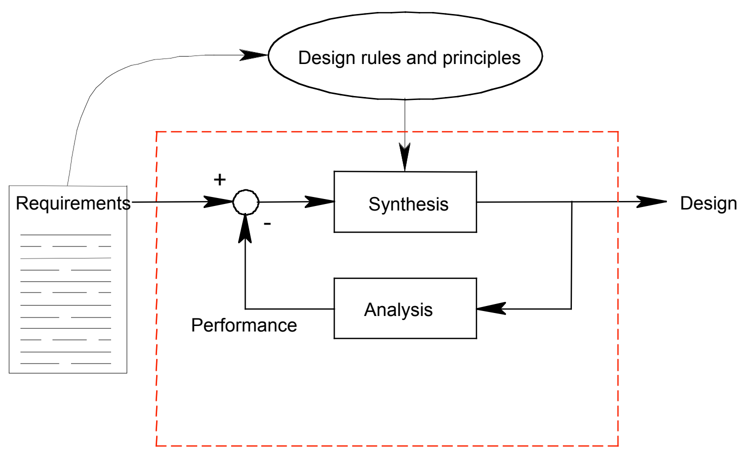
A Design Engineer communicating with an expert will tell him his requirements and will not take a single solution for granted. There are more problems that need to be solved. So he will try to find out which are the constraints most difficult to satisfy, which are critical points in the solution and if there are alternatives. This is the way the Engineer should be able to work with synthesis software. There should not be a constraining one-way street to a single solution. The Engineer should be in charge, be able to explore possibilities and have alternatives at hand.

In linkage design, readily available synthesis methods exist. It is also obvious that implementing these methods in software is the way to go. This paper describes the principles to create more general synthesis tools and the application of WATT Mechanism Design Tool to the ASME Design Mechanism Challenge.

Design and Synthesis

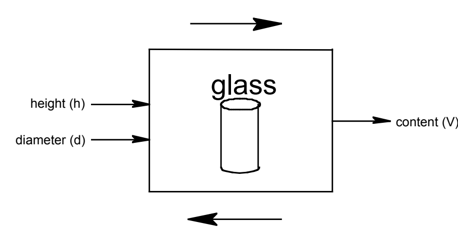
At the start of the Engineering Design Process, a set of qualitative requirements is translated into a list of specifications. Concepts of a possible Design are created. Each concept is a definition of sub tasks with their own set of specifications. For each sub task a possible solution will be proposed (synthesized) and through a loop of iterative analysis and modification a viable design may result.

Design process represented by a control loop



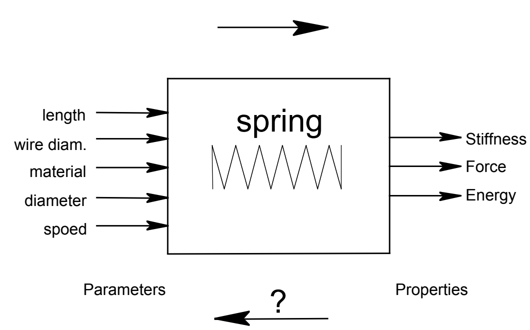
A synthesis software tool will only be able to work well with design tasks that allow the solution to be formulated as a mathematical model.

If the inverse of the mathematical model can be found, finding a solution would be easy. The design task of the glass (right) can be formulated as finding a height and diameter to create a certain required content. In this case the inverse of the mathematical model can easily be found.



However, in most cases the inverse model cannot be found. As the Engineer, the synthesis tool will need to go through the same loops of synthesis and analysis. The design task of finding a steel spring is just such a task.

Setting values for the **parameters** (length, wire diameter, material, diameter) of this model will define the spring. A manufacturer will be able to make it. Testing the spring will display its properties (stiffness, maximum force, elastic energy). The mathematical model will return its **properties** when all parameters are fully defined.



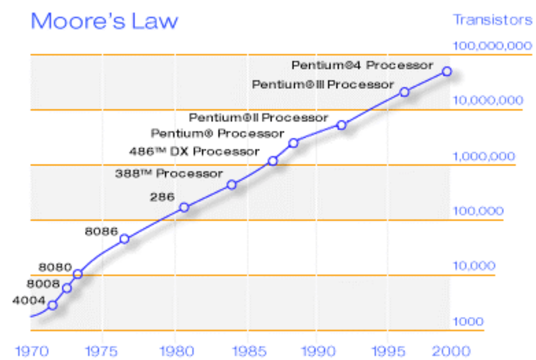
For a given spring stiffness, an enormous number of parameter combinations, must be evaluated on their suitability. A synthesis algorithm will be needed to find the alternatives in spring dimensions that exist for a required stiffness and known parameter restrictions. Finding one optimal solution is not the issue. Initially the Engineer needs different solutions more than the optimum.

Several search tactics can be used:

1. Brute force, checking every possible solution.
1. Randomly setting initial values and using local optimization to create different optimal solutions.
1. Random initial solutions, Genetic algorithms to improve the solutions available and local optimization to focus on each single solution.
1. Simulated Annealing to move around in solution space and gather potential solutions. Followed by optimization as the temperature drops.
1. Using search patterns, that step by step zoom in on areas of high potential

Some problems are small enough to be able to solve by brute force. Most problems remain too large even for the fast rising computer power available for the desktop. All other options consist of two stages:

1. Finding high potential initial solutions
1. Optimization



Reducing the search effort

All the above solutions ignore the fact that an experienced Engineer uses his knowledge to choose an initial solution. He knows which parameters to focus on and reduces the problem to manageable size.

Whatever methods used, to speed up time needed to find solutions or to tackle ever more complex problems two things need to be done:

1. Increase the ability to choose high potential initial solutions.
2. Reduce the number of parameters and the parameter range.

An initial solution does not have to be a good solution only better than choosing random values for each parameter. A system based on rules of thumb or principles of design offers good results.

If the number of parameters is n and the range of parameter values is k, the parameters search space is $P = k^n$. The parameter space determines the number of model evaluations needed to check every possible solution.

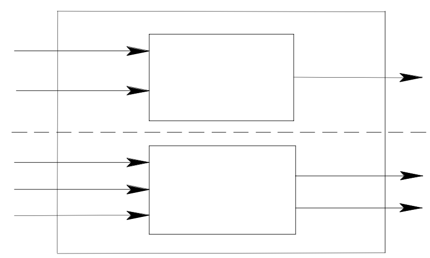
To reduce the number of parameters, rules of thumb may be used to create dependencies between say m parameters and reduce the parameter space to $P = k^{n+1-m}$. In the same way ignoring parameters with only marginal effect on the required properties aids in finding initial solutions faster.

Finding independencies or weak dependencies may substantially reduce parameter space by splitting the problem. If $n = m + p$ for two independencies, parameter space reduces significantly $P = k^m + k^p << k^n$

Apart from detecting real **independencies**, two approaches can be followed to split the task:

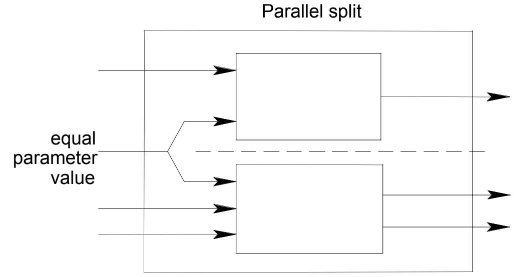
1. Parallel split
2. Sequential split

Independant



Parallel split can have effect when two parallel models seem to depend on a few of the same parameters. If there are i common parameters of $n = m + i + p$ parameters, search space reduces to
$P = k^{m+i} + k^{p+i} << k^n$

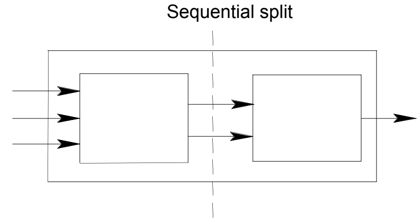
Parallel split



Another reduction in evaluation count is achieved if first one task is fully evaluated. In evaluation of the next task only the values of the common parameters are used that have resulted in valid solutions.

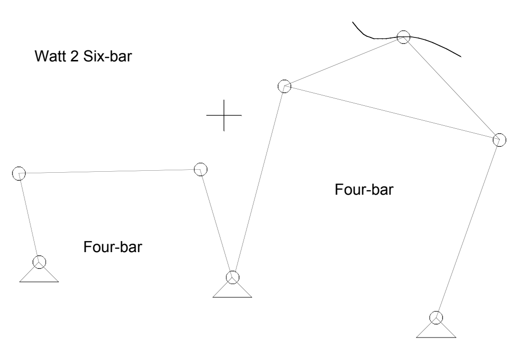
Sequential split is used where the design task can be seen as a sequence of tasks. The task is split and connecting constraints are formulated. These constraints may need to be changed if no solutions are forthcoming.

Sequential split



The Watt 2 function generator is a good example that allows this approach. The mechanism consists of a four-bar function generator mechanism connected to a four-bar path generator mechanism.

Function and path can be split into separate tasks.



User Interaction

The Engineer would like numerous Synthesis Tools for all kinds of knowledge domains at his fingertips. 80% of the time he should not need to consult either an expert in the field or a computer software specialist. The tool should be extremely easy to learn and use.

Part of the problem for the user is the interface. Studies, using focus groups have shown a clear User Interface should abide by the following rules.

- Single window interface.
- The interface must stay recognizable and should not change suddenly.
- No unclear error messages should be displayed in separate pop-up dialogs, that need to be clicked away to restore the problem.
- Dialogs only one level deep.
- Standard Windows layout.
- Primary functions within sight, specialist settings reached through Option Dialogs.

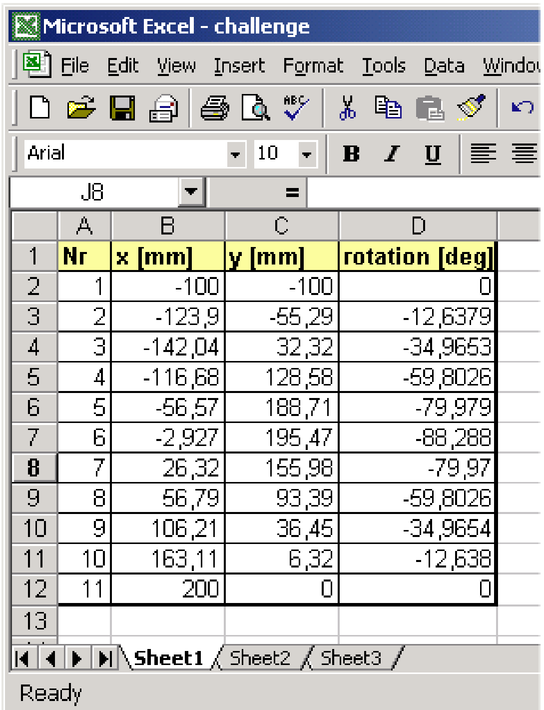
The user may be challenged by two other problems. The number of solutions presented may either be too large to comprehend or there may be none at all.

During the search process, the syntheses tool could investigate the parameter and properties relations. Resulting in advice or automatic adjustment of the most sensitive constraints.

If there are many solutions, there may be only few that are 'really' different for the Engineer. Clusters of look alike solutions confuse the engineer. The synthesis Engine could cluster look alike solutions and show only a representative to the user.

Design Challenge

For the ASME conference, a mechanism design challenge is posed. The object is to synthesize a four-bar linkage to follow a path defined by 10 points in a plane. At each point a certain rotation of the object being moved is desired.

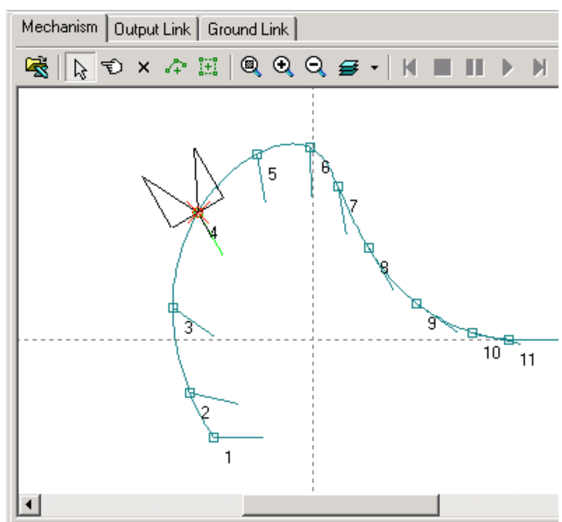


Nr	x [mm]	y [mm]	rotation [deg]
1	-100	-100	0
2	-123,9	-55,29	-12,6379
3	-142,04	32,32	-34,9653
4	-116,68	128,58	-59,8026
5	-56,57	188,71	-79,979
6	-2,927	195,47	-88,288
7	26,32	155,98	-79,97
8	56,79	93,39	-59,8026
9	106,21	36,45	-34,9654
10	163,11	6,32	-12,638
11	200	0	0

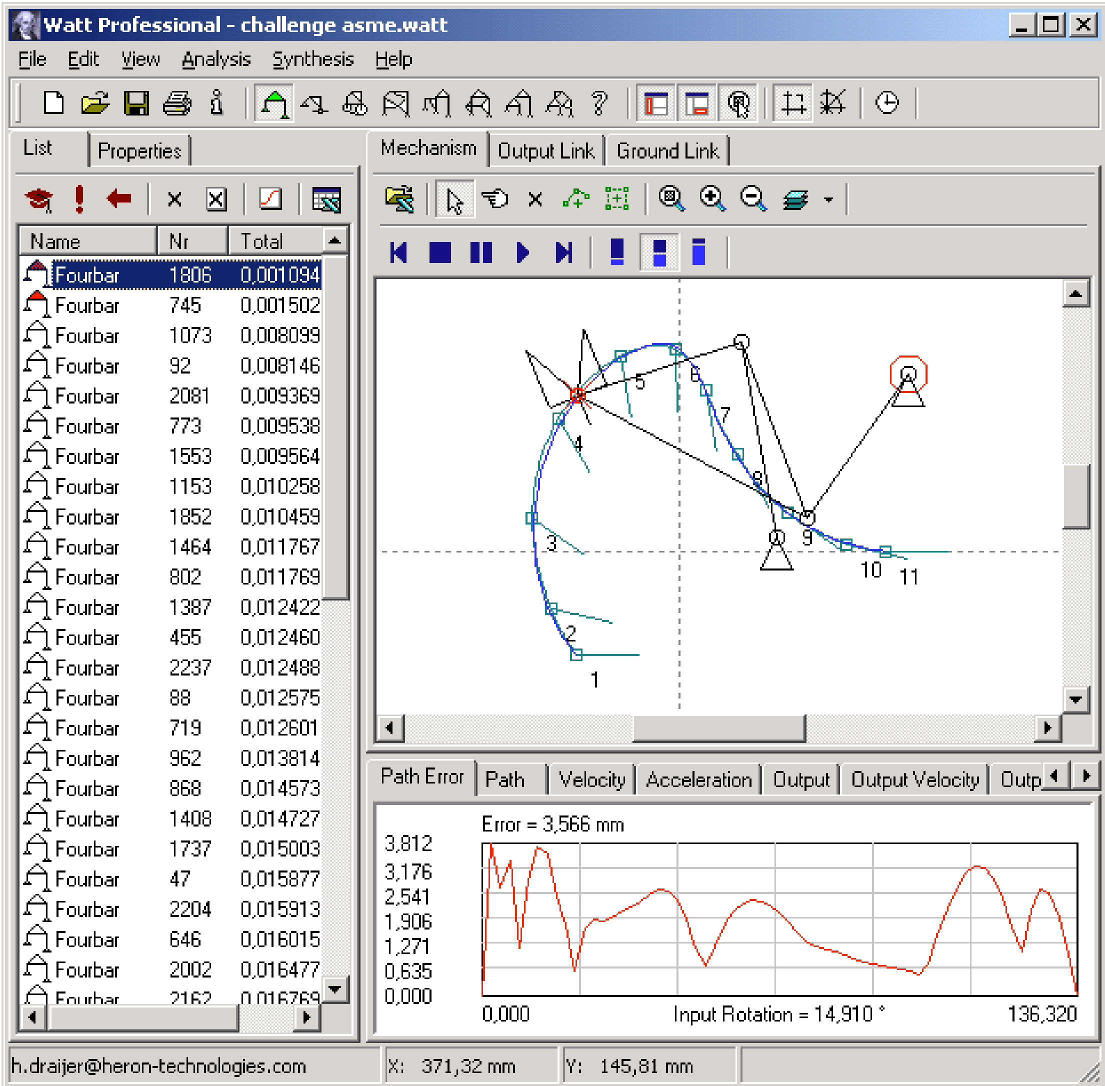
To tackle this challenge using Watt Mechanism Design Tool, the path settings were typed into an Excel spread sheet.

X,Y coordinates were multiplied by 100 to create reasonable [mm] settings. Rotations were adjusted to be referenced from the starting position.

This path definition can be imported from an Excel sheet. WATT uses the point definitions to create a spline, that defines the required path.



The synthesis process results in a list of several possible solutions. These can be viewed and animated for inspection. Suitable solutions can be optimized to fit the required path even better.



Optimization of a large quantity of the available mechanisms showed they all tended to converge to the same size and dimensions. Apparently the solution space is very limited for this problem.

Future Developments

Using the synthesis approach described, development has been started on a WATT alike Synthesis Engine. It will be able to help synthesize solutions to a diversity of problems. Every knowledge domain within Engineering (Electronics, Chemistry, Mechanics, etc.) could create its own Synthesis tool. Needed are:

- Mathematical Parameter - Property relations,
- In- and output representations and
- Synthesis Engine.

In the near future we expect Synthesis tools will be able to solve multidisciplinary- and higher order design problems. If Moore's Law holds true and if new synthesis methods are created the complexity of problems that can be solved will increase. This will impact the education of the student. It may be more focused on comprehension of basic principles and development of creativity. Engineers will be able to tackle more complex jobs and be more creative.

Literature

Davis, L. (editor), "Handbook of Genetic Algorithms", International Thomson Computer Press 1996, ISBN 1850328250

Erdman, G.A., Sandor, N.G., Kota, S., "Mechanism Design: Analysis and Synthesis", Volume 1 4th Edition

Hartenberg, R.S., and Denavit, J., "Kinematic Synthesis of Linkages", McGraw-Hill, New York 1961

Koster, M.P., "Constructie Principes", Twente University Press, Enschede 1996, ISBN 9036508320

Moore, G. E., "Cramming More Components Onto Integrated Circuits", Publication: Electronics, April 19, 1965

Petroski, H., "Design Paradigms", Press Syndicate of the University of Cambridge, New York 1994, ISBN 0-051-46649-0

Papalambros, P.Y., Wilde, D.J., "Principles of Optimal Design", Cambridge University Press 1988, ISBN 0-521-42362-7

Otten, R.H.J.M., and van Ginneken, L.P.P.P., "The Annealing Algorithm", Boston: Kluwer 1989