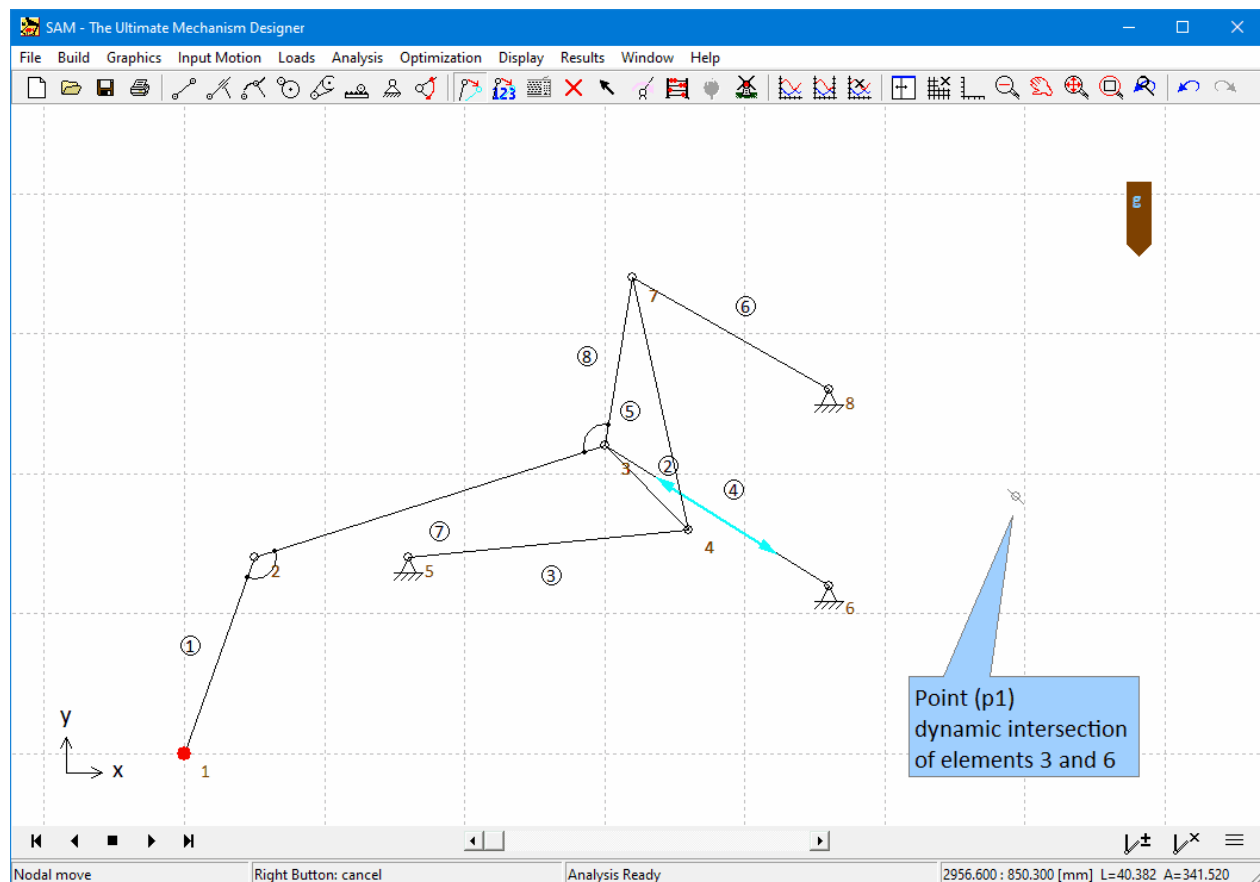


What is new in SAM 8.3?

Dynamic Points

The possibility to construct points via the intersection of lines and even the intersection of centerlines has become dynamic. When moving the objects used in the creation of a point, the xy coordinates of that point are adapted automatically. That holds for the situation that an object is move in the reference position, but also when performing an animation.

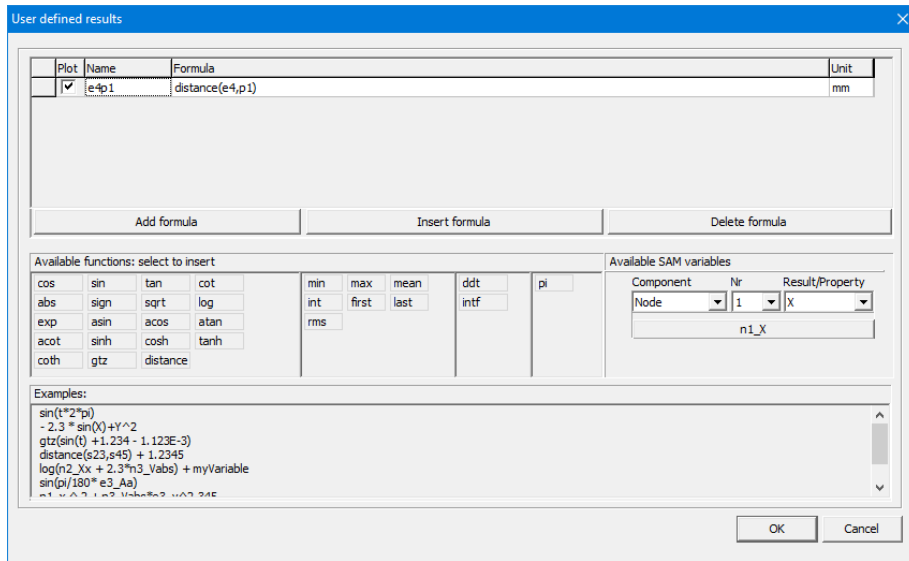
In this way one can for example create point p1 as the intersection of beam 3 and beam 6, which constitutes the momentary center of rotation of the triangle formed by the triangular coupler. When the mechanism is animated and moves from one position to the next, the location of point p1 is adapted automatically.



The point p1 is created as an intersection of the two (infinite) lines corresponding to the beam elements 3 and 6. The point is dynamic and moves during an animation.

Distance

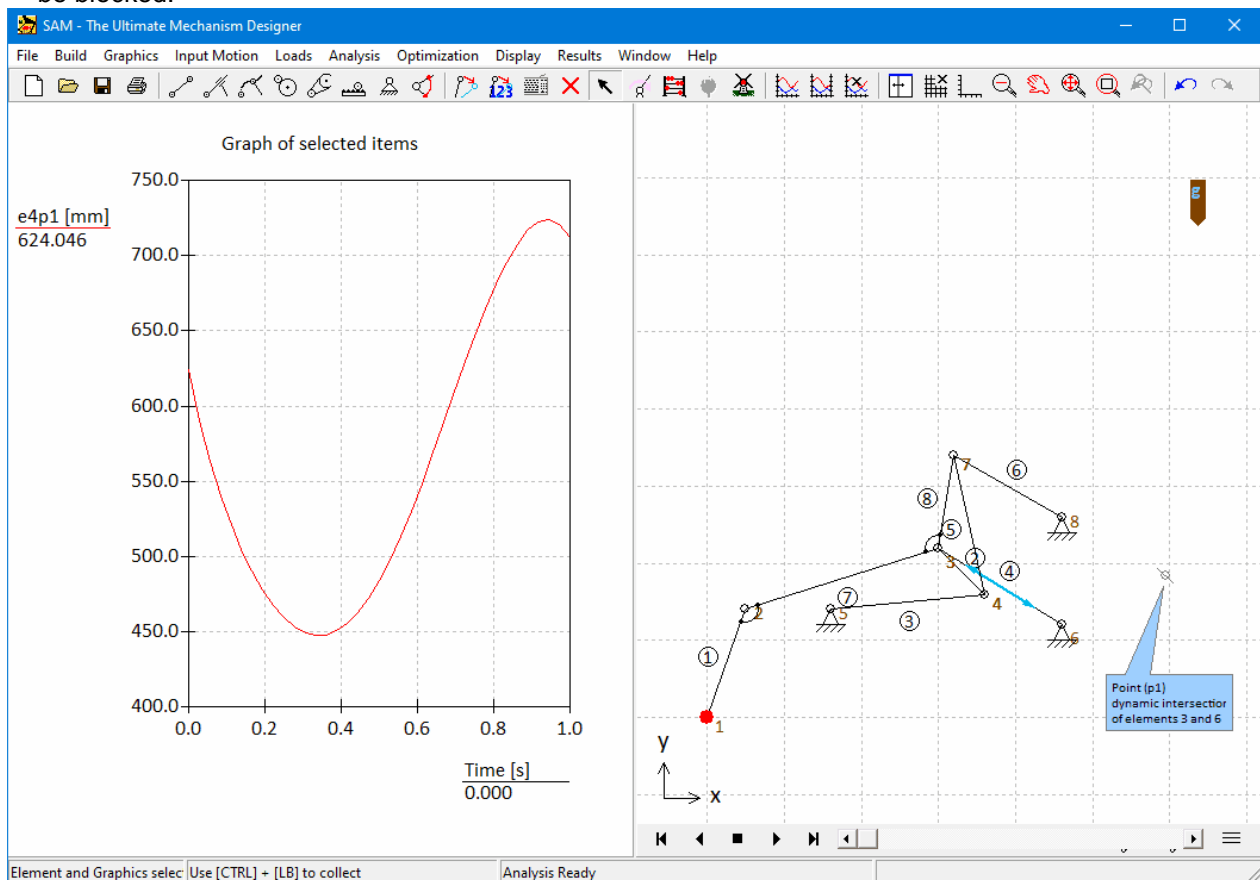
The postprocessing module (Results > User Defined ...) has been extended with a new function: `distance(object1,object2)`, in which an object can be either a node, element, point or shape.



Definition of the distance between element e4 and point p1.

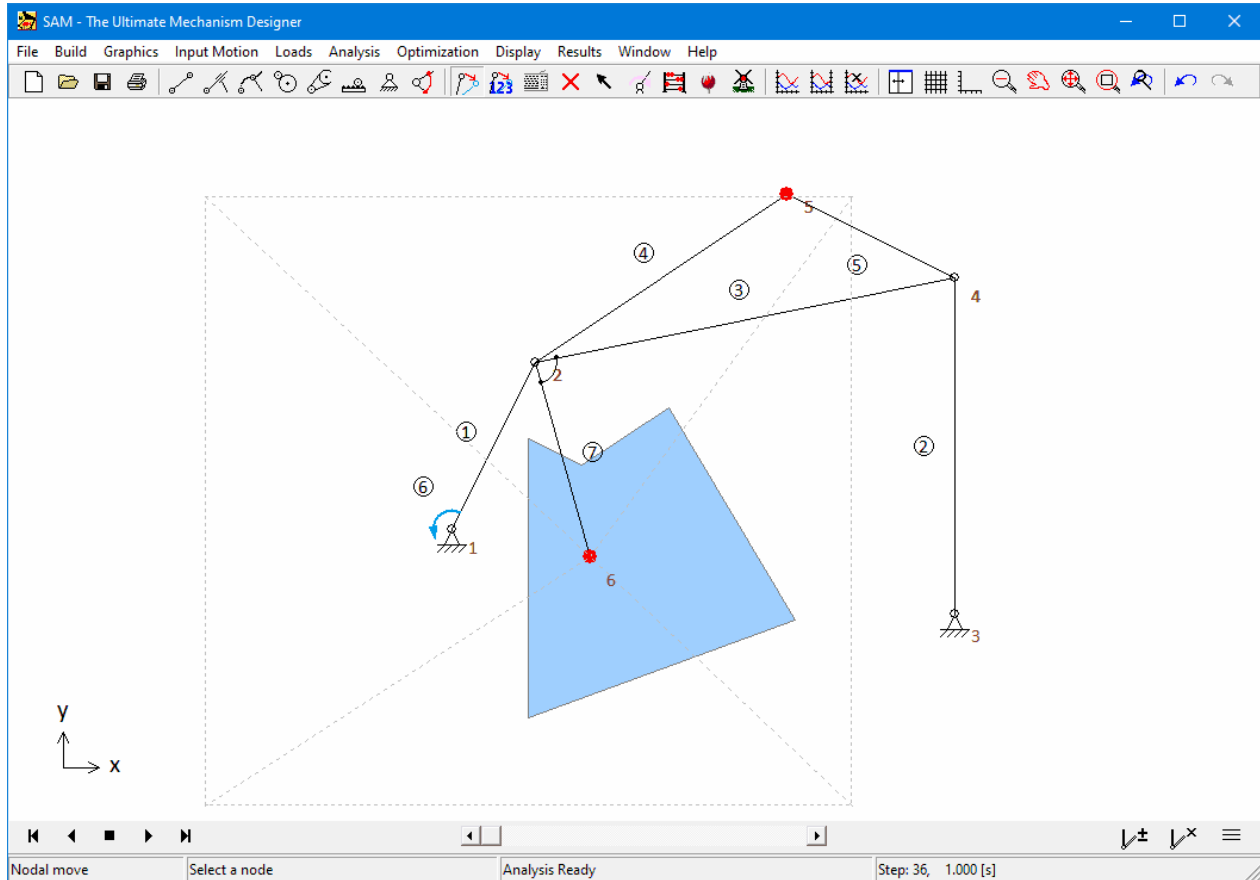
The use of this feature is manifold:

1. Plot the distance between two objects as function of time. In the previous example it might be useful to plot the distance between the driving cylinder (element e4) and the dynamic point p1 in order to monitor if that distance is always positive and does not reach zero, since in that situation the mechanism would be blocked.

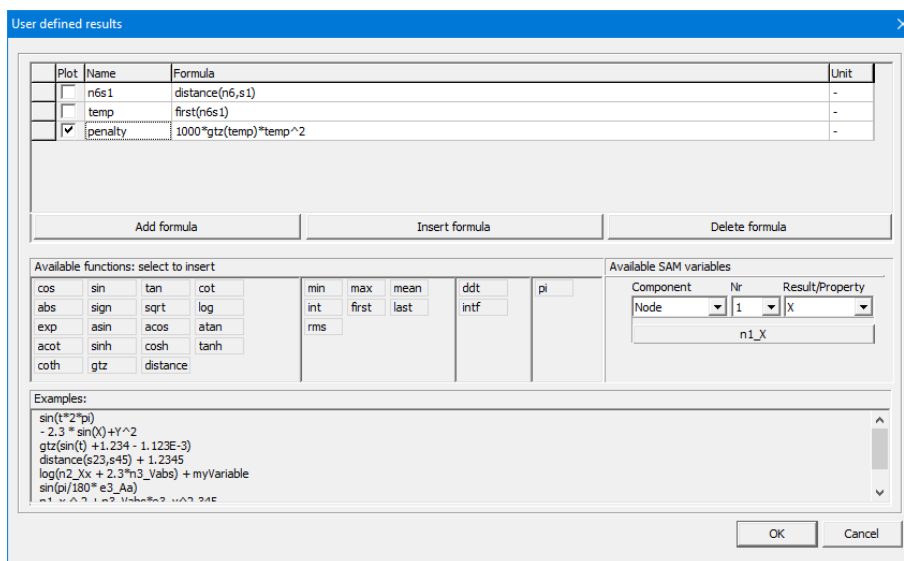


Distance between element e4 and point p1 as function of time is plotted next to the mechanism

2. During an optimization, the xy-location of a node can be an optimization parameter. Using the distance function one can define a penalty function for the optimization in order to constrain the xy-location of that node to any polygon



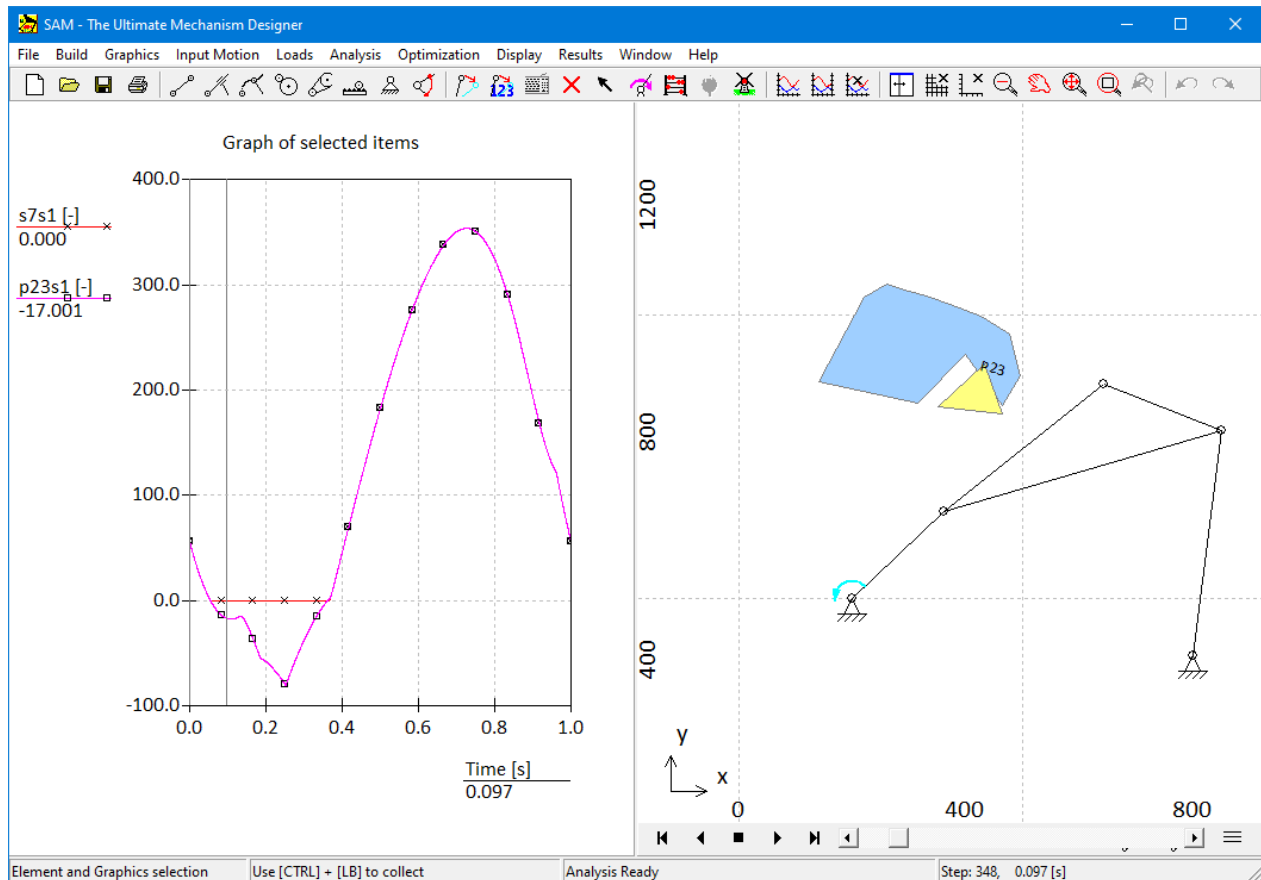
Optimization in which node n6 is a design parameter but should be constrained to the area of the polygon (shape s1).



Definition of the penalty function to constrain node n6 to the polygon s1. As soon as node n6 lies outside of the polygon in the initial position of the mechanism, the penalty is larger than 0.

3. Collision detection between objects

The distance function can be used to detect collision between objects and depending on the type of objects even the penetration depth (or minimum retraction distance to avoid collision). In the following example the blue polygon is a stationary shape $s1$, whereas the yellow triangle (shape $s7$) is attached to the coupler. In the plot one sees the variable $s7s1$, which corresponds to the distance between the two shapes and the variable $p23s1$, which corresponds to the distance between point $p23$ on the triangle and the blue polygon. In case of two colliding shapes the distance is set to zero, whereas in the case of a point or node moving into a shape both the positive and negative distance (negative distance = penetration depth) is shown in the plot.



Example of collision detection via use of the distance function.

Copy/Paste (Ctrl-C/Ctrl-V) for graphics/shapes

One can now use Ctrl-C and Ctrl-V to copy and then paste any shape.