

# Effective design and implementation of specific displacement diagrams to control kinetic sculptures

Pavel Dostrašil<sup>1,\*</sup>

<sup>1</sup>VÚTS a.s., Mechatronic Department, Svárovská 619, 460 01, Liberec XI, Czech Republic

**Abstract.** Electronic cams are used for different manufacturing systems, but in terms of displacement diagrams, they have common characteristics. The emphasis is usually placed on maximum accuracy, minimum machine cycle time and the displacement diagram has a simple shape. This paper addresses a completely different case, which shows that the use of electronic cams is very diverse. An Omron's electronic cam was used to control kinetic art sculptures. It was necessary to develop an implementation that would be able to accommodate a large number of very long and complex displacement diagrams. Some sculptures contained up to 147 interpolating axes and their programs took up to an hour. The proposal builds on the basic animation and designer's demands, but it must comply with all the limits of the mechanism (maximum speed, torque, etc.). For this purpose, an independent software tool was developed. The final displacement diagram is composed from polynomial of the 5th order by defining the 0th, 1st, and 2nd derivatives at the key points. This method of design has proved to be very effective, and in addition, this implementation brought a significant saving of memory and reduction of computational complexity.

## 1 Introduction

For production systems, displacement diagrams are usually supplied by the contracting authority or generated by specialized software tools (in VÚTS, we use our own tools KINZ [1] and KINZ2 [2]). Displacement diagrams (Fig. 1) can be composed from different functions (higher order polynomials, trigonometric functions, exponential, and combinations thereof). In the PLC (Programmable Logic Controller), data is stored in a table where the first column contains the coordinates of independent variables, i.e. the virtual master (hereinafter referred to as VM). The second column represents the rotation of the output shaft of the motor. The next columns may contain higher derivatives which can be used to improve the regulatory process. The step of the VM can be constant or variable.

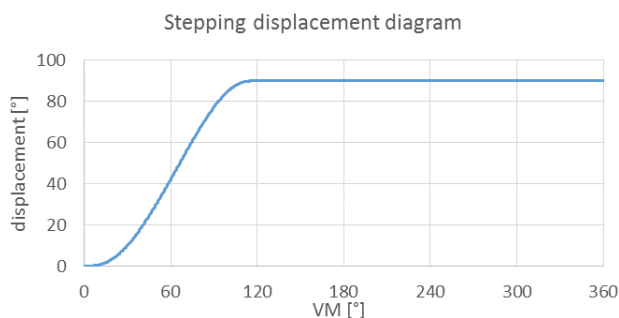


Fig. 1. Displacement diagram used e.g. for rotary table.

Based on the contract for the controlling of kinetic sculpture [3], it was necessary to revise the above approach. The customer usually delivers the data in the form of a motion function (dependency of the output member on time) (Fig. 2). Motion function were supplied in the form of data table, but it cannot be directly used for the electronic cam. Data was created in graphic software tools, so they did not apply the continuity of higher derivatives (velocity and acceleration) and they did not take into account the physical limits of the mechanism (maximum speed, tilt limitation of individual elements, etc.).

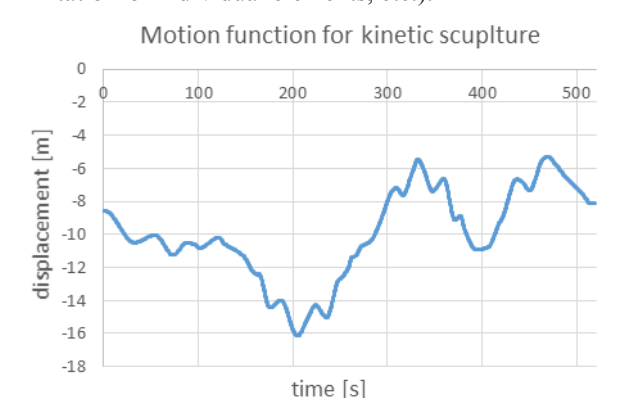


Fig. 2. Part of the motion function for one axis of the kinetic sculpture.

The last installed sculpture contained 147 controlled axes, 3 scenes, and each scene lasted approximately 20

\* Pavel Dostrašil: [pavel.dostrasil@vuts.cz](mailto:pavel.dostrasil@vuts.cz)

minutes. This is a large amount of data that needs to be checked and implemented in the PLC control program.

If we want to use the same approach we used for the production systems, the implementation of the PLC management program would be difficult because we would have to work with a huge amount of data. The biggest problem would be the extreme time requirement for checking and correcting data from the customer. For these reasons, a specialized software tool for data preparation for kinetic sculptures was created. A displacement diagram from the customer serves as an input and is interpreted as a sample curve. The tool enables the creation of a new displacement diagram that is composed of several polynomials. The tool implements a number of procedures to manually or automatically create a new displacement diagram, which is the most similar to the pattern curve. Before using a new displacement diagram for a specific mechanism, a number of checks are required. Some constraints are simple, such as adherence to displacement diagram limits and its derivations. However, some mechanisms require more complicated collision control algorithms. These checks will usually not pass, and minor interventions must be made in the shape of the displacement. For these corrections, polynomial implementation is very convenient. The shape of the curve can be intuitively modified using the 0<sup>th</sup>, 1<sup>st</sup> and 2<sup>nd</sup> derivative values at the boundary points of the individual polynomials. In detail, the issue of a displacement diagram design is discussed in the next chapter.

The output is no longer a data table, but the parameters of the individual polynomials. The first version of the software tool cooperated with Yaskawa PLC and wrote coefficients of polynomials directly to the PLC memory via Modbus protocol. Current projects are built on the Omron platform, which allows us to read data files from an SD card. The software tool uses a more general approach using binary data files. The method of data transmission and basic description of the PLC control system is outlined in Chapter 3.

The first kinetic sculptures included less controlled axes. Our first attempts were based on the use of the on commonly available software tools. The first kinetic sculpture was solved by the composition of the harmonic function and the polynomial [4]. MS Excel was used for the solution. The solution was built on a specific case and other adjustments according to the current customer requirements were difficult. We also tested using the freely available GeoGebra tool [5]. This solution was not practically used because it had similar drawbacks as the previous approach. After realizing the first sculpture, we received orders for other significantly more complex mechanisms. Implementation of its own general software tool was a logical process.

## 2 Creating a displacement diagram

The displacement diagrams of the kinetic sculptures are subject to different requirements in comparison to production machines. Maximum accuracy is not critical. Important is the ability to quickly design and easy

modifiability. It is often the case that the last modification is made to the customer on a completely finished product. It must be possible to immediately adjust the displacement diagram and check whether the mechanical limits are exceeded or there is no collision.

These are the main reasons why we stopped using the other software tools for designing displacement diagrams and developed a specialized tool for kinetic sculptures. The first versions of these tools did not support any of the automatic features described in this chapter. They were always determined for one particular application built on the Yaskawa platform with which they communicated directly through the Modbus protocol. [6] The currently-used tool exports binary data files for PLC on Omron. The tool is much more general and only new control functions are added for specific applications (for example, control of collisions that are specific for the design).

### 2.1 Division on polynomials

Working with the original customer data table quickly proved to be ineffective. Even minor adjustments were very difficult. The table occupied a large amount of PLC memory. Sometimes they contained discontinuities and could not be used directly. The software tool will only load the data table from the customer, performs the filter to a reasonable number of values and uses it only as a pattern curve. We divided the displacement diagram into a shorter section, which we replaced with a selected function. We chose a 5<sup>th</sup> degree polynomial, because it can easily be defined by the boundary conditions 0<sup>th</sup>, 1<sup>st</sup> and 2<sup>nd</sup> derivatives. Neighboring polynomials always share their boundary conditions, which ensures the continuity of the resulting displacement diagram to the second derivative. The function is determined by a conventional notation [7].

$$y = c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \quad (1)$$

For implementation in the control system, notation is used according to Horner scheme.

$$y = (((((c_5x + c_4)x + c_3)x + c_2)x + c_1)x + c_0 \quad (2)$$

This implementation guarantees the most effective method of calculation. It used only 5 additions and 5 multiplications. The calculation of individual coefficients is more complex as described in equations 3 to 7. The calculation of coefficients takes place only when designing in the PC and the control program in the PLC does not matter.

$$c_0 = y(0) \quad (3)$$

$$c_1 = y'(0) \quad (4)$$

$$c_2 = \frac{y''(0)}{2} \quad (5)$$

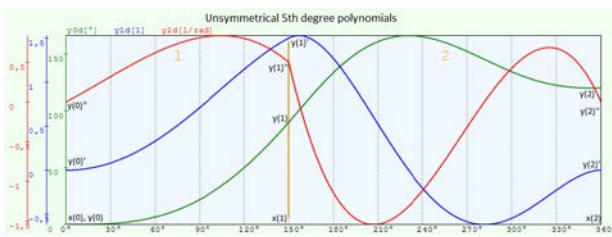
$$c_3 = \frac{1}{2(x(1)-x(0))} \left( 20 \frac{y(1)-y(0)}{(x(1)-x(0))^2} - 4 \frac{2y(1)' + 3y(0)'}{x(1)-x(0)} + y(1)'' - 3y(0)'' \right) \quad (6)$$

$$c_4 = \frac{1}{2(x(1)-x(0))^2} \left( -30 \frac{y(1)-y(0)}{(x(1)-x(0))^2} + 2 \frac{7y(1)' + 8y(0)'}{x(1)-x(0)} - 2y(1)'' + 3y(0)'' \right) \quad (7)$$

$$c_5 = \frac{1}{2(x(1)-x(0))^3} \left( 12 \frac{y(1)-y(0)}{(x(1)-x(0))^2} - 6 \frac{y(1)' + 8y(0)'}{x(1)-x(0)} + y(1)'' - y(0)'' \right) \quad (8)$$

**2.2 Definition of boundary conditions**

Individual functions are defined by the boundary conditions. To maintain good quality control, the end and initial boundary conditions of adjacent functions are always the same. The software tool allows you to change the boundary points by using form controls or keyboard shortcuts. This method is useful if the customer's sample curve has very few points or if manual adjustments are required. After brief training, the user quickly learns the curve behaviour. Small adjustments to the displacement diagrams are very common and very fast for experienced users.



**Fig. 3.** The creation of displacement diagrams using boundary conditions.

**2.3 Snap to the pattern curve**

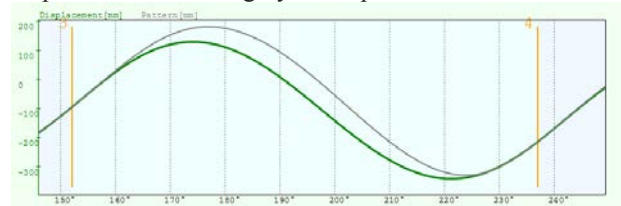
If the customer delivers a sufficiently dense pattern curve, the process can be further automated. The software tool implements an algorithm for the numerical calculation of the 1<sup>st</sup> and 2<sup>nd</sup> derivatives at any border point. The tool automatically snaps a 0<sup>th</sup> and 1<sup>st</sup> derivative. For a 2<sup>nd</sup> derivative, a scale of 0 to 100 percent can be set. Decreasing the influence of the 2<sup>nd</sup> derivative is advantageous for a small number of sample points.

If the user creates a sufficient number of appropriately selected borders, it is possible to create the resulting displacement diagram without any further

adjustments. However, unnecessarily large numbers of polynomials are not beneficial, because it increases the memory demands and complicates manual editing.

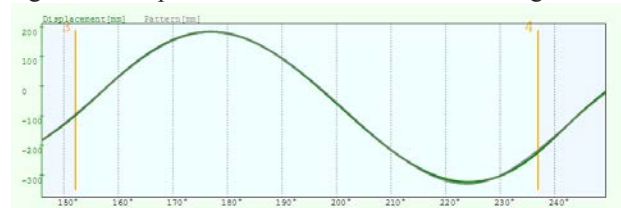
**2.4 Automatic minimization of deviations from the pattern curve**

If we want to achieve smaller errors (the area between the pattern and the resulting curve) with fewer polynomials, it is usually necessary to modify individual boundaries manually. Deviating from the calculated boundary conditions, a smaller overall error can be achieved. In the case of inappropriately selected boundary points, this error may be significant as shown in the following figures where the green curve is the displacement and the grey is the pattern.



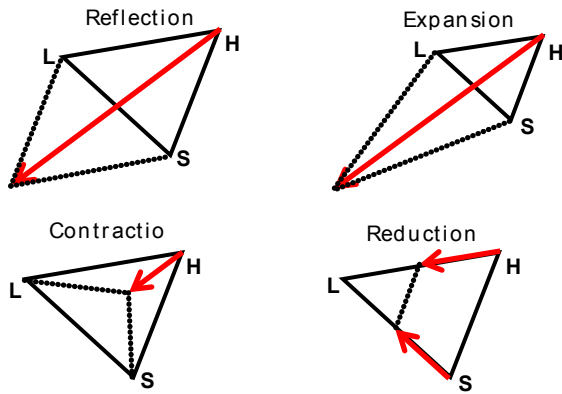
**Fig. 4.** Snap border points to the pattern curve.

Minor editing of boundary conditions may lead to significant improvements as shown in the next figure.



**Fig. 5.** Editing boundary points to reduce overall error.

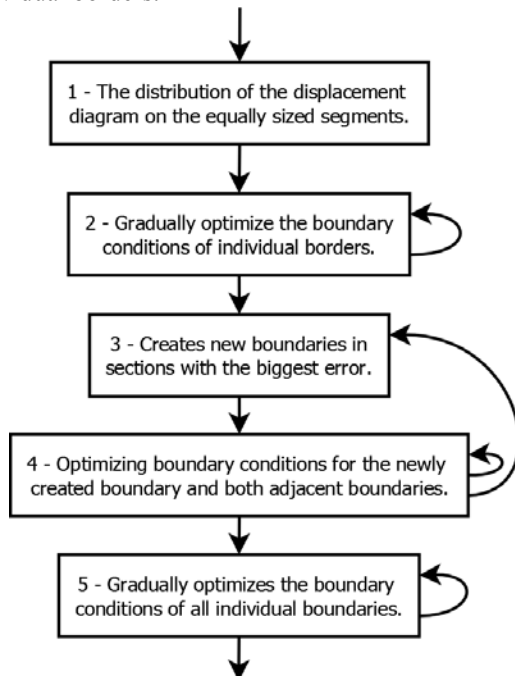
Due to the large amount of data, we decided to automate this process. To find optimal parameters, we used the Nealder-Mead iteration algorithm [8]. This method is one of the comparative methods. It searches for the minimum of the cost function, comparing their values in specific selected points in  $R^n$  and does not require the knowledge of their derivation. The method is generally n-dimensional. The following figure demonstrates the functioning of the algorithm in a two-dimensional role. The first step is the selection of the initial simplex (points L, S and H), where H (high) represents the worst point and L (Low), the best point. The following steps of the iterative algorithm are trying to replace the point with the highest value of cost function H with a better one. It uses several different transformations. In this case, four basic transformations are used, mentioned in Fig. 6. Detailed descriptions of the algorithm can be found in [8].



**Fig. 6.** Transformation of two-dimensional simplex.

The first step was the creation of a library, implementing a generally multidimensional iteration Nealder-Mead algorithm. A solver object runs in its own thread for better handling of the GUI (graphical user interface). The library contains many debug functions for easier tracking of the iterative process.

The second step was to develop a procedure that applies the solver to a specific problem. Simple optimization of individual sections is not enough, because the adjacent polynomials partly share their boundary conditions. One boundary is shared by two polynomials. The correction of one polynomial affects two adjacent ones. We have developed a process of gradual optimization of boundary conditions along individual borders.



**Fig. 7.** The procedure for automatically designing the displacement diagrams.

Individual phases of the solution:

- 1) Divide the entire displacement diagram in a user-defined number of segments. To accelerate the iterative process, snap the boundaries to the pattern curve.

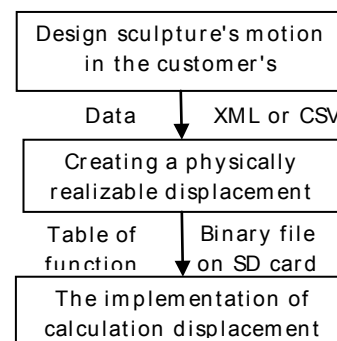
- 2) Gradually optimize all boundaries. The whole phase is repeated several times to reduce the impact of adjacent boundary conditions.
- 3) Calculate the error for individual sections. Select the worst and divide it into two.
- 4) The newly created border snaps to the model curve. Optimize the boundary conditions of the new border and neighboring borders. The optimization process is repeated several times. If the user-defined total number of boundaries is not reached, it returns to phase 3.
- 5) Gradually optimize all boundaries. The whole phase is repeated several times.

The number of repeated passages per phase can be selected by the user in the application window. Similarly, the number of polynomials created in the first phase and the number of polynomials added in the third phase.

The component implementing the Nealder-Mead solver was developed in general and did not have the limited dimension of the problem solved. Instead of a gradual optimization of individual boundaries, this allowed the modifying of all the boundary points at once. For simple displacement diagrams (about 5 polynomials), this method worked well. But for more complex displacement diagrams (20 and more polynomials), much more operations were needed to achieve a similar error. The time requirement rose up to several orders.

### 3 Implementation in the control system

Based on the data from the customer's animation software, physically realizable displacement diagrams were created from specialized software tools. The curves are usually made up of many functions of the 5th order polynomial. The displacement diagrams or more precisely the parameters of functions are saved as a binary file. The form of the binary file is determined directly by the data structure in the PLC. For a definition of one polynomial, there are seven necessary coefficients (x-coordinate of the end point and 6 polynomial coefficients).



**Fig. 8.** Data flow diagram in the creating of displacement diagram.



Kinetic sculptures are controlled by the NJ5 or NX7 PLC Omron series. The PLC has an SD card slot and supports data transfer via FTP. The main part of the control program works as a state machine (Fig. 9).

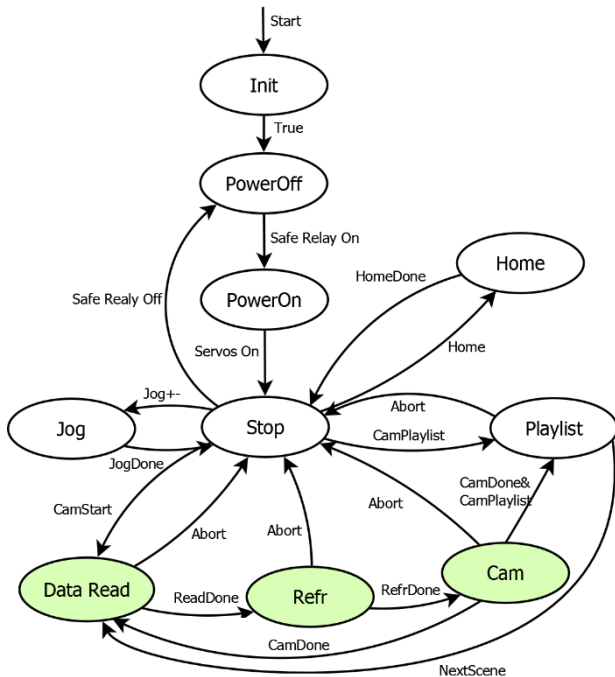


Fig. 9. Basic structure of the control system.

From the data processing point of view, especially the green highlighted states are significant:

**Data Read** – Status ensure load current scene data from the SD card into the memory structure.

**Refr** – Before starting the scene, all axes must be moved to their initial positions (the first coefficient of the first polynomial for each axis).

**Cam** – The Cam state provides two basic functions - it manages the VM and provides for the calculation of the individual axis positions.

The control of kinetic sculptures as the field of electronic cams tied to one VM proved to be a very robust solution. It is easy and safe to perform a smooth stop, reverse, etc. as the PLC treats the VM as a real axis. The calculation of the position of all axes relative to the VM is very fast (tens of microseconds). In contrast, the operation of communication for all EtherCAT drives (147 axis), safety CPU and other I/O takes almost one millisecond.

A detailed description of the selected kinetic sculpture deals with the prepared article [9].

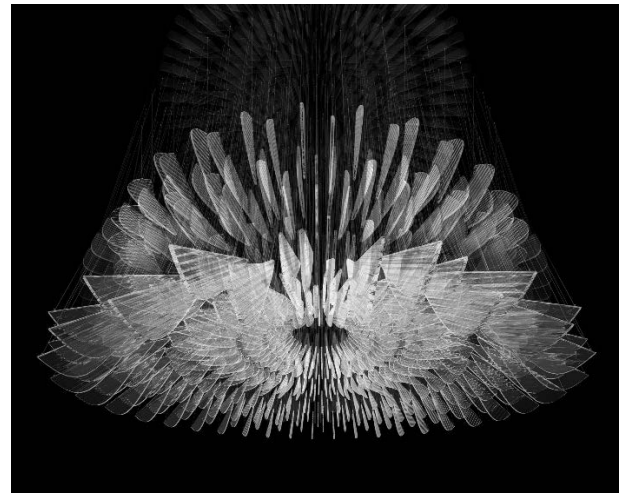


Fig. 10. Kinetic sculpture Supernova.

## 4 Conclusion

The realization of the kinetic sculptures brought completely new requirements for the implementation of displacement diagrams. In place of a simple but strictly defined displacement diagram, it was necessary to effectively implement complex movements whose purpose was not to create a product, but the artistic impression.

At first, the developed method divided the displacement diagram to individual sections which are replaced by the 5<sup>th</sup> order polynomial. This approximation is sufficient to preserve the character of the movement and also very efficient from a technical perspective. The displacement diagram keeps the continuity to the second derivative (acceleration) and its memory and computational requirements are relatively small. Another great advantage of this approach is the easy modification of the resulting displacement diagram by editing the boundary conditions of the individual polynomials.

The current version of the software tool for the displacement diagram design contains several features for quick design. In particular, the automatic displacement diagram design according to the pattern curve the iterative algorithm Nealder-Mead. This makes it is possible to match the pattern curve better for a fraction of the time compared to the manual design. As a result, data preparation for the kinetic sculptures has been shortened from days to hours.

This is already the third version of the software tool for the design of displacement diagrams. The structure of the application is very open to easily adapt it to the specific requirements of a particular project. For example, the last modification added a function to verify the maximum tilt of the glass elements of the sculpture.

The practical outcome is a graphical software tool for a fast design of a displacement diagram based on data from the customer graphical tool. Everything has been practically applied in the implementation of ten kinetic sculptures and is ready for use in other similar systems.

Future developments depend mainly on other orders. With more frequent use, it would be advisable to make a number of minor enhancements to increase user-

friendliness. For example, direct access to PLC, improved graphics outputs, etc. Making a bigger adjustment would be to improve the automatic design process so that it automatically chooses the position of the x-point boundary points. The transition from the manual design of the displacement diagrams to the current semi-automatic, saved a huge amount of time. However, the returnability of other algorithm improvements is very unclear.

## References

1. P. Jirásko, *Methodology of electronic cam applications in drives of working links of mechanisms of processing machines*. (Dissertation, Technical University of Liberec, Faculty of Mechatronics, Informatics and Interdisciplinary Studies, 2010)
2. P. Dostrašil, *Kinetostatic synthesis of stepping mechanism with a classical and electronic cam*. (Dissertation, Technical University of Liberec, Faculty of Mechatronics, Informatics and Interdisciplinary Studies, 2014)
3. Lasvit: *Supernova* [online]. [cit. 2018-03-08]. <http://lasvit.com/bespoke-glass-installations/emotions/supernova--309/about>
4. P. Jirásko. *SW prostředky řešení mechanismů s elektronickou vačkou - Prostředí MS Excel*. P. Jirásko et al. *Mechatronika pohonů pracovních členů mechanismů*. Liberec: VÚTS, 2015, p. 735-740. ISBN 978-80-87184-63-9.
5. P. Pátek. *SW prostředky řešení mechanismů s elektronickou vačkou - Geogebra*. P. Jirásko et al. *Mechatronika pohonů pracovních členů mechanismů*. Liberec: VÚTS, 2015, p. 741-748. ISBN 978-80-87184-63-9.
6. P. Dostrašil, *Implementation of Specific Displacement Diagrams for the Control of Kinetic Sculptures with Yaskawa Electronic Cams*. In: *Advances in Mechanism Design II*. Cham: Springer International Publishing, 2017, 2017-08-03, 377. *Mechanisms and Machine Science*. DOI: 10.1007/978-3-319-44087-3\_51. ISBN 978-3-319-44086-6. Online: [http://link.springer.com/10.1007/978-3-319-44087-3\\_51](http://link.springer.com/10.1007/978-3-319-44087-3_51)
7. Z. Koloc, M. Václavík, *Cam Mechanisms*, (1st Ed. Amsterdam: Elsevier, 1993. ISBN 04-449-8664-2)
8. L. Čermák, R. Hlavička, *Numerické metody* (Akademické nakladatelství CERM, 2008, ISBN 978-80-214-3752-4)
9. J. Antoš, M. Bušek, P. Dostrašil. *The Movable Mechatronic Kinetic Sculpture with Glass Pendants on Electronic Cams*. ICCS 2018 (to be published).