

SOFTWARE FOR GRAPHICAL INVESTIGATION OF PLANAR AND SPATIAL CLOSED KINEMATIC CHAINS FOR ROBOTS AND MANIPULATORS

GRIGOR ZHELEV

*Institute of System Engineering and Robotics, Bulgarian Academy of Sciences,
Acad. G.Bonchev Str, Bl 2, 1113 Sofia, Bulgaria,
e-mail: GrigorZhelev@vector-systems.eu*

[Received 19 November 2012. Accepted 21 January 2013]

ABSTRACT. The aim of this paper is to present a developed computational program for computer graphical modelling of planar and spatial robots and manipulators closed kinematic chains. In order to demonstrate the functionality of the program two types of 5-bars and two types of 8-bars closed manipulators are used for simulation of the kinematic models. The basic mathematical calculations for visual representation of kinematic models are made, as well as calculations and visualisation of the manipulators zone of reachability frame by frame. The presented program can be used for analyzing closed kinematic chains, solving tasks from the real practice or for students training.

KEY WORDS: Simulation, manipulators, visualization, graphical modelling, software.

1. Introduction

Simulation has been recognized as an important research tool since the beginning of the 20th century. We can work with the exact geometric characteristics of the objects using modern tools for simulation, to evaluate the dynamic characteristics of the system, to insert the man-machine interface and to visualize the object in 3D-space. For example, the algorithms for planning of trajectories depend strongly on kinematic models [1].

There are many software packages and many tools for simulation of kinematics and dynamics models in the robotics. The package SD/FAST can be used to perform analysis and design studies on any mechanical system which can be modelled as a set of rigid bodies interconnected by joints, influenced by forces, driven by prescribed motions, and restricted by constraints [2]. Open

dynamic engine (ODE) is open-source, highly-productive library for simulation of the dynamics of rigid body, virtual objects and creatures in games or in car-simulation. ODE is straightforward software used for building robot models where we can create all parts and connect them with proper joints [3]. Webots is software which can be used for creating robots of type Lego [4]. This software offers a fast prototype environment for modelling, programming and simulation of mobile robots [5, 6]. In addition the added libraries allow the user to transfer his new programs into a multitude of real commercial robots. MATLAB is one of the most used platforms for modelling and simulation of various kinds of systems [7]. As an extension to MATLAB, toolbox Simulink adds many features for easier simulation of dynamic systems, e.g. graphical models and the possibility to simulate in real time. With SimMechanics the user can model and simulate mechanical systems, to specify bodies and their mass properties, kinematic constraints, and coordinate systems, and to initiate and measure body motions [8, 9].

As a rule, the existing software products are either expensive, or with limited functionality. The young scientists and students need free software which allows creating and editing a chosen set of manipulators, and let them explore their functionality visually. The software should calculate the kinematic and dynamics of the mechanism, to export work into most popular file formats, as well as to work as a plug-in or extension to some popular engineering product.

Based on models and methods of the above mentioned programs a new program for computer graphical modelling of two types of 5-bars and two types of 8-bars closed manipulators was developed in the Institute of System Engineering and Robotics, Bulgarian Academy of Sciences.

The program is named RobotDesigner and is a very suitable programming tool for simulations that will allow the students and engineers easily to achieve best results in their work with manipulators. The program is developed in C# for Windows, requires only .Net 2.0 and is accessible for free download from the following location: <http://www.vector-systems.eu/Demo/RobotDesigner.exe>.

2. Methods

The developed software RobotDesigner extends the work in the area of the planar closed manipulators [10, 11] and contains modules for simulations of the 5-bar and 8-bar manipulators kinematics. It also allows work with kinematic chains with more than one degree of freedom (d.o.f.) which can be used for mechanic structures, for robots and manipulators. One of the widely

used methods for the particular type of calculations is “triangle method” [10] described below.

We use a local coordinate system OXY right with center at the point O and the OX axis passing through point A to minimize computational operations. If we denote the coordinates of point B in the local coordinate system Bx and By , then the following calculations are used:

2.1. Calculation of coordinates of a point, as a function of two points, angle and distance (Algorithm 1).

Calculating coordinates of point B , as a function of points A , O , angle β and distance b .

If we have points A and O in the plane OXY calculating the coordinates of point B is based on the angle AOB and the distance from O to B (Fig. 1(a)).

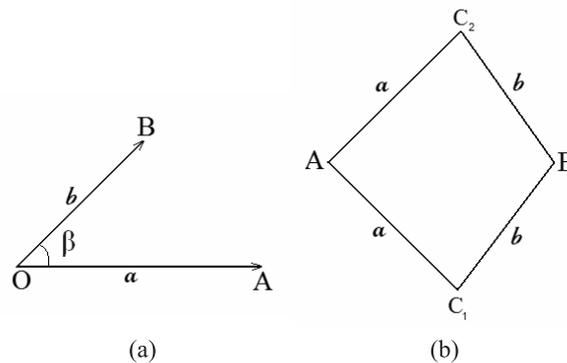


Fig.1. (a) Calculating coordinates of point B , as a function of points A , O , angle β and distance b ; (b). Calculating coordinates of point C , as a function of points A , B , and distances a and b

Let's define $OA = a$, $OB = b$ and angle $AOB = \beta$. In the given frame of reference-plane OXY the coordinates of point B will be found after applying scaling to OA and rotation of point A on angle β . So the result is:

$$(1) \quad \begin{aligned} Bx &= Ox + (\cos \beta(Ax - Ox) + \sin \beta(Ay - Oy))b/a, \\ By &= Oy + (-\sin \beta(Ax - Ox) + \cos \beta(Ay - Oy))b/a. \end{aligned}$$

2. 2. Calculation of coordinates of a point, based on the known distances from two other points (Algorithm 2).

Calculating coordinates of point C , as a function of points A , B and the distances a and b from A and B .

The solution to this problem is based on intersecting two circles – first circle is with centre $A(Ax, Ay)$ and radius a and the second circle is with centre $B(Bx, By)$ and radius b .

For clarity let's have $|Bx - Ax| > |By - Ay|$. Then we find the coordinates Cx and Cy of point C this way:

$$Cx = m + n.Cy,$$

where the parameters m and n are:

$$m = (a.a - b.b + Bx.Bx + By.By - Ax.Ax - Ay.Ay)/(2.(Bx - Ax)),$$

$$n = -(By - Ay)/(Bx - Ax).$$

The quadratic equation has two solutions with values:

$$(2) \quad Cy_{1,2} = (-k \pm \sqrt{D})/(n.n + 1),$$

where $k = (-Ay + n.(m - Ax))$ and $D = k.k - (n.n + 1).(Ay.Ay + (m - Ax).(m - Ax) - a.a)$.

Thus using values $Cy_{1,2}$ from (2) we calculate $Cx_{1,2}$: $Cx_{1,2} = m + n.Cy_{1,2} = (a.a - b.b + Bx.Bx + By.By - Ax.Ax - Ay.Ay)/(2.(Bx - Ax)) - ((By - Ay)/(Bx - Ax)).((-k \pm \sqrt{D})/(n.n + 1))$.

3. Results

The computational program for computer graphical modelling of planar and spatial closed kinematic chains for robots and manipulators was specifically designed to have a user-friendly interface. In order to demonstrate the functionality of the program two types of 5-bars and two types of 8-bars closed manipulators are used for simulation of the kinematics models. The manipulator type and the geometrical parameters for each bar can be set by buttons located in the top left corner of the menu, just before the buttons for operations with the manipulators:

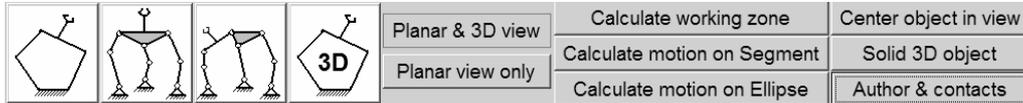


Fig. 2. Menu with buttons for manipulator generation and different operations

The operations with the manipulators are performed just by activating the short-cut menu, connected to each of the program buttons.

3.1. Description of the user's interface

The zone of reachability is divided into two parts: – the left part displays the planar view, while the right part displays the spatial view of the active object (Fig. 3). The buttons functions are set in the program as follows:

– allows definition of manipulator's trajectory of type segment;

– allows definition of manipulator's trajectory of type ellipse;

– The handler's motion over an already defined trajectory (an ellipse or a segment) can be shown dragging the slider,

– is only active when the program is simulating for a 5-bar closed manipulator, which has 2 contiguous segments with apple-like joints. The middle joint will move along a circle in a plane perpendicular to the manipulator's plane moving this slider.

The user can get a real idea of the spatial movement or presentation of the 5-bar's manipulator abilities and zone of reachability by combining the two sliders.

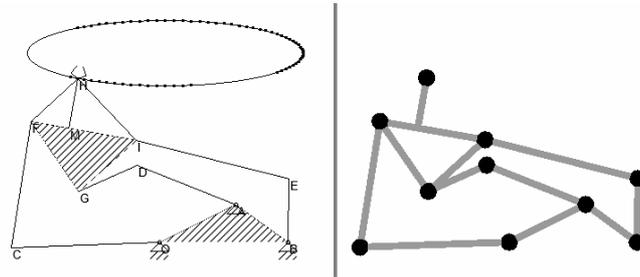


Fig. 3. Planar and spatial views in the centre, showing the handler's movement over an elliptic trajectory

The main functions that the program can perform are the following:

- The interface allows creating any of the 3 types of manipulators, just by choosing the lengths of bars and the handler's tilt. Then the program checks parameters for correction using the following idea: first the mechanism is split into triangles and pentagons and then each of them is checked if the length of each bar is bigger than the sum of the rest. For the 5-bar the polygons are OABCD and BCH, while for the 8-bar they are OAB, FGI, FIH, OADGFC and ABEIG (Fig. 4). The software calculates one sample disposition of the points if all the conditions are met and shows it on the screen for further investigation.

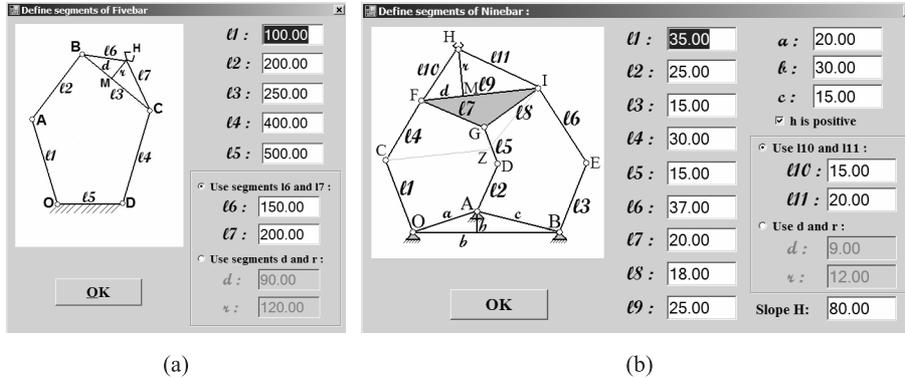


Fig. 4. Creation of (a) 5-bar manipulator; (b) 8-bar manipulator

- After the object has been created its sizes can be modified by activating the short-cut menu, connected to each of the manipulator's segments.
- The user can set an elliptic orbit for the handler, just by choosing the ellipse's top-left and bottom-right corners, or the focuses and radius of the ellipse. The values of these parameters can also be entered in the window, or the ellipse can be drag-and-dropped with the mouse on the screen. The elliptic orbit can be followed point-by-point, by dragging the bottom-left slider through its values. All the achievable points will be displayed on screen with black dots on the ellipse. (Fig. 3). All the manipulator's segments will be redrawn to display the change to the user, while the handler changes its position over the ellipse.
- The handler can move along a segment. The segment can be defined by drag-and-drop its ending points on the screen, or by entering their position on the dialog-window. Then the slider's tracking shows manipulator's behaviour when the gripper's position changes.
- The program can show the manipulator in 3D space – the user can choose between wireframe and solid view of the object using the buttons. He can also rotate the object around the horizontal and vertical axes and centre it on the screen.
- The zone of reachability can be calculated for each manipulator with just one click. The result is represented as a number of points for 5-bar in Fig. 5(a), and for 8-bar in Fig. 5(b).

3.2. Description of mathematical calculations, used in the software product

The following calculations are performed for the implementation of the

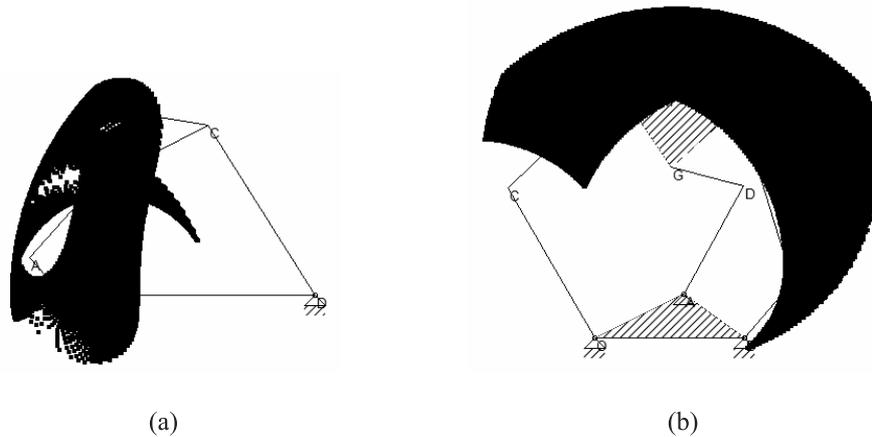


Fig. 5. Calculating the zone of reachability of a) 5-bar manipulator; b) 8-bar manipulator

basic functions of the program:

3.2.1. Calculating an initial frame of a 5-bar (Fig. 4(a))

Let's have a 5-bar whose segments are l_1, l_2, l_3, l_4, l_5 .

Calculating its initial frame starts with choosing the lengths l_{12} (OB) and l_{34} (BD). The calculation is based on the algorithm (2) where the lengths of the other segments, satisfying the triangles inequalities. Finally, finishing all the steps, we'll have the new positions of points A, B, C, D and H . The software uses these points to display the initial frame, just after the user has chosen the segments for testing.

3.2.2. Calculating frames of a 8-bar (Fig. 4(b))

Let's have the 8-bar whose segments are $l_1, l_2, l_3, \dots, l_9$. The algorithm for calculating each of its frames is based on two nested cycles, enumerating the values of two angles. The first angle is BOC (α) and the second is angle OBE (β). We'll show that a couple (α_i, β_j) fully defines one frame of the 8-bar, and one of these frames is used as an initial frame. So, for a couple of two fixed angles (α_i, β_j) we have to make the following calculations:

- We find point $C(Ox + l_1 \cdot \cos \alpha_i, Oy + l_1 \cdot \sin \alpha_i)$;
- We find point $E(Bx + l_3 \cdot \cos \beta_j, By + l_3 \cdot \sin \beta_j)$;
- Calculation of point $A(Ax, Ay)$ using algorithm (2) for points $O(Ox, Oy)$, $B(Bx, By)$ and distances OA and BA .

- Based on the length of the segment $FI = l_9$ and its tilt against the basis OB , we find the vector FI as $(l_9 \cdot \cos(\text{tilt}H - 90), l_9 \cdot \sin(\text{tilt}H - 90))$;

- e) We draw the point Z , so that the figure $CZIF$ is a parallelogram. So, we find $Z(Zx, Zy) = C(Cx, Cy) + FI$ vector;
- f) Calculation of point $I(Ix, Iy)$ using algorithm (2) for points $Z(Zx, Zy)$, $E(Ex, Ey)$ and distances l_4 and l_6 ;
- g) Calculation of point $F(Fx, Fy) = I(Ix, Iy) - FI$ vector;
- h) Calculation of point $G(Gx, Gy)$ using algorithm (2) for points $I(Ix, Iy)$, $F(Fx, Fy)$ and distances l_8 and l_7 ;
- i) Calculation of point $D(Dx, Dy)$ using algorithm (2) for points $G(Gx, Gy)$, $A(Ax, Ay)$ and distances l_5 and l_2 ;
- j) Calculation of point $H(Hx, Hy)$ using algorithm (2) for points $F(Fx, Fy)$, $I(Ix, Iy)$ and distances l_{10} and l_{11} ;
- k) We find point $M(Fx + (Ix - Fx).d/l_9, Fy + (Iy - Fy).d/l_9)$;

We'll know all the points of the 8-bar after calculating all these steps. We won't have a solution for this couple (α_i, β_j) if one of the steps doesn't produce result. If the step finds 2 possible positions for a point, we can skip one of them, comparing the solution with the one found in the previous couple for angles (α, β) .

3.2.3. ShowWorkZone – algorithm for a pentagon (Fig. 5(a))

We use two nested cycles, enumerating the values of two angles. The first angle is $BOC(\alpha_i)$ and the second is angle $OBE(\beta)$. Then, for a couple of two fixed angles (α_i, β_j) we'll calculate the following steps:

- a) We find point $A(Ox + l_1 \cdot \cos(\alpha_i), Oy + l_1 \cdot \sin(\alpha_i))$;
- b) We find point $C(Dx + l_4 \cdot \cos(\beta_j), Dy + l_4 \cdot \sin(\beta_j))$;
- c) Calculation of point $B(Bx, By)$ using algorithm (2) for points $A(Ax, Ay)$, $C(Cx, Cy)$ and distances l_2 and l_3 .
- d) Calculation of point $H(Hx, Hy)$ using algorithm (2) for points $B(Bx, By)$, $C(Cx, Cy)$ and distances l_6 and l_7 .

With this, we have all the points for the pentagon and we can draw its zone of reachability, frame by frame, on the screen.

3.2.4. ShowWorkZone – algorithm for a eight-bar (Fig. 5(b))

We use two nested cycles, enumerating the values of two angles. The first angle is $BOC(\alpha_i)$ and the second is angle $OBE(\beta)$. Then, for a couple of two fixed angles (α_i, β_j) we'll calculate the following steps:

- a) Point $A(Ax, Ay)$ using algorithm (2) for points $O(Ox, Oy)$, $B(Bx, By)$ and distances OA and AB .
- b) We find point $C(Ox + l_1 \cdot \cos(\alpha_i), Oy + l_1 \cdot \sin(\alpha_i))$;
- c) We find point $E(Bx + l_3 \cdot \cos(\beta_j), By + l_3 \cdot \sin(\beta_j))$;
- d) Calculation of point $Z(Zx, Zy) = C(Cx, Cy) + FI$ vector;

- e) Calculation of point $I(Ix, Iy)$ using algorithm (2) for points $Z(Zx, Zy)$, $E(Ex, Ey)$ and distances l_4 and l_6 ;
- f) Calculation of point $F(Fx, Fy) = I(Ix, Iy) - FI$ vector;
- g) Calculation of point $G(Gx, Gy)$ using algorithm (2) for points $I(Ix, Iy)$, $F(Fx, Fy)$ and distances l_8 and l_7 ;
- h) Calculation of point $D(Dx, Dy)$ using algorithm (2) for points $G(Gx, Gy)$, $A(Ax, Ay)$ and distances l_5 and l_2 ;
- i) Calculation of point $H(Hx, Hy)$ using algorithm (2) for points $F(Fx, Fy)$, $I(Ix, Iy)$ and distances l_{10} and l_{11} .

With this, we have all the points for the pentagon calculation and we can draw its zone of reachability, frame by frame, on the screen.

3.3. Module for backlash calculation of the mechanism in 3D-space

One of the important features of the software is its ability to calculate the backlash in the joints of the 5-bar mechanism in 3D-space. This was actually a task that came from the practice and we proposed a solution for it in our program.

The calculations method includes one more argument – an angle of backlash for each of the bars (in degrees). Then the program performs a cycle through all the possible rotations for each bar and tries to join it with the other bars. The result at the end is a cloud-like area, similar to that one in Fig. 6. The calculations for defining this cloud area grow exponentially when the angle of backlash expands.

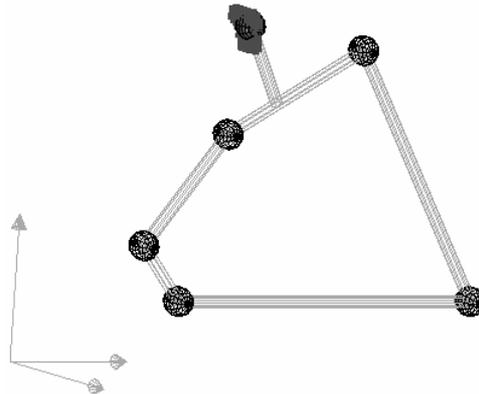


Fig. 6. Representation of a 3° backlash for the spherical joints in a 5-bar in 3D

3.4. Module for calculation and visualisation of joints velocity for a given mechanism

The application is very useful, when the engineer wants to calculate the velocity of the joints for a given mechanism. This feature becomes available after the specification of trajectory of the mechanism, on which the handler moves with constant speed. The calculations for line, ellipse and the new Bezier spline trajectories are similar.

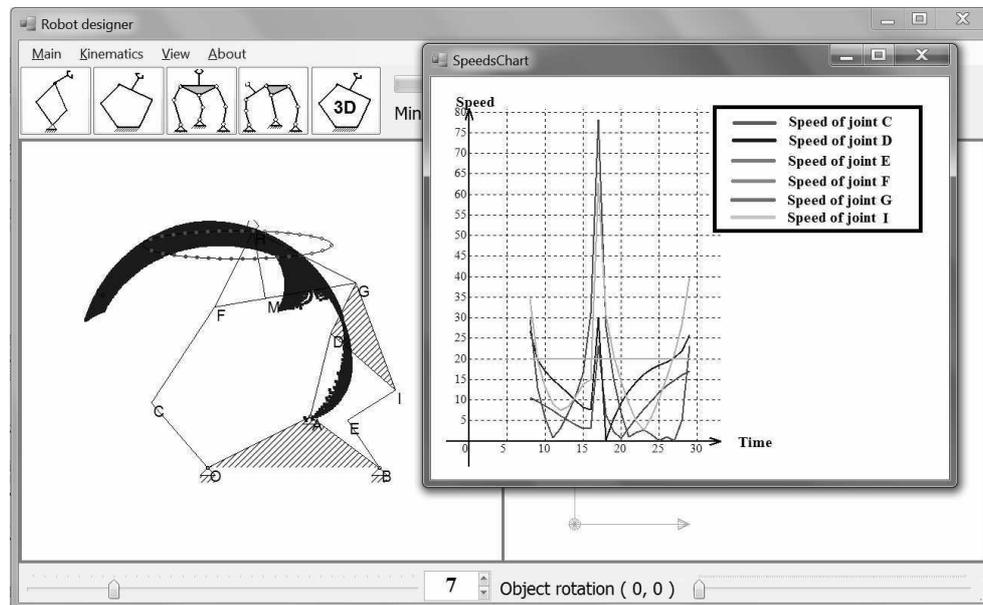


Fig. 7. Representation of the speeds of the joints for 8-bar mechanism

For example, the speeds of the joints for 8-bar mechanism are shown in Fig. 7 with different colour. It is to note two important facts, visible in this figure:

- The inaccessible points in the trajectory are always skipped in the chart.
- The step of the handle over the trajectory can be also set - bigger steps will result in more sparse points on the screen. Thus, the length of the OX axis of the chart dynamically changes while changing the step of the handle on the trajectory. As a result the lines become smoother but the calculations are increased.

4. Conclusion

The currently developed programming tool can be of a great help to any student or engineer, eager to create a new manipulator, or just to investigate the possibilities that the world of robots offers. Any of the proposed manipulators can be created just by changing the input parameters, the main kinematics tasks solved and the manipulator can be displayed in both 2D and 3D views, easily rotatable and movable on the scene. Furthermore, the program can be used to solve tasks from the real practice and can be used successfully while teaching students or developing new structures for Robots and mechatronic systems.

In addition, the programming package RobotDesigner is freely available for download from internet, user-friendly and sufficient to create and modify manipulators and inspect its functionality. In the near future, this software should calculate the dynamics of the mechanisms, to export it into some well-known file format in order to be later used in different engineering software and convert the results into a plugin or extension for other packages.

REFERENCES

- [1] ZLAJPAH, L. Simulation in Robotics, Jozef Stefan Institute, 2008.
- [2] Symbolic Dynamics Inc., SD/FAST User's Manual, 1994, <http://www.sdfast.com/>.
- [3] Open Dynamics Engine, <http://ode.org/ode.html>.
- [4] Cyberbotics Ltd. Webots User Guide, 2005, <http://www.cyberbotics.com/>.
- [5] MICHEL, O., A. WEBOTS. Powerful Simulator for the Khepera Robot, <http://www.ccnacht.de/ccnacht2/media/pdf/kheperawebots.pdf>.
- [6] STORM, T. KiKS is a Khepera Simulator, User Guide, <http://www.tstorm.se/projects/kiks/>.
- [7] CORKE, P. I. A Robotics Toolbox for MATLAB. *IEEE Robot. Autom. Mag.* **3** (1996), No. 1, 24–32.
- [8] VENEVA, I. Device for Control of Active Ankle-Foot Orthosis and Monitoring System for Gait Analysis. *Journal of Theoretical and Applied Mechanics*, **40** (2010), No. 4, 81–92, ISSN: 0861-6663.
- [9] VENEVA, I. Intelligent Device for Control of Active Ankle-foot Orthosis, Proceedings of the 7th IASTED International Conference on Biomedical Engineering “BioMed 2010”, 17–19 February, ISBN: Book: 978-0-88986-827-4, CD: 978-0-88986-828-1, Austria, Innsbruck, 2010, 100–105.

- [10] KONSTANTINOV, M., Y. MARINOV, C. GANOVSKI, I. AVRAMOV, C. ZAMANOV. Application of Robots in Industry, Technique, 1980 (in Bulgarian).
- [11] RAYKOV, P. Planar Robots and Manipulators, Ph.D. Thesis, 1986 (in Bulgarian).