

Variable Robot Geometry Optimization Method to Avoid Tip Over Situations During Slow Motion on Unknown Terrains

Ferenc TAJTI,¹ Bence KOVÁCS,² Géza SZAYER,³ Péter KORONDI,⁴
Zoltán SZÉKELY⁵

This paper presents a parametrized stability control method for special slow motion field mobile robots, based on use cases from border surveillance. The concept uses the centre of gravity (COG) as the virtual centre of motion (VCM). The simplified robot geometry is an input parameter of the model, so it can work with different types of mobile robots, like holonomic-wheeled, differential-wheeled, steered, tracked, wheeled-tracked, segmented, etc. structures. This method resulted in the implementation of a flexible and universal control algorithm for transformable and hybrid drive mobile robots, where every parameter can be changed and recalculated for different applications or even in discrete time steps during the motion at a 3D path. The velocity reference, the angular velocity reference and the optimization parameter (for example gravity compensation) of the robot can be prescribed. The model was implemented in MATLAB and can be compiled to C for measurements and validation with test robots.

Keywords: mobile robot, hybrid drive, transformer, variable geometry, police robots

Introduction

In most of the cases concerning mobile robot control theory the only implemented algorithm is robot kinematics. These algorithms do not take into consideration the inertias and the masses of the robot (robot dynamics). This method is parallel with industrial robot control methods. At linearly independent joints (for example an XYZ Cartesian TTT organized 3 DOF milling machine) the usual control method is the implementation of inverse kinematics (decentralized control method). [1] [2] With decentralized control method the masses and inertias of the segments do not have significant effects on each other (they have effects just on the position and the orientation of TCP). (They are linearly independent.) In the case of decentralized control theory, the system can be controlled without dynamical equations perfectly. With the centralized control method, we take into consideration the dynamical equations of motion, the interaction between the links, masses, and inertias. [3] [4]

-
- 1 MSc, PhD candidate, Budapest University of Technology and Economics; e-mail: tajti@mogi.bme.hu
 - 2 MSc, PhD candidate, Budapest University of Technology and Economics; e-mail: bence.kovacs@mogi.bme.hu
 - 3 MSc, PhD candidate, Budapest University of Technology and Economics; e-mail: geza.szayer@mogi.bme.hu
 - 4 DSc, Head of Department, Budapest University of Technology and Economics, TAMOP-4.2.2.B-10/1-2010-0009; e-mail: korondi@mogi.bme.hu
 - 5 JD, Major of the Hungarian National Police, Assistant Lecturer, National University of Public Service; e-mail: szekely.zoltan@uni-nke.hu

Mobile robots have usually only one link, and the robot cannot be divided into different parts for centralized motion control method, so the whole mobile robot can be considered as a point (the centre of gravity, virtual centre of motion). The velocity and the angular velocity of the centre of gravity define the motion of the robot (at constant speed). Along the path of the motion we prescribe the velocity and the angular velocity of the robot. With the equations of the robot kinematics we can calculate the angular velocities of the wheels from the equations of motion of the centre of gravity. Parts of transformable mobile robots can be considered different links. These links usually have negligible dynamic effects on each other (because of the mass, inertia and velocity ratios between the whole system and one link). The geometrical transformation has many more important effects on the robot, like turning radius, length, width, height of the structure, stability, and etc. These parameters modify the whole robot kinematics. We can categorize the mobile robots for groups in the view of control theory. (For example: holonomic type mobile robots, non-holonomic type, differential type, non-differential type, etc.) In each group the control methods are very similar. Transformable mobile robots, the basic control method can be more difficult than having extra equations, additional parameters, or special equations of motion.

Use Case

Use cases can serve as a good platform to assume the challenges and needs as well as to present the necessity of discussed solutions. In this article the use case is border surveillance. Border surveillance is defined by the Schengen Border Code. Besides protecting the external borders of the Schengen Area, an area almost overlapping with the territory of the European Union, it also has a traditional key role in protection of the national security interests of the country, including fight against terrorism. [5] Border surveillance means “the surveillance of borders between border crossing points and the surveillance of border crossing points outside the fixed opening hours, in order to prevent persons from circumventing border checks.” [6: 16]

This means border surveillance has three key aspects:

- solutions to supervise border sections between border crossing points (at land [including rivers and lakes] as well as at sea);
- solutions to supervise border crossing points (border gates) outside opening hours (when they are usually unmanned);
- movement control capabilities in order to prevent persons from circumventing border checks.

In the last year, a vast number of persons attempted to cross the Schengen external border into Hungary illegally. There were 10,046 persons arrested on the peak day in 2015, 430,607 persons were caught crossing the border illegally in total in 2015. For comparison, the total number in 2014, was 61,664 (according to statistical reports of the Hungarian National Police – published on [40].) Detection itself did not help authorities who faced a huge challenge only because of the sheer numbers. Moreover, if the police stopped them, they attacked the border gates (Röszke, September 16, 2015), but were driven off with tear gas and water cannons. First in realizing the nature of the current migration trend, meaning that the masses of migrants may start a riot at any time, Hungary erected a border fence along its southern border. Additional large waves are expected later in 2016 and some signs show that attempts

are becoming more and more aggressive (on February 19, 2016, an individual pointed a gun at the patrols, threatening them with shooting, on the same day a car broke through and tried to run down the officers trying to stop it). Every day, smaller or larger groups try to cut their way through the border fence. This means, that the border surveillance has to be able to cope with such mass breakthrough attempts. To prevent injury to officers and spare manpower for critical events, the Hungarian National Police launched a series of research and innovation actions, aimed at developing autonomous border surveillance capabilities. One of the possible solutions is the authorization of ground robot patrols in the form of slow moving ground robots equipped with sensor arrays, radio and extended power sources with hot-swap function. Some sensors shall be able to capture multi-modal biometric identifiers (e.g. motion picture with movement patterns in normal and infrared [IR] light), enabling later identification of trespassers. [7]

Infrastructure close to the border fence is slightly underdeveloped, as it had to be erected hastily in one month, especially manoeuvre roads are in bad condition on rainy days, tents and mobile toilets were hastily deployed to cover the patrol lines along the border fence. Currently, power lines are under construction at several border sections which can serve as a solid base infrastructure for ground surveillance robots covering areas which cannot be covered by power lines and static surveillance equipment. Therefore, control methods for ground robots applicable under such conditions had to be investigated.



Picture 1. Manoeuvre road after a rainy day.

[Courtesy of the Hungarian National Police. Made exclusively for the authors by Martin MEYER.]



Picture 2. Temporary infrastructure along patrol lines.

[Courtesy of the Hungarian National Police. Made exclusively for the authors by Martin MEYER.]

Background

Several types of mobile robots exist, such as tracked, wheeled, legged, wheeled-legged, leg-wheeled, segmented, climbing or hopping. Transformable mobile robots proved that they can be much more effective on terrain, than the single link ones, and can overcome obstacles 2–3 times their wheel diameter. [8] [9] [10] The control methods of these robots are implemented individually for every construction. In most of the cases the control algorithms uses the virtual centre of motion method and the path of the motion describes the path of this point. [8] [11] Usually the centre of gravity is not exactly the same with the virtual centre of motion. Mostly the virtual centre of motion is the geometrical centre of the structure. [8] At the design of the robot mechanics the geometry of the robot can be optimized for the expected terrain. [8] [9] [11] In some cases the control method is also optimized for the surface of the ground. [8] [9] The design and control of a robot for an unexplored terrain requires a different approach. [13] [14] Transformable mobile robots are designed for motion in special environments. With the variable geometry, the structure can optimize its shape to overcome the current obstacle. For example, after the attack of the World Trade Center these types of mobile robots were used in searching for victims. These machines can work in hazardous environments without any risk to human life. [11] [15] The robots can be smaller than dogs and go deeper in small tunnels. The most significant barriers are stairs, stones, and pits. [14] The key factor of this movement is the stability of the robot. In case of an impossible path a new route can be designed, but in case of a tip over situation the mission is unsuccessful. The main parameter of the stability is the effect of gravity. (In case of a relatively slow motion robot the dynamics can be neglected.) Variable transformation shapes give opportunity for the compensation of the gravity effect. [16] The low-level control algorithms are implemented in each structure individually. [17–26] For the high-level robot action control we can find general structures, and path controller algorithms. [22] [25] [27] The path of the robot motion is also different for example at a differential drive and a holonomic drive. [25] The low-level control algorithm (kinematics) of a transformable and a non-transformable differential drive is also different. At the transformable version of the differential drive the shape of the structure is continuously under optimization during motion. [27] [28] [29] [30] [31] The aim of the paper is to describe a parameterized low level mobile robot control algorithm for slow motion on unknown terrain. [32] [33]

Model Description

Simplifying the robot geometry, we can find a way to define the required parameters for the calculation of inverse kinematics generally. Consider the mass of the robot as a point (centre of gravity). This point is the centre of the robot coordinate system. Draw vectors from the centre of the coordinate system to the contact points between the floor and the wheels. The model can calculate with constant and variable coordinates also. For example, from a holonomic type robot model (Figure 1) we can sketch a simplified structure (Figure 2). Of course, the geometrical transformation can change the centre of gravity relatively to the wheels. In this case, all of the structure can be recalculated but this effect is usually negligible. (With future development a segmented robot with more concentrated masses will be defined. [34] [35]) A contact point can be geometrical and constrains perpendicular to the

plane of the floor (contact plane), and at a direction in parallel with the contact plane. These parameters also can be variables or constants. (For example, in radio control [RC] cars the steering angle, modifies the direction of the geometrical constrains at the contact points of the steered wheels.) Additional input parameters are the mass of the robot, the gravitational vector and the coefficient of adhesion friction. With this parameter description, we can calculate the normal forces at the contact points and the components of the acceleration (mainly the mass acceleration). This will result in a simple statically model that can be optimized in real-time during motion.

The path planning algorithm defines the velocity and the angular velocity of the COG(s) or centre of a multi robot system. [36] With these data and the parameters mentioned above we can calculate the angular velocities of the wheels (inverse kinematics). With the definition of the acceleration and the angular acceleration of the robot we can calculate the angular velocities of the wheels at discrete time steps. During the control of transformable mobile robots, we can prescribe optimums, maximal, or minimal values of robot parameters, like maximal acceleration utilization, maximal velocity utilization, stability optimum, energy consumption minimum, etc. (Figure 3) This way we can also make sure the robot is going to move slowly in case of a critical tip-over situation to avoid robot damage caused by high motional energy (and we can also neglect dynamic effects of robot motion).

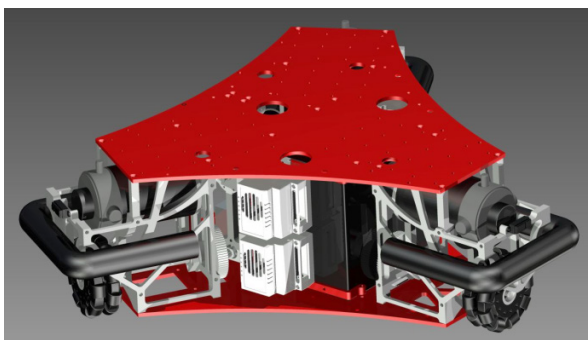


Figure 1. The rendered image of the holonomic robot base.
[Edited by the authors.]

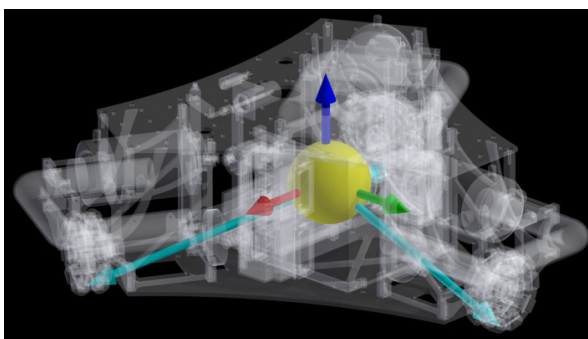


Figure 2. The rendered image of the holonomic robot base with the robot coordinate system, the centre of gravity, and the vectors of the contact points.
[Edited by the authors.]

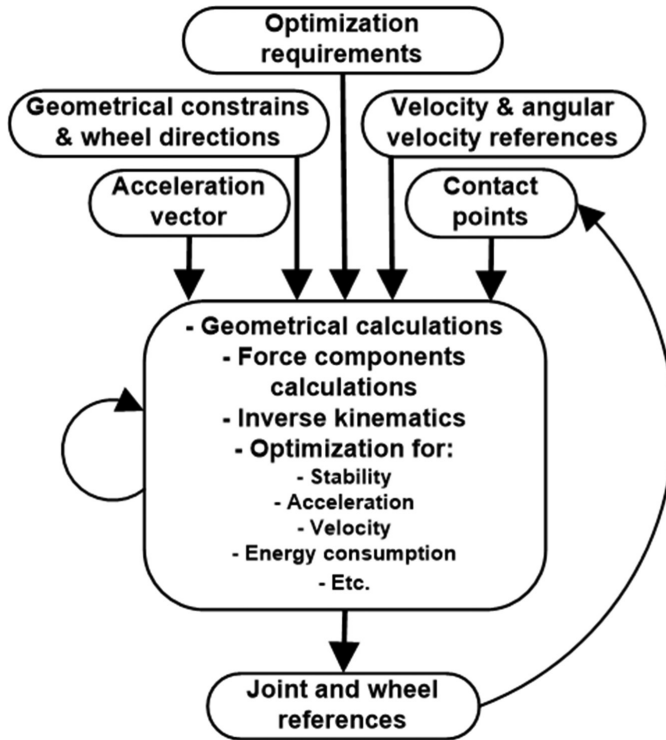


Figure 3. The block diagram of the system.
[Edited by the authors.]

Calculation Method

The implemented model can be divided into two main parts: basic calculations and optimizations. These can be also divided into different loops and iterations. At the end of the development the code must run on an embedded system or on an embedded computer. [37] The calculation throughput these hardware elements is limited, so the code must be speed optimized. [39] The program details have different priorities. The inverse kinematics has to run on the highest frequency in discrete time. The tangent plane (orientation of ground relatively to the robot) calculations, the normal forces and gravitational force calculations, the geometry transformation, or the stability optimization can run on different priorities and frequencies. (For example, in case of a constant acceleration vector the recalculation of the tangent plane function is unnecessary. It means the robot is moving on the same oriented flat terrain for a long time.) The time constants of the application define these priorities, frequencies, and the minimal calculation throughput.

Basic calculations

This part of the code requires the input parameters. These parameters can be given in the same frame for different robots. The values of the frames are constants, but the constants can

be calculated from any type of MATLAB compatible functions. (For example, the inverse kinematics of multi DOF legs or arms. [38] [39]) The required data of the robot are the followings in robot coordinate system:

- vector of the acceleration (provided by a 3-axis accelerometer);
- coordinates of the contact points (for example the output of the robot leg's direct geometry functions);
- direction of the geometrical constrains and the drives (direction of the steered wheels);
- mass of the robot;
- value of the adhesion friction;
- transformation matrix between the robot coordinate system and the world coordinate system (provided by navigation module);
- velocity and angular velocity references of the robot (provided by path planner);
- values of the acceleration and the angular acceleration of the robot (provided by path planner).

Every parameter is in robot coordinate system (except the references). (The references are usually given in world coordinate system and transferred to robot coordinates.) The program generates the simplified 3D model of the robot from the input parameters. It contains the acceleration vector, the components of the acceleration vector, the concentrated mass, the vector of the contact points, the tangent plane and the normal forces. (Figure 4) The method is implemented in 3D for contact points (e.g. wheels), location and positive integer.

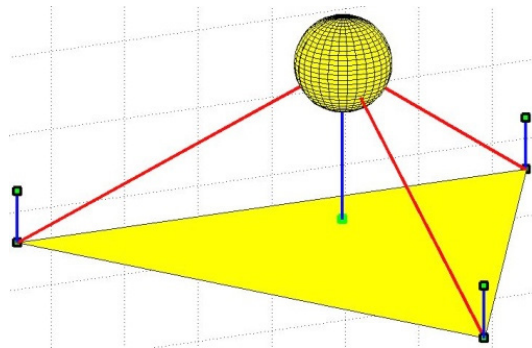


Figure 4. The 3D MATLAB model of the simplified robot (same as Figure 1) geometry. (In this case the acceleration has only perpendicular component [only gravity effect].)
[Edited by the authors.]

As it was mentioned above in most of the cases the acceleration of the robot from the drives is negligible in the ratio of the gravity effect. The input parameters are dimensionless ratios. (Of course, these parameters can be easily changed to real dimensions.) The first part of the code generates the 3D model, the tangent plane, and the force components. The output of this process is a 3-dimensional free body diagram. The program calculates the distances between the gravitational vector and the contact points. Every distance is projected to the XY plane. With these distances, we can define a first iteration ratio for the normal forces at the contact points. (Normal forces are between the wheels and the ground.) The program selects three points to find the ground plane in robot coordinate system. Three points define a 3D plane.

Equation (1), where x,y,z are the coordinates of a general point on the plane, x_0,y_0,z_0 are the coordinates of a wheel contact point, and a,b,c are the coordinates of the normal vector of the plane (can be expressed from three points of the plane).

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \tag{1}$$

These three points can be selected from any of the points with different combinations. The program calculates the maximal number of these combinations, where k_{max} in Equation (2) is the maximal number of the combination (and the number of the iterations), and n is the number of the contact points.

$$k_{max} = \frac{n(n - 1)(n - 2)}{6} \tag{2}$$

For the test of the first iteration, (the validation of the plane) we have to calculate, that the other points are lies on the plane, over the plane, or under the plane from the Equation (1). (I.e. point error.) This step also helps us in case of multiple wheel robots to figure out which 3 wheels are defining a plane. In case the gravitational vector goes through the plane the robot is standing on those three wheels and we can get estimation about the rest of the wheels. (We want to see if those are touching the ground, a stone, being on the first step of a stair, etc.)

The next part of the plane validation is the investigation of the relationship between the plane and the continuance of the gravitational vector. The continuance of the vector (3D line) should intersect the plane inside the triangle defined by the three main contact points. A rigid body rolls over when the impact line of the gravity is outside of the contact surface. (I.e. intersection error.) The coordinates of this point will satisfy the equation of the 3D plane [Equation (1)] and the equation of the 3D line [Equation (3)] at the same time, where r is the vector of an arbitrary point of the line (in this case the intersection point), r_0 is the vector of a known point of the line (in this case the gravitational vector), t is a real parameter ($t \in R$) and v is the direction vector of the line.

$$r - r_0 = tv \tag{3}$$

Equation (3) can be expressed in coordinate form like Equations (4)–(6), where d_{gx}, d_{gy}, d_{gz} are the coordinates of the intersection point, $(D_g), g_x, g_y, g_z$ are the coordinates of the gravitational vector, t_1 [Equation (7)] is a real parameter ($t_1 \in R$) and x_1, y_1, z_1 are the coordinates of the main points from the plane (so as x_2, y_2, z_2 and x_3, y_3, z_3).

$$d_{gx} = g_x(1 - t_1) \tag{4}$$

$$d_{gy} = g_y(1 - t_1) \tag{5}$$

$$d_{gz} = g_z(1 - t_1) \tag{6}$$

$$t_1 = - \frac{a(x_1 - g_x) + b(y_1 - g_y) + c(z_1 - g_z)}{ag_x + bg_y + cg_z} \tag{7}$$

In case of a calculation error related to the coordinates of the wheel (x_1, y_1, z_1) the program creates a new combination with a different wheel. In case of a calculation problem with the intersection point (D_g) the program also creates a new combination of wheels to find the three wheels where the robot lies on the ground. The iteration runs maximally once with every combination (n times). These attempts are running in probability sequences. At the first successful case the next step will be the calculation of the gravitational components (force perpendicular to the ground plane and force parallel to the ground plane) and the normal forces (forces between the wheels and the ground plane). The normal component of the gravitational (and the normal) forces will be parallel with the normal vector of the plane. The point where the normal component intersects the plane can be calculated with the same method, where d_{cx}, d_{cy}, d_{cz} are the coordinates of the intersection point, (D_g) , t_2 is a real parameter ($t_2 \in R$), defined by the normal component of the gravitational force. [See Equations (8)–(11).]

$$d_{cx} = a(1 - t_2) \tag{8}$$

$$d_{cy} = b(1 - t_2) \tag{9}$$

$$d_{cz} = c(1 - t_2) \tag{10}$$

$$t_2 = - \frac{a(x_1 - a) + b(y_1 - b) + c(z_1 - c)}{a^2 + b^2 + c^2} \tag{11}$$

The parallel component of the gravitational vector will be parallel with a direction vector calculated from the $D_c D_g$ vector. The sum of the normal vectors at the contact points results the normal component of the gravity. From the distances between intersection point of the gravitational vector and the contact points we can define the ratios for the calculation of the normal vectors in Equation (12), where L is the ratio number, n_a is the number of the points on the plane (some of the wheels can be over the plane), N_g is the normal component of the gravitational vector and $|F_{ni}|$ is the absolute value of normal vector of the i^{th} contact point.

$$|F_{ni}| = - \frac{|N_g|}{\sum_{j=1}^{n_a} L_j} \times \left(\frac{\sum_{j=1}^{n_a} L_j}{\frac{n_a}{2}} - L_i \right) \tag{12}$$

As further development, the inverse kinematics module is under implementation. The program categorizes the structure:

- 3 DoF mobile robots:
 - steered,
 - non-steered.
- 2 DoF mobile robots:
 - steered,
 - non-steered.

A mobile robot usually has two or three DOF on the ground plane. The wheels of the robot can be steered or non-steered.

Optimizations

The firstly implemented optimization of the model is the 3D gravity compensation method. At transformable mobile robots, the geometry can be optimized for the stability of the robot.

A rigid body rolls over when the impact line of the gravitational acceleration is outside of the contact surface. The static torque of the body calculated to the centre of the coordinate system (S_{MO}) can be expressed as Equation (13), where m is the mass and r is the vector of the elementary points of the mass.

$$S_{MO} = \int_m r \, dm \tag{13}$$

The vector of the centre of the body (r_s) can be expressed from Equation (4.13) as Equation (14).

$$r_s = \frac{1}{m} S_{MO} \tag{14}$$

The vector between the centre of the mass and the elementary points (R) can be expressed as Equation (15).

$$R = r - r_s \tag{15}$$

The static torque of the body calculated to the centre of the rigid body (S_{MS}) can be expressed as Equation (16).

$$S_{MS} = \int_m R \, dm = S_{MO} - mr = 0 \tag{16}$$

The connection between the torque and the static torque can be expressed as Equation (17), where M_A is the torque, calculated to the point A and g is the gravitational vector.

$$M_A = S_{MO} \times g \tag{17}$$

From Equations (16), (17) Equation (18) can be expressed. (Torque calculated to the centre of the polygon is zero.)

$$M_S = S_{MS} \times g = 0 \tag{18}$$

The torque of the force in the point P (F_P) at point B (M_B) can be expressed as Equation (19), where r_{BP} is the vector between point P and B .

$$M_B = r_{BP} \times F_P \tag{19}$$

If the distance between the intersection point and centre of the polygon (r_{SDg}) is zero [Equation (20)], the continuance of the g intersects the plane at the center of the polygon. This case results normal distribution of the robot mass on each of the wheels.

$$r_{SDg} = 0 \tag{20}$$

In this case the torque of the gravitational force is zero, so the stability of the robot is maximal. (See Equation 21.)

$$M_{Dg} = r_{SDg} \times (mg) = 0 \tag{21}$$

The other key factor of the robot stability is the height of the structure.

The wheels of the robot define a polygon on the ground plane. (Figure 5) The centre of the polygon is the average of all points of the polygon. The polygon can be divided to triangles. The centre of the polygon can be calculated from the centres of the triangles. The centre of a triangle projected to the XY plane can be calculated from the coordinates with the Equations (22) and (23), where s_{1x}, s_{1y} are the coordinates of the center of the first triangle.

$$s_{1x} = \frac{x_1 + x_2 + x_3}{3} \tag{22}$$

$$s_{1y} = \frac{y_1 + y_2 + y_3}{3} \tag{23}$$

If we project all of the coordinates to the XY plane we can calculate the area of the triangles in 2D with Equation (24), where T_1 is the area of the first triangle.

$$T_1 = \frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)| \tag{24}$$

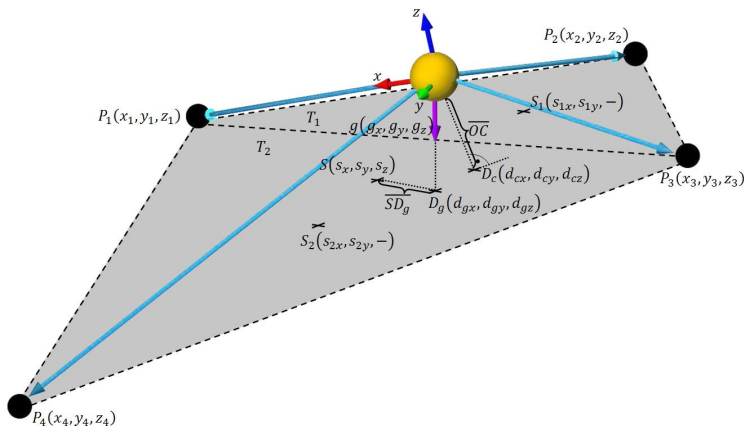


Figure 5. The 3D model of the stability optimization method.
[Edited by the authors.]

From the centres and the areas of the triangles the centre of the polygon can be calculated in 2D with area weighted average with Equations (25) and (26), where s_x, s_y are the x and y coordinates of the centre of the polygon.

$$s_x = \frac{\sum_{i=1}^n T_i S_{ix}}{\sum_{i=1}^n T_i} \tag{25}$$

$$s_y = \frac{\sum_{i=1}^n T_i S_{iy}}{\sum_{i=1}^n T_i} \tag{26}$$

The z coordinate of the point can be expressed from the plane Equation (1) as Equation (27) form, where s_z is the z coordinate of the center of the polygon.

$$s_z = ax_1 + by_1 + cz_1 - \frac{as_x + bs_y}{c} \tag{27}$$

The distance between the centre of the coordinate system and the ground plane (the height of the robot structure) can be expressed as Equation (28), where $|OC|$ is the height.

$$|OC| = \sqrt{d_{cx}^2 + d_{cy}^2 + d_{cz}^2} \tag{28}$$

The distance between the centre of the polygon and the point, where the continuance of the gravitational vector intersects the plane (i.e. stability factor) can be expressed as Equation (29), where $|SD_g|$ is the “stability factor”.

$$|SD_g| = \sqrt{(d_{gx} - s_x)^2 + (d_{gy} - s_y)^2 + (d_{gz} - s_z)^2} \tag{29}$$

Implementation of the control method

The block diagram of the control method can be seen on Figure 6. The blue boxes of the diagram are coded and simulated. These are the basic calculations, the stability optimization and an additional transformation matrix for the simulation of the gravitational vector during the optimization. The grey boxes are the inverse kinematics, the kinematics optimization and final reference control box. The method can be extended with these further optimizations in the future.

The input variables of the block diagram are the followings:

- $\max |SD_g|, |OC|$ are the optimization requirements for the stability;
- $\min \dot{s}, \ddot{s}, \dot{\theta}, \ddot{\theta}$ are the optimization requirements for the velocity (\dot{s}), angular velocity ($\dot{\theta}$), acceleration (\ddot{s}) and the angular acceleration ($\ddot{\theta}$);
- $P_{i=1\dots n}$ are the functions of the contact points (depends on different parameters like $\alpha_p, \gamma_p, \vartheta_p, \dots$);
- $f_{i=1\dots n}$ are the geometrical constrains of the contact points;

- S_o, θ_o are the current position and orientation of the robot;
- m, μ are the mass and the adhesion friction (the centre of gravity $M(0,0,0)$ is in the center of the robot coordinate system);
- $g(x,y,z)$ is the gravitational vector;
- $s, \dot{s}, \theta, \dot{\theta}$ are the position (s), velocity (\dot{s}), angular (θ) and angular velocity references ($\dot{\theta}$).

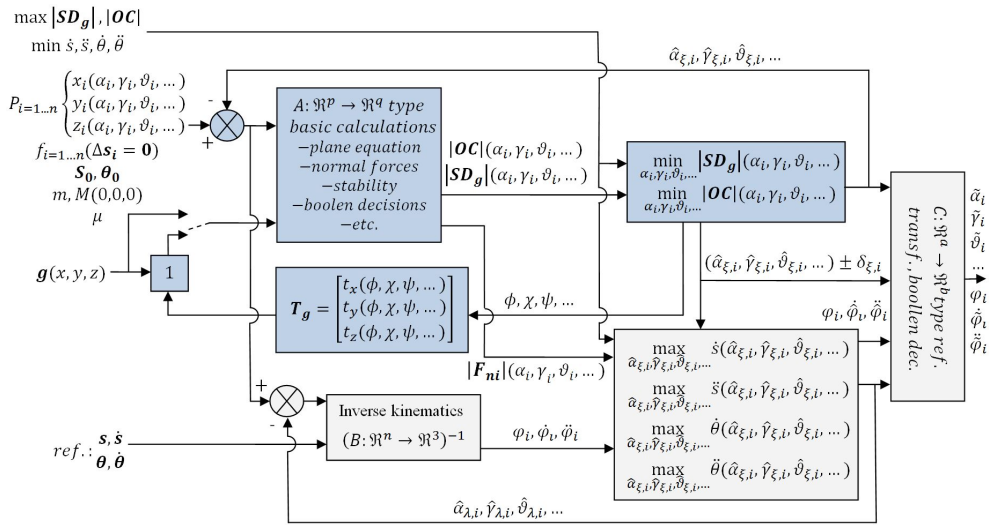


Figure 6. The block diagram of the control method.
[Edited by the authors.]

The variables in the block diagram are the followings:

- T_g is the transformation matrix of the gravitational vector (when the stability factor is under optimization the change of the gravitational vector must be also simulated);
- ϕ, χ, ψ, \dots are the variables of T_g ;
- $\hat{\alpha}_{\xi,i}, \hat{\gamma}_{\xi,i}, \hat{\vartheta}_{\xi,i}, \dots$ are the estimated parameters of the contact points (estimated by the stability optimization, indicated with ξ);
- $\pm \delta_{\xi,i}$ is the range of the estimated parameters, where the parameters are acceptable;
- $\hat{\alpha}_{\lambda,i}, \hat{\gamma}_{\lambda,i}, \hat{\vartheta}_{\lambda,i}, \dots$ are the estimated parameters of the contact points (estimated by the kinematics optimization, indicated with λ);
- $\varphi_i, \dot{\varphi}_i, \ddot{\varphi}_i$ are the angular position, the angular velocity and the angular acceleration references of the wheels;
- $\varphi_i, \hat{\varphi}_i, \check{\varphi}_i$ are the angular position, the estimated angular velocity and the estimated angular acceleration references of the wheels (mostly these estimated values are the same with the original values).

The output variables of the block diagram are the followings:

- $\tilde{\alpha}_i, \tilde{\gamma}_i, \tilde{\vartheta}_i, \dots$ are the final estimated values of the contact point parameters (contact point references);
- $\varphi_i, \hat{\varphi}_i, \check{\varphi}_i$ are the final angular position, the estimated angular velocity and the estimated angular acceleration references of the wheels (mostly these estimated values are the same with the original values).

Simulation results

The first simulation of the results is the MATLAB simulation of a basic three wheeled transformable robot geometry. (Figure 7) The position of the P_1 point is variable with the change of α parameter. (Can be tuned between α_A to α_B .) The distances between the wheels and the centre of gravity are equal at every wheel. Between the first two wheels and the XY plane there is 30.964° . In case of $\alpha = 30.964^\circ$ ($\alpha_{opt.}$) and with a horizontal plane $|SD_g| = 0$.

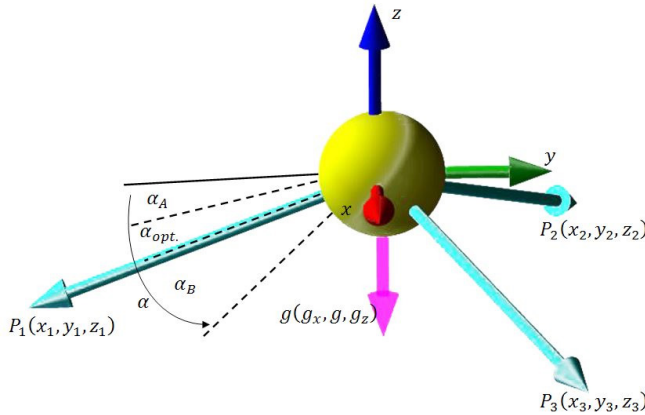


Figure 7. 3D model of the simulated robot structure. [Edited by the authors.]

Assuming a horizontal plane the blue function on (Figure 8) is $|OC|(\alpha)$, and the red one is $|SD_g|(\alpha)$. As we can see on $|OC|(\alpha)$ the increase of α increases the height of the robot. The red $|SD_g|(\alpha)$ function is more important. It has a minimum value at $\alpha \approx 31^\circ$. It means that simulation could find the value of α to get back the original optimal structure of the robot.

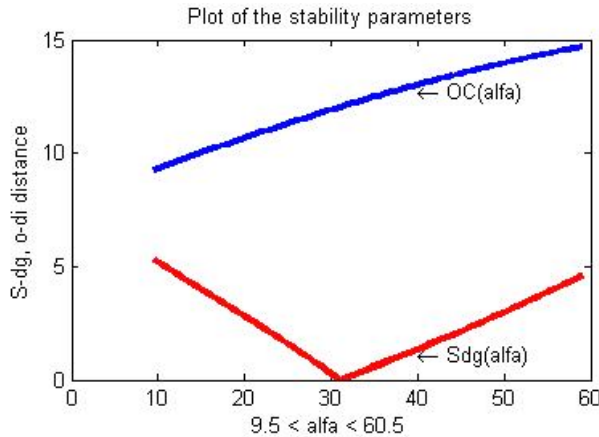


Figure 8. The result of the simulation. [Edited by the authors.]

The implementation generates different $|SD_g|(\alpha, \gamma, \vartheta_p, \dots)$ functions. During the function generation, the program makes logical decisions so the $|SD_g|(\alpha, \gamma, \vartheta_p, \dots)$ function cannot contain symbolic variables.

The second simulation made with the same geometry, but in this case two of the parameters were variables (α, γ) . (Figure 9) In case of $\alpha = \gamma = 30.964^\circ$ ($\alpha_{opt}, \gamma_{opt}$) and a horizontal plane $|SD_g| = 0$ (the maximum of the stability).

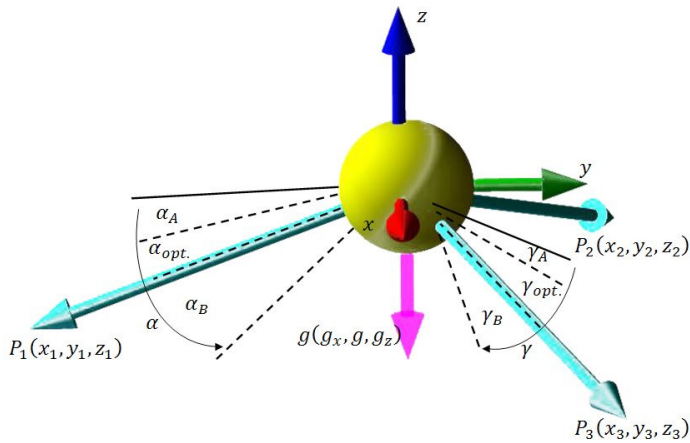


Figure 9. 3D model of the second simulation.
[Edited by the authors.]

In this case three methods were programmed and tested for the calculation of the optimal α and γ values. For comparison: at the worst case in case of $|SD_g|(\alpha, \gamma) > 10$ the robot tips over. Of course, on a ramp less stability can be reached, than in case of a flat and horizontal ground.

At the first method, the program calculates the $|SD_g|(\alpha, \gamma)$ values at 2,500 times. (50 times α and (multiplied by) 50 times γ) As we can see on (Figure 8) and (Figure 9) the $|SD_g|(\alpha, \gamma)$ function has a minimal value at $\alpha = \gamma = 31^\circ$. In this case $|SD_g|(\alpha, \gamma) = 0.0022$. This value is only the 0.022% of the tip over situation. This method could provide the expected results around the original optimal α, γ values. The disadvantage of this method is that the calculation of these points at a normal PC is more than an hour, because of the 2,500 iteration steps. As an advantage of the method it did not use symbolic variables in the $|SD_g|(\alpha, \gamma)$ function and we can visualize the whole “stability map” for different gravitational vectors. At an embedded system, a look up table can contain the values of these maps, although it requires an additional flash memory to store the data points.

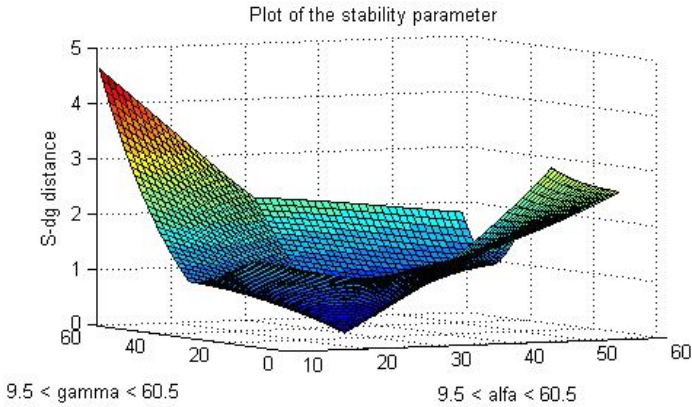


Figure 10. The result of the second simulation (stability map) in 3D. [Edited by the authors.]

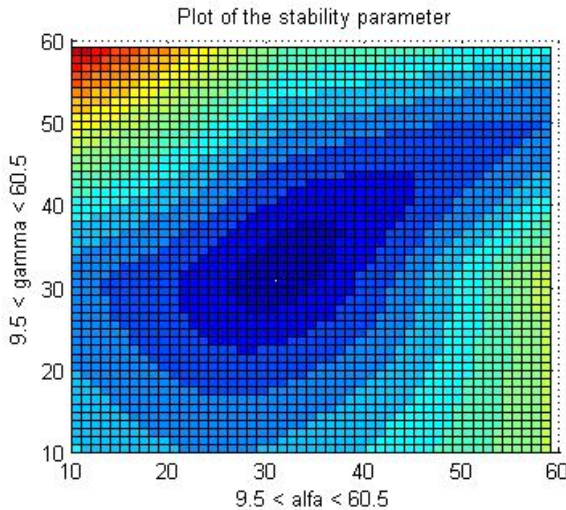


Figure 11. The result of second simulation (stability map) in 2D. [Edited by the authors.]

The second method is the use of the “fminsearch” MATLAB function (so it solves the $\min_{\alpha, \gamma} |SD_g|(\alpha, \gamma)$ problem). The results have been the followings: $\alpha = 30.9628$; $\gamma = 30.9605$ and $|SD_g|(\alpha, \gamma) = 0.0009092$. This value is the 0.0009092% of the tip over situation. It is an advantage; that MATLAB could solve this problem in less than a second with 49 iteration steps. The problem with this method is that α and γ where symbolic variables at the program. At a more difficult case the symbolized equation cannot be generated by the MATLAB implementation, and it could not handle the logical decisions even in this simple case.

The third method is the use of a simple search algorithm in 3 steps. At the first search the program calculates $|SD_g|(\alpha, \gamma)$ values at 25 times. (5 times α and (multiplied by) 5 times γ) This iteration is resulted the followings: $\alpha = 30$; $\gamma = 30$ and $|SD_g|(\alpha, \gamma) = 0.0821$. This value is the 0.821% of the tip over situation. (Figure 12)

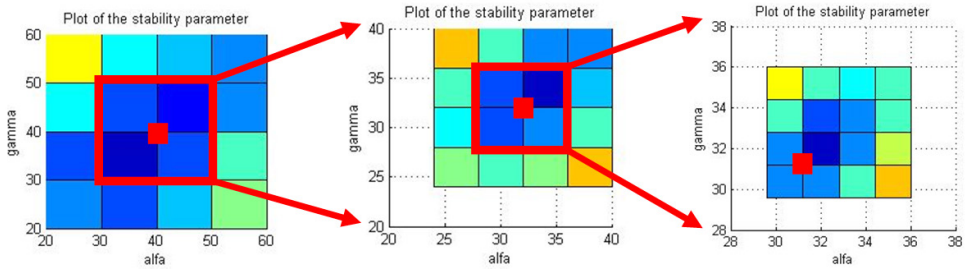


Figure 12. The results of the 1st, 2nd, and 3rd steps at the searching algorithm. [Edited by the authors.]

At the following search the program refines the mesh around the previous minimal point. (Figure 12, 1 and 11)

The second step required 50 iterations (25 for the first and 25 for the second step) The results are the followings: $\alpha = 32$; $\gamma = 32$ and $|SD_g|(\alpha, \gamma) = 0.0566$. This value is the 0.566% of the tip over situation. (Figure 13)

The third step required 75 iterations (25 for the first, 25 for the second and 25 for the third) The results are the followings: $\alpha = 31.2$; $\gamma = 31.2$ and $|SD_g|(\alpha, \gamma) = 0.0131$. This value is the 0.131% of the tip over situation. (Figure 14)

After 150 iteration (6×25) the algorithm could find a solution, where $|SD_g|(\alpha, \gamma) = 0.00037889$. This value is the 0.0037889% of the tip over situation.

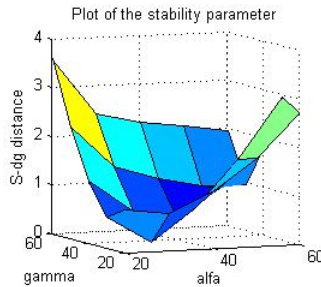


Figure 13. The result of the 1st step. [Edited by the authors.]

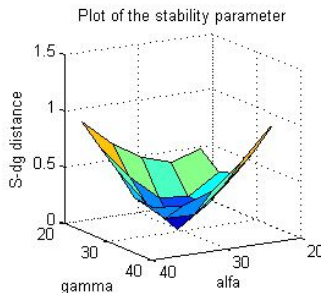


Figure 14. The result of the 2nd step. [Edited by the authors.]

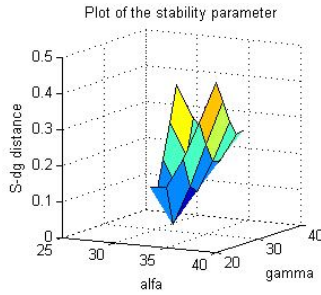


Figure 15. The result of the 3rd step.
[Edited by the authors.]

Table 1. The results of the different iteration methods.
[Edited by the authors.]

Method	Stability map	fminsearch	Searching algorithm			
Num. of iter.	2500	49	25	50	75	150
	31	30.9628	30	32	31.2	30.9568
	31	30.9605	30	32	31.2	30.9568
	0.0022	0.00009092	0.0821	0.0566	0.0131	0.00037889
Error %	0.022	0.0009092	0.821	0.566	0.131	0.0037889

This method worked without symbolic variables and could run in less than a second. The main disadvantage of the algorithm is that it can find a local minimum value instead of the optimal solution. This problem can be solved with further developments.

At the next simulation, MATLAB simulated the robot on a rough terrain. The 3D path on the terrain can be described as Equations (30)–(33), where $r(s)$ is the distance function of the 3D path and the gravitational vector and can be described as Equation (34).

$$r(s) = x(s)i + y(s)j + z(s)k \tag{30}$$

$$x(s) = 1 \times s \tag{31}$$

$$y(s) = 0,8 \frac{\sin x(s)}{x(s)} \tag{32}$$

$$z(s) = 15 \frac{\sin^2 x(s)}{x(s)} \tag{33}$$

$$g = \begin{bmatrix} 0 \\ 0 \\ -9,81 \end{bmatrix} \tag{34}$$

During the 3D path simulation MATLAB worked with the implemented searching algorithm to estimate the optimal geometrical structure arrangement.

At the first test robot performed a slow motion on the path without any stability control methods. The simulation calculated the number of tip over situations at 100 points (with normal distribution of the path) and it resulted, that the robot would tip over on the 72% of the path. (Figure 16, where the red parts mark the tip over on the path.)

At the next step the robot performed a slow motion with the stability control method, where the phase shift of the discrete system was implemented, so the control algorithm used the $(i - 1)^{\text{th}}$ gravitational vector to control the robot at the i^{th} point. This simulation resulted that the robot would tip over only on the 22% of the path. (Figure 17, where the red parts mark the rollover on the path.) The robot had the same orientation along the path so the robot could not turn its variable wheels towards an obstacle. Changing the robot orientation can also improve this number.

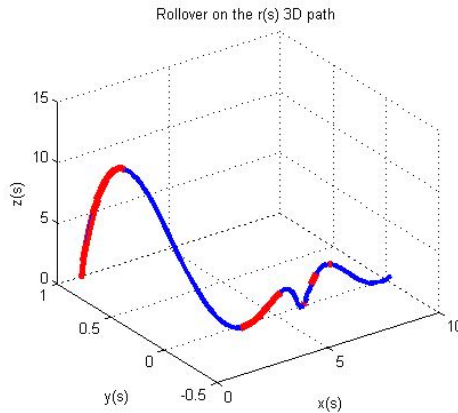


Figure 16. Tip over on the 3D path without stability control.
[Edited by the authors.]

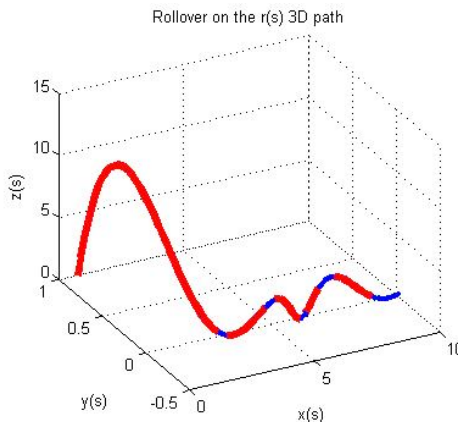
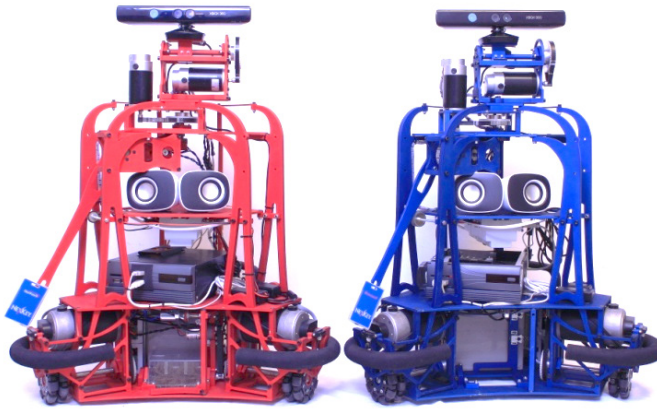


Figure 17. Tip over on the 3D path with stability control.
[Edited by the authors.]

Conclusion

This paper presented a universal mobile robot control method for transformable and hybrid drive mobile robots performing slow motion on unknown terrains, inspired by use case of border surveillance. The basic calculations and the stability optimization method could provide the expected results at the MATLAB simulations. We designed two ethnologically inspired holonomic mobile robots called “Ethon” (Picture 3) and these were the bases of the validation. Unfortunately, these robots are having fix geometry parameters so we could validate the first simulation only with a 3-axis accelerometer and they were not tested in live environment (at the border) yet.



Picture 3. Ethons.
[Made by the authors.]

Acknowledgement

The authors wish to thank the support to the Hungarian Research Fund (OTKA K100951), and MTA-ELTE Comparative Ethology Research Group (MTA: 01 031). The results also discussed above are supported by the grant TÁMOP-4.2.2.B-10/1-2010-0009.

References

- [1] LIANGJI, C.: Kinematics Modelling and Post-Processing Method of Five-Axis CNC Machine. *International Workshop on Education Technology and Computer Science*, 1 (2009), 300–303.
- [2] HANENDA, A., OKADA, K., INABA, M.: Interpreter based interactive robot simulator to realize dynamics effects caused by robot kinematic actions. *IEEE International Conference on Mechatronics and Automation*, 08 (2008), 994–999.

- [3] LI, B., GUO, Y., SHAO, X., WANG, W., YI, J.: Novel inverse dynamics control strategy with different phases for the quadruped robot. *World Congress on Intelligent Control and Automation*, 07 (2012), 3601–2606.
- [4] MUTOH, N., NAKANO, Y.: Dynamics of Front-and-Rear-Wheel-Independent-Drive-Type Electric Vehicles at the Time of Failure. *IEEE Transactions on Industrial Electronics*, 59 3 (2012), 1488–1499.
- [5] NÉMETH J.: The role of the Bureau of Border Patrol and its tasks against the terrorism: Domestic and international opportunities. *Academic and Applied Research in Military Science*, 6 4 (2007), 623–629.
- [6] VARGA J.: The system of Hungarian border police administration. In. SABAYOVÁ, M., KRÁLIKOVÁ, K. (Eds.): *Racionalizácia verejnej správy: zborník vedeckých prác z medzinárodnej vedeckej konferencie konanej 4–5. februára 2016 v Piešť'anoch*. 369–373. Bratislava, 2016.
- [7] BALLA J.: Applying Biometric Data for Personal Identification. *Biztonságpolitika.hu*, 13 8 (2013), 1–11.
- [8] LI, S., LIU, J., SUN, M., WEN, W., BI, S.: Optimal Design for Self-Adaptive Mobile Robot Based on Virtual Center of Motion. In. *International Conference on Robotics and Biometrics*. 414–419. Guillin, 2009.
- [9] UDENGAARD, M., IAGNEMMA, K.: Kinematic Analysis and Control of an Omnidirectional Mobile Robot in Rough Terrain. In. *Intelligent Robots and Systems*. (Conference) 795–800. San Diego, 2007.
- [10] ADACHI, H., KOYACHI, N., ARAI, T., SHIMIZU, A., NOGAMI, Y.: Mechanism and Control of a Leg-Wheel Hybrid Mobile Robot. In. *International Conference on Intelligent Robots and Systems*. 1792–1797. Kyongju, 1999.
- [11] LIM, S. K., PARK, D. I., KWAK, Y. K., KIM, B-S., JEON, S-W.: Variable Geometry Single-Track Mechanism for a Rescue Robot. In. *International Workshop on Safety, Security and Rescue Robotics*. 111–115. Kobe, 2005.
- [12] LI, Z., MA, S., LI, B., WANG, M., WANG, Y.: Parameter Design and Optimization for Mobile Mechanism of a Transformable Wheel-Track Robot. In. *International Conference on Automation and Logistics*. 158–163. Shenyang, 2009.
- [13] LEE, W., KANG, S., KIM, M., PARK, M.: ROBHAZ-DT3: Teleoperated Mobile Platform with Passively Adaptive Double-Track for Hazardous Environment Applications. In. *International Conference on Intelligent Robots and Systems*. 33–38. Sendai, 2004.
- [14] DAI, Y-J., NAKONO, E., TAKAHASHI, T., OOKUBU, H.: Motion Control of Leg-Wheel Robot for an Unexplored Outdoor Environment. In. *Intelligent Robots and Systems '96*. (Conference) 402–409. Osaka, 1996.
- [15] KANG, S., CHO, C., LEE, J., RYU, D., PARK, C., SHIN, K-C., KIM, M.: ROBHAZ-DT2: Design and Integration of Passive Double Tracked Mobile Manipulator System for Explosive Ordnance Disposal. In. *International Conference on Intelligent Robots and Systems*. 2624–2629. Nevada, 2003.
- [16] KIM, J., LEE, C.: Variable transformation Shapes of Single-Track Mechanism for a Rescue Robot. In. *International Conference on Control, Automation and Systems*. 1057–1061. Seoul, 2007.
- [17] LI, Z. MA, S., LI, B., WANG, M., WANG, Y.: Kinematics Analysis of a Transformable Wheel-Track Robot with Self-Adaptive Mobile Mechanism. In. *International Conference on Mechatronics and Automation*. 1537–1542. Xi'an, 2010.

- [18] MARAU, R., LEITE, P., VELASCO, M., MARTI, P., ALMEIDA, L., PEDREIRAS, P., FUERTES, J. M.: Performing Flexible Control on Low-Cost Microcontrollers Using a Minimal Real-Time Kernel. *IEEE Transactions on Industrial Informatics*, 4 2 (2008), 102–124.
- [19] ROBERGE, V., TARBOUCHI, M., LABONTE, G.: Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Transactions on Industrial Informatics*, 9 1 (2013), 132–141.
- [20] ZHANG, Q., LAPIERRE, L. XIANG, X.: Distributed Control of Coordinated Path Tracking for Networked Nonholonomic Mobile Vehicles. *IEEE Transactions on Industrial Informatics*, 9 1 (2013), 472–484.
- [21] LEI, Z., WEI, G., XIAOLIN, D.: Real-time ground dynamics simulation method of a wheeled mobile robot in virtual terrain. In. *International Conference on Computer Science and Information Processing*. 355–358. Xian (Shaanxi), 2012.
- [22] KIM, G., CHUNG, W.: Tripodal Schematic Control Architecture for Integration of Multi-Functional Indoor Service Robots. *IEEE Transactions on Industrial Electronics*, 53 5 (2006), 1723–1736.
- [23] AROGETI, S. A., WANG, D., LOW, C. B., YU, M.: Fault Detection Isolation and Estimation in a Vehicle Steering System. *IEEE Transactions on Industrial Electronics*, 59 12 (2012), 4810–4820.
- [24] LI, Z., MA, S., LI, B., WANG, M. WANG, Y.: Parameters of the Drive System for a Transformable Wheel-Track robot with Self-Adaptive Mobile Mechanism. In. *International Conference on Robotics and Biometrics*. 339–344. Tiajin, 2010.
- [25] MAZUR, A.: A new universal adaptive tracking control law for nonholonomic wheeled mobile robots moving in R^3 space. In. *First Workshop on Robot Motion and Control. RoMoCo'99*. 113–121. Kiekrz, 1999.
- [26] KIM, J., RYU, J., LEE, C.: Variable shapes single-tracked of a belt type wheel mechanism. In. *International Conference on Advanced Intelligent Mechatronics*. 1065–1069. Xi'an, 2008.
- [27] NOREILS, F. R., CHATILA, R. G.: A General Structure for Mobile Robot Action Control. In. *International Workshop on Intelligent Robots and Systems '89*. 550–556. Tsukuba, 1998.
- [28] KWON, H-J., SHIM, H., KIM, D. G., PARK, S-K., LEE, J.: A Development of a Transformable Caterpillar Equipped Mobile Robot. In. *International Conference on Control, Automation and Systems*. 1062–1065. Seoul, 2007.
- [29] VYSIN, M., KNOFLICEK, R.: The Hybrid Mobile Robot. In. *Industrial Technology*. 262–264. Brno, 2003.
- [30] LI, Z., MA, S., LI, B., WANG, M., WANG, Y.: Design and Basic Experiments of a Transformable Wheel-Track Robot with Self Adaptive Mobile Mechanism. In. *International Conference on Intelligent Robots and Systems*. 1334–1339. Taipei, 2010.
- [31] CHOI, K. H., JEONG, H. K., HYUN, K. H., CHOI, H. D., KWAK, Y. K.: Obstacle Negotiation for the Rescue Robot with Variable Single-tracked Mechanism. In. *Advanced intelligent mechatronics. 2007 IEEE/ASME International Conference*. 1–6. Zürich, 2007.
- [32] BIGLARBEKIAN, M., MELEK, W. W., MENDEL, J. M.: Design of Novel Interval Type-2 Fuzzy Controllers for Modular and Reconfigurable Robots: Theory and Experiments. *IEEE Transactions on Industrial Electronics*, 58 4 (2011), 1371–1384.

- [33] BATURONE, I., MORENO-VELO, F. J., BLANCO, V., FERRUZ, J.: Design of Embedded DSP-Based Fuzzy Controllers for Autonomous Mobile Robots. *IEEE Transactions on Industrial Electronics*, 55 2 (2008), 928–936.
- [34] LILJEBÄCK, P., PETTERSEN, K. Y., STAVDAHL, Ø., GRAVDAHL, J. T.: Snake Robot Locomotion in Environments with Obstacles. *IEEE/ASME Transaction on Mechatronics*, 17 6 (2012), 1158–1169.
- [35] YU, J., DING, R., YANG, Q., TAN, M., WANG, W., ZHANG, J.: On a Bio-inspired Amphibious Robot Capable of Multimodal Motion. *IEEE/ASME Transaction on Mechatronics*, 17 5 (2012), 847–856.
- [36] MAS, I., KITTS, C.: Obstacle Avoidance Policies for Cluster Space Control of Nonholonomic Multirobot Systems. *IEEE/ASME Transaction on Mechatronics*, 17 6 (2012), 1068–1079.
- [37] XIE, Y., WANG, W., GUO, J., LEE, K-M.: Edge detection using structured laser pattern and vision for mobile robot navigation. In. *Advanced Intelligent Mechatronics (AIM)*. 910–915. Budapest, 2011.
- [38] CHANG, Y-C., KIM, W-J.: Aquatic Ionic-Polymer-Metal-Composite Insectile Robot with Multi-DOF Legs. *IEEE/ASME Transaction on Mechatronics*, 18 2 (2013), 547–855.
- [39] DING, J., GOLDMAN, R. E., XU, K., ALLEN, P. K., FOWLER, D. L., SIMAAN, N.: Design and Coordination Kinematics of an Insertable Robotic Effectors Platform for Single-Port Access Surgery. *IEEE/ASME Transaction on Mechatronics*, 18 5 (2013), 1612–1624.
- [40] www.police.hu/a-rendorsegrol/statisztikak/hatarrendeszet (Downloaded: 21 12 2016)

