TRACER ROBOT WITH A PROPORTIONAL CONTROL

ELENA PIVARCIOVA, TIBOR CSONGRADY

Technical University in Zvolen
Faculty of Environmental and Manufacturing Technology
Department of Machinery Control and Automation
Zvolen, Slovak Republic

DOI: 10.17973/MMSJ.2016_11_201690

e-mail: pivarciova@tuzvo.sk

The paper deals with the analysis of the proportional control of a discrete process in a transport robot model. The issue was addressed by determination of relationships between individual external and internal variables and parameters of the process. The gained results provide abetter understanding of the relations which will lead to better results in bigger and more complex processes. Such results can be used in above mentioned algorithms programming for the automatic guided vehicles (AGV) used for logistic purposes and for manipulating with specific materials.

KEYWORDS

tracer robot, proportional control, PID, algorithm

1 INTRODUCTION

Significant expansion of automatic guided vehicles AGV application mainly for logistics purposes, but also for special materials manipulation is currently taking place in the production sphere. A commonly used technique for navigation is thephysical marking of the requested move path trajectories on a floor (optical, magnetic, by operators in a floor or by other means). That can be applied in production systems, where robots cart components, eventually reload goods in stores etc. These transport vehicles present palette conveyors, towing vehicles, conveyors of components and materials operating at assembly lines. Such robot equipment have to operate autonomously, they have to bypass possible barriers and consequently find again the guide track and complete their tasks [JBT 2016], [Pirnik 2015].

In the case of the guide track marking by an optical symbol, a steering mechanism can be simple electronic components such as phototransistors or more complex equipmentsuch as cameras, a smart perceptive system. To locate possible barriers it is necessary to supplement the steering mechanism with other suitable sensors, too.

The Research and Education Laboratory for Roboticsis currently being built at the Department of Machinery Control and Automation at the moment. It is being equipped with robot LEGO education kits and a model robot of Arduino.

Considering possible broad applications of the automatic guided vehicles, building and design of a simple autonomous moving robot that follows a drawn line with a number of curves, straight parts, alternatively witha crossing or a short break, too.

There is plenty of such systems management examples (especially on the web), but they mostly describe their programming and controlling in a non-systematic and accidental way, by a trial and error method application. Therefore, we tried to define, modify and mathematically describe some of the chosen methods in compliance with the theory of automatic controlling and regulating.

2 BASIC METHODS OF CONTROLLING AND REGULATING

We understand the term „control“ as purposeful and systematic actionlaid on a controlled plant so that defined goals are achieved. A number of controlling systems exist including manipulation, opened,closed (with a feedback), programmedetc.

We understand the term „regulation“ as a specific controlling type when the requested behavior or status is gained by a stabilization of output quantities of the plant being controlledconsidering requested values. Regulation is carried out by the „feedback“ application, which enables obtaining information about the plant and the comparison of a current plant status with the defined control aims [Gvozdiak1990], [Franklin 2002], [Jurisica 2002], [Dorcak 2006], [Valisek 2007], [Nascak 2014], [Sucik 2014].

The control is a more general category and we understand the control as automatic control using automatic regulation with feedback principles.

Basic control diagram with a feedback by the automatic controlling principles application is shown in Fig. 1.

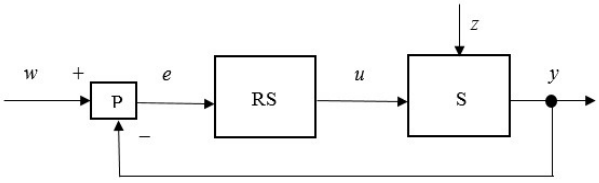


Figure 1. Block diagram of automatic controlling with a feedback

Single elements in Fig. 1 represent as follows:

S – controlled plant, object of controlling, RS – control device, controller, regulator, P –summing point, comparator, *y* –plant output, controlledvalue, *u* –control action, controlled output, *z* –disturbance, external impact on the controlled plant, *w*– reference or command input, desired value for *y*, *e* – tracking error, control error: $e = w - y$.

The feedback provides information on thecurrent value of input value *y* which is then compared to desired output value *w*. That enables to determine a continuous regulation deviation *e*.

A controller RS, which transforms the control error *e* into a control action *u*, performs a fundamental role in the mentioned system.

The controlling can take place in a discrete or continuous way. A basic element of the discrete controlling is such nonlinear element whose output operating value*u*takes up two or three discrete values. It is a „two- or three-position control“.

Three basic controller types (Proportion, Integration, and Derivation) are applied for the continuous controlling. Control action proportion*u* is always determined from a regulation deviation*e*and a difference of the referencevalue *w* and the actual value of the plant output*y*.

– Proportional – P-controller, where the regulating control intervention is in a linear (proportional) relationship to the control error.
– Integrating – I-controller, where the controlling intervention is derived from a control error integral which means it is derived from previous changes, i.e.the history.
– Derivation – D-controller, where the controllingintervention is set by the control error derivation, what is rather the control error running outline and it represents a prediction forward i.e. „forecasts“ the future.

These basic controller types combination are also used in practice as controller and then we meet PI, PD a PID controllers, where interventions of the individual types are summed to total controlling intervention.

2.1 *PID (PSD) controller*

General form of the proportional-integral-derivative (PID) controller(a form with time constants) is:

$$u(t) = r_0\left[e(t) + \frac{1}{T_i}\int_0^t e(t)\,\mathrm{d}t + T_d\,\frac{\mathrm{d}\,e(t)}{\mathrm{d}\,t}\right], \quad (1)$$

where*t* is a time, r_0 is aparameter, $e(t)$ is a control error and $u(t)$ is a control action of the controller, T_i is integrating and T_d is derivation time constant.

Or after adjusting:

$$u(t) = r_0 e(t) + \frac{r_0}{T_i}\int_0^t e(t)\,\mathrm{d}t + r_0 T_d\,\frac{\mathrm{d}\,e(t)}{\mathrm{d}\,t} = K_p e(t) + K_i\int_0^t e(t)\,\mathrm{d}t + K_d\,\frac{\mathrm{d}\,e(t)}{\mathrm{d}\,t}, \quad (2)$$

where$K_p = r_0$; $K_i = r_0 / T_i$; $K_d = r_0 . T_d$ is a proportional, integrating and derivation constant or parameter.

For the discrete version, it is then: $t = k.T$, where T is a sampling period and k is a tact.

The integral component I(kT) is then replaced with a summary and the derivation component $D(kT)$ is replaced with a differential formula:

$$I(kT) = \int_0^{kT} e(t)\,\mathrm{d}t \cong T\sum_{i=1}^{k} e(i)\,, \quad (3)$$

$$D(kT) = \frac{\mathrm{d}\,e(t)}{\mathrm{d}\,t} \cong \frac{1}{T}\left[e(k) - e(k-1)\right]. \quad (4)$$

Then we get the discrete version of the PID controller an expression:

$$u(kT) = r_0\left[e(kT) + \frac{1}{T_i}I(kT) + T_d D(kT)\right], \quad (5)$$

or:

$$u(k) = r_0\left\{e(k) + \frac{T}{T_i}\sum_{i=1}^{k} e(i) + \frac{T_d}{T}\left[e(k) - e(k-1)\right]\right\} + u(0) \quad (6)$$

where$u(0)$ represents the starting value.

It is so called the positional algorithm for the discrete PID controller formulation. Since the integral is gained by summarization and derivation is calculated by a differential, the name of "proportional-summarizing-differential PSD controller" is also used [Huba 2008]. The most frequently used controller is the proportional P-controller.

To serve the purpose of the relations application we will use the following formula:

$$u(k) = K_p.e(k) + K_i\sum_{i=1}^{k} e(i) + K_d\left[e(k) - e(k-1)\right], \quad (7)$$

where individual symbols represent: $u(k)$ –control action, $e(k)$ – current control error, i – a tact serial number, k – actual tact, K_p – proportional constant, K_i – integrating (summarizing) constant, K_d – derivation (differential) constant.

A program-controlled transporting robot model will beused to test the robot´s motion by various regulating methods. The program will process all the information and determine the controlling actions, though it will operate as a controller.

3 THE TRANSPORTING ROBOT MODEL CONSTRUCTION

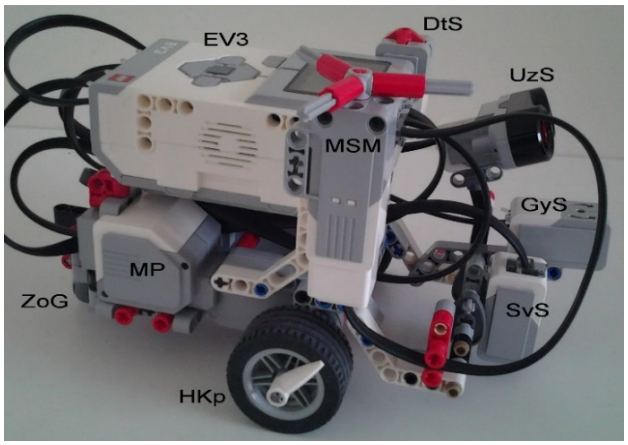
The LEGO Mindstorms EV3 kit represents the latest LEG robots generation, being currently the most wide-spread robot kit in schools of all the types because of their variability, variety, and modifiability [Hlinovsky 2015].

The LEGO Mindstorms EV3 kit combines a programmable „intelligent cube" with input sensors, output operating elements, communication environment and program software. The robot core is represented by the intelligent cube with a microprocessor ARM 9 (of Harvard architecture) on the basis of Linux operating system, with 64 MB RAM and 16 MB Flash, with a Mini SDHC counter card up to 32 GB for a memory expansion, with four input gates for data gathering with a speed of 1000 operating members a second, with an output to a graphic display and with a sound output. The cube can communicate via USB, Wi-Fi, and Bluetooth interfaces.

Optical, ultrasonic, gyroscopic and two contact sensors are included in the basic set for information gathering from an environment. The output operating members are represented by two big and one smaller interactive servomotors with inbuilt rotating sensors. A graphical program software is also a part of the set enabling to use all inbuilt components, including the possibility of parallel programming.

We used the construction designed by the LEGO (45544 Core Set LEGO Education) company as „tricycle", using input and output elements from the basic and the extended kit sets, to analyze and test the individual methods of controlling. Therefore, only one color sensor was used, which reduces theamount of received information on the robot location on the working floor (Fig. 2).

The robot turning was designed with a differential drive mechanism. It consists of 2 drive wheels mounted on a mutual axis, and each wheel can be driven independently either forward or backward [Sakshat 2016].



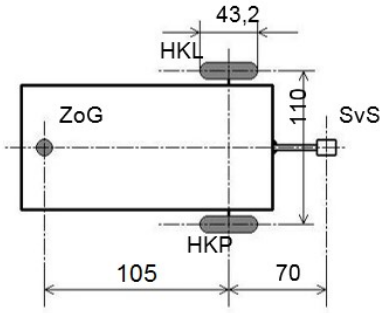


Figure 2. Robot model and its schematic diagram

EV3 – intelligent cube, SvS – light sensor, DtS – contact sensor, UzS – ultrasonic sensor, GyS – gyroscopic sensor, MSM – small servomotor, HKL, HKP – left and right driving wheel, MP, ML – right and left motor, ZoG –rollerball wheel.

The original construction was modified as follows:

– the light sensor SVS was arranged in front of the wheels axle in thecenter to ensure symmetric reactions in the moving robot oscillation; it was mounted firmly, approximately 7 mm above a surface and 70 mm in front of the wheels axle,
– the sensors wereadded:
  - a small MSM servomotor acting as a sensor of manual rotating of the motor axle which enables changing individual parameters at the robot´s start and partly also during its operation,
  - a contact sensor DtS to approve/reconcile the parameter values and their changes,
  - a gyroscopic GyS sensor to monitor the robot oscillation during movement,
  - an ultrasonic UzS sensor to be able to stop the robot in case of a barrier approaching,
– HKL an HKPdrive wheelsof a diameter of 43.2 mm were used to enable controlling action in a wider values range; the distance between the wheels was 110 mm.

The controlled system (robot) has two operating members (motors), whose performance is controlled by autonomous operating quantities, to carry out a motion in a direction. The input values can determine some of the trajectory parameters (e.g.light reflection of the line color and reflection of the surface out of the line, the chosen level of the allowed minimal and maximal performance of driving motors, proportionality area size etc). The output quantity is the robot position or its optical reflection of the current position (a color scanned by a light sensor). The aforementioned implies that the robot controlling is a solution of multi-parameter controlling.

The paper is mainly aimed at the use of a proportional controlling principle.

4 DIFFERENTIAL DRIVE KINEMATICS

A differential drive robot control enables independent velocity control of the left (V_L) and right (V_R) wheels (Fig. 3).

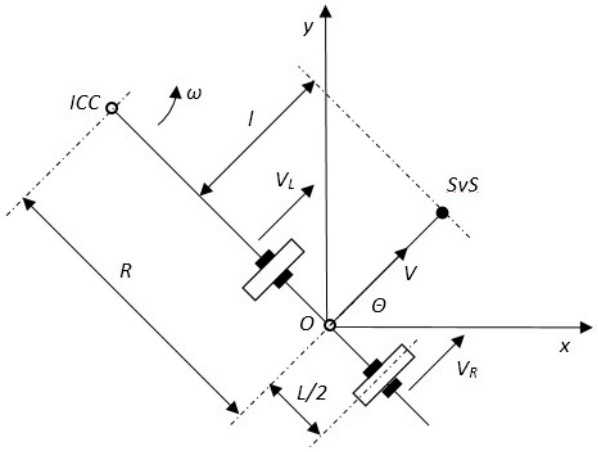


Figure 3. Differential drive kinematics (by[Dudek 2010]

ICC – instantaneous center of curvature, ω – angular rate, V_L – velocity control to the left wheel, V_R – velocity control to the right wheel, L – distance between the wheels, R – turning radius, SvS – light sensor, l – distance from the light sensor to the midpoint between the wheels, O – the midpoint between the wheels, V – velocity of the center between the wheels, Θ – direction of velocity

Under the differential drive, the robot must rotate around a point of ICC (instantaneous center of curvature) that lies on the common axis of the two drive wheels to perform a rolling motion of each of the two drive wheels. The point of this rotation can be changed by changing the relative velocity of the two wheels and then different vehicle trajectories can be chosen. At each moment of time, it must apply that the point at which the robot rotates that the left and right wheels follow the trajectory that goes around the ICC at the same angular rate ω, and thus

$$\omega\left(R+\frac{L}{2}\right)=V_R\,,\ \omega\left(R-\frac{L}{2}\right)=V_L\,. \tag{8}$$

VL, V_R, ω and R are all functions of time. At any moment of time, solving for R and ω leads to:

$$R=\frac{L}{2}\frac{(V_L+V_R)}{(V_R-V_L)},\ \omega=\frac{V_R-V_L}{L}. \tag{9}$$

If $V_L = VR$, then the radius R is infinite, and the robot moves in a straight line. If $V_L = -V_R$, then the radius is zero, and the robot rotates around a point O (midpoint between the wheels), that is, it rotates in the place. This makes a differential drive attractive for robots that must navigate in narrow environments. For other values of V_L and V_R, a trajectory is curved with a turning radius R, changing both the robot's position and orientation.

The kinematic structure of the vehicle does not allow certain vehicle´s motions. For example, there is no combination of V_L and V_R such that the vehicle moves directly along the wheels' common axis.

To compute a robot's position x in the idealized error-free case with a velocity vector dx/dt, we use

$$x=\int_{t_0}^{t_f}\frac{dx}{dt}dt \tag{10}$$

where the motion takes place over a time interval t_0 to t_f. More generally, we can integrate x repeatedly to recover the position for motion information from higher-order derivatives (such as acceleration). However, this also implies that errors in the sensing or integration process are manifested as higher-order polynomials of the time interval over which we are integrating.

For discrete motions, where a positional change is expressed by a difference vector δ_i, we can calculate the absolute position as

$$x = \sum \delta_i \quad (11)$$

This method of position estimation plays an important part in most mobile robot systems. It can have acceptable accuracy over sufficiently small steps, assuming a suitable terrain and drive mechanism. Thus, the final position after several motions can be calculated from a known starting position. For large or complex motions, however, the unavoidable errors in the individual position estimates have a major consequence, Simple scaling errors can be readily corrected, but more complex errors are essentially impossible to eliminate.

Forward kinematics for differential drive robots

Suppose that the robot is at a position (x, y) and facing along a line making an angle Θ with the x-axis. Through manipulation of the control parameters V_L and V_R, the robot can take different positions. Determining the position is available if the control parameters are known as the **forward kinematicsproblem** for the robot. ICC location is given by:

$$ICC = \left(x - R\sin(\theta), y + R\cos(\theta)\right) \quad (12)$$

and at the time $t + \delta t$ the position of the robot is given by:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix} \quad (13)$$

This equation describes the motion of a robot rotating a distance R around its ICC with an angular velocity given by ω.

It is possible to compute by integrating the equation from an initial condition (x_0, y_0, θ_0), where the robot will be at any time t based on the control parameters $V_L(t)$ and $V_R(t)$, in other words, to solve the forward kinematics problem for the vehicle. In general, for a robot capable of moving in a particular direction $\theta(t)$ at a given velocity $V(t) = (V_R(t) + V_L(t))/2$, for the special case of the differential drive vehicle it applies [Dudek 2010]:

$$\begin{aligned} x(t) &= \frac{1}{2}\int_0^t (V_R(t) + V_L(t))\cos(\theta(t))\,dt \\ y(t) &= \frac{1}{2}\int_0^t (V_R(t) + V_L(t))\sin(\theta(t))\,dt \\ \theta(t) &= \frac{1}{L}\int_0^t (V_R(t) - V_L(t))\,dt \end{aligned} \quad (14)$$

The resultant path would be a circle and the equation to get the position of the robot is possible after integration.

5 BASIC ALGORITHM OF CONTROLLING

The goal of robot monitoring is tracing the line that is „drawn“ on a flat surface. According to the scheme,the information about the robot´s position is provided only by the light sensor SvS by the optical reflection of its position: whether it is above the line or out of the line, rest. near the line border. Then, the basic control principle can be written down in a symbolic language as follows:

```
repeat  // start of cyclic repeating activity
        Read (position);
        //position is given by the position color
        if you are on the line
        then go out of the line
        //  if you are on the line
        else go to the line
        // if you are not on the line
        *if
        // continuing of the conditional activity
*repeat // return to the cyclic activity start
```

This basic process has to be described more precisely. The motion direction in 2D space has to be specified by a direction as „go right“ or „go left“. The robot has to receive an unambiguouscommand whether it shallreturn to the linein the direction to the left or the right. In other words, it must be stated clearly, whether it should follow the right or left border of the line.

We chose tracing the right line border for next procedure and so the basic controlling algorithms can be expressed as:

```
repeat
        Read (position);
        // determine the position color
        if you are on the line
        then go right
        // will go out of the line
        else go left
        // will return to the line
        *if
*repeat
```

The robot movement will thenbe oscillating about the right line border. In thecase of choosingthe left line border tracing, the individual directions in the branch of „then” and „else” would be contrary.

The block diagram in Fig. 1 and the basic principle of automatic controllingboth use a feedback for a control action. Therefore, it is necessary to define parameters: reference value and plant output, of which the control errorcan be calculated and the controlling action determined.

As the previous basic control algorithm description suggests, the regulated output quantity which value can be monitored is scanned by the current robot position color reflection (more accurately of robot light sensor). Let us mark this current value as: **fposition** (color position). Let us assume that the line will be drawn in a dark (e.g. black) color on thelighter (e.g. white) background. The light reflection will be the value for the lighter area: **white** (out of the line) always bigger than the darker line reflection: **black**. Aborderline can then be taken as the requested value: **border** and can be determined as an arithmetic average of both the white and black values:

$$border = (white + black) / 2 \quad (15)$$

A difference between the requested value and the current light reflection will be thecontrol**error**:

$$error = border - fposition. \quad (16)$$

This information can be a base for the controller output calculation.

The robot movement is made by two motors´ application and the operating quantity for their control can be the requested performance of these motors **VL** for the left motor and **VP** for the right motor. The strength of these operating interventions is set by a controlleraccording to the chosen type and method of the control. We will choose the allowed interval for performance**V** of the motors from the range: V ϵ <–100, 100>. While doing so, we will consider a performance limit within the range <**Vmin**, **Vmax**> during controlling, so that it applies:

$$-100 \le Vmin < Vmax \le 100. \quad (17)$$

5.1 *Two-position controlling*

We chosenonlinear two-position controlling as a base for various robot controlling methods. The control was chosen in the way that only one of the motors is running in time in a performance Vmax and the second will have zero performance, it will be turned off.

The algorithm of the two-position control can be written as:

```
//Alg.:01-NTPC (Nonlinear Two-Position Controlling)
Read (black);  // line reflection
Read (white);  // surface out of line reflection
border = (white + black)/2;   // requested value
Vmin = 0;      // we choose minimal power = 0
Vmax = 55;     // we choose maximal power
repeat  // start of (infinite) control cycle
        Read (fposition);
                // trace where is located
        error = border - fposition;
                // control error
        if(error > 0)
                // are you at the line?
        then    VLM = Vmax;     // turn right
                VPM = Vmin
        else    VLM = Vmin;     // turn left
                VPM = Vmax
        *if;
*repeat;        // control cycle closing
```

Commands VLM and VPM are symbolic scripts for commands for left and right motors control with the stated performance.

The motor movement will be oscillating-clonic around the line right border. A motion speed and oscillation width will be proportional to the Vmax value. Depending on the robot construction and on the trajectory geometry that has to be taken by the robot, in certain Vmax value, there can occur an unstable status when the robot is not able to follow the line and starts sudden, mostly circular, moves on the surface.

6 PROPORTIONAL LINEAR CONTROL

6.1 *Symmetric algorithms*

The basis of the symmetric algorithms is determining the requested value as an arithmetic average of the light line and out of line area reflection values and symmetric motor controlling.

The principle of the proportional control lies in the fact that the controlling action –controller output(**RZ**) – strength is linearly proportional to the control errorstrength. If the requested value border will be determined by (15) and the control error by (16), then the control action is calculated as:

RZ = Kp × error. (18)

The control can be carried out by choosing a proportionalitycoefficient**Kp** – proportional parameter (a proportional constant) and interventions in individual motors are executedsymmetrically in opposite directions:

```
Read (black);  // black color reflection
Read (white);  // white color reflection
border = (white + black)/2;
                // the requested value
Kp = 1;         // proportional parameter
repeat
        Read (fposition);
                // determines its position
        error = border - fposition;
        RZ = Kp * error;
        VLM = RZ;
        VPM = -RZ;
*repeat;
```

During the algorithm testing, the robot appears to oscillate around the line right border but is not moving forward. It is necessary to implement another input request, e.g.to determine basic performance**Vz** for motors towards which the controlling action RZ will be superposed, as follows:

```
// Alg.: 02-SPC (Symmetric Proportional Control)
Read (black);  // black color reflection
Read (white);  // white color reflection
border = (white + black) / 2;
        // the requested value
Kp = 1;         // proportional parameter
Vz = 40;        // basic power
repeat
        Read (fposition);
                // determines its position
        error = border - fposition;
        RZ = Kp * error;
        VLM = Vz + RZ;
        VPM = Vz – RZ;
*repeat;
```

We can use the functional interpretation of the motors´performance to choose a proportional constant Kpdepending on the error variable strength:

$$VLM = f1(error) = Vz + Kp \times error \quad (19)$$

VPM = f2(error) = Vz – Kp × error

which can be presented graphically by Fig. 4.

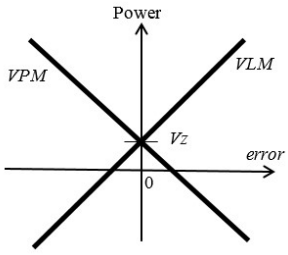


Figure 4. Graphical representation of performance proportional controlling

The higher Kp value means the bigger intervention to the controlled motor performance, the bigger sensitivity tothe change of thedifference between the requested value (at the line border) and the current value scanned by the light sensor (*border* – *fposition* = *error*). The lesser *K*p value enables finer robot movement control, will have smaller amplitudes in the line following. The *VLM* and *VPM* lines cross the performance axis in the value of the basic performance *Vz*.

Assuming that the reflected light value *fpositions*canned by the light sensor is:

fposition ϵ <*black*, *white*>, (20)

which can actually be changed by the environment illumination change, then *error* gets values from the interval ±(*white* – *black*)/2.

If the motors performance is limited by a minimal and maximal Vmin and Vmax performance, then the line representing the performance of the (left) motor will be passing through {A, B} points as in Fig. 5.

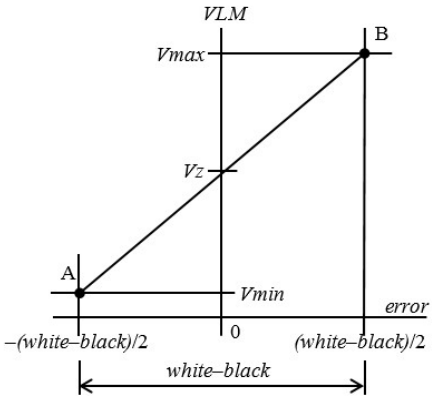


Figure 5. Diagram of controlling line for the left motor VLM

The intersection point of the A-B line with the *VLM* axis is the point: *Vz* = (*Vmax* – *Vmin*) / 2. The *Kp* line slope (the proportional parameter) can be calculated as:

$$Kp = \frac{Vmax - Vmin}{white - black} \tag{21}$$

and it gives sothe biggest constant *K*p value that the ***rRZ***controlling action range

$$rRZ = Vmax - Vmin \tag{22}$$

does not exceed *Vmin*, *Vmax* limits.

The *K*p size depending on the **distance** values of line and surrounding surface out of line light reflection (the black and white color) for different values of thedifference value between maximal and minimal performance allowed or chosen motors performance*Vmax* and *Vmin*are presented in Fig. 6.

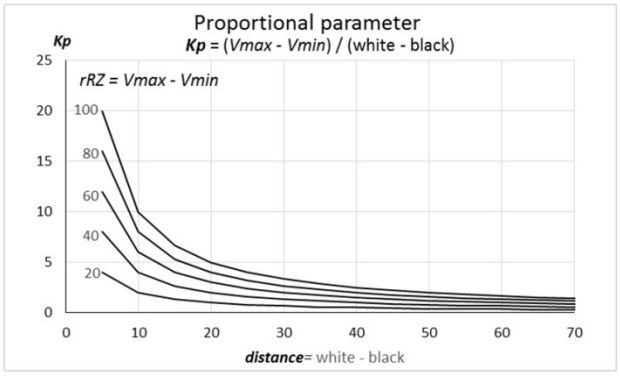


Figure 6. Proportional parameter dependency on white and black color light reflection difference

The diagram shows that the proportional parameter decreases by increasing the distance between out-of-line and line reflections

$$distance = (white - black), \tag{23}$$

as well as by decreasing the controlling action/interventionchosen range *rRZ* = (*Vmax* – *Vmin*):

$$Kp = rRZ / distance. \tag{24}$$

If using just positive motors performance in the interval of <0, 100> then the basic performanceVzdepending on Vmax and Vmin can be chosen: $Vz \in \left\langle \frac{rRZ}{2}, 100 - \frac{rRZ}{2} \right\rangle$, as it is shown in Fig. 7.

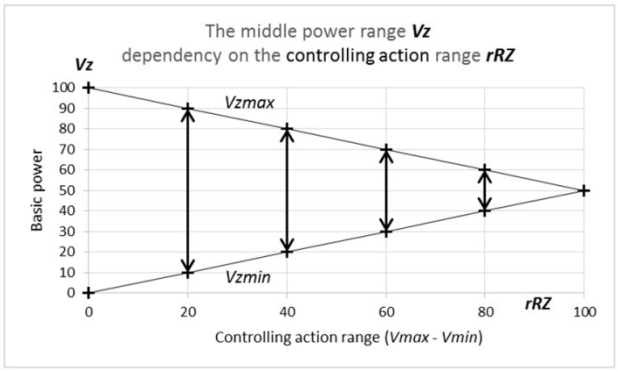


Figure 7. Possible basic performance range depending on the controlling action range

where*Vzmin* = *rRZ* / 2 is the minimal value and *Vzmax* = 100 – rRZ /2 is the maximal value of possible basic performance.

Necessary parameters can be obtained as follows:

– the *distance* = *white* – *black* is calculated from actually found data from the light sensor,
– the allowed controlling actionrange size *rRZ* = *Vmax* – *Vmin* is set considerably to the robot´s motion requested fineness (oscillation size and speed),
– the basic performance*Vz* is set from interval <*Vzmin*, *Vzmax*> from allowed range and by the controlling action*rRZ* range size.

The robot controlling symbolic algorithm can be as follows:

```
//Alg.:03-PCSRIR Proportional Controlling
//Specified by the controlling action range
// constants initialization
rRZ = 30;       // controlling action
                // range Vmax-Vmin
Vz = 10;        // basic power
        // border and distance requested
        // value determination: white-black
Read (black);  // black color reflection
Read (white);  // white color reflection
border = (white + black)/2;
        // middle value for ta border
        // line border is requested value
distance = white - black;
        // proportional parameter
Kp = rRZ/distance;
        // the motor control
repeat
        Read (fposition); // determines position
        error = border - fposition;
                // current control error
        RZ = Kp * error;
                // control action
        VLM = Vz + RZ; //operating intervention
                       // for the left motor
        VPM = Vz - RZ; //operating intervention
                       //for the right motor
*repeat;
```

Optimal parameters value *rRZ* and Vz depends on the robot construction as well as on trajectory parameters to be taken by the robot. Their value can be set gradually. It is recommended to set *rRZ* = 10, 20, 30..., *Vz* = *rRZ*/2 for the beginning until theinstability status is reached (the robot will not be able to follow the trajectory anymore). Then rRZis decreased and Vzis possible to be increased gradually, whilean increased robot´s oscillation is also achieved.

6.2 *Competitive algorithms*

To achieve good results the method of competition is frequently used. Such principle can also be used for the robot movement controlling when following the line. Let us assume, the line right border following.

Method of aggressive competition

The competition strategy can be verbally described as follows:

The left motor will aim to force out the robot from the line to the right, to the surface out of the line and the right motor, contrary, to get the robot out of the surface to the left, back to the line. This can be understood as a form of attacking the other.

So when the robot is above the line, the left motor has to raise its performance relative to the distance from the surface out of the line: *white – fposition* (the bigger distance from the surface out of the line, the higher performance). When the robot is above the surface out of the line, the right motor has to increase its performance relative to the distance from the line: *fposition – black* (the bigger distance from the line the higher performance). Meanwhile, we assume thatfposition <black, white>.

In thecase of a small difference *white – fposition,* the left motor performance will be low, similarly in thecase of a small difference *fposition – black* the right motor performance will be low (Fig. 8). The mentioned diagrams enable to derive equations for individual motors controlling easily.

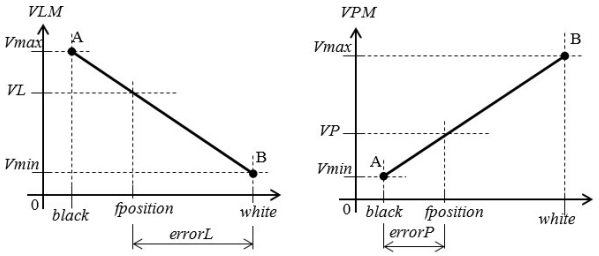


Figure 8. Principle of controlling by the competitive algorithm

Such defined control error*white – fposition* and *fposition – black* are nonnegative ones, which will ensure the robot´s motion in a forward direction (it is possible to use also: *abs(white – fposition)* and *abs(black – fposition))*.

The requested value for the left motor controlling will be defined *white* and the control error size (*errorL*) will be a difference between the requested value and the actual light reflection scanned value:

errorL= white – fposition. (25)

Similarly, we get the control error for the right motor:

errorP = fposition – black. (26)

If the left and right motors both are set for the highperformance*Vmax*, (it will be achieved for the left motor when the sensor is above the line and for the right motor when itis above the surface out of the line) and the basic performance will be set as the value of low performance*Vmin*, then we can formulate for both the left and right motors:

Vmax = Vmin + Kp × (white – black), (27)

When proportionalityparameter*Kp* will be:

Kp = (Vmax – Vmin) / (white – black). (28)

The relation (28) is the same as the relation (21), which means that the proportional parameter can be calculated as a division of the control action range: *rRZ = Vmax – Vmin* and the black line color reflection and white out of the line surface color distance: *distance = white – black*.

Operating quantities for individual motors controlling can be then calculated as:

VLM = Vmin + Kp × errorL and (29a)

VPM = Vmin + Kp × errorP. (29b)

It is necessary to enter the values to control: *Vmax* and *Vmin*, from which theparameter*Kp* will be calculated. Another possibility of control modification can be defined as entering low-performance*Vmin* and the control action*rRZ*, then it will be calculated: *Vmax = zRZ + Vmin*.

At such controlling, various values of the control errorare applied by forms (25) and (26) for individual motors. The controlling algorithm then can be as:

```
//Alg: 04-ACA (Attacking Competitive Algorithm)
Read (black);  // black color reflection
Read (white);  // white color reflection
Vmin = 10;     // minimal power
rRZ = 30;      // control action range
Vmax = rRZ + Vmin      // maximal power
Kp = rRZ/(white-black);// ratability parameter
// proportional controlling
repeat
       Read (fposition); //determines position
       errorL = white-fposition;
       // control actionleft
       errorP = fposition-black;
       // control actionright
       VLM = Vmin + Kp * errorL;
       VPM = Vmin + Kp * errorP;
*repeat
```

Fair competition method

The second, more fair way of competition can be described as follows.

Let the left motor follow „its“ side, i.e. the line. When it is „at home“, it goes full performance, when it is out, it slows.

The right motor follows „its“ surface out of the line. When it is „at home“, it goes full performance, when it is above the line, it slows.

Control errors will be also different for such control, similarly to the previous case:

errorL = fposition – black and (30a)

errorP = white – fposition. (30b)

A proportionality parameter will be the same as in the previous cases:

Kp = (*Vmax – Vmin*) / (*white – black*).

It is possible to derive forms for the operating motor interventions:

VLM = Vmax – Kp × errorL and (31a)

VPM = Vmax – Kp × errorP. (31b)

Combined competitive methods

Competitive methods of controlmayalso be combined, e.g. the fair competition controlling (31a) and (32a) is used for the left motor and the aggressivecompetition controlling (26) and (29b) for the right motor. In such case, the control *error*

will be the same for both the left and right motor, but the basic performance will be different. It will apply that:

error = fposition – black, (32)

And the performance for the motors will be as:

VLM = *Vmax* – *Kp* × *error*, (33a)

VPM = *Vmin* + *Kp* × *error*. (33b)

6.3 *Position information normalization*

Light sensor provides information on current position during the line following in the range *fposition*ϵ<*black*, *white*>. This numeric range can be transformed by simple adjusting:

$$f100\,position = \frac{fposition - black}{white - black} \times 100 \quad (34)$$

into a range: *f100position*ϵ<0, 100>. We labeled the difference in the denominator (23) as *distance* = *white* – *black*.

In this transformation application the proportional parameter*Kp* is calculated as:

Kp = (*Vmax* – *Vmin*) / 100 = *rRZ*/100 (35)

andthe control error will be:

error = *f100position*, (36)

that means it is not necessary to be calculated separately, the transformed value from the light scanner will be applied.

It is suitable to supplement the algorithm with thecheck of range exceeding to prevent accidental exceeding of this range (e.g.in the trajectory surrounding illumination change).

```
if (f100position > 100) then f100position = 100;
*if;
if (f100position < 0)  then f100position = 0;
*if;
```

If the *fposition* range is depicted into the interval<0, 1> by using the transformation relation:

f1position = (*fposition* – *black*) / *distance*, (37)

thentheproportionalityparameter*Kp* will be equal numerically to the control action range*rRZ*, and performances for the motors will be calculated as:

VLM = *Vmax* – *rRZ* × *f1position*, (38a)

VPM = *Vmin* + *rRZ* × *f1position*. (38b)

7 THE ALGORITHMS TESTING AND ACHIEVED RESULTS

The designed algorithms were programmed and verified.

We used a graphical programming system includedin the basic set of the robot kit. We usedthe technique of parallel programming with more program threads.

We used bidirectional wireless communication of Bluetooth for the communication between the robot and PC.

We built two test trajectories to verify functionality and quality of the algorithms (Fig. 9).

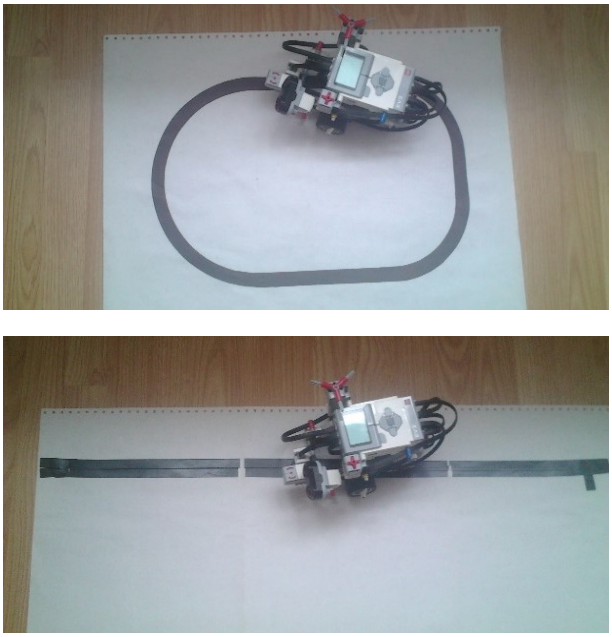


Figure 9. Trajectories for the controlling algorithms testing

Simple closed trajectory with thesize of 440 × 300 mm, the black line width of 20 mm on the white surface. The trajectory has straight segments and segments with the radius of curvature of 100 and 150 mm. The algorithms parameters were set so that the robot passes the trajectory in a steady way, on the outer border of the line (in an anticlockwise direction) and on the inner border of the line (a clockwise direction).

All the algorithms worked well.

By visual evaluation of the robot´s behavior on the trajectory, the algorithms with proportional controllingproduced more subtle oscillation and more fluent robot´s motion, compared with the two-position control. Under an increased speed, critical points were the curve corners with the radius of 100 mm (the value is lesser than the wheels track width: 110 mm). The most important input parameters for the motion stability with good passing in the corners, with the lowest oscillation of the robot and the highest possible speed, was basic or minimal performance (*Vz*, or *Vmin*) and the control action range (*rRZ*).

Values of thedistance of the line black color and the white surface out of the line were not entered manually but scanned by a program.

Parameters were possible to be entered and their adjustment done before the start and during the program running by a small installed servomotor (in Fig. 2: MSM) functioning as a sensor which rotated its axis. Data agreement was performed by a contact sensor (in Fig. 2: DtS).

The best results in a visual evaluation of individual algorithms were achieved for the basic parameters size stated in Tab. 1.

Individual visual evaluation,not even with time measurement, does not provide a complex view of the "quality" of the robot´s ride,as the speed issuewill not probably play the most important parameter in specific materials transporting. Therefore, a method of recording of the robot´s oscillation was designed and test-verified.

To monitor and measure the oscillation linewidth during the robot´s ride,we prepared a straight trajectory (Fig. 8) with a length of 800 mm and two short breaks for the robot´s forced oscillation. A gyroscopic sensor (in Fig. 2: GyS) is used for the vibration size monitoring and the values of the robot angle

deflection from straight ride direction are recorded. We chose the straight trajectory so that the gyroscope data would not be influenced by the trajectory curves,though, they could characterize the robot´s behavior dynamic more accurately.

Table 1. Entered **(in bold)** and calculated values of the controlling parameters

Algorithm	01-NTPC	02-SPC	03-PCSRIR	04-ACA
Vmin	**0**	21,5	21	**20**
Vmax	**40**	58,5	51	55
Vz	---	**40**	**36**	---
rRZ	40	37	**30**	**35**
Kp	---	**0,5**	0,42	0,50
white	92	89	82	83
black	12	15	10	13
distance	80	74	72	70
border	52	52	46	---

The measured robot´s oscillation development are illustrated in Fig. 9 and Fig. 10. A comparison of these developmentspoints out the significance of control use. Not only the faster robot's movement was achieved but also lower robot´s oscillation was recorded in angular degrees.

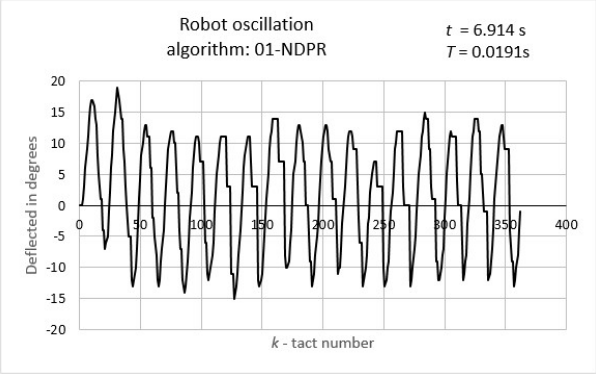


Figure 9. Diagram of the robot deflection process under the two-position control use

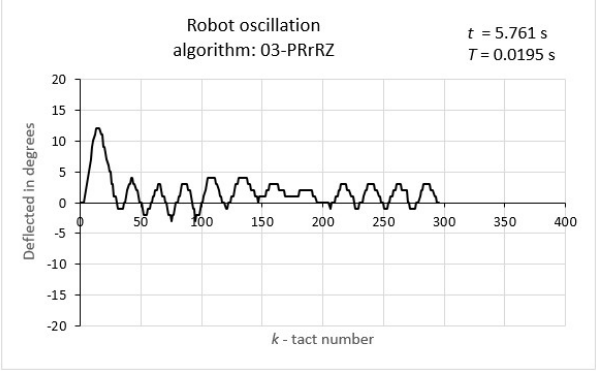


Figure 10. Diagram of the robot deflection process under proportional control

In the diagrams, there are presented values t – a time needed to completean800 mm trajectory and T – a sampling interval. In the horizontal axis, there is presented tact number – k of a discrete process in the algorithm performance. Diagrams data represent oscillation of the light sensor. The conversion of angles to oscillation in mm is possible by applyinga simple formula, stemming from a sensor positioning geometry (Fig. 2):

$$x = L \times \sin(\Theta), \quad (39)$$

where$x$ is a deflection from the straight motion direction in mm, L = 70 is avertical distance of the light SvS sensor from the motors axis, Θ – deflection value in degrees.

8 DISCUSSION

The proportional parameter *Kp* by the formulas (24) and (28) directly proportional to the control action range (*rRZ* = *Vmax* – *Vmin*) and indirectly proportional to the black from white color reflection distance (distance = white – black). That results in a fact that increasing *rRZ* will lead into a proportionally increased robot deflection.

For the robot´s ability to move straight forward, the average of minimal performance *Vmin* and maximal performance *Vmax* has to be positive.

Visual evaluation of the line following implies:

– Robot´s oscillation is bigger in the two-position control than in the proportional control, confirmed by diagrams in Fig. 9 and Fig. 10, too. We can see from the comparison of the robot´s deflections in degrees that while in the two-position control the amplitude is approximately the same, in the proportional control they decrease gradually. But the proportional control principles say that their size remains greater than zero.
– Oscillation increases with increasing of the control actionrange.
– When a trajectory curves radius is significantly smaller than the robot´s drive wheelstrack width, then motors minimal performance*Vmin* has to be negative.

9 CONCLUSIONS

In the paper, we aimed at the analysis of the discrete process of the proportional control in a transport robot simple model. We dealt with the issue by characterizing the relations between individual outer and inner variables and the process parameters. The received results suggest abetter understanding of the relations which will lead to better results in larger and more complex control algorithms.

The study suggests that models, even on basis of simple kits, can be useful and advantages for the practice. The aim of testing the individual robot controlling algorithms was avalidation of the most significant characteristics of the robot´s movement during the line following.

The results obtained from the model robots will be applicable and useful in formulating the control algorithms for commercial automated guided vehicles for logistic purposes and for manipulating with specific materials.

ACKNOWLEDGMENTS

This paper was prepared within the work ona research project KEGA MS SR 003TU Z-4/2016: Research and education laboratory for robotics.

REFERENCES

[Dorcak 2006] Dorcak, L., Terpak, J. and Dorcakova, F. The automated control theory: Analogue linear systems (in Slovak). Kosice: Technical University, 2006. ISBN 80-8073-025-3

[Dudek 2010] Dudek, G., Jenkin, M. Computational Principles of Mobile Robotics. Cambridge University Press 2010 – 2nd ed. ISBN 978-0-521-87157-0

[Franklin 2002] Franklin, G. F., Powel, J. D., Emami-Naeini, A. Feedback Control of Dynamic Systems. Prentice Hall New Jersey 2002 – 4th ed. ISBN 0-13-032393-4

[Gvozdiak 1990] Gvozdiak, L., Borsc. M. and Vitko, A. Cybernetics basis (in Slovak).Bratislava: Alfa Publisher, 1990. ISBN 80-05-00677-2

[Hlinovsky 2015] Hlinovsky, M. Subject: Robots (in Czech). Prague: Czech Technical University in Prague [online]. 2010–2015. [May 2016]. Available from<http://www.robosoutez.cz>

[Huba 2006] Huba, M., Hubinsky, P. and Zakova, K. The automated control theory (in Slovak). Bratislava: Slovak technical university, 2006. ISBN 978-80-227-3000-6

[JBT 2016] JBT Automated Guided Vehicle. Special Application Automatic Guided Vehicles [online].2016. [May 2016]. Available from: <http://www.jbtc-agv.com/en/Solutions/Products/Special-Application-Automatic-Guided-Vehicles-AGVs>

[Jurisica 2002] Jurisica, L. and Huba, M. Automatisation introductory. The study introductory (in Slovak). Bratislava: Slovak Technical University, Faculty of electro technique and informatics, 2002

[Nascak 2014] Nascak, L. and Koleda, P. Machines and equipment control systems (in Slovak).Zvolen: Technical university, 2014. ISBN 978-80-228-2667-9

[Pirnik 2015] Pirnik, R., Hrubos, M., Nemec, D., Mravec, T. and Bozek, P., 2015. Integration of inertial sensor data into control of the mobile platform. In: Federated Conference on Software Development and Object Technologies. Zilina

[Sakshat 2016] Sakshat Virtual Labs, Cinematics Experiment [online]. 2016. Available from: <http://iiith.vlab.co.in/index.php?sub=21&brch=72&sim=511&cnt=1>

[Sucik 2014] Sucik, G. and Popovic, L. Measuring, control and regulation (in Slovak). Kosice: Technical university, Faculty of metallurgy, 2014

[Valisek 2007] Valisek, K. and Kostal, P. Mechanisation and automatisation (in Slovak). Bratislava: Slovak Technical University, 2007. ISBN 978-80-227-2753-2

CONTACTS:

doc. Mgr. Elena Pivarciova, PhD.
Ing. Tibor Csongrady, CSc.
Technical University in Zvolen
Faculty of Environmental and Manufacturing Technology
Department of Machinery Control and Automation
Masarykova 24, Zvolen, 960 01, Slovak Republic
tel.: +421 45 5206 477
e-mail: pivarciova@tuzvo.sk, cst.0608@gmail.com