1/2011

atp journal plus

Systems of automatic control

reviewed slovak professional magazine for scientific and engineering issues

Systémy automatického riadenia

recenzované periodikum vedeckých a inžinierskych publikácií

ISSN 1336-5010

Systémy automatického riadenia

Systems of automatic control

Odborný garant

prof.Ing.Ladislav Jurišica,PhD.,
Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky, Ústav riadenia a priemyselnej informatiky. Oddelenie robotiky a umelej inteligencie, Ilkovičova 3, 812 19 Bratislava, Slovensko,
Tel.: +421 2 60 291 351,
e-mail: ladislav.jurisica@stuba.sk

Technical guarantee

prof.Ing.Ladislav Jurišica,PhD.,
Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky, Ústav riadenia a priemyselnej informatiky. Oddelenie robotiky a umelej inteligencie, Ilkovičova 3, 812 19 Bratislava, Slovensko,
Tel.: +421 2 60 291 351,
e-mail: ladislav.jurisica@stuba.sk

Vydavateľ

HMH s.r.o.
Tavarikova osada 39
841 02 Bratislava 42
IČO: 31356273

Publisher

Spoluzakladateľ

Katedra ASR, EF STU
Katedra automatizácie a regulácie, EF STU
Katedra automatizácie, ChtF STU
PPA CONTROLL, a.s.

Co-founder

IEEE Czechoslovakia Section

www.ieee.cz

Redakcia

Editors office

ATP Journal
Košická 37
821 09 Bratislava 2
tel.: 02/5026 1752 – 5
fax: 02/5026 1757
e-mail: vydavatelstvo@hmh.sk
www.atpjournal.sk

Ing. Anton Gérer
šéfredaktor – editor in chief
gerer@hmh.sk

Ing. Martin Karbovanec
vedúci vydavateľstva – editorial office manager
karbovanec@hmh.sk

Bc. Zuzana Bakošová
marketingová manažérka – marketing manager
mediamarketing@hmh.sk

Ing. Branislav Bložon
odborný redaktor – editor
blozon@hmh.sk

Peter Kanda
DTP grafik – DTP graphic designer
dtp@hmh.sk

Dagmar Votavová
asistentka redakcie – editorial assistant
atp_podklady@hmh.sk

Mgr. Bronislava Chocholová
jazyková redaktorka – text corrector

Datum vydania: júl 2011

ISSN 1336-5010
ATP Journal *PLUS* 1/2011

Obsah

TEÓRIA RIADENIA A APLIKÁCIE

RIADENIE ROBOTICKÝCH SYSTÉMOV

SERVISNÁ A MOBILNÁ ROBOTIKA

Content

CONTROL THEORY AND APPLICATIONS

CONTROL OF ROBOTIC SYSTEMS

SERVICE AND MOBILE ROBOTICS

Input and output constraints in model predictive control design

Jozef Škultéty, Eva Miklovičová, Marián Mrosko

Abstract
Model predictive control represents the class of advanced control techniques widely applied in process industries. One of its main advantages is that it allows to incorporate the constraints of process variables into the control design procedure. The aim of this paper is to analyze how the input/output limits influence the control system performance and to show the performance improvement when the model predictive control is used.

Key words: model predictive control design, generalized predictive control, input and output constraints, quadratic programming, real-time control

Introduction

Research in the area of control theory has brought a number of advanced control techniques that can be used to improve the control performances. Model predictive control (MPC) is well known control design approach, which makes explicit use of process model to predict the future process behavior and based on this prediction future control sequence minimizing the objective function is calculated [1]. The MPC popularity in both the research and industrial community is mainly due to its relatively simple time-domain formulation and good performance. As in practice all processes are subject to the input and output constraints, another important feature is that these constraints can be systematically incorporated into the design procedure.

In predictive control design there are several tuning parameters that influence the resulting closed loop performance and stability properties. In case of constrained input/output signals these tuning parameters can be chosen so as the constraints are not exceeded during the standard plant operation which on the other hand can deteriorate the performances inside the operating range. If the constraints are taken into account in the control design procedure, the resulting algorithm is more complex and time consuming.

In the paper the bounds in the amplitude and in the slew rate of the control signal and limits in the output have been considered. Besides the simulations also the real time experiments have been performed. To their realization the Matlab environment together with the programmable logic controller Simatic S7-200 have been used.

The paper is organized as follows. First, standard GPC algorithm is briefly presented. Then its modification taking into account the process variables constraints is described. The properties of both control algorithms are analyzed using the simulation examples and the real-time control of a laboratory plant.

1 Generalized predictive control

Generalized predictive control (GPC) [2] is one of the most popular predictive algorithms based on the parametric input/output process model. It is applicable to the wide class of processes including the instable or inversely instable processes or processes with unknown or variable time delay. The GPC implementation is relatively simple and thanks to several control design parameters the control law can be tuned to specific applications.

1.1 Plant model

Consider that the process behavior can be described by the following CARIMA model

$$A(z^{-1})y(t) = B(z^{-1})u(t-d-1) + v(t) + w(t) \quad (1)$$

$$D(z^{-1})v(t) = C(z^{-1})\xi(t) \quad (2)$$

where *u(t)* is the control signal, *y(t)* is the measured plant output, *d* denotes the minimum plant model time-delay in sampling periods, *v(t)* represents the external disturbances and *ξ(t)* is the random variable with zero mean value and finite variance. Of practical importance $D(z^{-1}) = 1 - z^{-1}$ allows incorporating an integral action into the controller. For simplicity in the following the $C(z^{-1})$ polynomial is chosen to be 1.

1.2 j-step ahead prediction

The GPC control design procedure requires the calculation of the sequence of *j*-step ahead future values of the plant output (predictions) which are composed of the following two parts:

- free response (depends on data available up to the time *t*, namely the past control signal and output values),
- forced response (depends on the future control signal values that can be obtained by the minimization of the GPC control objective),

while the elements that depend on the random disturbance *ξ(t)* are unpredictable and are omitted. The calculation of the *j*-step ahead output predictions necessitates the iterative solutions of the following Diophantine equations for $j = sh, \ldots, ph$

$$1 = A(z^{-1})D(z^{-1})E_j(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (3)$$

$$E_j(z^{-1})B(z^{-1}) = G_{j-d}(z^{-1}) + z^{-j+d}H_{j-d}(z^{-1}) \quad (4)$$

1.3 Standard control design

The control objective consists of minimizing in a receding horizon sense the following cost function

$$J(t, ph, ch, sh, \rho) =$$

$$E\left\{ \sum_{j=sh}^{ph} \left(\hat{y}(t+j/t) - y^*(t+j)\right)^2 + \rho\left(D\left(z^{-1}\right)u(t+j-sh)\right)^2 \right\} \quad (5)$$

subject to

$$D(z^{-1})u(t+i) = 0 \text{ for } ch \le i \le ph \quad (6)$$

sh, *ph* and *ch* are positive scalars defining the starting horizon, prediction horizon and control horizon, ρ is a nonnegative control weighting scalar. $\hat{y}(t+j/t)$ denotes the j-step ahead prediction of *y(t)* based on the data available up to time *t* and $y^*(t+j)$ is the future reference value which is supposed to be known.

The GPC control design consists in performing the following three steps:

- compute the j-step ahead output predictions $\hat{y}(t+j/t)$ for $j = sh, \ldots, ph$,
- minimize with respect to the future control signal sequence the objective function (5) - (6),
- only the first component out of this sequence is used for control

$$D\left(z^{-1}\right)u(t) = -\sum_{j=sh}^{ph} \gamma_j \left(y_0(t+j/t) - y^*(t+j)\right) \quad (7)$$

where the coefficients γ_j depend on the solutions of Diophantine equations (3) – (4).

The whole process is repeated in the next sampling period.

The control law (7) may be implemented using the standard pole-placement control structure (depicted in Fig. 1)

$$S\left(z^{-1}\right)D\left(z^{-1}\right)u(t) + R\left(z^{-1}\right)y(t) = T\left(z^{-1}\right)y^*(t) \quad (8)$$

The polynomials $R(z^{-1})$, $S(z^{-1})$ and $T(z^{-1})$ need to be calculated only once before the control law implementation and for the control signal calculation at the time *t* only the output and control signals available up to the time *t* as well as the reference value $y^*(t)$ are necessary. The orders of $R(z^{-1})$, $S(z^{-1})$ and $T(z^{-1})$ polynomials are given by orders of the plant model numerator and denominator and their coefficients depend on the plant model parameters as well as on the choice of the tuning parameters *sh*, *ph*, *ch* and ρ .

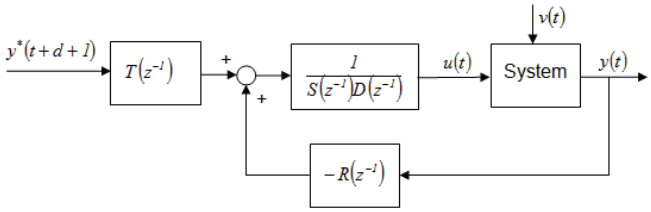


Fig. 1 Control scheme

1.4 Control design taking into account the input/output constraints

In practice all process variables (process output, control signal) are subject to constraints. Constructive, safety or environmental reasons and sensor scopes can cause the limits on the process variables. If these constraints are not taken into account in the control design but they are active during the control implementation, it can lead to deterioration of the control performances or to instability.

For this reason it is reasonable to introduce the constraints into the function to be minimized. Normally, bounds in the amplitude and in the slew rate of the control signal and limits in the output are considered

$$u_{\min} \le u(t+j/t) \le u_{\max} \quad j = 0,1,\ldots,ch-1 \quad (9)$$

$$\left|\Delta u(t+j/t)\right| \le \Delta u_{\max} \quad j = 0,1,\ldots,ch-1 \quad (10)$$

$$y_{\min} \le \hat{y}(t+j/t) \le y_{\max} \quad j = 0,1,\ldots,ph \quad (11)$$

By adding the constraints (9) – (11) to the objective function (5) - (6) the minimization becomes more complex, so the solution can not be obtained explicitly as in the unconstrained case and the control law can not be expressed in the simple RST form (8). The quadratic programming (QP) methods have to be used to solve this optimization problem. The numerical optimization is carried out each sampling instant and the obtained optimal value of *u(t)* is sent to the process. In the following this control design will be denoted as GPC QP.

1.5 Implementation

Implementation of standard GPC control law (8) not considering the input/output constraints in the design procedure is very simple. As it has already been mentioned, the calculation of the polynomials $R(z^{-1})$, $S(z^{-1})$, $T(z^{-1})$ necessitates the recursive solution of Diophantine equations (3) – (4) which can represent a great amount of calculations especially in case of large prediction horizons, but these calculations are performed offline before the control application. The control signal in each sampling instant is computed as a linear combination of past output and control signal values, current output value and the reference value.

GPC QP implementation needs the calculation of the process free response which has been realized using the S-function. The computational effort is much higher comparing to the implementation of the standard constraint-free control law because the solution of the QP optimization problem has to be found each sampling instant.

2 Case study

In order to evaluate the influence of the input/output constraints on the control performance the simulation of simple cylindrical tank depicted in Fig. 2 has been performed. The tank parameters are given in Table 1. The tank is a nonlinear system whose time constant and gain vary considerably throughout the operating range.

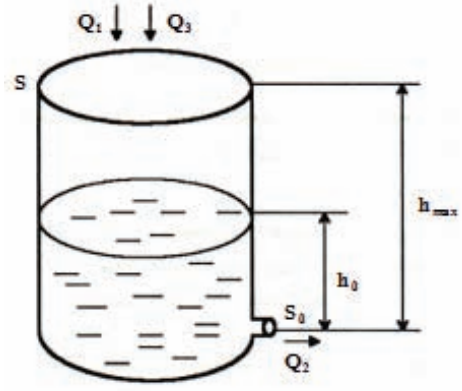


Fig. 2 Cylindrical tank

The controlled variable is the liquid height *h* and the control variable is the inlet flow rate Q_1 with the operating range from *0* to *0.04 m³/s*. The outlet flow rate Q_2 depends on the liquid height according to the Torricelli's law. The control objective is to follow the changes of liquid height reference value as well as to reduce the effect of disturbance represented by the additional inlet flow rate Q_3 subject to the input/output constraints as well as changing plant dynamics caused by the operating point changes.

$S = 1\ m^2$	$S_0 = 100\ cm^2$
$\mu = 0{,}62$	$\rho = 1000\ kg\,m^{-3}$
$g = 9{,}81\ m\,s^{-2}$	$h_{\max} = 2\ m$

Tab. 1 Tank parameters

For the control design purposes the following model of the liquid height dynamics has been identified around an operating point given by $h_0 = 0.5\ m$ with the sampling time $T_s = 10\,s$

$$G_s\left(z^{-1}\right) = \frac{8.772z^{-1}}{-0.8297z^{-1}+1} \qquad (12)$$

2.1 Constraints of the control signal amplitude

In this section, three simulations have been performed to evaluate the influence of the control signal amplitude limits on the closed loop performance.

2.1.1 First simulation – standard GPC

First the standard GPC controller has been designed based on the plant model (12) using the following values of the control design parameters:

- $ph = 3,$
- $ch = 1,$
- $ro = 0.$

Setting the prediction horizon to small value results in fast controller response manifested by large liquid height overshoot. If the calculated control signal – the inlet flow rate Q_1 – exceeds its limits, it is truncated to the corresponding limiting value, which, as a consequence, leads to the worse control performances. If the control law includes an integrator, the calculated control signal will continuously rise although its real implemented value is constant and equal to the limiting value. This behavior is known as the wind-up effect.

The influence of the control signal magnitude constraints on the control performances can be seen in Fig. 3. The additional inlet flow rate $Q_3 = 0{,}017\ m^3\ s^{-1}$ acts as a disturbance from the time 850 s. The blue line shows the simulation results of the unconstrained case (i.e. constraints are not active), while the simulation results where the control signal has been truncated to its limiting values are represented by red line.

2.1.2 Second simulation – standard GPC with modified parameters

To avoid the undesirable effects of the control signal truncation the control tuning parameters have been chosen so as the calculated control signal do not exceed given constraints during the standard operation

- $ph = 15,$
- $ch = 1,$
- $ro = 200.$

The simulation results (blue line) compared to the first simulations results (red line) are in Fig. 4.

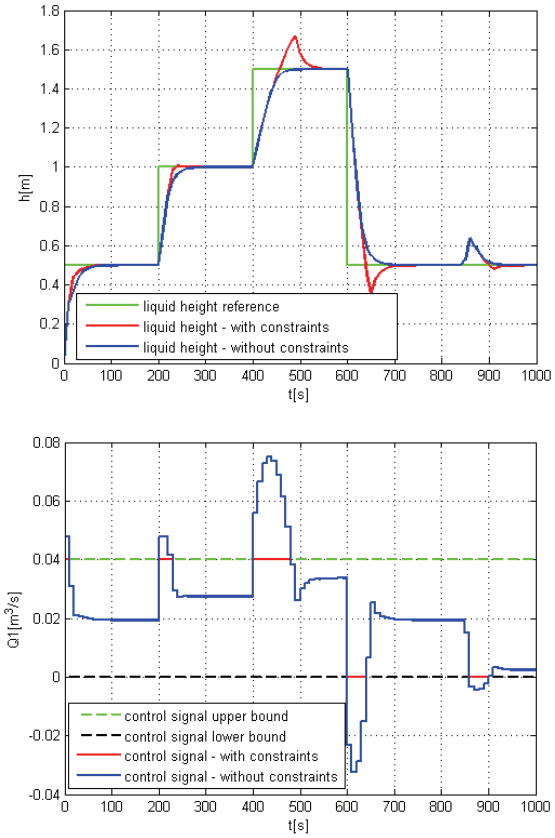


Fig. 3 First simulation results

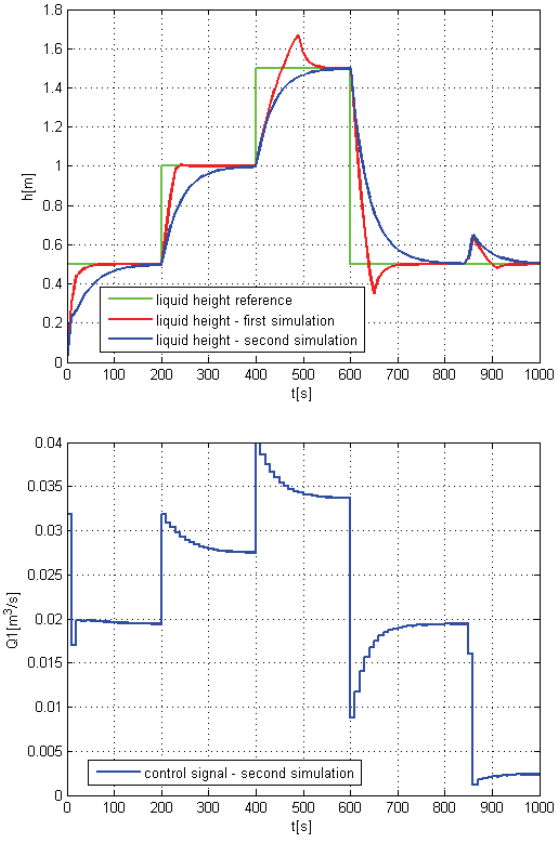


Fig. 4 Second simulation results

It can be seen that in the second simulation the wind-up effect has been eliminated, but the settling time has been much longer (for the liquid height reference value step change from 0.5 m to 1 m it has been about 90 s, while in the first simulation approximately 27 s).

2.1.3 Third simulation – GPC QP

In the third simulation the GPC QP has been used where the following input and output constraints have been taken into account:

- $u_{\min} = 0\, m^3 s^{-1}$,
- $u_{\max} = 0{,}04\, m^3 s^{-1}$,
- $y_{\min} = 0\, m$,
- $y_{\max} = 2\, m$.

$Q_{10} = 0.0194\ m^3 s^{-1}$ is the input flow rate in operating point corresponding to the liquid height 0.5 m. The control signal slew rate limit has not been considered. The incorporation of these bounds into the control design procedure results in such control signal that will not violate its amplitude limits and at the same time the calculated control signal will not give rise to the output outside its operating range.

In the following the GPC QP control performances are compared to those obtained using the standard GPC. In order to achieve the fast controller response the control tuning parameters of both controllers have been set to the same values as in the first simulation in section 2.1.1. The simulation results are in Fig. 5. It can be seen that the GPC QP preserves the fast controller response while respecting given input/output constraints, so the calculated control signal does not need to be truncated.

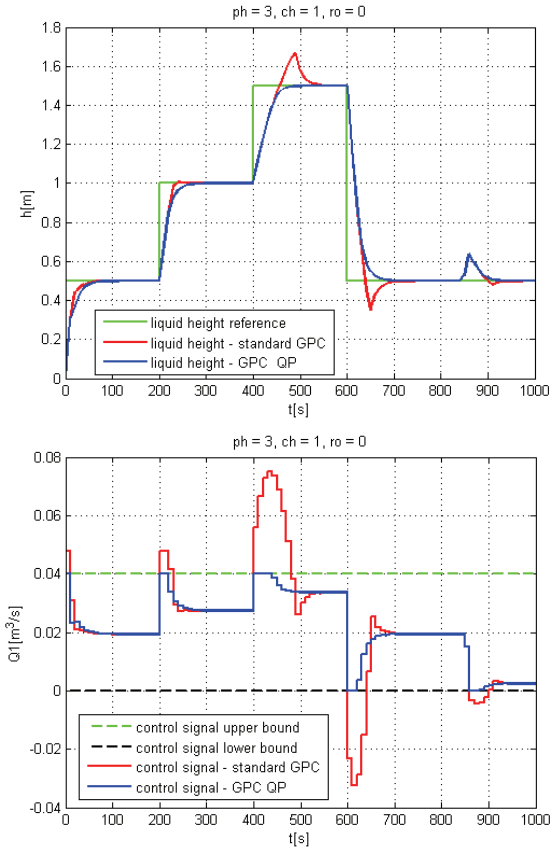


Fig. 5 Third simulation results

For the liquid height reference step change from 0.5 m to 1 m the GPC QP settling time is slightly worse than that of standard GPC. On the other hand, when the reference value changes from 1 m to 1.5 m, the output response using the GPC QP controller is much better (both the settling time and the maximal overshoot) than the output response obtained using the standard GPC controller.

If the control tuning parameters are chosen as in the second simulation, the performances of both standard GPC and GPC QP controllers are identical.

2.2 Constraint of the actuator slew rate

In this section, the influence of the actuator slew rate is investigated. For this purpose a simulation model of the valve has been created.

Two simulations have been performed. For both simulations the input constraints have been set to the following values:

- $u_{\min} = 0\, m^3 s^{-1}$,
- $u_{\max} = 0{,}04 m^3 s^{-1}$,
- $\Delta u_{\max} = 0{,}01 m^3 s^{-1}$.

2.2.1 Fourth simulation – standard GPC

In the fourth simulation the standard GPC controller has been used. The control tuning parameters have been set to the same values as in section 2.1.1.

The obtained simulation results (blue line) compared to the first simulations results (red line) are in Fig. 6. It can be seen that the actuator slew rate limit significantly deteriorates the controller tracking capability.

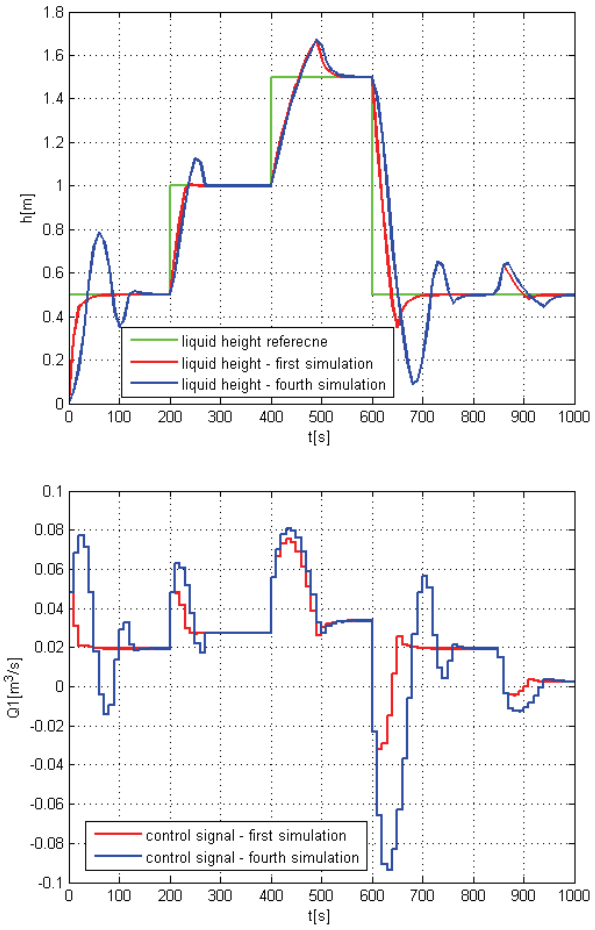


Fig. 6 Fourth simulation results

The comparison of the calculated and actually applied flow rate can be seen in Fig. 7. Note that if the control tuning parameters are chosen as in section 2.1.2, the influence of the actuator slew rate constraint on the control performance is low.

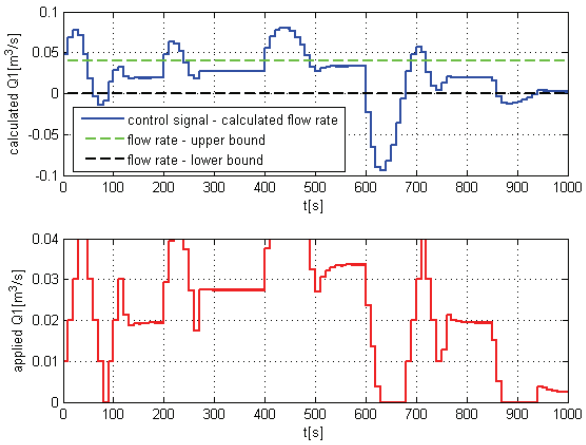


Fig. 7 Fourth simulation results –calculated and applied flow rate

2.2.2 Fifth simulation – GPC QP

In this simulation the GPC QP controller has been used with the same control tuning parameters as in section 2.1.3. Moreover, the maximal possible value of the actuator slew rate has been introduced:

- $\Delta u_{\max} = 0{,}01 m^3 s^{-1}$.

The simulation results (blue line) compared to the third simulations results (red line) are in Fig. 8. It can be seen that if the control signal slew rate constraints are taken into account in the control design procedure then only slight slowing down of the tracking capability occurs.

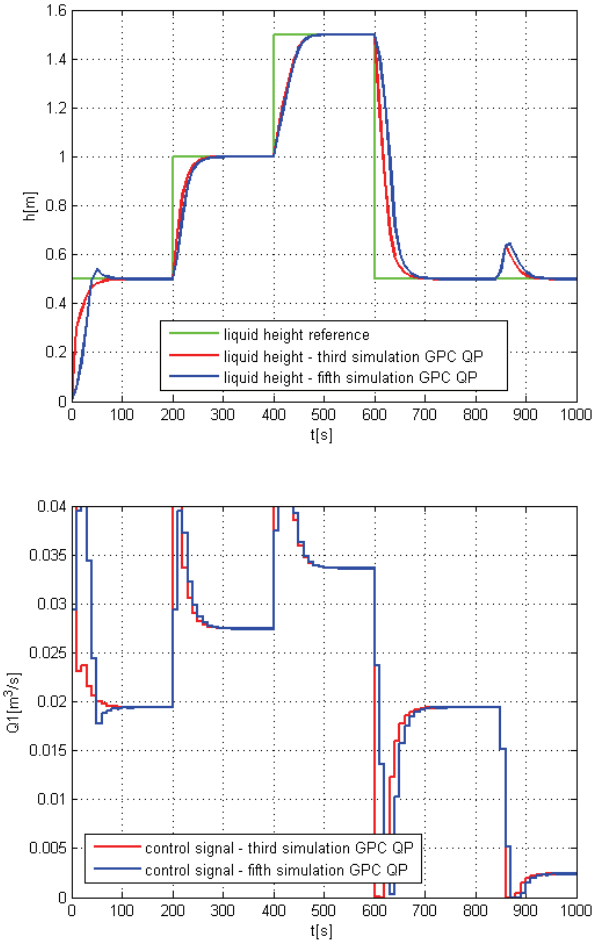


Fig. 8 Fifth simulation results

3 Experimental evaluation

The effectiveness of the GPC QP controller has also been evaluated by real-time control of a cylindrical laboratory tank depicted in Fig.9. The control signal is the inflow servo valve opening and the output is the liquid height measured by a pressure sensor. The outflow servo valve opening has been used to generate a disturbance. The servo valves are governed by voltage 0 – 10 V. The pressure sensor range is also 0 – 10 V.

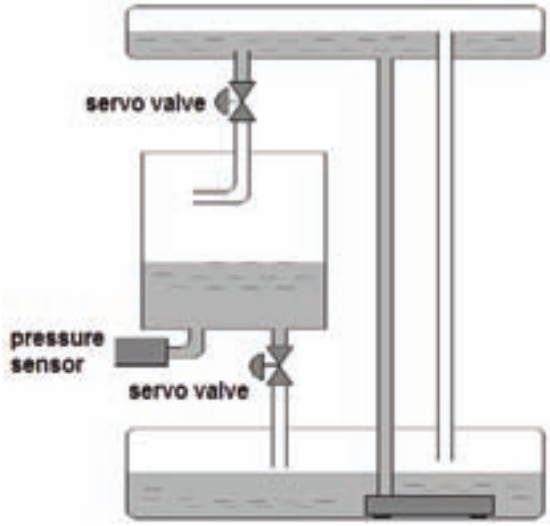


Fig. 9 Cylindrical laboratory tank

The control signal has been calculated in PC and implemented using the programmable logic controller (PLC) Simatic S7-200. For communication between PC and PLC the OPC (OLE for Process Control) communication standard has been used [3]. This standard defines methods for exchanging real-time automation data between PC-based clients using Microsoft operating systems [4]. The control scheme is shown in Fig. 10.

The following model of liquid height dynamics has been identified around an operating point 4.5 V with the sampling time $T_s = 1s$

$$G_R\left(z^{-1}\right) = \frac{-0{,}002254z^{-1} + 0{,}003291z^{-2}}{1 - 1{,}916z^{-1} + 0{,}917z^{-2}} \qquad (13)$$

The control design parameters have been set for both standard GPC and GPC QP controllers to the same values

- $ph = 30,$
- $ch = 2,$
- $ro = 10,$

and the input constraints are the following

- $u_{\min} = 0V$,
- $u_{\max} = 10V$,
- $\Delta u_{\max} = 1V$.

The real-time control results are in Fig. 11. For the liquid height reference step change from 3 V to 5.5 V the overshoot of the liquid level obtained with the GPC QP controller is considerably less then that obtained with the standard GPC. After the next liquid height reference step change to 3 V, when the standard GPC control signal does not violate the limits, the performances of both controllers are equivalent. In order to evaluate the regulation capability the disturbance in the form of the outflow valve closing has been introduced at time 450 s and after 200 s the valve returned to its original position.

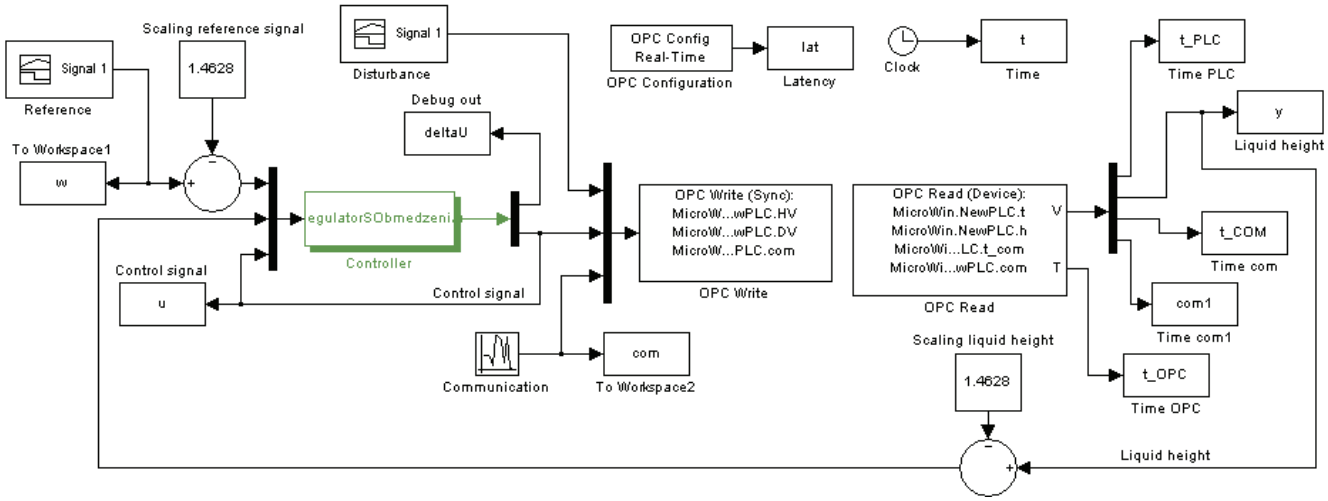


Fig. 10 Control scheme

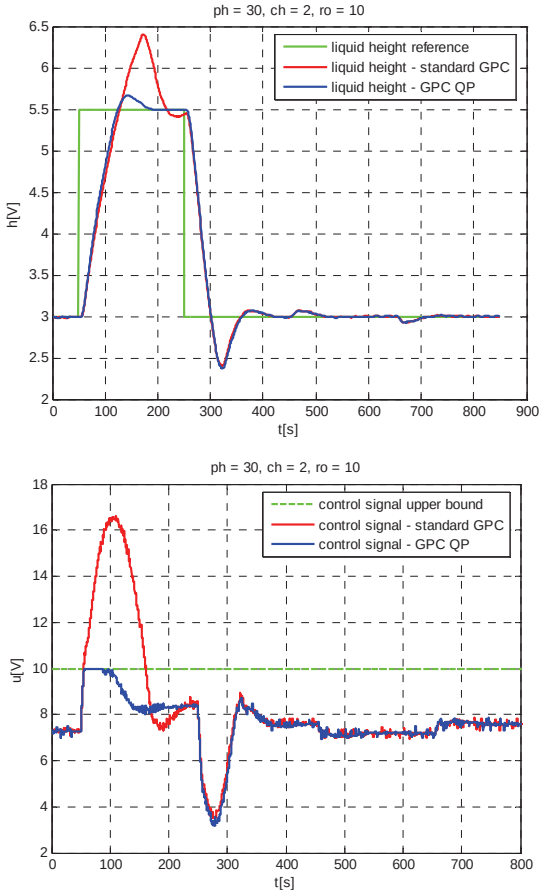


Fig. 11 Real-time control results

Conclusion

Model predictive control represents a family of advanced control methods which are popular in both the research and industrial community. In this paper two generalized predictive control design strategies have been compared: the standard one and the modification taking into account the input/output constraints. It can be concluded from the simulations as well as from the real-time experiments that the incorporation of the process variables limits into the control design procedures leads to improvement of the control system performances although, on the other hand, the implementation of this algorithm is more complex and time-consuming.

Acknowledgement

This work has been supported by the Slovak Scientific Grant Agency, Grant No. 1/0592/10.

References

[1] Camacho E.F., Bordons, C. *Model predictive control*. Springer-Verlag, London, 2004.

[2] Clarke, D.W., Mohtadi, C., Tuffs, P.S. Generalized predictive control – Part I. The basic algorithm. Part II. Extensions and interpretations. Automatica, 23, 137-160, 1987.

[3] Mrosko, M., Mrafko L., Körösi L.: Real time control. 9th International Conference Process Control 2010, Kouty nad Desnou, Czech Republic.

[4] OPC FOUNDATION 1998-2010, Specifications of OPC, Available on: http://www.opcfoundation.org

Ing. Jozef Škultéty

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics (ICII)
Ilkovičova 3
812 19 Bratislava
Phone: +421 2 60291 876
E-mail: jozef.skultety@stuba.sk

doc. Ing. Eva Miklovičová, PhD.

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics (ICII)
Ilkovičova 3
812 19 Bratislava
E-mail: eva.miklovicova@stuba.sk

Ing. Marián Mrosko, PhD.

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics (ICII)
Ilkovičova 3
812 19 Bratislava
E-mail: marian.mrosko@stuba.sk

Robust controller design methods for the laboratory model of DC motor

Mária Hypiusová, Slavomír Kajan

Abstract
The paper deals with comparison of robust controller design methods for uncertain SISO systems. The controller designed using the Kharitonov systems and Bode characteristics guarantees required phase margin, and the controllers designed using the Edge Theorem and Small Gain Theorem guarantee the required degree of stability. Practical application is illustrated by the robust controller design for a DC motor with various parameters.

Keywords: Robust Control, Kharitonov Theorem, Edge Theorem, Small Gain Theorem

Introduction

For many real processes, a controller design has to cope with the effect of uncertainties which very often cause a poor performance or even instability of closed-loop systems. The reason for that is a perpetual time change of parameters (due to aging, influence of environment, working point changes *etc.*), as well as unmodelled dynamics. The former uncertainty type is denoted parametric uncertainty and the latter one dynamic uncertainty. A controller guaranteeing closed-loop stability under both these uncertainty types is called a robust controller. A lot of robust controller design methods are known from the literature [1, 2] in the time-as well as in the frequency domains.

In this paper three approaches to robust controller design have been applied for three working points of a DC motor with varying parameters. The first method is based on the Kharitonov systems and Bode diagram for interval model. This controller guarantees the required phase margin. The second approach is accomplished with the Edge Theorem and the Neymark D-partition method for the affine model. The last controller design method is based on the Small Gain Theorem considering uncertain system model with additive uncertainty. For the second and third methods, the designer can specify a required closed-loop stability degree.

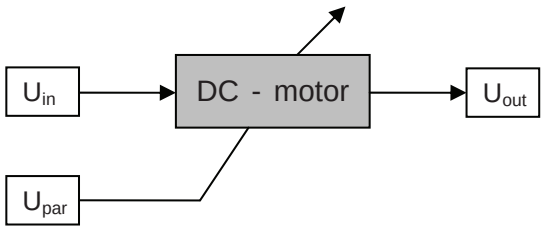
1. The laboratory model of a DC-motor

The laboratory model of a DC-motor in Fig. 1 consists of two co-operating real DC servomotors, where the first one is connected as a motor and the other one as a generator.

In our paper, manipulated variable is the input voltage of DC motor and controlled variable is the speed, however, only output voltage of generator has been considered. The mechanical interconnection is realised through inertia load and spring. Power supply, measurement of signals and motor control are supplied by motor electronics. In electronics, there is a RC component connected to the input of motor to enable changing the time constant and gain of controlled system. System dynamics parameters can be tuned with a potentiometer. The measured voltage value of potentiometer is a variable called U_{par}. The motor input variable U_{In} ranges within the interval 0 – 10 [V]. The motor output variable U_{out} is velocity ranging between 0 – 10 [V].

For interconnection of the controlled process with the computer and software LABREG for identification and control of real process [3], the Advantech data acquisition card of type PCI 1711 is used.

The focus of this paper is to show robust PID controller design to control angular velocity for selected area specified by several working points. For three working points (jump up and jump down) the identified models have been used to design of robust controllers.



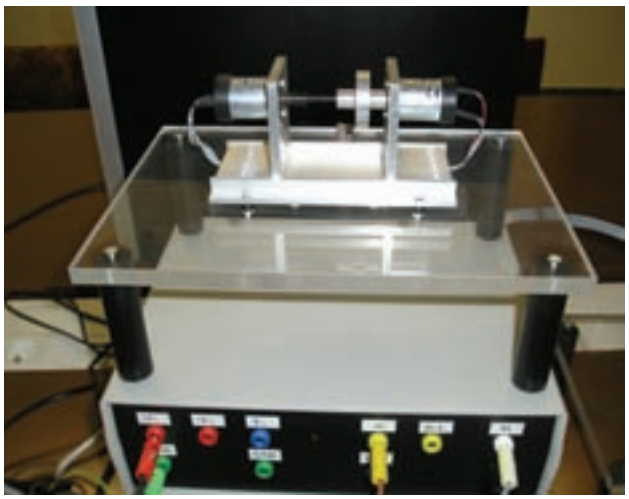


Fig. 1 The laboratory model of a DC motor with principial scheme

2. Robust controller design

2.1 The Kharitonov systems

Consider the controlled system identified in three working points. Comparing coefficients of equal powers of „s" of the three transfer functions we obtain intervals for the coefficients as follows

$$b_i \in \left\langle \underline{b}_i, \overline{b}_i \right\rangle \qquad \text{and} \quad a_j \in \left\langle \underline{a}_j, \overline{a}_j \right\rangle$$

$$\text{for } i = 0,1,m\,;\ j = 1,2,\ldots,n \tag{1}$$

where $a_n \neq 0; b_m \neq 0; m \leq n$.

Hence, interval polynomials of the system numerator and denominator respectively are given

$$\mathbf{N}(s) = \left\{ B(s) : b_0 + \ldots + b_m s^m, b_i \in \left\langle \underline{b}_i, \overline{b}_i \right\rangle, i = 0,1,\ldots,m \right\} \tag{2}$$

$$\mathbf{D}(s) = \left\{ A(s) : a_0 + \ldots + a_n s^n, b_i \in \left\langle \underline{a}_i, \overline{a}_i \right\rangle, i = 0,1,\ldots,n \right\} \tag{3}$$

The interval system is defined in the form

$$\mathbf{G}(s) = \left\{ \frac{B(s)}{A(s)} : (B(s), A(s)) \in (\mathbf{N}(s), \mathbf{D}(s)) \right\} \tag{4}$$

Consider $I(s)$ to be the set of closed-loop characteristic polynomials of degree *n* for the interval systems (4)

$$p(s) = A(s) + B(s) = p_0 + p_1 s + p_2 s^2 + \ldots + p_n s^n \tag{5}$$

where $p_0 \in \left\langle \underline{p}_0, \overline{p}_0 \right\rangle$, $p_1 \in \left\langle \underline{p}_1, \overline{p}_1 \right\rangle$, ..., $p_n \in \left\langle \underline{p}_n, \overline{p}_n \right\rangle$.

Such a set of polynomials is called *interval family* and we refer to $I(s)$ as to interval polynomial. The necessary and sufficient condition for the stability of the entire family is formulated in the Kharitonov Theorem.

Theorem 1 (Kharitonov Theorem)

Every polynomial in the family $I(s)$ is stable if and only if the following four extreme polynomials are stable:

$$K^1(s) = \underline{p}_0 + \underline{p}_1 s + \overline{p}_2 s^2 + \overline{p}_3 s^3 + \underline{p}_4 s^4 + \ldots = p^{--} \tag{6}$$

$$K^2(s) = \underline{p}_0 + \overline{p}_1 s + \overline{p}_2 s^2 + \underline{p}_3 s^3 + \underline{p}_4 s^4 + \ldots = p^{-+}$$

$$K^3(s) = \overline{p}_0 + \underline{p}_1 s + \underline{p}_2 s^2 + \overline{p}_3 s^3 + \overline{p}_4 s^4 + \ldots = p^{+-}$$

$$K^4(s) = \overline{p}_0 + \overline{p}_1 s + \underline{p}_2 s^2 + \underline{p}_3 s^3 + \overline{p}_4 s^4 + \ldots = p^{++}$$

If the coefficients p_i vary dependently, the Kharitonov Theorem is conservative. In such a case we can use the set of Kharitonov systems as follows:

$$\mathbf{G}_K(s) = \left\{ \frac{K_B^i(s)}{K_A^j(s)} : i, j = 1,2,3,4 \right\} \tag{7}$$

where $K_B^i(s)$, i = 1,2,3,4 and $K_A^j(s)$, j = 1,2,3,4 denote the Kharitonov polynomials associated with **N**(s) and **D**(s), respectively.

Theorem 2

The closed loop system containing the interval plant **G**(s) is robustly stable if and only if each of the Kharitonov systems in $\mathbf{G}_K(s)$ is stable.

According to Theorem 2, the controller is to be designed for the „worst" Kharitonov systems (i.e. with the least phase margin). Standard methods can be applied to the controller design, e.g. Bode characteristics to guarantee performance in terms of the required phase margin [4].

2.2 Robust Controller design using the Edge Theorem

If a part of plant coefficients vary dependently, then it is better to use the affine model of the plant in the form:

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_0(s) + \sum_{i=1}^{p} b_i(s) q_i}{a_0(s) + \sum_{i=1}^{p} a_i(s) q_i} \tag{8}$$

where $q_i \in \left\langle \underline{q}_i, \overline{q}_i \right\rangle$ are uncertain coefficients. The coefficients depend linearly on uncertain parameter vector $\mathbf{q}^T = [q_1, \ldots, q_p]$; the parameters q_i vary within a *p* - dimensional box

$$\mathbf{Q} = \left\{ \mathbf{q} : q_i \in \left\langle \underline{q}_i, \overline{q}_i \right\rangle, i = 1,\ldots,p \right\}. \tag{9}$$

If we vary parameters $q_i = \underline{q}_i$ or $q_i = \overline{q}_i$ then is possible to obtain 2^p transfer functions with constant coefficients; inserting them to the vertices of a *p* - dimensional polytope, the transfer function (8) describes a so-called *polytopic system*.

Consider the controller transfer function in the form

$$G_R(s) = \frac{F_1(s)}{F_2(s)} \tag{10}$$

where $F_1(s)$ and $F_2(s)$ are polynomials with constant coefficients. Then the characteristic polynomials with the polytopic system are

$$p(s, \mathbf{q}) = b_0(s) F_1(s) + a_0(s) F_2(s) + \sum_{i=1}^{p} q_i \left[b_i(s) F_1(s) + a_i(s) F_2(s) \right] \tag{11}$$

or in a more general form

$$p(s, \mathbf{q}) = p_0(s) + \sum_{i=1}^{p} q_i p_i(s) \qquad q_i \in \mathbf{Q} \tag{12}$$

Theorem 3 (Edge Theorem)

The polynomial family (12) is stable if and only if the edges of **Q** are stable.

A simple stability analysis method for families of polynomials (edges of **Q**) is given in the following theorem.

Theorem 4 (Bialas)

Let $\boldsymbol{H}_n^{(a)}$ and $\boldsymbol{H}_n^{(b)}$ be the Hurwitz matrices of

$$p_b(s) = p_{b0} + p_{b1}s + p_{b2}s^2 + \ldots + p_{bn}s^n \quad p_{bn} > 0,$$
$$p_a(s) = p_{a0} + p_{a1}s + p_{a2}s^2 + \ldots + p_{an}s^n \quad p_{an} > 0, \tag{13}$$

respectively. The polynomial family

$$p(s,Q) = \{\lambda p_a(s) + (1-\lambda)p_b(s), \quad \lambda \in [0,1]\} \tag{14}$$

is stable if and only if:

1) $p_b(s)$ is stable

2) the matrix $\left(\boldsymbol{H}_n^{(b)}\right)^{-1}\boldsymbol{H}_n^{(a)}$ has no nonpositive real eigenvalues.

Using the Edge Theorem, the controller is designed for the 4 vertices of the polytopic system, e.g. using the Neymark D-partition method that guarantees required closed-loop stability degree. Then, stability of each edge of the box **Q** is checked by e.g. the Bialas Theorem. If any of the edges is unstable, new controller coefficients are to be designed.

2.3 Robust controller design using the Small Gain Theorem

Consider a perturbed plant with unstructured additive uncertainties in the form

$$G_p(s) = G_{nom}(s) + \partial G(s) \tag{15}$$

where $G_{nom}(s)$ is the nominal model and $\partial G(s)$ are additive uncertainties.

The nominal model can be obtained e.g. by *N* identifications of the plant (in *N* working points) by taking mean values of the nominator and denominator coefficients, respectively:

$$G_{nom}(s) = \frac{(B_1(s) + \ldots + B_N(s))/N}{(A_1(s) + \ldots + A_N(s))/N} \tag{16}$$

For each ω the uncertainties are found by substituting $s = j\omega - \alpha$ where α is the required stability degree:

$$\delta G(\omega) = \max \left| G_{nom}(s) - G_{p_i}(s) \right|_{s=j\omega-\alpha} \quad \text{for } i = 1,\ldots,N \tag{17}$$

Theorem 5 (Small Gain Theorem)

Assume that the open-loop system is stable. The closed-loop system is stable if and only if the open-loop magnitude satisfies

$$\left| G_R(j\omega)G_p(j\omega) \right| < 1 \quad \text{for } \omega \in \langle 0, \infty \rangle \tag{18}$$

Theorem 6

Consider an auxiliary characteristic polynomial in the form

$$1 + F_{URO}(s)\frac{\delta G(s)}{G_{nom}(s)} \tag{19}$$

where $F_{URO}(s) = \dfrac{G_{nom}(s)G_R(s)}{1 + G_{nom}(s)G_R(s)}$. (20)

Assume that the open-loop system (nominal model and controller) and the auxiliary characteristic polynomial (19) are stable. Then closed-loop characteristic polynomial $p(s) = 1 + G_p(s)G_R(s)$ with unstructured additive uncertainties (15) is stable if and only if the following condition holds:

$$\left| F_{URO}(j\omega - \alpha) \right| < \frac{1}{\left| \dfrac{\delta G(\omega)}{G_{nom}(j\omega - \alpha)} \right|} = M_0(\omega) \text{ for } \omega \in \langle 0, \infty \rangle \tag{21}$$

Condition (21) is verified graphically. The robust controller design using Small Gain Theorem is realized in the following steps:

1. Specify the closed – loop system magnitude corresponding to the transfer function:

$$W(s) = \frac{G_{nom}(s)G_R(s)}{1 + G_{nom}(s)G_R(s)} \tag{22}$$

If the nominal model is of second order then $W(s) = \dfrac{as+1}{bs+1}$ and $G_R(s) = \dfrac{W(s)}{G_{nom}(s) - W(s)G_{nom}(s)}$ is a PID controller.

2. Choose the numerator of $W(s)$ equal to the numerator of G_{nom}.

3. Choose $b > a$ and design the robust controller so that (21) is satisfied.

3 Design of robust controller for the DC motor laboratory model

Consider the transfer functions of a DC motor obtained by identification in three working points:

WP1: manipulated variable $u = 2.5$ [V];

regulated variable $y = 2.85$ [V]; load $z = 2$ [V]

„jump up“: $G_{p_{1u}}(s) = \dfrac{-0.2464s + 1.605}{s^2 + 2.036s + 1.117}$

„jump down“: $G_{p_{1d}}(s) = \dfrac{-0.2626s + 2.139}{s^2 + 2.882s + 1.551}$

WP2: $u = 5$ [V]; $y = 6.8$ [V]; $z = 2$ [V]

„jump up“: $G_{p_{2u}}(s) = \dfrac{-0.212s + 1.791}{s^2 + 2.087s + 1.217}$

„jump down“: $G_{p_{2d}}(s) = \dfrac{-0.1539s + 1.723}{s^2 + 2.359s + 1.192}$

WP3: $u = 5$ [V]; $y = 5.8$ [V]; $z = 5$ [V]

„jump up“: $G_{p_{3u}}(s) = \dfrac{-0.3675s + 2.543}{s^2 + 2.577s + 1.831}$

„jump down“: $G_{p_{3d}}(s) = \dfrac{-0.2058s + 2.598}{s^2 + 4.496s + 1.885}$

3.1 Robust controller design using Kharitonov systems

The Kharitonov approach uses the interval model:

$$G_p(s) = \frac{B(s)}{A(s)} = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + 1}$$

where $b_0 \in \langle 1.3782,\ 1.4717 \rangle$, $b_1 \in \langle -0.30197,\ -0.10918 \rangle$, $a_1 \in \langle 1.7149,\ 2.3851 \rangle$, $a_2 \in \langle 0.5305,\ 0.89526 \rangle$.

Consider controller design for required phase margin $\Delta\varphi_z = 65[°]$ and the phase added by PI controller $\varphi_r = 30[°]$. We have designed a robust PI controller for the worst Kharitonov system (with the least phase margin) using Bode characteristics in the form

$$R(s) = K\left(1 + \frac{1}{T_i s}\right)$$

where the gain $K = 0.85941$ and the integration time constant $T_i = 2.0758[s]$.

3.2 Robust controller design using the Edge Theorem

The Edge Theorem based approach uses the polytopic model:

$$G_p(s) = \frac{b_0(s) + b_1(s)q_1 + b_2(s)q_2}{a_0(s) + a_1(s)q_1 + a_2(s)q_2}$$

where:

$b_0(s) = -0.28665s + 2.1945$, $a_0(s) = s^2 + 3.2915s + 1.551$,

$b_1(s) = -0.02595s + 0.4965$, $a_1(s) = 1.23s + 0.384$,

$b_2(s) = -0.1068s + 0.093$, $a_2(s) = 0.0255s + 0.05$,

q_i - uncertain coefficients,

required degree of stability: $\alpha = 0.5$

The robust PID controller $G_R(s)$ has been designed in 4 vertices of the polytopic system by Neymark D-partition method:

$$G_R(s) = K\left(1 + \frac{1}{T_i s} + T_d s\right)$$

where the gain $K = 1.3$, the integration time constant $T_i = 1.3[s]$ and the derivative time constant $T_d = 0.30769\ [s]$. For each edge of the box **Q,** stability was verified using the Bialas Theorem - all eigenvalues of the Bialas matrices were not nonpositive real, therefore, the closed-loop with polytopic systems and robust controller is stable and the achieved degree of stability in 4 vertices is $\alpha = 0.5226$.

3.3 Robust controller design using Small Gain Theorem

The Small Gain Theorem based approach uses the uncertain model with additive uncertainties.

The nominal model: $G_{nom} = \frac{-0.2607s + 2.1015}{s^2 + 3.266s + 1.501}$,

uncertainty for the required degree of stability $\alpha = 0.4$ is depicted in Fig. 2.

For the desired transfer function: $W(s) = \frac{-0.12405s + 1}{bs + 1}$, we have designed a robust PID controller with following coefficients:

b = 0.7, $K = 1.886$, $T_i = 2.1758[s]$, $T_d = 0.3062[s]$

Robust stability condition (21) was satisfied (see Fig. 3.).

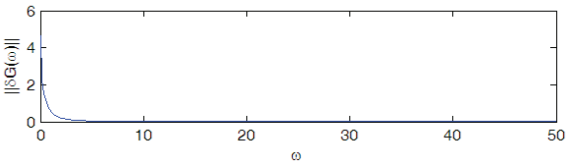


Fig. 2 $\delta G(\omega)$ - versus - ω plot

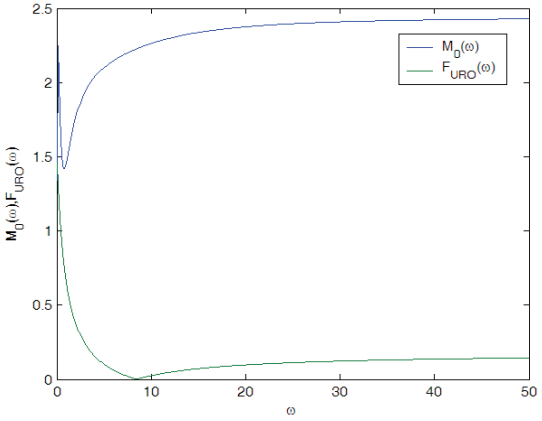


Fig. 3 Robust stability condition for SGT

3.4 Verification of robust controller for the DC motor laboratory model

Figures 4, 5 and 6 show closed-loop step responses in the three working points under designed robust controllers.

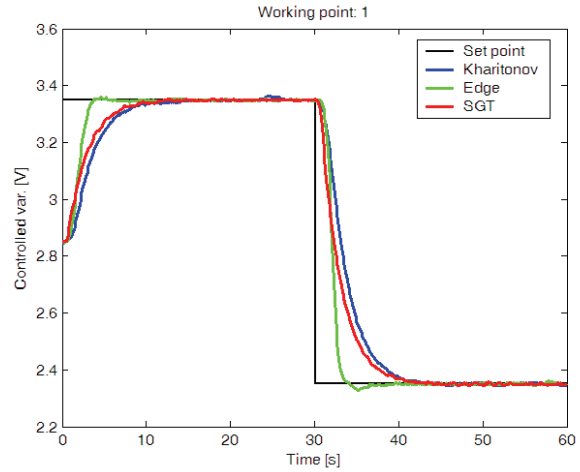


Fig. 4 Step responses in the first working point

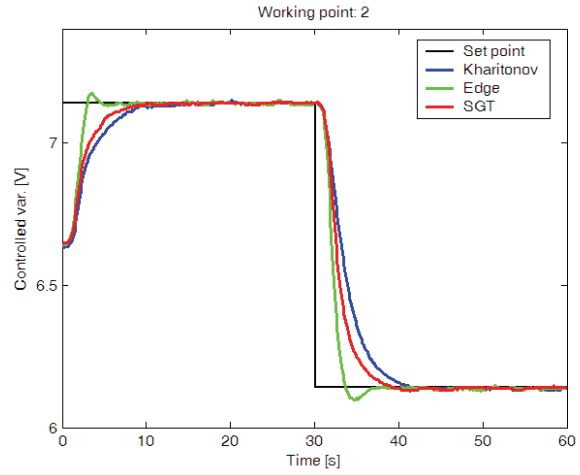


Fig. 5 Step responses in the second working point

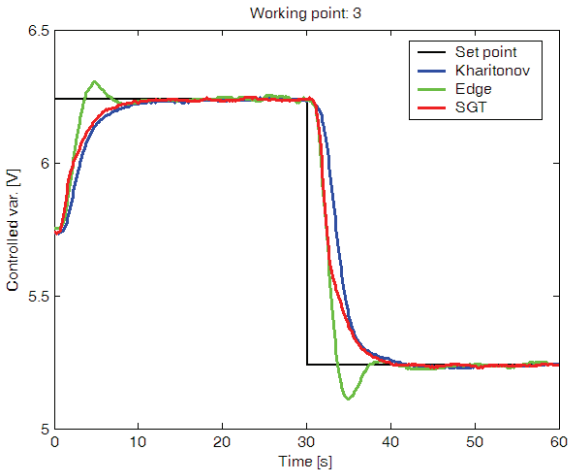


Fig. 6 Step responses in the third working point

Conclusion

The paper deals with application of three robust controller design methods to a laboratory model of a DC motor. The presented methods are based on robust stability analysis results of Kharitonov systems, and guarantee robust stability of the interval system specified by three working points with jump up and jump down. For the considered system, the Edge Theorem based method proved to be the best approach. This method and the Small Gain Theorem based method guarantee the required closed-loop stability degree. The results obtained by verification on DC motor with varying parameters prove the effectiveness of the proposed approaches.

Acknowledgments

The work on this paper has been supported by the VEGA Grant No. 1/0544/09 and No. 1/0690/09.

References

[1] Ackerman, J. 1997. Robust Control - Systems with Uncertain Physical Parameters. *Springer - Verlag London Limited*, 406 s. ISBN 0-387-19843-1, 1997

[2] Bhattacharyya, S. P.; Chapellat, H.; Keel, L. H. 1995. Robust Control: The parametric Approach. *Prentice Hall,* 647 s. ISBN 0-13-781576-X, 1995

[3] Kajan, S., Hypiusová, M.: Labreg Software for Identification and Control of Real Processes in Matlab. *Technical Computing Prague 2007*: 15th Annual Conference Proceedings. Prague, Czech Republic, 14.11.2007. ISBN 978-80-7080-658-6, 2007

[4] Kuo, B.C.: Automatic Control Systems. Sixth Edition. *Prentice-Hall International*, Inc. Englewood Cliffs, New Jersey, 1991

[5] The Mathworks. Matlab ver. 7.1 (R14), user documentation. 2006

Ing. Mária Hypiusová, PhD.

Slovak University of Technology
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Systems and Signals
Ilkovičova 3
812 19 Bratislava
E-mail: maria.hypiusova@stuba.sk,

E-mail: maria.hypiusova@stuba.sk
Ing. Slavomír Kajan, PhD.

Slovak University of Technology
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail: slavomir.kajan@stuba.sk

Robust controller design using genetic algorithm

Slavomir Kajan, Mária Hypiusová

Abstract
This paper deals with the robust controller design using genetic algorithm for uncertain SISO systems. The genetic algorithm represents an optimisation procedure, where the cost function to be minimized comprises the closed-loop simulation of the controlled process and a selected performance index evaluation. Using this approach, the parameters of the PID controller were optimised in order to obtain the required behaviour of the controlled process. The presented approach is illustrated by the robust controller design for a nonlinear system with one uncertain parameter.

Key words: robust controller, genetic algorithm (GA)

Introduction

For control of non-linear dynamic systems with uncertain parameters usually robust or adaptive controllers are used. In this paper, robust controller design approach using genetic algorithm have been applied for nonlinear system with one uncertain parameter, which is considered in several working points. The genetic algorithm represents an optimisation procedure, where the minimization of cost function comprises the closed-loop simulation of the control process and a selected performance index evaluation. Control performance indices corresponding to robust controllers designed for several required closed-loop stability degree are compared in several working points.

1. PID controller design using GA

As mentioned above, the aim of the control design is to provide required static and dynamic behaviour of the controlled process. Usually, this behaviour is represented in terms of the well-known concepts referred in the literature: maximum overshoot, settling time, decay rate, steady state error or various integral performance indices [1, 5, 7].

Without loss of generality let us consider a feedback control loop (closed-loop) (Fig.1), where angular velocity *y* is the regulated variable, input voltage *u* is the manipulated variable, *w* is the reference variable of angular velocity and *e* is the control error (*e*=*w*-*y*).

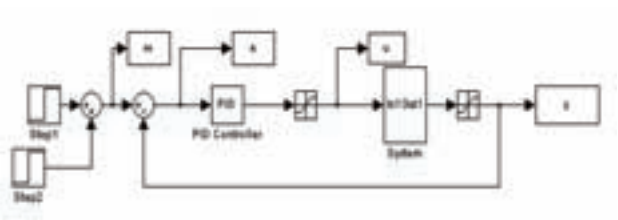


Fig.1 The control loop simulation scheme

The controller design principle is actually an optimization task - search for such controller parameters from the defined parameter space, which minimize the performance index. The cost function (fitness) is a mapping $R^n \rightarrow R$, where *n* is the number of the designed controller parameters. The cost function can be represented by a sum of absolute control errors (SAE) in the following form:

$$J = \sum_{i=1}^{N} |e_i| \quad = \quad \sum_{i=1}^{N} |w_i - y_i| \tag{1}$$

where *w* is reference variable, *y* is controlled output, *e* is control error and *N* is number of samples. Fitness is represented by the cost function or in the case of control, by the modified cost function, which can be penalized for example by derivation of process output *y* or by overshoot of process output or by saturation of control action *u*.

The evaluation of the cost function consists of two steps. The first step is the computer simulation of the closed-loop time-response, and the second one is the performance index evaluation.

Genetic algorithms are described in e.g. [1-8] and others. Each chromosome represents a potential solution, which is a linear string of numbers, whose items (genes) represent in our case the designed controller parameters. Because the controller parameters are real-valued variables and in case of complex problems the number of the searched parameters can be large, real-coded chromosomes have been used.

Without loss of generality let us consider a PID controller with feedforward structure, described in the continuous time domain by the equation (2), where *P*, *I*, *D* are controller parameters and *t* is a time.

$$u(t) = P.e(t) + I.\int_0^t e(t)dt + D.\frac{de(t)}{dt} \tag{2}$$

The searched PID controller parameters are $P \in R^+$, $\boldsymbol{I} \in R^+$, $D \in R^+$. The chromosome representation in this case can be in form ***ch***={*P, I, D*}.

A general scheme of a GA can be described by following steps (Figure 2):

1. Initialisation of the population of chromosomes (set of randomly generated chromosomes).

2. Evaluation of the cost function (fitness) for all chromosomes.

3. Selection of parent chromosomes.

4. Crossover and mutation of the parents → children.

5. Completion of the new population from the new children and selected members of the old population. Jump to the step 2.

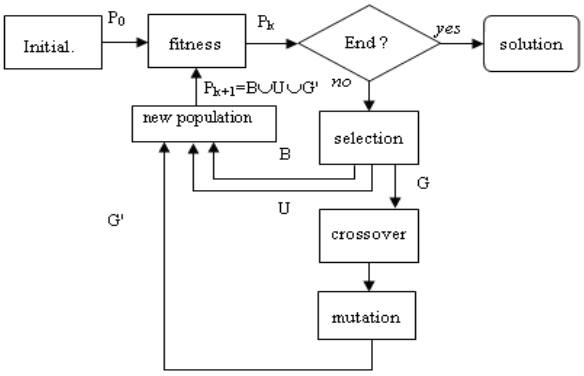


Fig.2 Block diagram of the used genetic algorithm

A block diagram of a GA-based design is in Figure 3. Before each cost function evaluation, the corresponding chromosome (genotype) is decoded into controller parameters of the simulation model (phenotype) and after the simulation the performance index is evaluated.

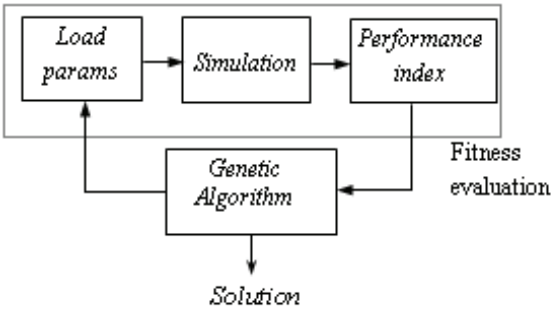


Fig.3 Block diagram of the GA-based controller design

2. Robust controller design using GA

2.1 Design in fixed defined working points

Consider $c = \{c_1, c_2, \ldots, c_q\}$ to be the set of designed controller parameters and let $s = \{s_1, s_2, \ldots, s_r\}$ is the set of parameters of the controlled system. During the operation of the plant, the parameters s_i can vary within some uncertainty domain

$$S : s_{i,\min} \le s_i \le s_{i,\max} \; ; i = 1,2,...,r \qquad (3)$$

where $s_{i,min}$ and $s_{i,max}$ are the minimum and maximum possible values of the i-th system parameter, respectively. Consider W different (physical) working points of the controlled process, defined by different vectors s, which are to be controlled by the robust controller. For that case consider the cost function in the additive form

$$J = \sum_{i=1}^{W} J_i \qquad (4)$$

comprising performance evaluation (for instance (1)) in all W working points. It is also recommended to include the measured noise from the real system or other possible disturbances or expected situations in the simulation model. Note, that alternatively to the set of W defined physical working points we can use a set of $2r$ system parameter vectors located in the vertices of a polytope representing bounds of the parameter space S [3, 7].

2.2 Random generating of working points

Alternative to the previous method, the following method can be considered [8] for the working points selection. In each generation of the GA, n random working points (for all chromosomes of the population the same ones) are generated i.e. n vectors (say $n=100$) of system parameters s become random values from the domain S. For each individual of the population the fitness function is calculated using the performance index (1). In the next generation other n random parameter vectors are generated. In each generation the cost function evaluation is as follows:

1. random generation of n system parameter vectors s (working points),
2. closed-loop simulation and evaluation of the performance index for each individual and each working point,
3. evaluation of the cost function (4) for each individual.

3. Examples of robust PID controller design using genetic algorithm

Testing of performance of selected nonlinear system with the designed robust PID controller has been realized using simulation in Matlab Simulink. For the testing purposes, we have used simulation model of nonlinear dynamic system described by following differential equation:

$$y''' + 10y'' + 5y'y^2 + ay - 0.1au = 0 \qquad (5)$$

where 0,5 < a < 5 is measurable varying parameter, u is input to the system and y is output from the system.

The above nonlinear system has nonlinear static characteristic and dynamics of the system changes according to working point (WP), where the range of system input is from 0 to 25. Step responses of the system for different step changes of system input u, when a = 2.5 are depicted in Fig. 4.

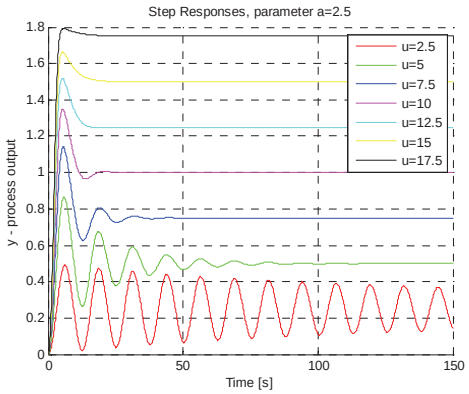


Fig. 4 The step responses of system (5), for a = 2.5

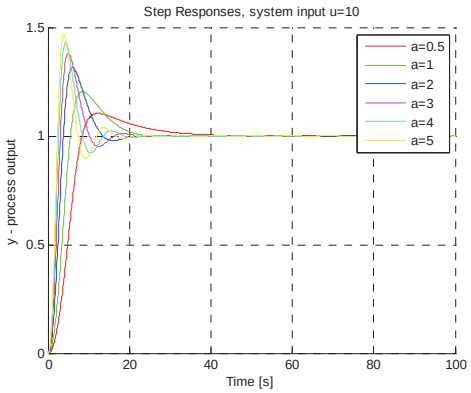


Fig. 5 The step responses of system (5), when u is changing from 0 to 10

Step responses of system (2) for step change of system input *u* from 0 to 10 for different values of parameter *a* are depicted in Fig.5.

In Matlab environment, we used simulation scheme (Fig. 1), which consists of PID controller and model of system (5). For evaluation of the performance index for each individual working point, closed-loop simulation using this scheme was used.

Robust PID controller has been designed using genetic algorithm for several working points of the system: from a = 0.1 to a = 5 and from y = 0.2 to y = 1.8. Range of the search space for controller parameters has been selected as: P = <0; 200>, I = <0; 10> and D = <0; 200>.

For the design of PID controller parameters using GA, two groups with 25 and 100 randomly generated working points (Fig. 7) and two groups of fixed defined working points (Fig. 6) were used:

- 4 extreme working points from defined space of system parameters
- 25 equally placed working points from defined space of system parameters

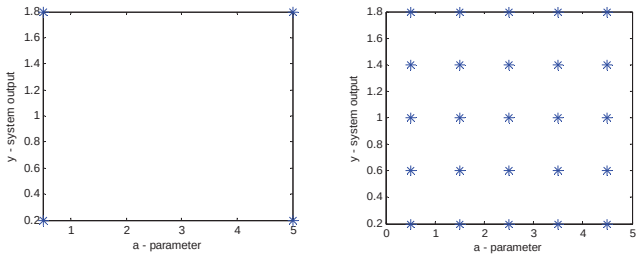


Fig.6 Defined working points of system

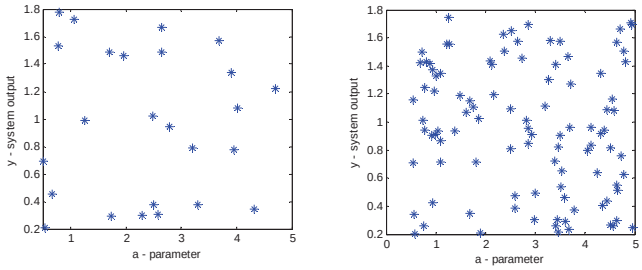


Fig.7 Randomly generated working points of system

Consider that the GA finds the optimal (sub-optimal) solution in the user-defined search space of controller parameters, where these optimal PID controller parameters for several working points (Fig.6 and Fig.7) are shown in Tab.1.
The proposed *Statistical robustness measure* (SRM) quantifies statistical evaluation of a set of closed-loop simulation experiments (in our case 1200) with randomly generated working points from the uncertainty space *S*. The SRM can be expressed by a scalar value calculated as

$$SRM = \frac{1}{N}\sum_{i=1}^{N} J_i \qquad (6)$$

where N is the number of closed-loop simulation experiments and J is a selected performance index (1). Working points for testing the designed PID controllers are displayed in Fig. 8.

Time responses of the controlled variable and reference value for 4 extreme working points of the system for designed PID controllers are shown in Fig. 9, the results for other choices of working points are in Fig.10 and 11.

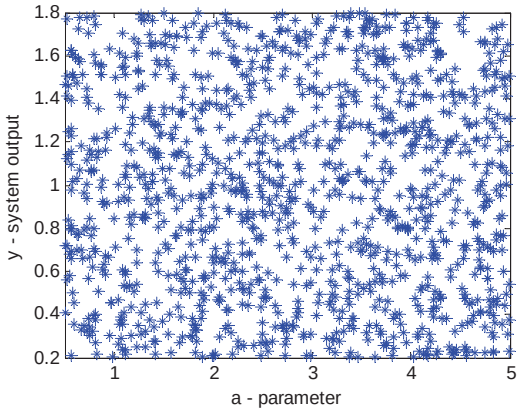


Fig.8 Randomly generated 1200 working points of system for testing control performance

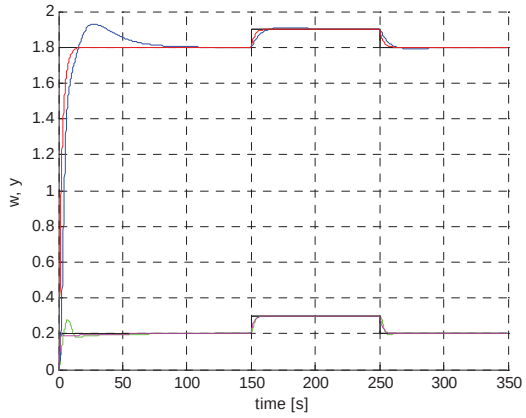


Fig.9 Time responses of the controlled variable and reference value for 4 extreme working points

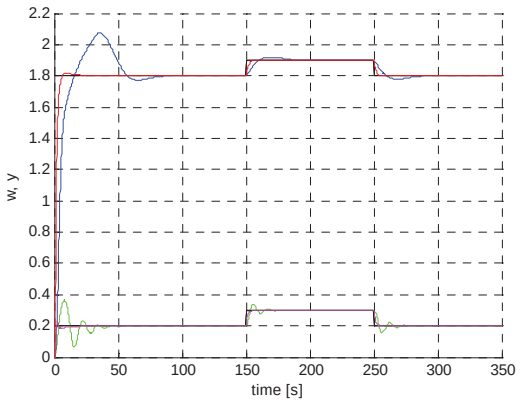


Fig.10 Time responses of the controlled variable and reference value for 100 random working points

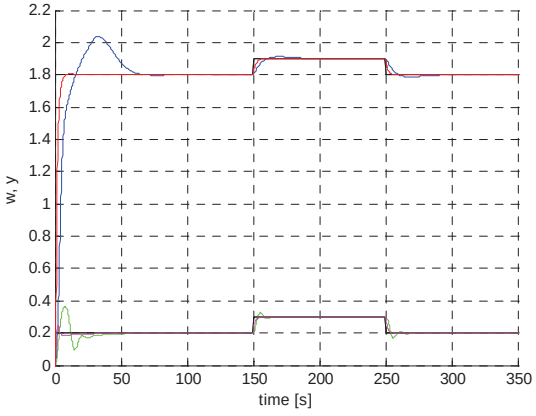


Fig.11 Time responses of the controlled variable and ref. value for 25 equally placed working points

Method	P	I	D	SRM
4 WP	82.6789	3.8416	192.761	0.2970
25 equally placed WP	74.2669	5.8959	98.4345	0.2177
25 random WP	56.8388	6.2724	82.2015	0.2405
100 random WP	55.6419	6.4148	70.4997	0.2310
Inverse dynamic	25.2144	2.5290	0.6322	99.7321

Tab.1 The PID controller parameters designed through several methods

Results for robust controller were compared with those for PID controller designed for nominal parameters: a=2.5 and controlled variable y=1. For the design of PID controller, inverse dynamic method [9] was used. The linearized system model for a=2.5 and y=1 was identified using genetic algorithm, where transfer function of linearized system is

$$G(s) = \frac{0.1003}{3.583\,s^2 + 2.529\,s + 1} \qquad (7)$$

where system poles are $p_{1,2} = -0.3529 \pm 0.3931i$.

For system transfer function (7) PID controller parameters were designed (Tab.1), using inverse dynamic method. The respective time responses of the controlled variable and reference value for 4 extreme working points are depicted in Fig.12. This controller does not guarantee the stability for all working points.

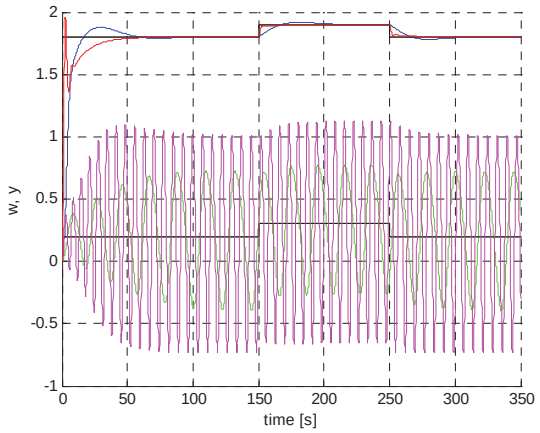


Fig.12 Time responses of the controlled variable and reference value for inverse dynamic method

Conclusion

The main aim of this paper has been to design a robust PID controllers using genetic algorithm for nonlinear system with uncertain parameters. The presented method is based on genetic algorithm, where a fitness is evaluated as a sum of performance indices in all working points. Using genetic algorithm the parameters of the PID controller were optimised in order to achieve the required behaviour of the control process. The results obtained by verification on nonlinear system with various parameters show the effectiveness of the proposed method.

Acknowledgement

The work has been supported by the Slovak Scientific Grant Agency VEGA No. 1/0544/09 and No. 1/0690/09. This support is very gratefully acknowledged.

References

[1] DORF, R.C.: Modern Control Systems. Addison-Wesley publishing Company, 5th edition, 1990
[2] GOLBERG, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addisson-Wesley, 1989
[3] HYPIUSOVÁ, M., KAJAN, S.: Robust Controller Design by Genetic Algorithm. In: Technical Computing Prague 2009 : 17th Annual Conference Proceedings. Prague, Czech Republic, 2009. - Prague : Humusoft, Ltd., - ISBN 978-80-7080-733-0. 2009
[4] MAN, K.F., TANG, K.S., KWONG, S.: Genetic Algorithms, Concepts and Design. Springer, 2001
[5] SEKAJ, I.: Evolučné výpočty a ich využitie v praxi, Iris, Bratilava (in slovak). 2005
[6] SEKAJ, I.: Genetic Algorithm Based Controller Design, In: 2nd IFAC conference Control System Design'03, Bratislava. 2003
[7] SEKAJ, I.: Evolutionary Based Controller Design, Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos, ISBN 978-953-307-008-7, pp. 239-260, October 2009, I-Tech, Vienna, Austria. 2009
[8] SEKAJ, I., ŠRÁMEK, M.: Robust Controller Design Based on Genetic Algorithms and System Simulation In: *Proceedings on the conference CDC-ECC'05*, Seville, Spain, December 2005
[9] Šulc, B., VÍTEČKOVÁ, M.: Teorie a praxe návhu regulačních obvodů (in czech), ISBN80-01-03007-5. 2004

Ing. Slavomír Kajan, PhD.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail: slavomir.kajan@stuba.sk

Ing. Mária Hypiusová, PhD.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Systems and Signals
Ilkovičova 3
812 19 Bratislava
E-mail: maria.hypiusova@stuba.sk

Cluster analysis using Self-Organizing Maps

Slavomir Kajan, Miloš Lajtman, Zuzana Dideková

Abstract
This paper deals with the Kohonen Self-organizing maps for cluster analysis applications. The cluster analysis represents a group of methods whose aim is to classify the objects into clusters. For solving the cluster analysis applications, there have been used many new algorithms using neural networks. This paper deals with the use of an advanced method of neural network represented by Kohonen self-organizing maps for cluster analysis. There have been presented example of one case study for cluster analysis in software Matlab.

Key words: cluster analysis, Self-organizing maps (SOM), Kohonen network

Introduction

The cluster analysis represents a group of methods whose aim is to classify the objects into clusters. This paper deals with the use of an advanced method of neural network represented by Kohonen self-organizing maps for cluster analysis. The basic principle of their function is cluster analysis, i.e. ability of algorithm, network, to find out certain properties and dependencies just in the offered training data without presence of any external information. The idea of just network structure self-organizing has been formed for the first time at the beginning of seventies by von der Malsburg and later has been followed by Willshaw. All of their works from that time are characteristic with orientation to biological knowledge, mainly from the field of neuron and cerebral cortex research [4, 6, 7].

1. Kohonen neural network

The basic idea of Kohonen network emanates from knowledge, that brain uses inner space data representation for storing information. At first, data received from the environment are transformed to vectors which are encoded to the neural network.

The extension of competitive learning rests on the principle that there are several winners permitted. The output from such NN is geometrically organized to the some arrangement, e.g. abreast, or to the rectangle and thus there is a possibility of neighbour identification. This layer is called Kohonen layer. Number of inputs entering to the network is equal to the input space dimension. On the Figure 1, the topology of Kohonen network with 2 inputs is depicted.

The neuron structure in Kohonen network is different from the neuron structure in perceptron network. Number of inputs entering to the neuron is equal to the number of inputs entering to the Kohonen network. The weights of these inputs serve for the encoding of patterns, which represent submitted patterns as well as in the case of perceptron. These neurons don't have actual transfer function. The only operation, which neuron executes is calculation of distance (error) *d* of submitted pattern from pattern encoded in the weights of given neuron according to the equation:

$$d = \sum_{i=0}^{N-1} \left[x_i(t) - w_i(t) \right]^2 \qquad (1)$$

where *N* is a number of submitted patterns, $x_i(t)$ are individual elements of input pattern and $w_i(t)$ are appropriate weights of neuron which represent the encoded patterns [4, 6, 7].

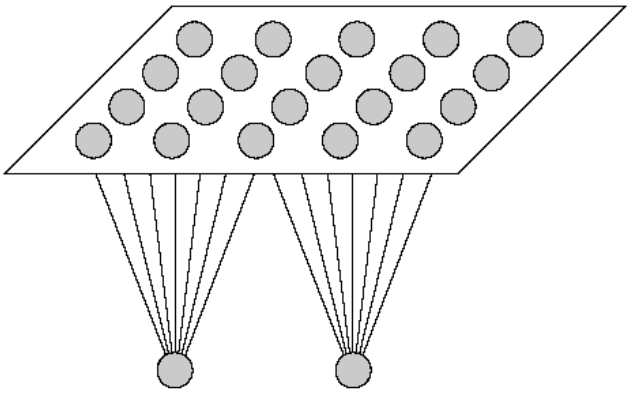


Fig.1 Topology of Kohonen network with two inputs

The learning algorithm tries to arrange the neurons in the grid to the certain areas so that they are able to classify the submitted input data. This organization of neurons can be imagined as unwrapping of primarily wrinkled paper on the plane so that it covers the whole input space.

The process of learning is autonomous, i.e. without the presence of external additional information and it's performed iteratively, i.e. weights are adapted in the each learning step. This adaptation is based on comparison of input patterns and vectors embedded in each neuron (see equation 1). Whenever the vector, which fits input pattern the best is found, it's adapted as well as all the vectors of neurons situated near this neuron. Entire grid is gradually optimized to fit input space of training data the best.

So-called surroundings of neuron take a great role in the learning. The surroundings are defined for each neuron particularly. In initialization phase, the surroundings are usually chosen to cover all the neurons in the grid, i.e. radius of the surroundings is equal to the number of neurons

on the one side of the grid. The surroundings are gradually reduced and also parameter of learning is reduced similarly. The surroundings of neuron are calculated for given so-called winning neuron, whose vector of weights fits the input pattern the best. For this neuron and their surroundings, the weights are adapted. On the Figure 2, a neighbourhood function (Gaussian surroundings) which represents adaptation rate of surroundings weights is depicted. This function has been recognized from the real biological neural networks and it is called neighbourhood function (sometimes also Mexican Hat function). The neighbourhood function (NF) can be described by the equation:

$$\lambda(j^*, j) = h(t).\exp\left(\frac{d_E^2(j^*, j)}{r^2(t)}\right) \quad (2)$$

where $d_E(j^*,j)$ represents Euclidean distance of winner neuron j^* and another compared neuron, $r(t)$ is a radius of neighbourhood function and $h(t)$ is a height of neighbourhood function which decreases to the zero during the time whereby it provides decreasing the surroundings during the learning [4, 6, 7].

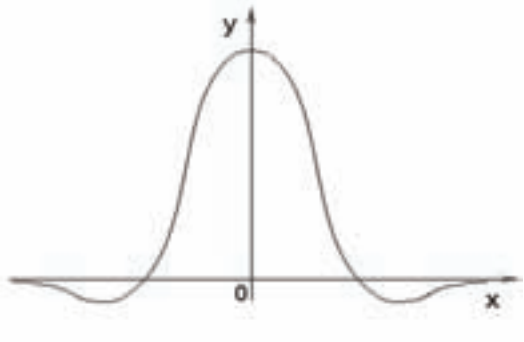


Fig.2 Gaussian neighbourhood function

In the process of learning, the best results are achieved when the size of surroundings r is discreetly decreasing during the time and also it's useful to reduce the size of learning parameter $\eta(t)$ too. The parameter of learning is used to control the learning rate, whose value is from the interval $0 \le \eta(t) \le 1$. At the beginning it is efficient if the parameter is the maximum so that the network of neurons fans out as fast as possible and it covers the largest area of definition range of input vectors. This parameter is gradually decreasing and it allows the network the finer setting of output neurons weights. Method which is available to modify the size of radius surroundings r and also parameter of learning η is shown in the rough on the Figure 3.

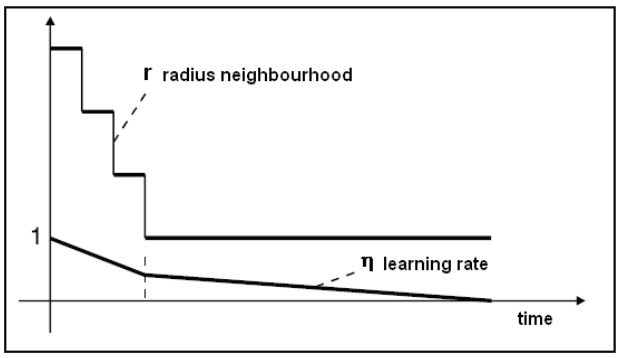


Fig.3 Graph of modification with time functionality for learning algorithm parameters

For the learning process of Kohonen network, let assume a network with N inputs and M neurons which have adjusted weight w_{ij}, then the learning process of Kohonen network can be described by next steps of algorithm:

Step 1: Initialization:
- Adjustment of weights w_{ij}, $0 \le i \le N-1$, $0 \le j \le M-1$, on the small initial values for all the synaptic joints from N inputs to M output neurons.
- Adjustment of learning parameter $\eta(0)$ on the value near the one. Value of the parameter is from interval $0 \le \eta(t) \le 1$ and it's used to control of learning rate.
- Next, the initial size of neighbourhood surroundings of output neurons has to be adjusted. It's also necessary to adjust the minimal value of surroundings, on which the mentioned decreasing of surroundings will be stopped (usually, the minimal surroundings is right the only given neuron around which the surroundings is reduced).

Step 2: Submission of pattern:
- Submission of training pattern $X(t) = \{ x_0(t), x_1(t),..., x_{N-1}(t) \}$ to the inputs of the neural network.

Step 3: Calculation of distances of patterns:
- Calculation of distance (similarity) d_j between the submitted pattern and all the output neurons j according to equation

$$d_j = \sum_{i=0}^{N-1} \left[x_i(t) - w_{ij}(t)\right]^2 \quad (3)$$

where $x_i(t)$ are respective elements of input pattern $X(t)$ and $w_{ij}(t)$ are weights between i-th input and j-th output neuron, which represents encoded patterns.

Step 4: Selection of winning (most similar) neuron:
- Selection of output neuron j^*, which satisfy the next condition and thus it fits the most similar neuron:

$$d_{j^*} = \min(d_j) \quad (4)$$

Step 5: Adaptation of all the weights:
- Calculation of neighbourhood rate from winning neuron according to equation (2).
- Adaptation of weights of neurons according to following equation

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t).\lambda(j^*, j)\left[x_i(t) - w_{ij}(t)\right] \quad (5)$$

Step 6: Continue of learning process:
- Decreasing of learning parameter η and size of neighbourhood surroundings r after the submission of all the patterns.
- End of learning, when the required number of training steps is done or the required accuracy is attained, otherwise continue with the step 2.

After the phase of network learning follows the phase of execution, in which the network responds to the submitted pattern with classifying the pattern to the appropriate class. In the phase of execution only steps 2 to 4 from the learning process are executed, where the winning neuron representing the recognized cluster is received [2, 4, 6, 7].

2. Cluster analysis using Kohonen network

Principle of classification (or cluster analysis) is ability of learning algorithm of Kohonen network to set respective neurons to the clusters of submitted patterns and thus to distribute submitted patterns into the clusters. On the Figure 4, the common scheme of Kohonen network allowing classification of inputs $X=(x_1,x_2,...,x_m)$ into the classes is depicted. For debugging the learning algorithm of Kohonen network, program Matlab has been used on the example of classification of points with coordinates x, y into the clusters. For testing, different numbers of neurons there have been ap-

plied. As an example, there has been used a Kohonen network with 2 inputs and with 9 neurons in the grid 3x3 [1, 3, 5].

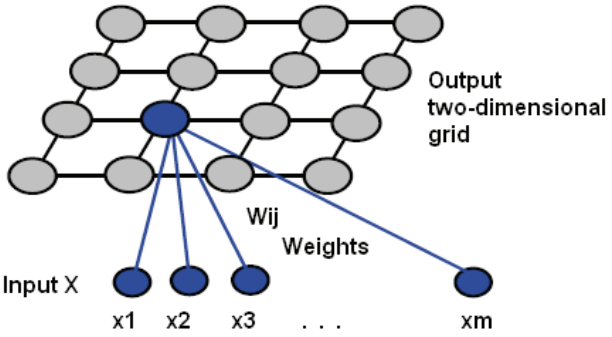


Fig.4 Kohonen network for cluster analysis

The aim of experiments was to set the initial parameters: learning step η_0, size of neighbourhood surroundings r_0, height of neighbourhood function h_0, number of learning cycles *num_cycles*, so that the learning algorithm described in the previous chapter performs for the best. For decreasing of learning step η, radius of neighbourhood function r and height of neighbourhood function h in dependence on number of learning cycle, the following exponential dependencies have been used:

$$\eta(t) = \exp(\eta_0 / cycle) - 1 \quad (6)$$

$$r(t) = r_0 . \exp(-cycle) \quad (7)$$

$$h(t) = \exp((1 - h_0)/cycle) \quad (8)$$

On the Figure 5, there are shown training points with optimal distribution of neurons of Kohonen network. Optimal setting of parameters of learning algorithm is noted in the Table 1. There are shown courses of decreasing of learning step η, size of neighbourhood surroundings r and height of neighbourhood function h on the Figure 6. Dependencies of efficiency of cluster analysis on the choice of initial parameters of learning algorithm are listed in the Tables 2 until 5.

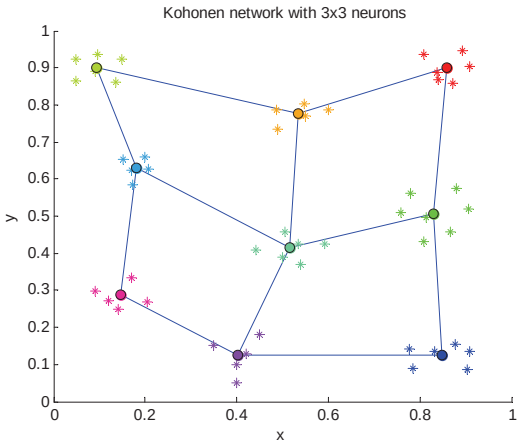


Fig.5 Result of cluster analysis using Kohonen network

Number of cycles	Learning rate η_0	Height NF h_0	Radius NF r_0
8	0.7	1.2	10

Tab.1 Optimal initial parameters of learning algorithm

Number of cycles	2	4	6	8	10	12	14	16
Efficiency [%]	1	87	89	88	88	87	89	90

Tab.2 Relation of efficiency from number of cycles

Learning rate η_0	0.1	0.3	0.5	0.7	0.9	1.5
Efficiency [%]	10	93	90	95	87	77

Tab.3 Relation of efficiency from learning rate

Height NF h_0	0.2	0.4	0.6	0.8	1	1.2	1.4
Efficiency [%]	81	88	87	86	90	90	87

Tab.4 Relation of efficiency from height of NF

Radius NF r_0	1	2	4	6	8	10	12
Efficiency [%]	2	63	72	82	84	89	89

Tab.5 Relation of efficiency from radius of NF

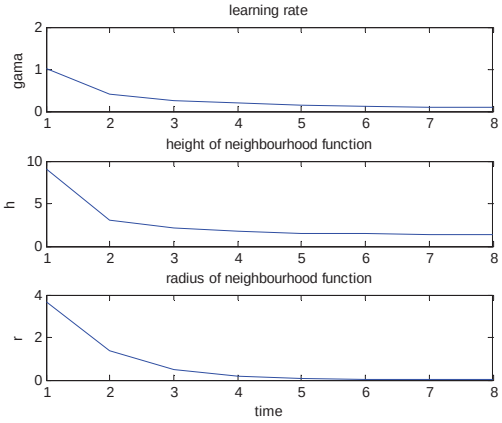


Fig.6 Charts of algorithm parameters functions

2. Practical example for cluster analysis

The aim was to apply Kohonen algorithm to the practical problem. Designed application solves the problem of business distributor, which needs to arrange their distributional centres in a such way that the distance from each centre to the surrounding customers is minimal and then the total freightage of transported goods will be minimal too. For this application, there has been created graphical environment in the program MATLAB and for demonstration there has been used a map of Slovakia. Neurons in neural network serve as distributional centres and input data are used as representation of customer location on the map. The task of algorithm is thus optimisation of location of distributional centres. But this program don't regard the road system, not even level of traffic or other factors, the algorithm thus optimises only location of distributional centres considering the distance from customers. On the Figure 7, there is shown graphical environment of the program. On the Figures 8 and 9, the results of optimisation of distributional centres location are depicted.

Fig.7 Program for distributive centres optimisation using Kohonen network

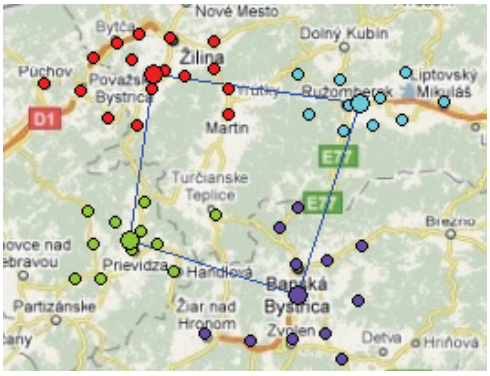


Fig.8 Result of distributive centres optimisation

Fig.9 Result of distributive centres optimisation

Conclusion

The main objective of this article was to demonstrate very good properties of Kohonen network for cluster analysis. The created program for cluster analysis using Kohonen network has been used for setting of initial parameters of learning algorithm. Due to good properties, the Kohonen network can solve classification problems very effectively. The demo program in Matlab for distributive centres optimisation presents only one practical application from many uses of Kohonen network.

Acknowledgement

The work has been supported by the grants agency VEGA no. 1/0544/09 and no. 1/0690/09. This support is very gratefully acknowledged.

References

[1] DEMUTH, H., BEALE, M.: Neural Network Toolbox, For use with Matlab, User's guide, 2003

[2] DOSTÁL, P., POKORNÝ, P.: Cluster analysis and neural network. In: Technical Computing Prague 2009 : 17th Annual Conference Proceedings. Prague, Czech Republic, 2008.

[3] GORGONIO, F., COSTA, J.: Privacy-Preserving Clustering on Distributed Databases: A Review and Some Contributions, In: book Self Organizing Maps - Applications and Novel Algorithm Design, Published by InTech, Rijeka, Croatia, january 2011, p. 33, ISBN 978-953-307-546-4

[4] KVASNIČKA, V. a kol.: Úvod do teórie neurónových sietí. Bratislava: IRIS, 1997. 142 s. ISBN 80-88778-30-1

[5] MEZA, L., NETO, L.: Modelling with Self-Organising Maps and Data Envelopment Analysis: A Case Study in Educational Evaluation, In: book Self Organizing Maps - Applications and Novel Algorithm Design, Published by InTech, Rijeka, Croatia, january 2011, p. 33, ISBN 978-953-307-546-4

[6] SINČÁK, P., ANDREJKOVÁ, G.: Neurónové siete Inžiniersky prístup (1. diel). Košice: ELFA, 1996. ISBN 80-88786-38-X

[7] ŠNOREK, M., JIŘINA, M.: Neuronové sítě a neuropočítače. 1. vyd. Praha: ČVUT, 1996. 39 s. ISBN 80-01-01455-X

Ing. Slavomír Kajan, PhD.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail: slavomir.kajan@stuba.sk

Miloš Lajtman

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Applied Informatics and Information Technology
Ilkovičova 3
812 19 Bratislava
E-mail: lajtman@gmail.com

Ing. Zuzana Dideková

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail: zuzana.didekova@stuba.sk

Hybrid Predictive Controller for Nonlinear Process

Jana Paulusová, Mária Dúbravská

Abstract
In this paper hybrid predictive controller for nonlinear process is addressed, proposed and tested. Model predictive controllers are based on dynamic model of the process. The proposed hybrid fuzzy convolution model consists of a steady-state fuzzy model and a gain independent impulse response model. The proposed model is tested in model based predictive control of the concentration in the chemical reactor, manipulating its flow rate. The paper deals with theoretical and practical methodology, offering approach for intelligent neuro-fuzzy control design and its successful application.

Keywords: hybrid models, neuro-fuzzy models, predictive control

Introduction

Predictive control has become popular over the past twenty years as a powerful tool in feedback control for solving many problems for which other control approaches have been proved to be inefficient. Predictive control is a control strategy that is based on the prediction of the plant output over the extended horizon in the future, which enables the controller to predict future changes of the measurement signal and to base control actions on the prediction.

Model-based predictive control (MPC) is a particular class of optimal controller [4]. The first main advantage of MPC is that constraints (due to: manipulated variables physical limitations, operating procedures or safety reasons) may be explicitly specified into this formulation. The second main advantage of MPC is its ability to be used for both simple and complex model based processes (time delays, inverse responses, significant nonlinearities, multivariable interaction, modelling uncertainties).

Advanced predictive control techniques use neural networks and fuzzy logic for modeling and control [3]. The neuro-fuzzy approaches used in model-based predictive control are efficient tool for handling plants with complex dynamics as well as unstable inverse systems, time varying time delays, plant model miss-matches and other various uncertainties of complex nonlinear systems. The strength of neuro-fuzzy approaches is in using fuzzy knowledge based systems in combination with learning abilities of neural networks allowing to obtain higher accuracy of the required output in a shorter time in comparison with classical systems because of parallel computing algorithm application. By including of the fuzzy and neuro techniques into predictive algorithms complex system control problems can be solved effectively [6].

The aim of this paper is to develop a hybrid neuro-fuzzy predictive control modelling technique for nonlinear process with constrained input. Firstly, the hybrid neuro-fuzzy convolution model is proposed. The next part presents predictive control algorithm with neuro-fuzzy model. Then an application example is presented. The last section offers the relevant conclusions.

1. Neuro-Fuzzy Convolution Model (NFCM)

The output of the model can be formulated as [1]

$$y_m(k+1) = y_s + K(u_s, x_2, \ldots, x_n) \cdot \sum_{i=1}^{N} g_i(x_2, \ldots, x_n) \cdot (u(k-i+1) - u_s) \quad (1)$$

where the first part represents steady-state part, which is described by neuro-fuzzy model (ANFIS), the second part is dynamic part of model (the impulse response model). The gain independent impulse response model is $g_i(x_2,\ldots,x_n)$, the previous input values are $u(k\text{-}i\text{-}1)$ over N horizon, K is steady-state gain, u_s and y_s are steady-state input and output, $x_2,\ldots,\ x_n$ are other operating parameters having effects on the steady-state output.

The convolution is multiplied by steady-state gain

$$K = \frac{\partial f(u_s, x_2, \ldots, x_n)}{\partial u_s} \quad (2)$$

1.1 The steady-state part

The steady state part is described by neuro-fuzzy model. A nonlinear discrete system can be expressed by neuro-fuzzy model (ANFIS) with n rules. The i-th rule of the model is described as follows [5]:

$$R^i : if \ \ x_1 \ is \ A_{1,i} \ and \ \ldots \ and \ \ x_n \ is \ A_{n,i} \ then \ \ y_s = d_i \quad (3)$$

where n is the number of inputs, $x=[x_1,\ldots,x_n]^T$ is a vector of inputs of the model, $A_{j,i}(x_j)$ is the $i=1,2,\ldots,M_j$ –th antecedent fuzzy set referring to the j-th input, where M_j is the number of the fuzzy set on the j-th input domain.
The first element of the input vector is the steady-state input $x_1=u_s$.
The output is computed as weighted average of the individual rules consequents

$$y_s = \frac{\sum_{i=1}^{n} \mu_i d_i}{\sum_{i=1}^{n} \mu_i} \quad (4)$$

where the weights $0 \le \mu_i \le 1$ are computed as $\mu_i = \prod_{j=1}^{n} A_{ji}(x_j)$, where Π is fuzzy operator, usually been applied as the *min* or the *product* operator and *n* is number of rules.

Triangular membership functions were examined in our research as shown in Fig. 1.

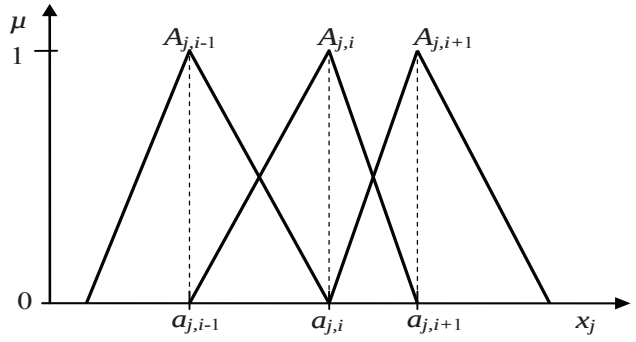


Fig.1 Membership functions used for the fuzzy model

The triangular membership functions are defined as follows:

$$A_{j,i}(x_j) = \frac{x_j - a_{j,i-1}}{a_{j,i} - a_{j,i-1}}, \qquad a_{j,i-1} \le x_j < a_{j,i}$$
$$A_{j,i}(x_j) = \frac{a_{j,i+1} - x_j}{a_{j,i+1} - a_{j,i}}, \qquad a_{j,i} \le x_j < a_{j,i+1} \tag{5}$$

where $x_j \in (a_{j,m_j}, a_{j,m_j+1}\rangle$.

The gain of the steady-state fuzzy model can be computed as

$$K_j = \frac{\partial y_s}{\partial u_s} = \sum_{i=m_j}^{m_j+1} \left[\left(\frac{\Gamma_{i-1}(u_s)}{a_{1,i} - a_{1,i-1}} - \frac{\Gamma_i(u_s)}{a_{1,i+1} - a_{1,i}} \right) \prod_{j=2}^{n} A_{j,i}(x_j) d_i \right], \tag{6}$$

$$K = \sum_{j=1}^{n} K_j$$

where

$$\Gamma_i = 1 \quad if \quad u_s \in (a_{1,i}, a_{1,i+1}\rangle$$
$$\Gamma_i = 0 \quad if \quad u_s \notin (a_{1,i}, a_{1,i+1}\rangle \tag{7}$$

1.2 The dynamic part

The dynamic part is described by the impulse response model (IRM). Parameters of the discrete IRM g_i (i=0,…, N, where N is the model horizon) can be easily calculated from the input-output data (u_i and y_i) of the process.

$$y(k) = \sum_{i=0}^{N} g_i u(k-i) \tag{8}$$

In matrix form

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} u_0 & 0 & \cdots & 0 & \cdots \\ u_1 & u_0 & \cdots & 0 & \cdots \\ \vdots & \vdots & & & \vdots \\ u_N & u_{N-1} & \cdots & u_0 & \cdots \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \end{bmatrix}$$

The parameters are given as follows

$$g = \left(U^T U\right)^{-1} U^T y \tag{9}$$

2 Hybrid Neuro-Fuzzy Model Based Predictive Controller

The nonlinear NFCM can be easily applied in model based predictive control scheme.
In most cases, the difference between system outputs and reference trajectory is used in combination with a cost function on the control effort. A general objective function is the following quadratic form [5]

$$J = \sum_{i=1}^{p} [\hat{y}(k+i \mid k) - r(k+i)]^2 \Gamma_y + \sum_{i=1}^{m} (\Delta u(k+i-1))^2 \Gamma_u \tag{10}$$

Here, r is desired set point, Γ_u ($\Gamma_u = \gamma K^2$) and Γ_y are weight parameters, determining relative importance of different terms in the cost function, u and Δu are the control signal and its increment, respectively. Parameter p represents length of the prediction horizon, m is the length of the control horizon. Output predicted by the nonlinear fuzzy model is $\hat{y}(k)$.

$$\hat{y}(k) = K \sum_{i=1}^{\infty} s_i \Delta u(k-i) \tag{11}$$

where $s_i = \sum_{j=1}^{i} g_j$ are the step response coefficients and the change on the control variable is $\Delta u(k) = u(k) - u(k-1)$.

Model predictions along the prediction horizon p are

$$\hat{y}(k+j \mid k) = K \sum_{i=1}^{\infty} s_i \Delta u(k+j-i) + e(k+j \mid k) \tag{12}$$

Disturbances are considered to be constant between sample instants

$$e(k+j \mid k) = y(k \mid k) - K \sum_{i=1}^{\infty} s_i \Delta u(k+j-i) \tag{13}$$

where $y(k \mid k)$ represents the measured value of the process output at time k.

So

$$\hat{y}(k+j \mid k) = K \sum_{i=1}^{N} s_i \Delta u(k+j-i) + f(k+j \mid k) \tag{14}$$

where

$$f(k+j \mid k) = y(k \mid k) + K \sum_{i=1}^{N} (s_{k+1} - s_i) \Delta u(k-i) \tag{15}$$

Prediction of the process output along the length of the prediction horizon, can be written compactly using matrix notation

$$\hat{y}(k) = KS\Delta u(k) + f(k) \tag{16}$$

Matrix S is called the system's dynamic matrix (17) [4]

$$S = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ s_2 & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_m & s_{m-1} & \cdots & s_1 \\ \vdots & \vdots & \ddots & \vdots \\ s_p & s_{p-1} & \cdots & s_{p-m+1} \end{bmatrix}_{p \times m} \tag{17}$$

By minimizing its objective function (10) the optimal solution is given by (18), if there are any constraints on variables.

$$\Delta u(k) = \frac{1}{K}(S^T \Gamma_y S + \gamma I)^{-1} S^T \Gamma_y e(k) \tag{18}$$

2.1 Quadratic Programming (QP) Solution

In many control applications the desired performance cannot be expressed solely as a trajectory following problem. Many practical requirements are more naturally expressed as constraints on process variables such as manipulated variable constraints, manipulated variable rate constraints or output variable constraints.

The solution calls into existence of quadratic programming solution of the control problem.

This paper deals with manipulated variable constraints and manipulated variable rate constraints.
The resulting vector of control contains not only the current moves to be implemented but also the moves for the future m intervals of time. Although violations can be avoided by constraining only the move to be implemented, constraints on future moves can be used to allow the algorithm to anticipate and prevent future violations thus producing a better overall response. The manipulated variable value at a future time $k+l$ is constrained to be

$$u_{\min}(l) \le \sum_{j=0}^{l} \Delta u(k+j \mid k) + u(k-1) \le u_{\max}(l); l = 0,1,\ldots,m-1 \tag{19}$$

where $u(k\text{-}1)$ is the implemented previous value of the manipulated variable. For generality, we allowed the limits $u_{\min}(l)$,; $u_{\max}(l)$ to vary over the horizon.
These constraints are expressed in matrix form

$$\begin{bmatrix} -I_L \\ I_L \end{bmatrix} \Delta U(k) \ge \begin{bmatrix} u(k-1) - u_{\max}(0) \\ \vdots \\ u(k-1) - u_{\max}(m-1) \\ u_{\min}(0) - u(k-1) \\ \vdots \\ u_{\min}(m-1) - u(k-1) \end{bmatrix} \tag{20}$$

where

$$I_L = \begin{bmatrix} I & 0 & \ldots & 0 \\ I & I & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \ldots & I \end{bmatrix} \tag{21}$$

Often predictive control is used in a supervisory mode where there are limitations on the rate at which lower level controller setpoints are moved. These are enforced by adding constraints on the manipulated variable move sizes:

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U(k) \ge \begin{bmatrix} -\Delta u_{\max}(0) \\ \vdots \\ -\Delta u_{\max}(m-1) \\ -\Delta u_{\max}(0) \\ \vdots \\ -\Delta u_{\max}(m-1) \end{bmatrix}; \quad \Delta u_{\max}(l) > 0 \tag{22}$$

The manipulated variable constraints (20) and manipulated variable rate constraints (22) can be combined into one expression

$$C^u \Delta U \ge C(k+1 \mid k) \tag{23}$$

where

$$C^u = \begin{bmatrix} -I_L \\ I_L \\ -I \\ I \end{bmatrix}$$

$$C(k+1 \mid k) = \begin{bmatrix} u(k-1) - u_{\max}(0) \\ \vdots \\ u(k-1) - u_{\max}(m-1) \\ u_{\min}(0) - u(k-1) \\ \vdots \\ u_{\min}(m-1) - u(k-1) \\ -\Delta u_{\max}(0) \\ \vdots \\ -\Delta u_{\max}(m-1) \\ -\Delta u_{\max}(0) \\ \vdots \\ -\Delta u_{\max}(m-1) \end{bmatrix} \tag{24}$$

By converting to the standard QP formulation the control problem becomes:

$$\min_{\Delta U(k)} = \Delta U(k)^T H^u \Delta U(k) - G(k+1 \mid k)^T \Delta U(k) \tag{25}$$

with condition (22).

Where the Hessian and the gradient vector of QP are

$$H^u = S^T \Gamma_y{}^T \Gamma_y S + \Gamma_u{}^T \Gamma_u$$
$$G(k+1 \mid k) = 2S^T \Gamma_y{}^T \Gamma_y (y(k) - f(k)) \tag{26}$$

Algorithm for the hybrid predictive control

The algorithm has the following steps:

- Initialization of parameters: m, p, Γ_y, Γ_u
- calculation of impulse response model g_i from (9),
- calculation of us from y_s=$y(k)$, considering the inversion of the fuzzy-neuro model,
- calculation of the value of the steady-state gain K by (6),
- calculation of S by (17) and e by (13),
- calculation of the controller output from the first element of the calculated Δu vector generated from Quadratic program formulation (25) and (26).

3 Case Study and Simulation Results

3.1 Case Study

The application considered involves an isothermal reactor in which the Van Vusse reaction kinetic scheme is carried out. In the following analysis, A is the educt, B the desired product, C and D are unwanted byproducts [2].

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$
$$2A \xrightarrow{k_3} D \tag{27}$$

From a design perspective the objective is to make k_2 and k_3 small in comparison to k_1 by appropriate choice of catalyst and reaction conditions. The concentration of B in the product may be controlled by manipulating of the inlet flow rate and/or the reaction temperature.

The educt flow contains only cyclopentadiene in low concentration, C_{Af}. Assuming constant density and an ideal residence time distribution within the reactor, the mass balance equations for the relevant concentrations of cyclopentadiene and of the desired product cyclopentanol, C_A and C_B, are as follows:

$$\begin{aligned} \dot{C}_A &= -k_1 C_A - k_3 C_A^2 + \frac{F}{V}(C_{Af} - C_A) \\ \dot{C}_B &= k_1 C_A - k_2 C_B - \frac{F}{V} C_B \\ y &= C_B \end{aligned} \tag{28}$$

This example has been considered by a number of researchers as a benchmark problem for evaluating nonlinear process control algorithm.

By normalizing the process variables around the following operating point and substituting the values for the physical constants, the process model becomes:

$$\begin{aligned} \dot{x}_1(t) &= -50\, x_1(t) - 10\, x_1^2(t) + u(10 - x_1(t)) \\ \dot{x}_2(t) &= 50\, x_1(t) - 100\, x_2(t) + u(-x_2(t)) \\ y(t) &= x_2(t) \end{aligned} \tag{29}$$

where the deviation variable for the concentration of component A is denoted by x_1, the concentration of component B by x_2, and the inlet flow rate by u.

3.2 Simulation Results

The comparison of time responses of output of NFCM with nonlinear plant is shown in Fig. 2. Time responses of the controlled, reference, manipulated and manipulated rate variables under hybrid predictive control (HPC) without constrained are shown in Fig. 3.

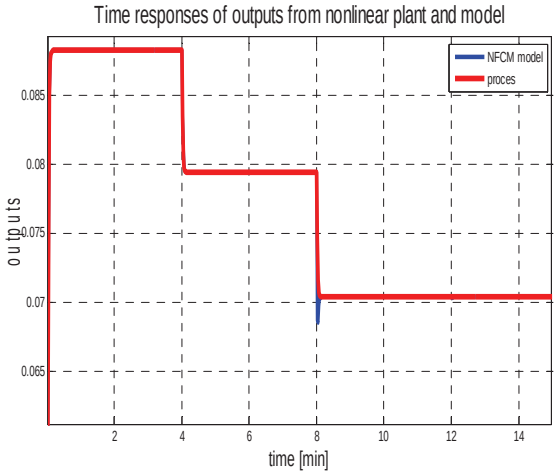
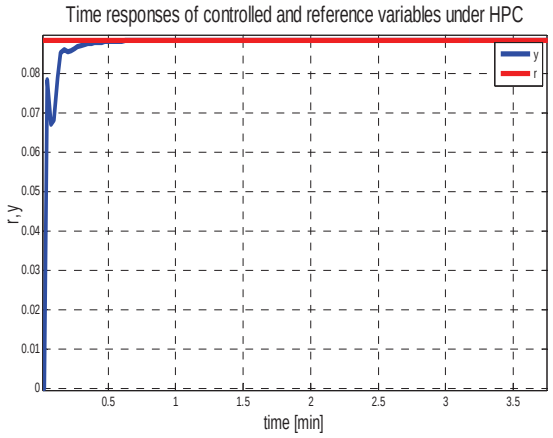


Fig. 2 Time responses of output from the nonlinear plant and the NFCM model

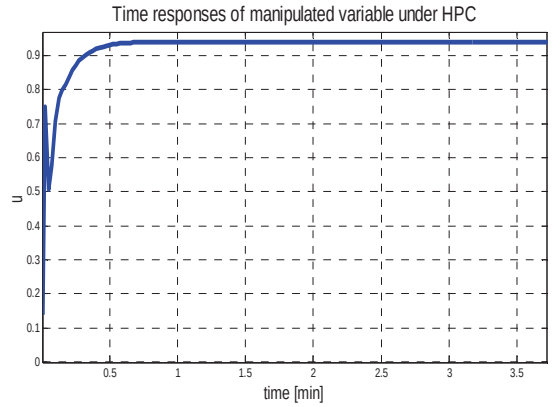
The parameters for predictive control are m=5, p=10, Γ_y=K, Γ_u=γK^2, γ=0.1.

As can be seen in Fig. 3 for this application, the manipulated variable $u\epsilon$<0;0.941> and manipulated variable rate are $\Delta u\epsilon$<-0.2452;0.1315>.

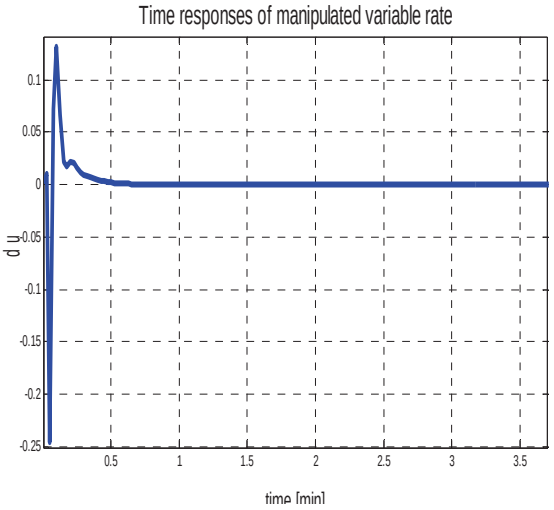
Time responses of the controlled, reference, manipulated and manipulated rate variables under hybrid predictive control (HPC) with constrained manipulated variable and manipulated variable rate are shown in Fig. 4. The quality of predictive control is very sensitive to constraints. So the constraints are $u\epsilon$<0;0.935> and manipulated variable rate is $\Delta u\epsilon$<-0.24;0.125>.



a)

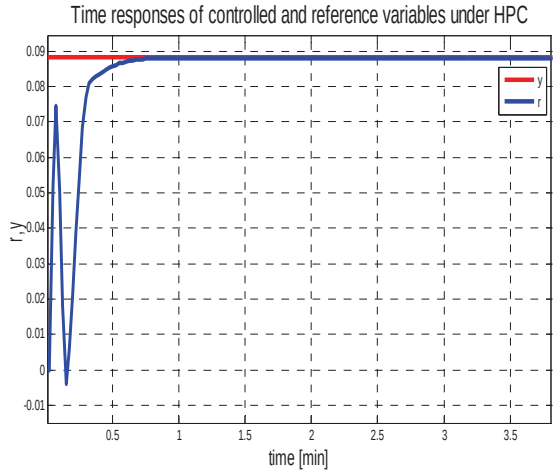


b)

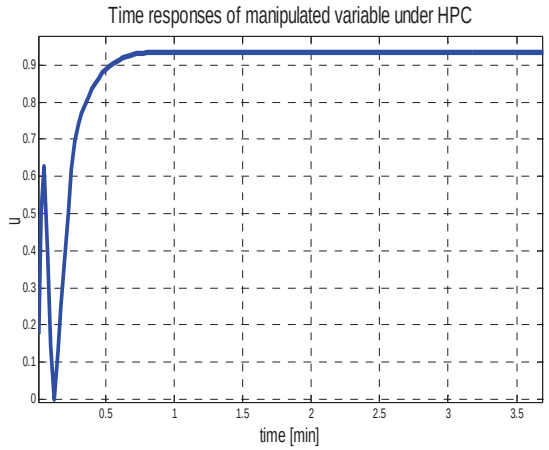


c)

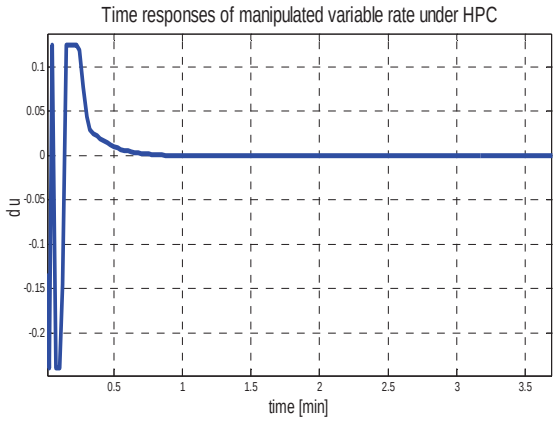
Fig. 3 Time responses of the controlled and reference a), manipulated b) and manipulated rate variables c) under hybrid predictive control without constraints



a)



b)



c)

Fig. 4 Time responses of the controlled and reference a), manipulated b) and manipulated rate variables c) under hybrid predictive control with constraints

Conclusion

The HPC uses the advantage of neuro-fuzzy systems in the representation of the steady-state behaviour of the system. Other advantage is that it combines knowledge of the system in a form of a priori knowledge and measured data in the identification of a control relevant model.

Simulation example illustrates the potential offered by the predictive control with NFCM.

Acknowledgments

This paper has been supported by the Slovak Scientific Grant Agency, Grant No. 1/0544/09.

References

[1] ABONYI, J., BÓDIZS, Á., NAGY, L., SZEIFERT, F.: Hybrid fuzzy convolution model and its application in predictive control. Chemical Engineering Research and Design 78 (2000), pp. 597-604.

[2] PAULUSOVÁ, J., KOZÁK, Š.: Robust Predictive Fuzzy Control. 7th Portuguese Conference on Automatic Control, CONTROLO'2006, Lisbon, Portugal: 11-13 September 2006, MA-7-1

[3] PAULUSOVÁ, J., KOZÁK, Š.: The comparison of the conventional controllers with fuzzy controllers. Tatranské Matliare, Slovac Republic, May 31-June 3, 1999, 168-171

[4] PAULUSOVÁ, J., KOZÁK, Š.: Nonlinear model-based predictive control. Control Systems Design 2003 CSD'03. Bratislava, Slovak Republic: 7-10 September, 2003, pp. 171-175.

[5] PAULUSOVÁ, J., KOZÁK, Š.: Robust and Fuzzy Dynamic Matrix Control Algorithm. 5th International Carpathian Control Conference 2004. Zakopane, Poland: 25-28 May, pp. 145-152

[6] PAULUSOVÁ, J., DÚBRAVSKÁ, M.: Hybrid Predictive Control of Nonlinear Process. International Slovak Society for Cybernetics and Informatics, Vyšná Boca, Slovak Republic: 10-13 February 2010, 21_Paulusova_Dubravska.pdf

Ing. Jana Paulusová, PhD.
Ing. Mária Dúbravská

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel.: +421 2 60291262
e-mail: jana.paulusova@stuba.sk

Designing and Implementation of Complex Software Solution for Processing of Measurement Rates

Mária Dúbravská, Jana Paulusová

Abstract
This paper deals with designing and implementation of a complex software solution which includes a server application for processing of measurement rates and software for agents carrying out electricity measurements. The management of measurements is the primary process of the organization. This paper defines the basic logical and technological procedure for carrying out measurements.

Keywords: data collecting, measurement batch, measuring equipment, field force, deductions management

Introduction

Collecting of primary data for network managers and then processing into information systems is nowadays an activity that rises up a requirement for well organized structure of data that are measured. Each director of network has its own database, which operates on a different geographic information system. But the principle always remains the same, to fill this database with current and reliable data, which required for the administrator. The aim of this work was to design and implement a comprehensive software solution including server software running on the annual target benefits and software for staff in the field to perform deductions in energy so that the system has sufficient control of the process, automating as many activities and that the system can display as much management information, the theoretical cost in particular, the countdown in real-time status of payments, process invoices and various statistics [1]-[4].

1. Analysis of the old system

To achieve the objectives it was necessary to analyze the original operating system in detail on the fly so it could run continuously in addition to analysis of readings while the system was running undisturbed by this activity. Analysis of the original system was one of the longest-running stages, because of the original system which has no official documentation and is made of a lot of smaller applications, which together create a kind of data collecting system. From beginning the project had a clear idea - implement an information system as a single centralized system without any other associated application.

The company employees (staff in the field, coordinator of the measurements) were needed to ensure that the analysis was successful. All recommendations and observations of people working every day with this system were included in the design strategy of the new system.

1.1 Measurement batch

The most important unit of the system is the measurement batch. Measurement batch is the data set in encrypted form which has its structure defined by the administrator of engineering network, containing the same itineraries, which are intended to deduct the measurement route. Number and choice of itineraries in the measurement batch are determined according to requirements of staff working in the field.

Measurement batch file in encrypted form contains**:**

- serial number,
- owner name of the measurement point,
- address of a measurement point,
- electrometer serial number,
- last measurement,
- actual measurement,
- anomaly at the point,
- note to the measurement point.

1.2 Lifecycle of measurement batch

The whole process of management starts with generated "hierarchically divided statistics on the number of measuring points" by a competent worker. Statistics are generated in electronic form as a file "xls" and workers are obliged to submit the statistics. The coordinator after obtaining updated statistics splits out the individual itineraries for workers.

The coordinator of measuring sets the "Monthly plan of itinerary" plan and “Plan for worker” for each worker in field. Competent power company administrator is obliged to generate itineraries for individual workers on the Monthly plan basis and send it to the server. Blank measurement batches are distributed in electronic form from server to workers by coordinator.

The next step is the actual measurement. Directly at the measurement point every worker is obliged to take action as required by a particular customer. Worker in the field must visually inspect the measurement device for any damages and he must record it through pre-defined list of anomalies. Collected data were then sent back to the energy company.

Coordinator of measurement, worker in the field and administrator in energy company, uses his own application to fill the data in or to send the data to another site. This system is simple, but does not have one centralized database.

2. Design proposal of the new system

The main objective in the design of the information system for reading the status of measuring equipment in the energy sector was to keep the idea of the original system, for example maintain the measurement batch benefits and format as closely as possible. Preserve user interface for workers. The aim was to split the life cycle activities deduction into logical units and their subsequent automation to the maximal possible extent. Create powerful tools for work coordinators, workers in the field and for administrator in electric company to ensure integrity, protection and archiving of the data.

This information system is designed to be flexible enough and to be easily integrated into existing environments, or if necessary, to replace this entire environment.

Great emphasis was given to the interface which users will be accessing the system through. This interface was designed to be similar to the original interface. This similarity ensures users intuitively use the system features they were used to. Information system has been enhanced with new features which, despite the user-friendly control, lead to necessity of organizing training sessions for individual users.

The biggest change was the replacement of mobile devices used in the field. Psion has been replaced by the personal digital assistants (PDA). PDA work with PC, connecting and synchronizing with PC is different than that with Psion. Some field workers have become accustomed to write on paper stock sheets. Psions were not used, because the old models did not have sufficient memory capacity to save several measurement batches at once.

2.1 New system architecture

The main idea behind this technology is that the measurement batches should be treated where they are stored. If the user wants to work with them, system splits the data between the basic computer operations - servers, which store the data and user station - clients. Competences are shared between two processes - the client is responsible for processing of the particular tasks assigned to it when planning itinerary for which it requires the services of the second process - the server.

The main computer - the server is not only used as an application server, but also as a file server, which means that it collects and stores all information. Manipulation and control access is managed by the user computer. If an user asks for a selection of records, his computer will provide it. Data change is performed at the user station and returned back as modified file. This architecture [5] is adjusted to the speed of movement of files over the network and its flawless processing.

The system has universal client, which is a hypertext browser called WEB browser. The browser contains a presentation function, because it interprets the content pages. Such architecture is sufficient only for passive disclosure of information in the form of hypertext pages and does not implement a full information system (IS). To achieve the functionality of IS, the architecture have been incorporated into the Web environment opportunities for dynamic work with information. To achieve such options programming language PHP have been used, which breathed life to the entire architecture from within HTML pages.

Interface allows user to save the actual monthly plan of measurement from administration site, distribution of the itineraries to the coordinator of measurement, planning and viewing statistics. That way it is possible to directly use the IS and work with data. Big advantage in this particular case is versatility and ease of client implementation. Client accesses the IS opportunities through WEB server, which is also the database server and the core of the whole system. User can work with IS through the web browser. Each user of the system has a limited amount of rights, which is sufficient for him to perform tasks. These rights are set by the log-in names.

2.2 Linking, system security and data protection

Used protocol for data exchange must have capability of packing and transmitting through SSL. Chosen protocol is a CIFS (SMB BLOCK PROTOCOL), which is a standard data exchange protocol and complies all safety criteria. Based on the selected protocol, SSL (encapsulation protocol) was used for the secure exchange of data between the end points of information systems technology tunneling ports. It is a highly reliable and maintainable solution that can be easily implemented into the systems.

The reason for this is, that the tunneling data exchange goes via a public computer network. Secure connection is ensured by software called Stunnel, which required configuration on the electric company site. Furthermore, safety certificates have been generated to help to pass the CIFS protocol through SSL connection works.

Other security features:

- The system for communicating with users is using secure communication channels.
- Transfer of data between systems and users or other parties is encrypted.
- Each user of the system is given only limited amount of rights, which is sufficient for him to perform tasks.
- When properly deployed, system and compliance with safety standards is guaranteed by the physical security of data.
- The system creates regular backups of all stored data.
- The system records all significant events which users have made including the timestamp (TIMESTAMP).

2.3 Functional analysis

User management in the system provides to field worker, coordinator of measurement, manager and third party graphical user interface (GR) for management of users who log into it - adding, removing users, changing their data. System has a mechanism for communication and notification of users that always appears on the front page. This communication takes place either within the system itself through short message or by e-mail messages. The system provides managers with GR to define areas and points deductions and provides a mechanism for receiving the data from third parties.

System manager has the tools to set the conditions of the countdown cycle, such as list of anomalies for measurement points, setting fees for various types of measurement points and fees for the detection of faults. System provides to system manager various GR mechanisms to monitor the current status of measurement batches, including statistics how many measurement batches have been measured by the worker, the number of points to be checked on the gross and net success measurements, also has a GR to allow monitoring on the progress of the whole month plan. The system manager provides mechanisms for operational changes (changes to the reading assignment and urban

areas). Finally, the system provides tools for creating finished documents and statistics for the field worker.

Field worker by GR has a mechanism for redistribution of urban areas or deductions to measurement batch. Field worker assigns theoretical measure date to the system - so he creates a timetable countdown for various areas.

For the top management system provides real-time overview of the field worker financing, respecting deadlines measure transfer and the quality of the measurement batches. Management has tools to predict the cost of the monthly plan cycle in any unit price for the measurement or detect anomalies, as well as the entire history of the annual target cycles system, including statistics and reports.

Conclusion

Based on the analysis of the original setup a new system for processing of measurements has been designed and implemented, information system which meets the requirements of all activities associated with the collection status of measuring equipment in the energy sector.

The system is based on stable system architecture and can be expanded in the future based on the actual needs. It is compatible with different operating systems and web browsers, as well as mobile devices for field work.

In the design phase of this information system a lot of time had been spent in consultation with staff and with field workers. The new information system meets all design conditions and it can be used as a substitute to the old system with lots of new functionality to automate administrative work and calculations. IS provides a comprehensive overview of the current course readings and increases their effectiveness. User-friendly environment makes the system work simple for workers who have to use it daily.

Acknowledgments

This paper has been supported by the Slovak Scientific Grant Agency, Grant No. 1/0544/09.

References

[1] E. NARAMORE, J. GERNER Y. L.SCOUARNEC,J. STOLZ, M. K. GLASS: Vytváříme webové aplikace v PHP5, MySQL a Apache, Computer Press, 2006

[2] P. ŠETKA: Mistrovství v Microsoft Windows Server 2003, Computer Press, 2003

[3] L. DOSTÁLEK: Velký průvodce protokolmi TCP/IP Bezpečnost, Computer Press, 2003

[4] L. JELÍNEK: Jádro systému Linux, Computer Press, 2008

[5] M. HINKA – Diploma thesis: Designing and implementation of complex software solution for processing of measurement rates and for agents carrying out electricity measurements.

Ing. Mária Dúbravská

Ing. Jana Paulusová, PhD.

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel: +421-2-60291 286
maria.dubravska@stuba.sk

The position prediction for bridge crane using input shaping methods

Peter Hubinský, Ľuboš Chovanec

Abstract
When the operator tries to control a bridge crane, we can observe after reaching the final position the residual oscillation. One of the most useful methods to control crane are input shaping methods. A lot of articles treat swinging suppression, but only a little effort has been placed on reducing the time lag. Shaper can efficiently reduce the vibration, however it causes a delay between operator commands and the output signal of shapers that is fed into crane drive system. So it is necessary to set up some kind of the prediction. Prediction can predict the stopping location when operator finishes with generating the control signal from joystick and crane is still moving because of shaping process is not finished yet. This paper presents a control method that assists the human operator when guiding the crane through the space with obstacles by visualization of prediction on the operator's screen while using input shaping method to eliminate swinging of a payload. Experimental results show that input shaping methods and prediction together can improve quality of online control of bridge crane by operator.

Keywords: input shaping methods, prediction, bridge crane

Introduction

Nowadays the bridge cranes use to be used in various branches of industry because of their big lifting capacity and rigidity of construction. However the possibility of appearing residual oscillation can be unsuitable for dynamics and accuracy of positioning the payload in the working area. The deflection can lead to achieve desired position with a big delay. Other problems are consisting in possible collisions of payload with obstacles due to exciting of enormous amplitude of swinging.

One of the most useful way how to manage position of the trolley is the feedback control that uses sensors to gather information from plant. But it is not easy to find suitable place for sensors. Considering that the dirty environment where crane is situated can acquire disturbances or damage sensors. Another way how to control bridge crane is consisting in application input shaping methods that can effectively reduce payload oscillations.

1. Input shaping methods

Shapers modify the input signal which is generated by operator to series of appropriate pulses. We are talking about the feedforward control so the block of shaper has to be placed before the plant as is illustrated in Fig.1.

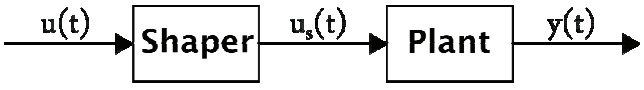


Fig.1 Scheme of input shaping

Input shapers are working on principle of convolving the step impulse and a series of Dirac impulses and thus generate suitable signal to eliminate residual vibration in the plant [1]. In the Fig.2 we can watch response of the plant on simple velocity rectangular impulse. Oscillations of the payload around desired value are unsuitable for positioning to the final position and it takes too much time to decrease swings using natural damping of the plant to zero amplitude.

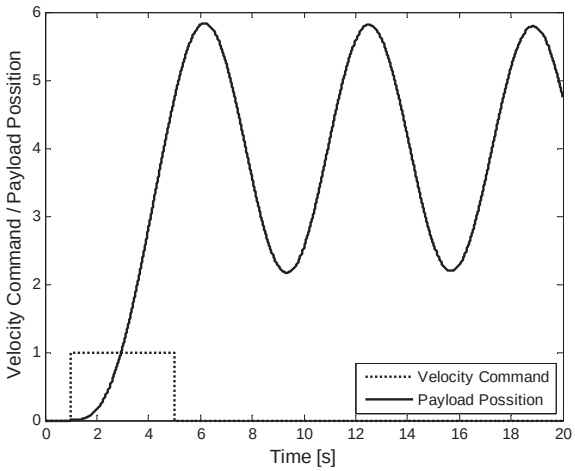


Fig.2 Unshaped input signal and plant response

As the Fig.3 shows, the shaper has delay which is modulated according to the natural frequency of plant. The increasing of command duration causes an additional trolley motion after the operator is stopping to generate the control signal. The operator has to estimate the time which is needed to stop the crane. Prediction displayed as a point on the operator screen includes delays of all shapers and relieve to avoiding obstacles carefully.

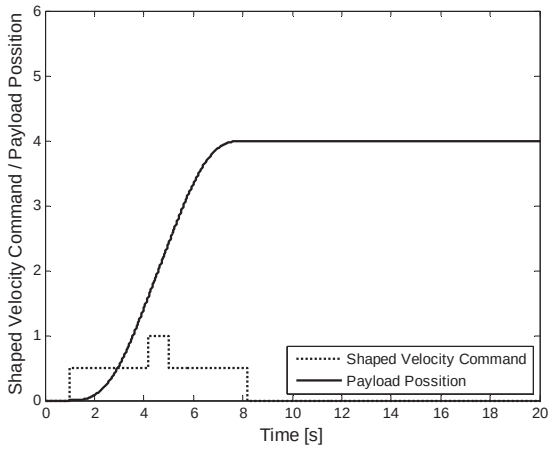


Fig.3 Shaped input signal and plant response

Bridge crane model

Analyzed bridge crane system has 3 degrees of freedom. (travelling along X-axis, traversing along Y-axis and hoisting along Z-axis).Fig.4

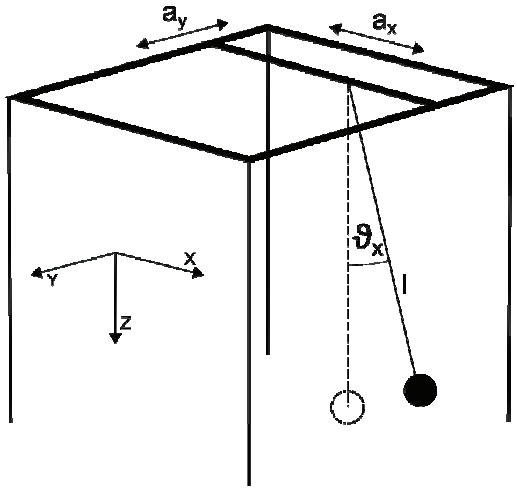


Fig. 4 Scheme of Bridge crane

We can describe it as a mathematical pendulum with moving base (travelling and traversing). After linearization the transfer function it could be represented in single axis as follows:

$$F_x(s) = \frac{\theta_x(s)}{a_x(s)} = \frac{K\omega_0^2}{s^2 + 2b\omega_0 s + \omega_0^2}$$

Where θ_x is swing angle amplitude, a_x is acceleration of trolley in x direction K is the gain, ω is natural angular frequency of the plant and b is the damping coefficient.

Prediction

To improve controlling and influence of the input shapers by the increasing the duration, it was implemented prediction block [1]. As the Fig.5 shows, the signal led to the prediction block from input of the shaper, is reduced by signal from the shaper's output. After that it is integrated to acquire the value of predicted position of the payload.

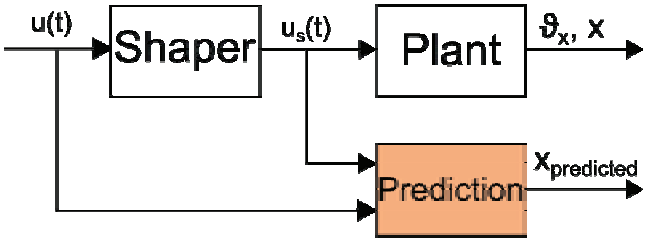


Fig. 5 Prediction block

The Fig.6 shows the response of the crane on impulse of velocity using ZV and ZVD shaper with prediction. Prediction is represented by red line and shows calculating stopping position. Here we can observe the difference between ZV prediction and ZVD prediction due to increasing the time delay of higher order shapers. Predicted positions which is connected to ZVD shaper, can show the greater distance before trolley position than predicted ZV shaper but the delayed time of stopping the crane is lengthened, too.

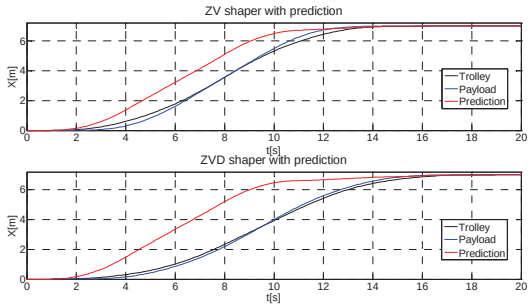


Fig.6 Prediction using ZV and ZVD shaper

Design of GUI

The graphic user interface has been created in Matlab software to help the crane operator to control the bridge crane through the space with obstacles in a warehouse using input shaping methods and prediction in the same time. The Fig.7 illustrates GUI which is divided to the control panel on the right side and 2D crane visualization on the left side. As a first operator can choose the following control algorithms: 1. No input Shaping, 2. ZV shaper, 3. ZVD shaper, some settings of the parameters that can be various for the crane system and the shaper (e.g. various length of rope for crane and for shaper which causes drag in to the plant more disturbance), showing the visualization of the prediction and after simulation. The user can observe maximal deflection during control process, time duration and all trajectories that were draw by trolley, payload and prediction. The position of trolley is represented on Fig.7 as an intersection of two green lines (green cross). The payload is illustrated as a blue square hoisting under the trolley. The position prediction is marked by red circle. Obstacles are presented by cyan colour. Graphical representation of the motor speed in X,Y axis is situated in upper right corner. That can help the user to understand the existence the delay between operator commands and real position of the trolley when the input shaping methods are applied.

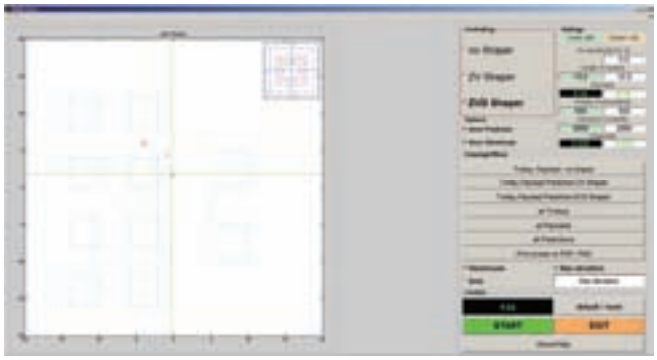


Fig. 7 Bridge crane GUI

Obstacles avoiding

Input shaping and prediction of stop point are now tested on the human operator using joystick to navigate the bridge crane through the warehouse and avoid the obstacles. Fig.8 illustrates the operator's view (cross represents trolley, blue square is the payload, red circle is the prediction and a pink square is the goal).

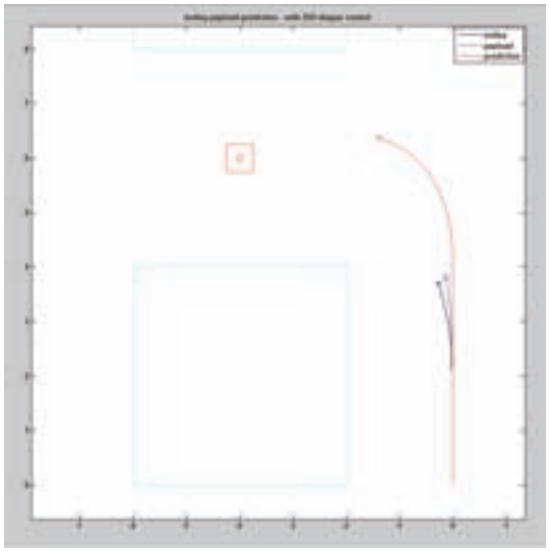


Fig.8 Visualization of the prediction for operator

When the operator guides the crane through the space of obstacles without input shaping, we can see the big payload swinging (blue waving trajectory in Fig.9-a). It is difficult to navigate and avoid to crash without any operator's skills. Input shaping with prediction allows to generate smooth trajectory without observing any swings (Fig.9-b and c).

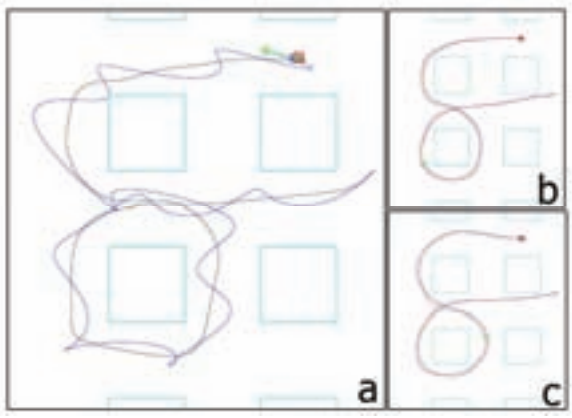


Fig. 9 Obstacles avoiding (a – without shaper, b ZV shaper, ZVD shaper)

As the graph shows in Fig.10, controlling without shapers is about operator's skills and user can achieve better time with each other attempt and a moderately decrease swinging of the payload. By contrast of input shaping, it cannot improve time duration (time both controlling is comparable) but it can evidently suppress the deflection. We can see pretty reduction of the vibration on Fig.9. Deflection is presented in Fig.10 as a distance between vertical projection of the trolley plain of the base and vertical projection of the payload to plain of the base in meters.

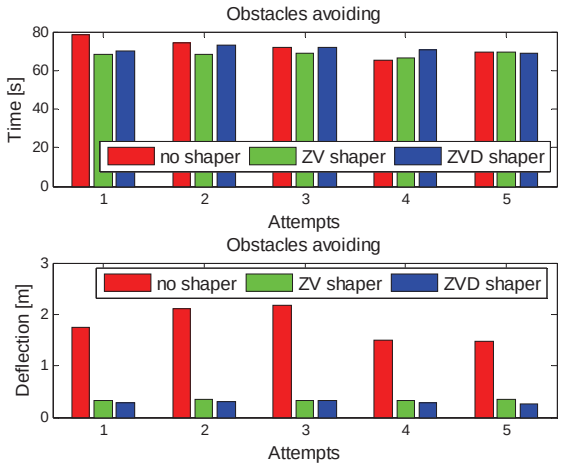


Fig.10 Experiment results for obstacle avoiding

Trajectory tracking

Bridge crane is considered as a low damped system. It is not easy to follow prescribed trajectory. The Fig.11 shows example of trajectory tracking (cyan colour).

When the human operator guides the crane along the desired trajectory, it is a little bit demanding due to time duration of shapers, if we choose controlling without prediction (see Fig.12). Other case, using input shaping with prediction can decrease the operation time significantly and suppress oscillation comparable with previous case (Fig.13).

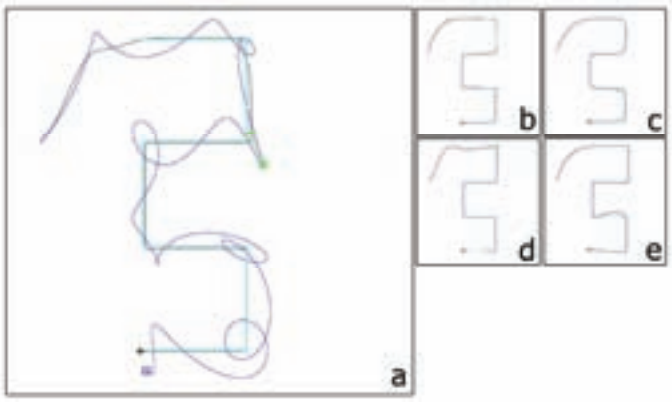


Fig. 11 Trajectory tracking without prediction (a-without shaper, b - ZV shaper with prediction, c - ZVD shaper with prediction, d – ZV shaper without prediction, e - ZVS shaper without prediction)

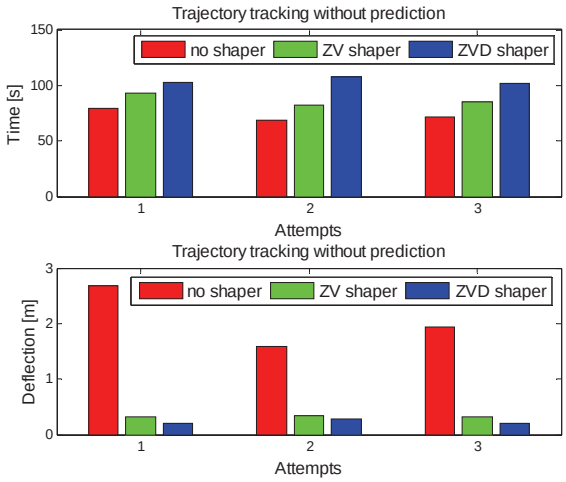


Fig.12 Experiment results for trajectory tracking without prediction

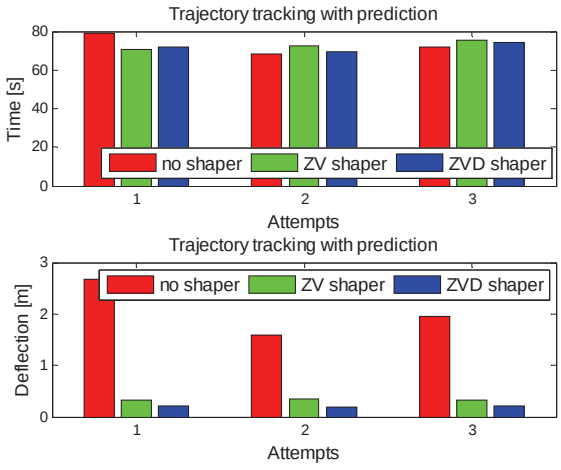


Fig 13 Experiment results for trajectory tracking with prediction

Conclusion

The paper presents the methods to enhance operator performance by combining a predictive user interface element with input shaping. It was demonstrated, that input shaping are useful feedforward methods which can suppress swinging of a payload but with remaining time duration.

As the results show, to set up prediction with shapers, it is very helpful to navigate crane with hanging payload through obstacles in warehouse with using the prediction spot, which indicates the place where the crane stop.

Acknowledgments

This work was supported by Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/0690/09. The authors are pleased to acknowledge this support.

References

[1] HUBINSKÝ, P.: Riadenie mechatronických systémov s nízkym tlmením. Vydavateľstvo STU, Bratislava 2010

[2] IVANOV, I., HUBINSKÝ, P.: Crane oscillation reduction, Selected Topics in Modelling and Control, vol. 4, 2004, STU Press, Bratislava, s.20-25

[3] Hubinský, P., KURILLA, J., PALKOVIČ, L.,: Estimation of the Gantry Crane Natural Frequency Using MEMS Accelerometer, AT&P Journal Plus, s.89-94, 2010

[4] Vaughan, J., Smith, A., Singhose, W.: Using a predictive graphical user interface to improve tower crane performance, Robotics and Applications, 2009

[5] Singhose, W.: Command generation for flexible systems, MIT, Stanford University, 1992

Prof. Ing. Peter Hubinský,PhD.

Slovak University of Technology

Faculty of Electrical Engineering

and Information Technology

Institute of Control and Industrial Informatics

Ilkovičova 3, 812 19 Bratislava

Tel.: +421 2 60291608

E-mail:peter.hubinsky@stuba.sk

Ing. Ľuboš Chovanec

Slovak University of Technology

Faculty of Electrical Engineering

and Information Technology

Institute of Control and Industrial Informatics

Ilkovičova 3, 812 19 Bratislava

Tel.: +421 2 60291305

E-mail:lubos.chovanec@stuba.sk

THE MANIPULATION EXTENSION WORKING IN A DIMENSIONAL CONSTRAINED SPACE

Rastislav Baláž, Mária Ádiová, Ľubica Miková

Abstract
Working spaces are very often dimensionally limited. Conventional manipulators are not able to operate in this workspace. This article shows the design of the manipulation extension, which is useful in these places. This device belongs between the important parts service robots useful for inspection operation.

Keywords: Inspection robot, Manipulation extension, Limited space

Introduction

A car bomb is an improvised explosive device placed in a car. The U.S. military and law enforcement agencies call often a car bomb as "VBIED", which is acronym standing for "Vehicle Borne Improvised Explosive Device" [3]. A car bomb can be placement in different parts of the vehicle. Nowadays, car bombs are fixed magnetically to the underside of the car or inside of the mudguard. Detonators are activated by the opening of the vehicle door, pressing the brake or the accelerator pedals. Detection of these explosives is made using the mobile service robot. The robot is made inspection of the vehicle by using a camera system. If the car bomb is fixed on the chassis, there is the problem with manipulation. A car bomb is stored in a confined space, which is dimensionally designed with ground clearance of car chassis. Manipulation in this area is very difficult because of its dimensionally limited.

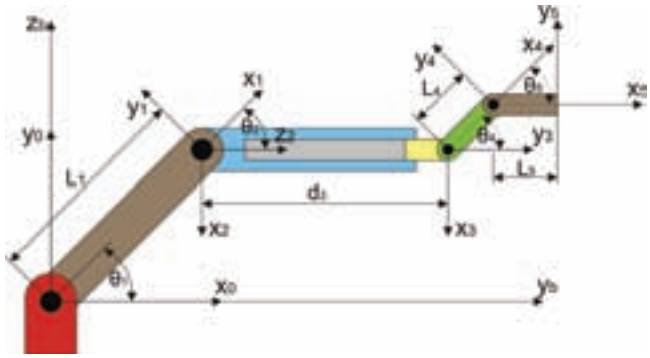
Service robot has a similar problem in the inspection of pipelines. The robot is deployed in a confined space, which is a dimensionally limited wall of pipe. The manipulation extension has problem with manipulation in such an environment, because if the link of manipulator are large, cannot be decomposed and robot cannot carried out inspection task.

1. Kinematics manipulation extension

The manipulation extension is designed as manipulator in manipulator. A smaller manipulator is hidden in link of larger manipulator. If it is necessary to use a smaller manipulator, this manipulator is ejected from a link of larger manipulator.

$$ {}^{1}T_{6} = {}^{1}T_{2}.{}^{2}T_{3}.{}^{3}T_{4}.{}^{5}T_{6} \tag{1} $$

$$ {}^{1}T_{6} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \tag{2} $$



Obr.1 Kinematika manipulačnej nadstavby
Fig.1 Kinematics manipulation extension

i	υ_i	d_i	a_i	α_i
0	$\pi/2$	0	0	$\pi/2$
1	θ_1	0	L_1	0
2	$\pi/2 + \theta_2$	0	0	$\pi/2$
3	0	d_3	0	$-\pi/2$
4	$\pi/2 + \theta_4$	0	L_4	0
5	θ_5	0	L_5	0

Tab.1 DH parameters

Inverse kinematics calculated by using Jacobian. The global position of the endpoint of the manipulator is at

$$ \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} r_{14} \\ r_{24} \end{bmatrix} \tag{3} $$

calculate the Jacobian

$$ J(q) = \left[\frac{\partial T_i}{\partial q_i} \right] \tag{4} $$

$$ J^{+} = \left(J^{T}.J \right)^{-1}.J^{T} \tag{5} $$

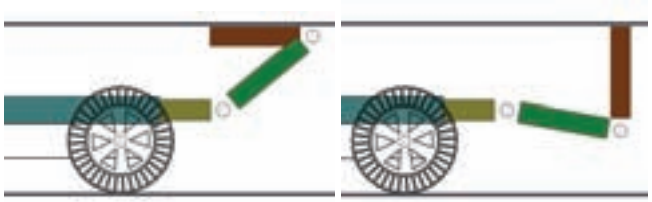
The kinematics of manipulation extension is seen that the manipulator is redundant - redundant manipulator. Whereas the Jacobian has a rectangular matrix and n > m, then

method of Moore-Penrose pseudoinverse is used for the calculation, equation (5), [1, 2].

For manipulation extension is defined two conditions for its design. The first condition is that the manipulation extension has to be able manipulated with the own body in space, which is defined with chassis car and the roadway. The second condition is that the manipulation extension has to able manipulated a larger space than the space in the first condition. This space is characterized by the size of air pipe, piping or canalization. The average distance to the lowest part of the chassis to the roadway varies from 100mm to 170mm. The average size of air pipes is from 100 mm to 1000mm [4].

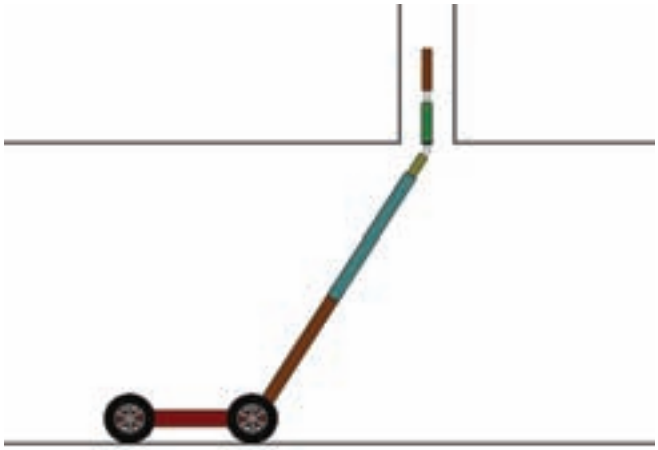
2. Manipulation extension in confined space

The designed kinematics manipulation extension can be working in a confined space which is defined with chassis car and the roadway (Fig. 2). For work in this space smaller manipulator is ejected from link of larger manipulator. If the height chassis is larger eg. for off-road cars, will be used to achieve the goal point also a larger manipulator. The designed manipulation extension is able to manipulate and work also in the pipeline (Fig. 3). Range of manipulating the superstructure is dependent on the length of the links of larger manipulator.



Obr.2 Manipulačná nadstavba pod podvozkom automobilu

Fig.2 The manipulation extension underneath chassis car



Obr.3 Manipulačná nadstavba v potrubí

Fig.3 The manipulation extension in pipeline

Conclusion

This article deals with the design of manipulation extension for manipulating in confined space. The design of manipulation extension is preliminary design of manipulation extension that to be able to manipulated in a confined space. Length of links of larger manipulator does not affect to the manipulation ability of a smaller manipulator. Another design will be directed towards design of a suitable end effector or reconfigurable wrist for different types of instruments.

Acknowledgement

This contribution is the result of the project implementation: Center for research of control of technical, environmental and human risks for permanent development of production and products in mechanical engineering (ITMS: 26220120060) supported by the Research & Development Operational Programme funded by the ERDF. This work was also supported in part by the Ministry of Education of the Slovakia Foundation under Grant VEGA 1/0464/09.

References

[1] CRAIG, J. J. Introduction to robotics: Mechanics and control. Addison-Wesley Publishing Company, 1989. p. 450.

[2] SKARUPA, J., MOSTYN, V. The theory of industrial robots. Kosice, 2000. p. 146. (in Czech)

[3] GLOBALSECURITY URL: <http://www.globalsecurity.org/military/intro/ied-vehicle.htm>, [cit. 2011-01-05].

[4] TUNWANNARUX, A. Design of a 5-Joint Mechanical Arm with User-Friendly Control Program. World Academy of Science, Engineering and Technology 27 2007, 43 – 48.

Rastislav Baláž, Ing.

Technická univerzita v Košiciach
Strojnícka fakulta, Ústav špeciálnych technických vied
Katedra aplikovanej mechaniky a mechatroniky
Letná 9
042 00 Košice
00421 55 602 2719

E-mail: rastislav.balaz@tuke.sk

Mária Ádiová, Ing.

Technická univerzita v Košiciach
Strojnícka fakulta, Ústav špeciálnych technických vied
Katedra aplikovanej mechaniky a mechatroniky
Letná 9
042 00 Košice
00421 55 602 2719

E-mail: maria.adiova@tuke.sk

Ľubica Miková, Ing.

Technická univerzita v Košiciach
Strojnícka fakulta, Ústav špeciálnych technických vied
Katedra aplikovanej mechaniky a mechatroniky
Letná 9
042 00 Košice
00421 55 602 2719

E-mail: lubica.mikova@tuke.sk

Robot Control Using Hand Gesture Recognition Overview

Michal Tölgyessy, Peter Hubinský

Abstract
This article is an overview of three distinct and complex algorithms developed for robot control. The basis of each algorithm is a set of hand gestures. Each gesture has a significant relationship with the controlled robotic system. A user shows a gesture in front of a webcam, this gesture is recognized by the system and a specific pre-defined action is carried out.

Keywords: Gesture recognition, Robot, Human Computer Interaction

Introduction

Since the origin of the first computers humans need to interact with them. Keyboard, mouse, joystick, gamepad or touch screen – all these devices are widely known and used to communicate with the computer. However, the most intuitive and natural way for a human to interact with the world around him or her is using voice and gestures. In such communication there is no barrier or mediator. No devices have to be used, no machine protocols have to be learned. Nowadays a lot of effort is invested in the research of gesture communication with robots [1][2][3]. In such communication gestures need to be captured and recognized. Subsequently a specific task is done by the robot. A specific gesture alphabet has to be invented in order to provide a smooth and reliable interaction. Usually, simple or more sophisticated cameras are used to monitor a person, frames are analyzed and expected gestures are identified. This approach introduces the need to master the field of computer vision, that is used in many scientific and industrial applications and is a subject of many research activities [4]. Convenient algorithms need to be selected, implemented and tested for a successful recognition process. This article is an overview of several latest approaches invented and documented by research teams from India and Macedonia.

1. Mobile robot's freight ramp control

In [1] the authors proposed an algorithm for hand gesture-based control of mobile robot's freight ramp. Their research was focused on solving some of the most important problems of current HRI (Human-Robot Interaction). Their final goal was to develop a chip and a robust control system without the need to use data gloves, colored gloves or other devices. Firstly, a set of gestures – a gesture vocabulary was defined. Secondly, a generic webcam was chosen for the purpose of image acquisition process. Finally, an algorithm was developed, where gestures are identified using motion detection and gesture recognition algorithm based on histograms. The result is implemented in a teleoperation client-server environment.

1.1 Hand gesture vocabulary

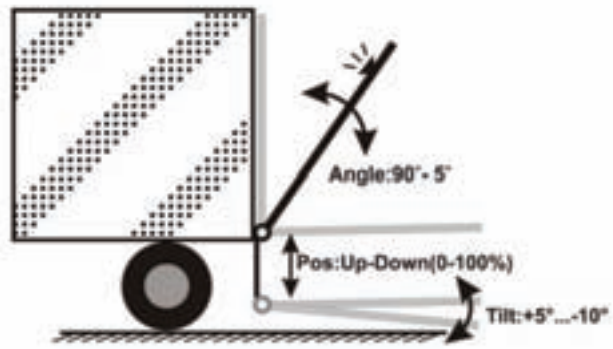


Fig.1 Freight ramp working principle

In order to control a robot freight ramp a vocabulary of 12 gesture poses was developed. The first two gestures control the opening and closing of the ramp. Next two gestures move the ramp up and down. The Stop hand gesture stops any action the ramp moving the system into the Stopped position. StopTilt gesture stops any tilt movement the ramp performs and takes the ramp into the Stopped position. The rest of the gestures control the tilt angle of the ramp.

Fig.2 Hand gesture vocabulary

1.2 Motion detection and object extraction

The output of a webcam is a set of 2D images that include a vast amount of data. First goal in successful gesture recognition is to detect a person in an image and detect and crop out those parts of his body that are supposed to show gestures the system is able to recognize. The authors of [1] have used motion detection to locate a potential region of interest (ROI). Their approach is based on analyzing consequent video frames and finding inter-frame differences. In such approach the background scene has to be updated regularly because of minor light condition changes, movements of small objects or appearance of new objects on the scene. This continuously updated background is stored as the reference image. The main problem with such technique is that over time a person's body is added to the reference image. Certain modifications are used to solve complications that arise from this factor.

The core movement detection mechanism works by applying a differentiation operation on the current and reference image. Hence, it is possible to see the absolute difference between the two images. Whiter areas on the output image show the areas of higher difference and black areas show the areas of no difference. A sample of the described process is in Fig.3.

Fig.3 Image differencing with reference image

On this image, some "thresholding" is done by classification of each pixel as significant change (most probably caused by moving object) or as non significant change. To get rid of false objects that are far from being a human body all objects in the image are examined and their size is checked. Only objects that are big enough and satisfy constrains relative to the human body proportions are considered as human body. In order to pass to the gesture recognition task the algorithm has to be sure that the detected object is not performing any movement and is showing a particular gesture. For this task a motion detector is used. It searches for motion absence using the between frames difference. If the absence of motion is present in a sequence of consecutive frames for a time period longer than a predefined threshold value, the gesture recognition procedure is performed.

1.3 Hand gesture recognition

The algorithm for hand gesture recognition assumes that target object occupies the entire image. It takes into consideration human body proportions and is based on histogram analysis. The core idea is based on analyzing two kinds of histograms – horizontal and vertical histogram (shown in Fig.4.).

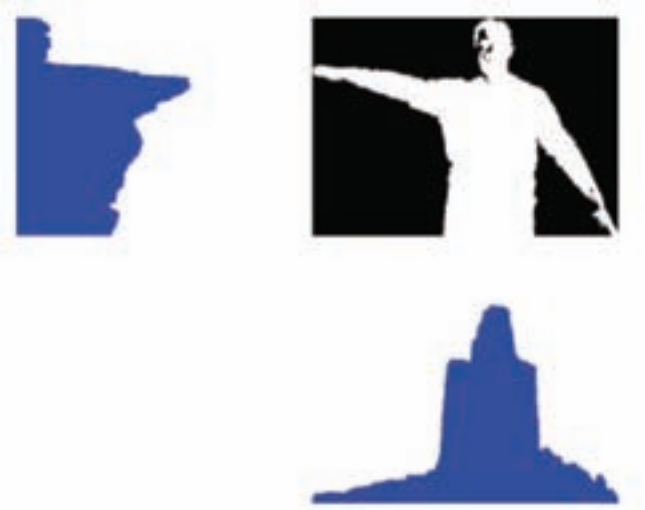


Fig.4 Horizontal and vertical histogram of the detected human body

By applying simple thresholding on the horizontal histogram hand areas and torso area can be easily classified and their length determined. Considering statistical assumptions about body proportions it can be determined whether the hand is raised or not. If the hand is not raised its width will not exceed some threshold value of torso's width. Both experimentally and analytically it was proven that the threshold value of 30% gives satisfactory results. If the hand is raised the vertical histogram is used to determine its exact position. In some cases the histograms may be cluttered with noise, which may be caused by light conditions or shadows. Hence, two additional preprocessing steps of filtering are performed on the vertical histogram:

- Values, which are lower than a defined threshold percentage of the maximal histogram value, are removed. This takes care of some artifacts caused by shadows.
- All the peaks except the highest one are removed. This removes strong shadows and other artifacts caused by the environment changing conditions.

After these preprocessing steps the exact hand position is determined.

2. Recognition techniques for HCI

In [2] the researchers propose an approach based on fast segmentation process to obtain the moving hand from the whole image. The final algorithm is able to deal with a large number of hand shapes against different backgrounds and lighting conditions and to recognize the hand posture from the temporal sequence of segmented hands. As the basic sensor to capture gestures a webcam was chosen.

2.1 System Components

The developed system is based on following processes:

Initialization

A visual memory is created in a start-up step and the recognizable hand gestures are stored in it.

Acquisition

A frame from the webcam is captured.

Segmentation

Each frame is processed separately before its analysis:
- The image is smoothed.
- Skin pixels are labeled.
- Noise is removed and small gaps are filled.
- Image edges are found.

After a blob (binary linked object) detection, the blob that represents the user's hand is segmented. A new image, which contains the portion of the original one where the human hand was found, is created.

Pattern recognition

Once the user's hand has been segmented, its gesture is compared with those stored in the system's visual memory using three stage recognition process:
- Kalman filtering process
- Hidden Markov models
- Graph matching

Executing action
In this process the system carries out the corresponding action according to the recognized hand gesture.

An image acquired by a low cost web cam is corrupted by random variations in intensity and illumination. For this reason a linear smoothing filter is applied in order to remove noisy pixels and homogenize colors. Best results were achieved using a mean filter. To normalize the color appearance a lighting compensation technique that uses "reference average" is used. The normalization operation subtracts the average of the whole image from each pixel color band (R, G, B) so that the odd colored images, for example reddish, are turned into more natural images.

Once the initial image is smoothed and normalized, a binary image is obtained in which every white pixel represents a skin-tone pixel from the original image. Blobs are groups of pixels that share the same label due to their connectivity in a binary image. After a blob analysis all the pixels that belong to the same object share a unique label. The developed system locates the user's hand using global features available for every blob, however it must be informed whether the user is left or right handed.

2.2 Hand gesture recognition

Gesture recognition is achieved using three stages (levels).

Level 1 (Kalman filtering process)

At this level the unknown gesture is projected into the eigenspace created during the Initialization phase, where all gestures from predefined alphabet were mapped onto a point. By finding the Euclidian distance of the point of the unknown gesture to all the representatives in the eigenspace a set of potential nearest representatives is determined. This group of gestures starts with the same shape as the unknown gesture and is passed to the next level.

Level 2 (Hidden Markov models)

By projecting the input gesture into the second common eigenspace (formed in the second phase of the Initialization process) the nearest sequence of symbols is extracted. The trained HMMs of the gestures forwarded from the first level are employed to calculate the likelihood of the extracted sequence. One can consider the HMM that results in the largest likelihood as the best match. The correct gesture has either the highest likelihood or a small deviation from the highest one. So, a well-chosen threshold gives a few gestures with a small deviation from the greatest likelihood. These are chosen to be compared carefully with the unknown input gesture.

Level 3 (Graph matching)

By projecting the unknown gesture into the eigenspaces of the selected gestures the authors try to find the best match. The projections of the gesture in the subspaces form the individual manifolds. Each manifold is estimated by a graph. Therefore, there are two graphs in every subspace.

3. Wireless robot control

In [3] the authors developed an algorithm for robot control based mainly on histograms. It is able to recognize static hand gestures with HSV color spaces as major parameters. Their experimental results are very encouraging because the system produces real-time responses and highly accurate recognition with various gestures used under different lighting conditions. A wireless camera is used to capture the hand gestures which are then registered, normalized and feature-extracted. Gestures are used to control a wireless robot which comprises of Zigbee module for receiving data from a remote personal computer. MATLAB is used throughout the project for real-time data processing, classification and controlling the robot by data transferred from the system. Once the user's hand gesture matches the predefined command programmed in the MATLAB environment, the command is issued to the robot control unit (ATMEL 89C52 Microcontroller) via a serial port. If an unknown gesture is issued, the system rejects it and notifies the user. Threshold filtering is applied to remove the non-skin components. The major advantage of the authors' approach is that the influence of luminosity can be removed during the conversion process. Segmentation is then less dependent on lighting conditions.

3.1 Gesture alphabet

Currently, the developed system needs only a few seconds to analyze the user's hand in order to determine the threshold value for skin segmentation and store it. The authors invented a set of gestures that are distinct from each other and easily recognizable by the system. The first gesture needed to initialize the hardware is Start followed by Front – these two gestures cause the robot to start and move forward. Any command from a predefined library can be issued randomly, however if the commands are not shown in a logical manner, a proper course of action cannot be taken.

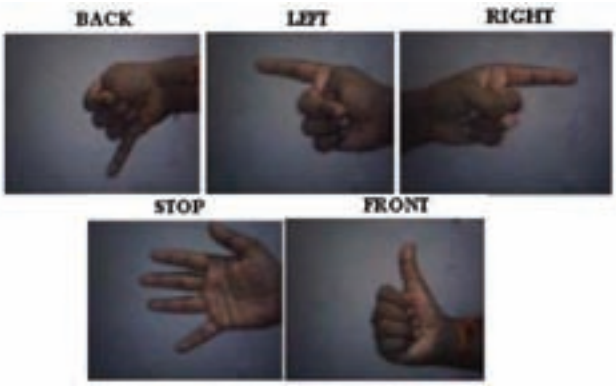


Fig.5 Hand gesture alphabet

3.2 Gesture normalization

The researches have given significant attention to the skin detection problem. Detecting skin-colored pixels has proven to be challenging for many reasons. The appearance of skin in an image depends on the illumination conditions and the choice of the color space affects the performance as well. In any given color space skin color occupies a part of it (usually called the skin color cluster) and can be a large region in the space. Another challenge comes from the fact that many objects in the real world, such as wood, leather, hair, sand can have skin colors. Skin detection depends on locating skin-colored pixels, therefore its use is limited to color images. It is not useful with gray-scale, infrared or other types of image modalities that do not contain color information.

Gesture normalization is done by the well-known morphological filtering technique – erosion combined with dilation. The output of this stage is a smooth region of the hand that is stored in a logical bitmap image. Image normalization is implemented using the morphological function available in MATLAB.

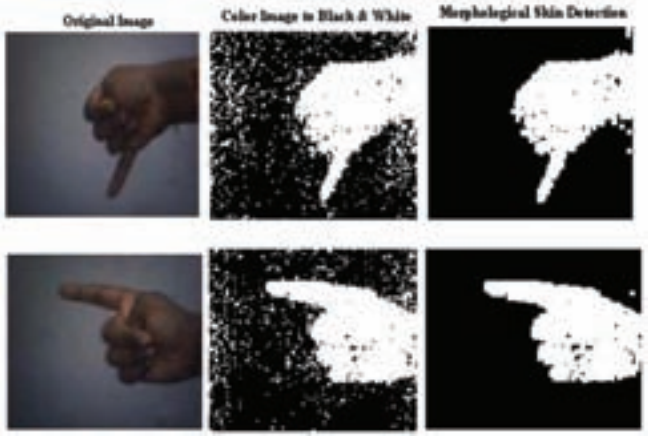


Fig.6 Smooth region output of the Morphological filtering technique

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Authors used the Canny edge detector to achieve noise reduction caused by lighting conditions.

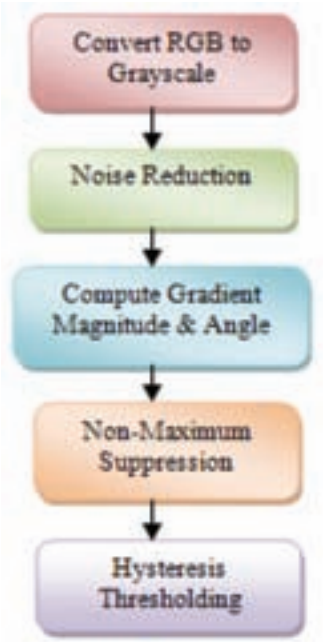


Fig.7 Steps involved in Canny edge detection

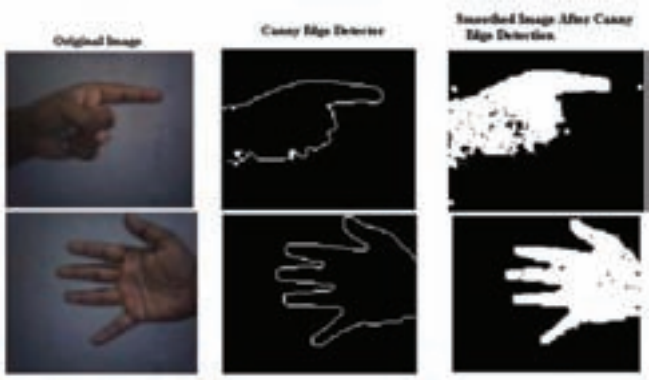


Fig.8 Smooth region output using Morphological filtering technique and Canny edge detector

3.3 Gesture recognition

Hand gesture recognition is carried out by improved centroidal profile created after proper gesture normalization as described in 3.2. Centroidal profile expresses a detected object with a set of vectors describing its contour. An algorithm computes the centroidal profile of the acquired gesture and compares it with a profile of an ideal object to recognize similarities. In the case of hand gestures every pixel of the contour is used to compute normalized distances. Chain code expresses the contour of an object with predefined set of vectors, moving along the edge of the object and evaluates the boundary. In the algorithm developed by the authors 8 directional codes are used.

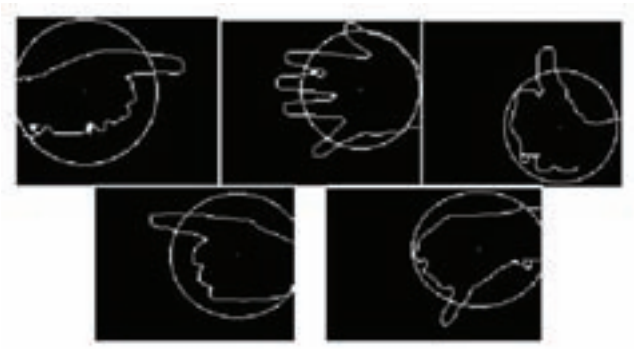


Fig.9 Center of gravity of various hand gestures

3. Conclusion

This article describes three distinct algorithms used to control robots with hand gestures. It is not an easy task because successful robot control requires real-time solutions. The research teams have managed to develop algorithms with quite high successful rates. Each approach is original and can serve as an inspiration for further development of hand gesture based algorithms in the field of robot control.

Acknowledgement

Support of the Grants VEGA 1/0690/09 is duly acknowledged.

References

[1] KOCESKI, S. - KOCESKA, N.: Vision-based Gesture Recognition for Human-Computer Interaction and Mobile Robot's Freight Ramp Control: Proceedings of the ITI 2010 32nd Int. Conf. on Information Technology Interfaces, 2010

[2] BURANDE, Chetan A. – TUGNAYAT, Raju M. – CHOUDHARY, Nitin K.: Advanced Recognition Techniques for Human Computer Interaction: IEEE ICACE, 2010

[3] MANIGANDAN, M. – JACKIN M.: Wireless Vision Based Mobile Robot Control Using Hand Gesture Recognition Through Perceptual Color Space: IEEE ACE, 2010

[4] BENICKÝ, P.: Motion classification and identification in robotics applications: FEI STU, 2010

prof. Ing. Peter Hubinský, PhD.
Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and
Information Technology
Institute of Control and Industrial Informatics
Department of Automation and Regulation
Ilkovičova 3
812 19 Bratislava
hubak@elf.stuba.sk

Ing. Michal Tölgyessy
Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and
Information Technology
Institute of Control and Industrial Informatics
Department of Automation and Regulation
Ilkovičova 3
812 19 Bratislava
michal.tolgyessy@stuba.sk

Robotic middleware based on protohreads

Michal Bachratý, Peter Hubinský

Abstract
Robots are based on lot of different hardware and software components. These components must communicate together with one common language to achieve one robotic application. Robotic middleware provide way to glue hardware components together. Components of robotic middleware communicate with one common communication protocol that enables application written in different languages or systems to communicate together. So that middleware helps system developers to composite large robotic applications without need to program all components. This paper presents new communication middleware based on protothreads. Protothreads provide linear code execution for event-driven systems implemented in C, sequential flow of control without complex state machines or full multi-threading and work under or without an underlying operating system. Because protothreads and middleware communication protocol is easy to implement, presented middleware can be put into 8 bit microcontroller with only 1KB RAM.

Introduction

Sensors and technology changed during the last ten years. Sensors include communication interfaces with data signal processing embedded on complex control units designed for specific purposes or designed as universal units. This step simplify system development, but each sensor or specific hardware from manufacturers have own communication protocol. Data output identical sensors from different manufacturers can be identical. Therefore it seems made uniform communication platform, which unloads developers from difficult programming of sensors communication protocols. Similarly, analog sensors can have similar interfaces as digital sensors. Robotic system consists of huge number of components. If we can take only space navigation and obstacle detection, sensors like ultrasonic camera or infrared are base. Robot navigation can be realized by different kinds of algorithm. All sensors, actuators and algorithms must communicate each other. Robotic constructor must be familiar with lot of technologies for creating complex robotic system. In the last time, research and development in robotic systems is focused on unit software architecture – middleware, where different devices, software algorithms modules, control units can talk together with common language – protocol. An advantage of uniform software is modular programming and possibility of code reusing. Modules communicate with other directly through inter-process communication on one processor or through different busses like Ethernet. Presented communication platform permits add or remove modules and devices while running system. The core of architecture is implemented in C language and so it can run on 8bit microcontroller with 1kB RAM and 16kB ROM. Code has only few hardware depended function, (communication, timer and loader) and can be easily ported to many platforms. All devices are precisely time synchronized, with 10us precision. Due to that the sensing of different sensors placed on different devices or different robots can be done and data can be fused from different places. This advantage can be used for sound processing or sound localization. Devices can also communicate wirelessly. Therefore multi-robot communication and swarm robotic can be easily created. Next paragraph roughly describes existing open-source robotic middleware initiatives.

Existing robotic middlewares

There are many robotic middleware, control architectures and development tool kits in use toady. Lot of them is freely available and distributed as open-source. Some open robotic middleware capable of implementing multi mobile robot system are described in the new few lines of this paper. The Orocos [2] project - middleware was developed specifically for industrial robots as platform for building modular framework for robots and machine control. The middleware is organized in to four libraries- i. the realtime toolkit (RTT), ii. Kinematic and Dynamics library (KDL), iii. The Bayesian Filtering library (BFL), iv. Orocos Component library (OCL). These libraries are implemented in C++. Another important toolkit is CARMEN [3], it is one of the most extensively used software in robotic research. It offers distributed collection of modules organized as three layered architecture. The hardware management layer/ base layer govern components interaction and control as well as presenting abstract interfaces to base and sensor systems. Navigation primitives are handled by the Navigation layer while the top layer is for high level task implementation. Modules communicate with each other over IPC (Inteprocess communication) protocol. This tool did not make the cut because it tailored for single robot systems. Humanoid robot uses YARP (Yet Another Robot Platform), is a multi-platform open-source framework that supports distributed computation. The main design focus is on robot control and efficiency [4]. It provides a set of protocols and a C++ implementation for inter-process communication on a local network. The ROS (Robot operating system) specification is at the messaging layer. Peer-to-peer connection negotiation and configuration occurs in XML-RPC, for which reasonable implementations exist in most major languages [5]. To support cross-language development, ROS uses a simple, language-neutral interface

definition language (IDL) to describe the messages sent between modules. The IDL uses (very) short text files to describe fields of each message, and allows composition of messages. Code generators for each supported language then generate native implementations which “feel” like native objects, and are automatically serialized and deserialized by ROS as messages are sent and received. ROS is designet as microkernel, where a large number of small tools are used to build and run the various ROS components. Urbi [6] is fully-featured SDK with environment for programming to orchestrate complex organizations of components. It relies on a middleware architecture that coordinates components named UObjects. The URBI platform is based on client/server architecture, built on top of a new programming language called URBI. Urbiscript is a scripting language that can be used to write orchestration programs. CORBA, RT-Middleware or openHRP objects can also be seen from within URBI, like in fact most existing distributed object architecture, thus making URBI a central platform to integrate various technologies.

Protohreads & modules scheduling

Protothreads are extremely lightweight stackless threads designed for severely memory constrained systems [7]. They provide linear code execution for event-driven systems implemented in C, sequential flow of control without complex state machines or full multi-threading and work under or without an underlying operating system. Only two bytes of memory is needed to work with protothreads. This attribute bring hardware independency and easy thread schedule. Because no thread stack is present, variables must be defined as static. Middleware is based on protothreads, where each module own one or more protothreads. Next figure describes modules scheduling functionality. Thread manager runs Pthread1 – module1 main function. Each protothread function starts with PT_BEGIN() statement and finish with PT_END(). Pthread1 runs while PT_WAIT_WHILE(condition1) statement isn’t reached. If condition1 is true, Pthread1 is released returns and afterwards thread Ptrhead2 bonded with module 2 starts running. Because this tread waits in last scheduling cycle, Pthread1 switch to PT_WAIT_WHILE(condidtion2) and check condition. If condition is false, thread runs and finish with PT_END(). This mechanism is implemented with swich statement and therefore, no code overhead is added. Each module has own memory space allocated with init function and its thread operates under this allocated space.

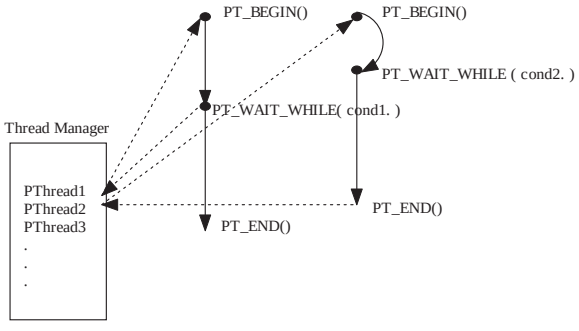


Fig.1 Modules scheduling

Communication principle and data binding

Modules communicate by sending messages. At first, each module must have set its own address. The Address is composed of device address (specific hardware board or software board) and the last 8 bit define specific module. Afterwards binding of module’s data and synchronization should to be done. The mask address manager is the core of binding system, which monitors whether there are data assigned to specific module. Each module have DataIn buffer, where data are stored in memory as structure defined by C programming language. These data are described by special definition language and are written in JSON format. Figure 2 describe data set process. Device gets data with specified address of sender. If the address matches in MASK_ADDR_MANAGER, data set process is initiated. The data payload from the specific position of the data stream and the data length is taken from the receiver buffer. This data are copied or set by specific operation (Reset, And, Or, Xor) to DataIn module memory space at specific position. Positions and length are defined in process of data binding with help of configuring software. The variable HasData of module indicate new data and afterwards data operation should be done in module thread. When module wants to send data, communication new packet is created. But when data needs other module inside one microcontroller (or board), there is no need of data packet. Only specific data are transported through MASK_ADD_MANAGER. For configuring and specific operations have modules command functions.

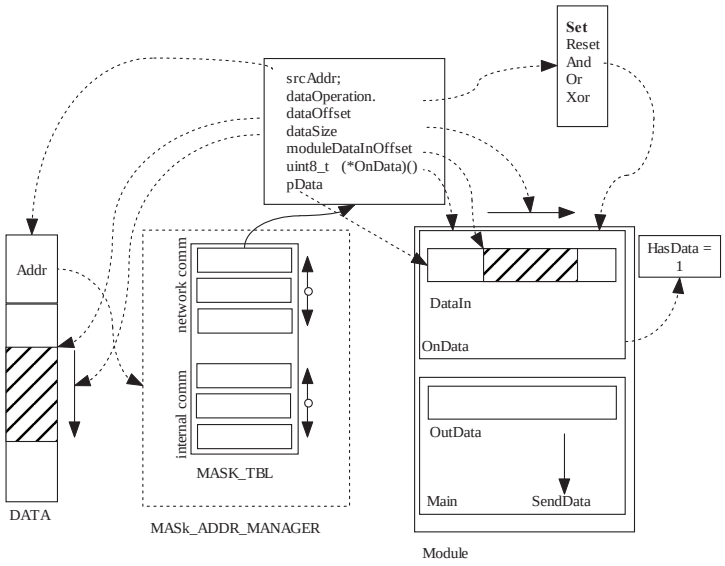


Fig.2 Module data set & data binding principle

Conclusion

Lightweight middleware presented in this paper make use of protothreads to create robotic modular system. Protothreads adds new view to programming modular systems and enables programming system to be independent of hardware architecture. Currently, the presented system works on AVR (8bit) and x86 computers under windows operating system. Within or future work we will tray to put architecture to ARM platform and write code in C++ for high power computers. There is no support for scripting languages today. Development of complex robotic system requires lot of effort and therefore this architecture have success only if robotic teams will start to program modules for this architecture. This platform has some advantages: it is scalable, multiplatform without need of operating system; time synchronized; and so can be used for other applications in the industry. Another remarkable aspect which can be remarked is that,

this system can be used for swarm robotic. Swarm robots are low cost robots composed of low power microcontrollers. This architecture is developed for lower expense for creating robotic system, where computing can be put onto low power microcontrollers. It might seem that minimalistic middleware architecture based on protothreads influences module programming comfort. But in the reality, it is easy to make new module.

Referenes

[1] Molaletsa Namoshe, N S Tlale, C M Kumile, G. Bright, Open middleware for robotics

[2] H. Bruyninckx. "Open robot control software: the OROCOS project," In IEEE International Conference on Robotics and Automation (ICRA'01) vol. 3, pp 2523–2528, 2001. [Online: http: //www.orocos.org].

[3] M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2436–2441, 2003.

[4] G. Metta, P. Fitzpatrick, and L. Natale. "YARP: yet another robot platform," International Journal on Advanced Robotics Systems, vol. 3, No. 1, pp. 43–48, 2006. [Online: http://yarp0.sf.net].

[5] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In International Conference on Robotics and Automation, 2009.

[6] Jean-Christophe Baillie. Design principles for a universal robotic software platform and application to URBI. In Davide Brugali, Christian Schlegel, Issa A. Nesnas, William D. Smart, and Alexander Braendle, editors, IEEE ICRA 2007 Workshop on Software Development and Integration in Robotics (SDIR-II), SDIR-II, Roma, Italy, April 2007. IEEE Robotics and Automation Society.

[7] Adam Dunkels, Oliver Schmidt, Thiemo Voigt, and Muneeb Ali. Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems. In Proceedings of the Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), Boulder, Colorado, USA, November 2006.

Ing. Michal Bachratý

Slovak University of Technology in Bratislava, Slovakia
Faculty of Electrical Engineering and
Information Technology
Institute of Control and Industrial Informatics
ICII (URPI) FEI STU Ilkovicova 3,
812 19 Bratislava, Slovakia
michal_bachraty@stuba.sk

prof. Ing. Peter Hubinský, PhD.

Slovak University of Technology in Bratislava, Slovakia
Faculty of Electrical Engineering and
Information Technology
Institute of Control and Industrial Informatics
ICII (URPI) FEI STU Ilkovicova 3,
812 19 Bratislava, Slovakia

Simple juggler manipulator

Stanislav Triaška, Milan Žalman

Abstract
The article deals with the description of manipulator whose task is to juggle - throwing and catching manipulator. The juggler manipulator is an open loop system without the feedback information about the actual position of the flying object. The core of the movement system is the position control of a servo drive. There is a mathematical model to set up desired parameters which are generated by the Master Generator to servo.

Key words: Juggling, Position Control, Master Generator, Movement equation of juggling system,

Introduction

One of the possible ways how to use servo drives in attractive way is a juggling with flying objects. Those systems whose task is to juggle with objects, we called juggling manipulators or simply the juggler. The throwing and catching task using servo drive can be realized on many different ways.

Fig. 1: The Ball Juggler

Comparing the juggler system – which are in details described in reference, their physical construction and the way it works we can divide into few categories:

Degree Of Freedom (DOF):
- 1 DOF – these kinds of manipulators are the most simplest; the physical construction is made by one servo drive; they can juggle only in one surface (x,y plane).
- 3 DOF – these kinds of manipulators can usually do more complicated kinematics task like 1 DOF manipulators.

Workspace:
- 2D workspace – manipulators which can juggle only in one surface (x,y, plane).
- 3D workspace - manipulators whose task is to juggle in x,y,z planes.

Actual information of flying object:
- Closed loop systems - using the feedback information about actual object position (mostly used industrial high speed camera).
- Open loop systems - working without the actual information of the flying object, the basis of these kinds of systems are a mathematical model.

Juggling algorithm:
- Simple algorithm – throwing and catching algorithm, juggling in vertical axes
- Cascade algorithm- one of standard algorithms to juggling in 3D workspace[7]
- Shower algorithm - one side of the manipulator is to throw an object and another side is only to catch the object
- Another algorithm – based on different mathematical models like those before.

These papers describe our method to juggle with the servo drive system with one ball in one DoF (x,y surface) using a simple juggling algorithm which is shown on Fig.4. The juggler is designed with a juggling arm, whose task is to throw the ball, and catch the flying ball. The ball juggler system is powered by one servo drive. The system of the ball juggler is shown on Fig. 3.

The papers are organized as follows:

- Physical and mechanical description and communication between individual devices is described in section I in detail. The control implementation ball tracking regulation and servo-drive position regulation are also included in this section of the article.

- Setting up desired waveform parameters of the Master generator, the mathematical model of the simple juggling algorithm and the juggling algorithm is a part of section II.

- The experimental part of the work and some results are shown on the section III.

- Some topics for the future work, conclusion and references are placed in the end of these papers.

I. The juggling system

The juggler system consists of six separate devices, which are connected between each other. The illustration of the devices connection is shown on Fig. 2.

The whole system is controlled by PC, by using program Matlab/Simulink. Communication card MF624 is installed on PC board and program Matlab can communicate with this card using the Real Time Toolbox. DGV700 frequency converter can be configured to accept external torque reference provided by communication MF624 card. This torque refer-

ence is output signal from the card and it is on range ±10V. This card is also used to process position signal from the converter which indicates the actual juggler shoulder position. The actual arm position is given by the resolver which is part of the used drive. We used interface card build on frequency converter, which can emulate the incremental encoder-like the signal on its output. The drive shaft is connected by the MPV 02 planetary gearbox, whose task is to increase the torque from PMSP brushless servomotor. Transmission of this planetary gearbox is 1:10 that means shaft of the servo is ten times faster than gearbox shaft. The juggling arm is fixed on the gearbox shaft.

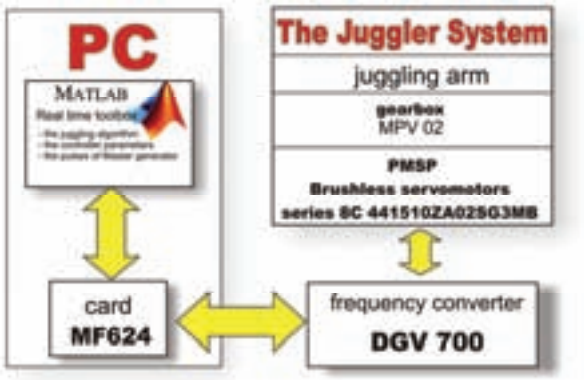


Fig. 2: The system hardware and devices connection

The mechanical ball juggler is the movement system without the actual feedback information about the actual ball position. The control implementation and all juggling algorithms are based on the servo drive system position control. We made the model of the juggler in Matlab/Simulink.

The juggler system is controlled by PIV controller. The input to the system is given by 4D master generator, whose task is to generate a signal to throw the ball with the arm, and generate the signal to catch the flying ball. Controlling with 4D master generator is called feedforward control. The base of the feedforward (Master-Slave) control can be classified as stock control or the control through the model.

Fig. 3: The mechanical ball juggler

The task of master generator is to generate the desired waveforms of the state variables to make the juggling algorithm. In our case state variables are (1.1): φ^* position, $\omega^* = \frac{d\varphi^*}{dt}$ velocity, $\varepsilon^* = \frac{d\omega^*}{dt}$ acceleration, $\tau^* = \frac{d\varepsilon^*}{dt}$ jerk.

$$\mathbf{V}^* = \left[\varphi^*, \omega^*, \varepsilon^*, \tau^*\right] \tag{1.1}$$

The method of pole-placement is one of possible methods to design parameters of PIV position controller. This means that by desired dynamics of system we choose the poles of the characteristic polynomial. By comparing the characteristic equation of closed control circuit of the actual transfer (1.2) and the transfer of the desired polynomial (1.3) in the same square we get parameters of the regulator (1.4).

$$G(s) = \frac{M(s)}{N(s)} = \frac{\frac{I.P}{J}}{s^3 + \frac{B+V.}{J}s^2 + \frac{I}{J}s + \frac{.I.P}{J}} \tag{1.2}$$

$$G^*(s) = \frac{M_0(s)}{N_0(s)} = \frac{k\omega_0{}^3}{\left(s^2 + 2\xi\omega_0 s + \omega_0{}^2\right)\left(s + k\omega_0\right)} \tag{1.3}$$

$$\begin{aligned} V &= (2.\xi + k).\omega_0.J - B \\ I &= (2.\xi.k + 1).\omega_0^2.J \\ P &= \frac{k.\omega_0}{(2.\xi.k + 1)} \end{aligned} \tag{1.4}$$

- J -Moment of inertia of the system
- B -Viscous friction
- ξ - Dumping coefficient
- k - Shift of the pole on the real axes
- ω_0 - Bandwidth
- P,I,V – Parameters of position controller

To control a flying ball we need to design a desired signal of the state variable generated by the Master generator.The state variable has to correspond to movement equations of the mathematical model of the transversal throw.

II. Simple juggling algorithm

The juggling algorithm is based on the mathematical model of the movement equations which are described by the Newton's law of motion. In this section we will show a movement of the ball in 2D surface.

The descriptions of these equations come from the physical model of the flying object – transversal throw, which can be explained by the Fig. 4 (c,d).

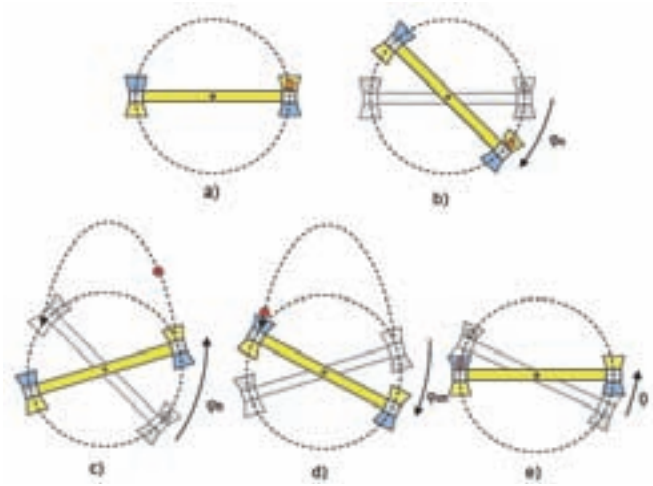


Fig. 4: Simple juggling algorithm (throwing and catching algorithm)

Assuming the ball is moving in the gravitational field of the well-known starting position [x0, y0], in the parallel surface with the vector of gravity acceleration. To simplify the model

we did not take into account the effect of the air resistance on the flying ball. II Newton motion law for x, y coordinates is given by the (2.1) equation.

$$m\frac{d^2\mathbf{r}}{dt^2}=\mathbf{F}+\mathbf{G} \qquad \begin{aligned} x:&\quad m\frac{d^2x_T}{dt}=ma_x \\ y:&\quad m\frac{d^2y_T}{dt}=ma_y+mg \end{aligned} \tag{2.1}$$

Solving these two equations we get the actual ball position and the velocity of the flying ball.

$$\begin{aligned} x(t)&=\int v_x dt=|v|.\cos(\varphi).t \\ y(t)&=\int v_y dt=|v|.\sin(\varphi).t+\frac{1}{2}.g.t^2 \end{aligned} \tag{2.2}$$

Comparing the juggling algorithm we can divide algorithm into five separate parts:

- Juggler is in stationary state (Fig. 4.a)
- Swing to starting position (Fig. 4.b)
- Accelerate with the arm and throw the object (Fig. 4.c)
- Catch the flying object (Fig. 4.d)
- Decelerate the movement of the flying object, and swing to the stationary position (Fig. 4.e)

For all of these juggler states we need to design desired waveform parameters of the servo drive. On the Fig. 5 it is shown the way to set up parameters of the Master generator to throw the ball from the starting position to the final catching position.

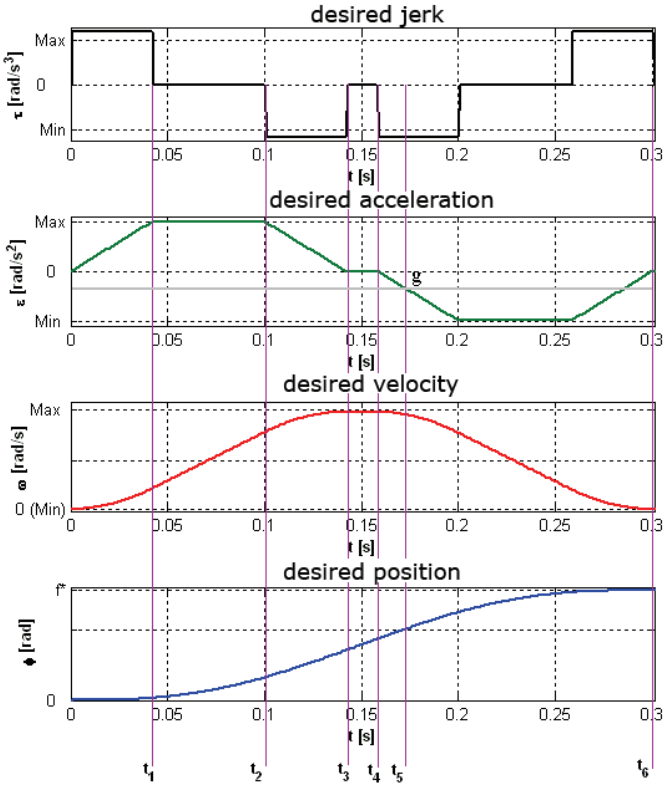


Fig. 5: Setting desired waveform Master parameters

The arm movement can be divided into six parts:

- The first part $0 \le t \le t_1$, arms start to move (start to accelerate), compare with Fig. 4.c
- Arm moves with constant acceleration $t_1 \le t \le t_2 \quad \varepsilon = \varepsilon_{\max}$
- Setting time to maximum velocity $0 \le t \le t_1,\ \omega_{34} \le \omega \le \omega_{\max}$
- Moving with constant velocity $t_3 \le t \le t_4 \quad \omega = \omega_{\max}$
- The throwing ball time, and throwing angle $t = t_5;\quad \varphi = \varphi_5$
- Desired master angle setting $\varphi = \varphi^*$

The ball has left the basket in the time when the arm starts to accelerate in reverse direction to set the desired position, when generator is set on value $\varepsilon = -g,\ \varphi = \varphi_5$.

The desired master angle can be set as:

$$\varphi^* = \varphi_5 + (\varphi_3 - \varphi_{45}) \tag{2.3}$$

Or: $\varphi^* = 2\varphi_3 + \varphi_{34}$ (2.4)

Where: - φ_5 is a desired angle for throwing the ball

$$\varphi_3 = \varphi_1 + \varphi_{12} + \varphi_{23} \tag{2.5}$$

$$\varphi_1 = \iiint_0^{t_1} \tau(t)\,dt = \frac{1}{6}\tau_{\max} t_1$$

$$\varphi_{12} = \int_{t_1}^{t_2} \omega_1 dt + \int\int_{t_1}^{t_2} \varepsilon(t)dt = \omega_1(t_2 - t_1) + \frac{1}{2}\varepsilon_{\max}(t_2 - t_1)^2$$

$$\varphi_{23} = \int_{t_2}^{t_3} \omega_2 dt + \int\int_{t_2}^{t_3} \varepsilon_2 dt + \iiint_{t_2}^{t_3} \tau(t)\,dt =$$

$$= \omega_2(t_3 - t_2) + \frac{1}{2}\varepsilon_2(t_3 - t_2)^2 + \frac{1}{6}\tau_{\max}(t_3 - t_2)^3$$

Generator times are:

$$t_1 = \frac{\varepsilon_{\max}}{\tau_{\max}}; \qquad t_2 = \frac{\omega_{\max}}{\varepsilon_{\max}}; \qquad t_3 = t_1 + t_2$$

$$\varphi_{34} = \int_{t_3}^{t_4} (\omega_0 + \omega(t))\,dt = \omega_{\max}(t_4 - t_3) \tag{2.6}$$

From equation (2.5) to express φ_{34} and equal with (2.6) we get the t_4 time parameter (2.7), and t_5 - throwing time parameter.

$$t_4 = \frac{\varphi^* - 2\varphi_3}{\omega_{\max}} + t_3 \tag{2.7}$$

$$\varphi_{45} = \omega_{\max}(t_5 - t_4) - \frac{1}{6}\tau_{\max}(t_5 - t_4)^3$$

$$\varepsilon_{45} = \int_{t_4}^{t_5} \tau(t)\,dt = -\tau_{\max}(t_5 - t_4) \tag{2.8}$$

$$t_5 = \frac{\varepsilon_{45}}{-\tau_{\max}} + t_4;\ \varepsilon_{45} = -g;$$

Those are all the necessary parameters needed to set 4D master generator to throw the ball to desired angle with desired velocity. The desired waveform Master parameters needed to catch the flying object we designed the same way like the parameters for throwing the object.

III. Experimental results

Juggling and master generator were tested on the juggler system with two juggling algorithms. The basis of these two algorithms was simple juggling algorithm for throwing and

catching a flying ball. The first algorithm was simple throwing from one side of the juggling arm and catching a flying ball on the other side of manipulator (Fig. 6 and Fig. 7).

The second algorithm was the same as the first one but with the arm rotation of 180°.

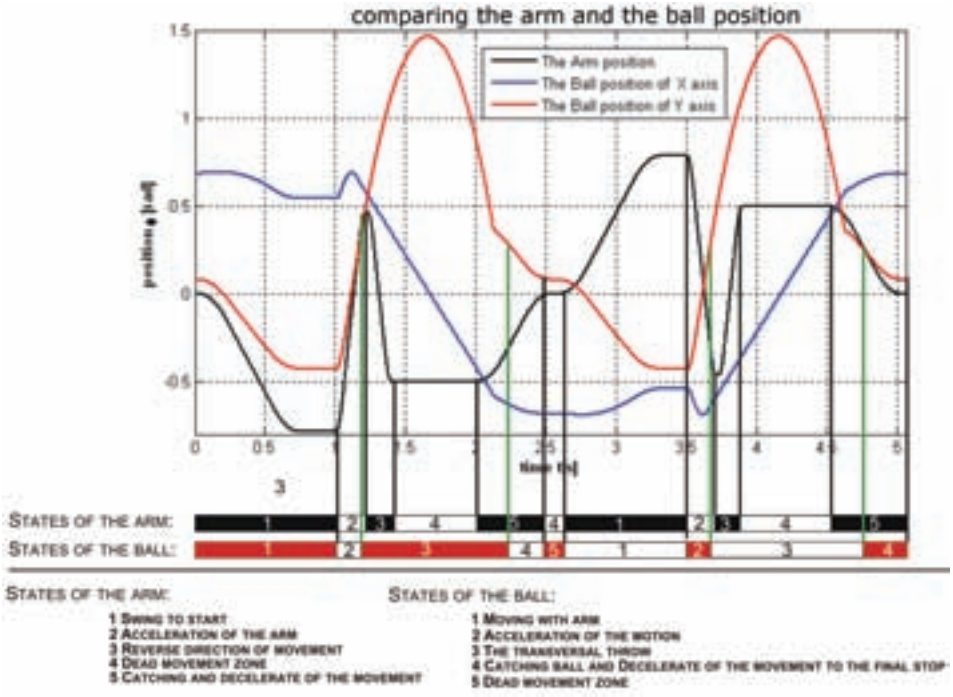


Fig. 6: Simple juggling algorithm

The juggler arm swings with a slow motion to the start position in the time: $0[s] \le t \le 0.7[s]$ with low velocity $\omega_{\max} = 1.5[rad/s]$.

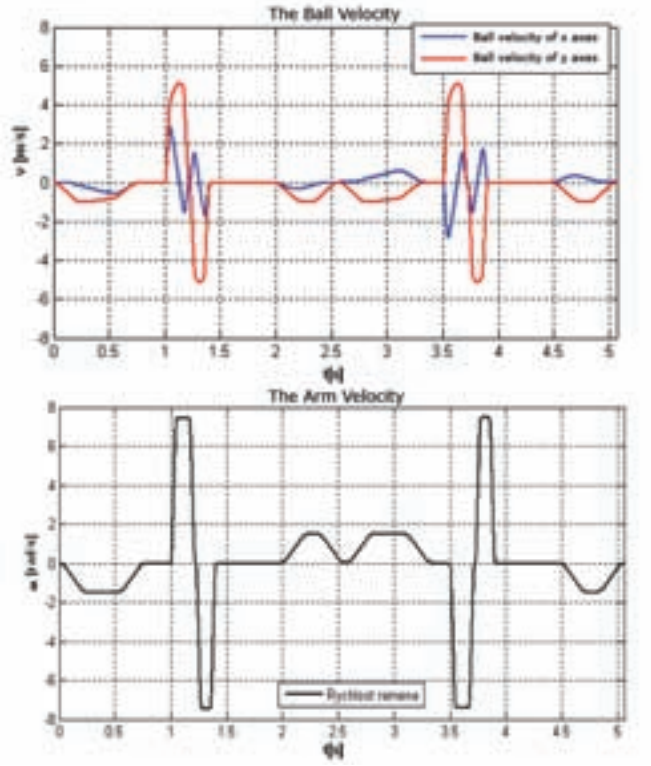


Fig. 7: Comparing the velocities

Comparing this graph and picture shown on Fig. 7, we can see that the maximum velocities of the flying ball are in the times $t = 1.33[s]$, $V_X = 5.1[rad/s]$, $V_Y = 0[rad/s]$, and at the $t = 3.63[s]$. In these times the ball is thrown from the right and left side of the juggler shoulder. The desired velocity for throwing the ball was set on value $\omega^* = 5.1[rad/s]$

The juggling algorithm with the arm rotation of 180º is shown on the Fig. 8. The velocity of the juggler arm is shown on the first graph. Comparing the actual positions between the ball and the juggler shoulder is shown on the graph below in the same picture.

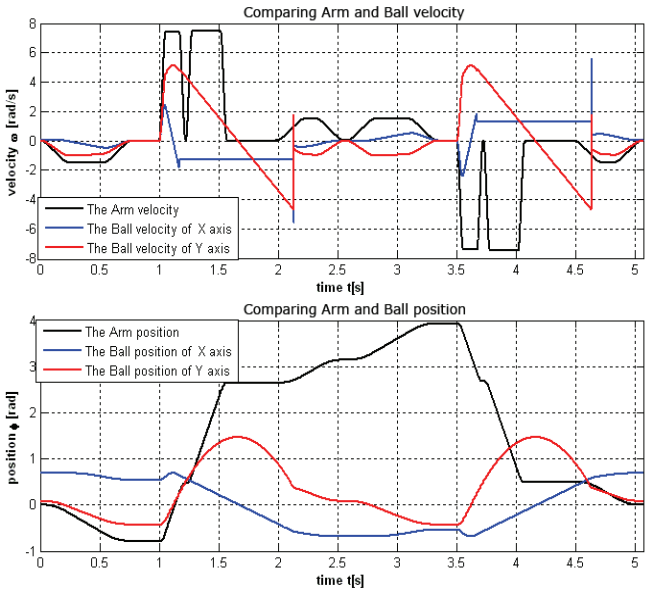


Fig. 8: Juggling algorithm with arm rotation of 180º

The ball is thrown and left the basket at $t = 1.17[s]$, after 0.1s the arm starts to rotate 180º and catches the flying ball at the same time as in the first algorithm (t=2.15s).

Conclusion and future work

The papers described juggling manipulator whose task is to simple juggle with the ball in x,y surface. The manipulator is made by 1DoF. The base of the juggling manipulator is the positional control of the servo drive system, which manipulator is powered by. The simple juggling algorithm comes from transversal throw, and can be described by II Newton's low motions. In experimental parts of this work we show some simulation results, which were tested on the real juggler manipulator. The ball position and velocity were numerically calculated because we don't have any feedback information of flying object.

The future work will be more focused on intelligent control of the ball juggler, and advanced juggling algorithms.

Acknowledgment

Since this work has been supported by the project VEGA reference number is: 1/0690/09. Therefore the authors of the article would like to give thanks for the support. This support is very greate-fully acknowledged.

Reference

[1] S.Triaška, M.Žalman "The control of the ball juggler", In 18th Telecommunications Forum TELFOR 2010: Belgrade, Serbia, 23.-25.11.2010. Belgrade: Telecommunications Society, 2010

[2] P. Burget, P. Mezera "A Visual-Feedback Juggler With Servo Drives", The 11th IEEE Internacional Workshop on Advanced Motion Control, March 21-24, 2010, Nagaoka, Japan

[3] A. Nakashima, Y. Sugiyama, Y. Hayakawa, "Paddle Juggling of one Ball by Robot Manipulator with Visual Servo", Control, Automation, Robotics, and Vision, 2006. ICRAV '06 .9th International Conference on 5-8 Dec. 2006

[4] B. B. Amor, N. K. Haded, F. Mnif, "Controllability Analysis of 1-DOF linear juggling system", 2009 6th International Multy-Conference on Systems, Signal and Devices, 2009

[5] S. Schaal, Ch. G. Atkenson, "Open Loop Stable Control Strategies for Robot Juggling", Robotics and Automation, 1993. Proceeding., 1993 IEEE International Conference on 2-6 May 1993

[6] A. Akbarimajd, "Optimal Cyclic Vertical Juggling Using 1-DoF Arm", Proceeding of the 2009 IEEE International Conference on Robotics and Biomimetics, December 19-23, Guilin, China

[7] P.J..Beek and A.Lewbel, "The Science of Juggling", *Scientific American,* vol. 273,pp 92-97, 1995

[8] T..Tabata and Y.Aiyama, "Tossing Manipulation by 1 Degree-of-freedom Manipulator", Proceedings of the 2001 IEEE/RSJ, International Conference on Inteligent Robots and Systems, Maui, Hawaii, USA, Oct 29- Nov. 03,2001

[9] T..Tabata and Y.Aiyama, "Passing Manipulation by 1 Degree-of-Freedom Manipulator", Proceedings of the 5th IEEE, International Symposium on Assembly and Task Planning, Besançon, France, July 10-11,2003

Ing. Stanislav Triaška
prof. Ing. Milan Žalman, PhD.

Institute of Control and Industrial Informatics
Slovak University of Technology in Bratislava,
Faculty of Engineering and Information Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
e-mails:
stanislav.triaska@stuba.sk
milan.zalman@stuba.sk

Service robots

František Duchoň, Ladislav Jurišica, Marián Kľúčik, Anton Vitko

Abstract
The paper presents a definition of a service robot, describes classes of service robots, their applications, components and basic control methods. Besides, the statistics of their applications and the concrete realization of the service robot are included.

Key words: Service robot, localization, navigation, subsystems

Introduction

Robots can be classified into various classes depending on their construction and utilization. One of possible classification distinguishes only two main classes: the industrial and service robots. The industrial robot is characterized as a multipurpose programmable manipulator with an automatic control of three or more motion axes, which can be fixed either on the ground or on a mobile platform. (Robotic Industries Association [1]). The service robots may be further divided as personal or professional ones. Internal Federation of Robotics [2] characterizes mobile robot as a robot which in a semi or fully autonomous regime provides a service useful for people and devices but without manufacturing operations. The tasks are often dangerous, uniform, dirty or "dull".

Personal service robots (households, aid, entertainment...) constitute a new part of service robotics in which the area of the so called socio-interactive robotics is highlighted. According to the one of several definitions, a social robot is an autonomously moving mechanism equipped with sensors, actuators and interfaces communicating with people in accord to the protocols reflecting both the robot and environment properties, while regarding the people's needs.

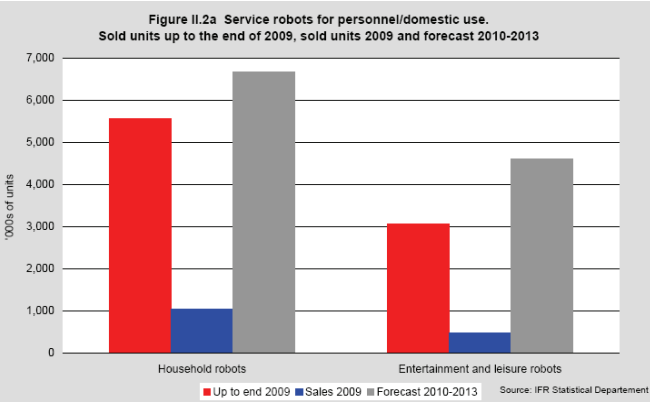


Fig.1 Statistics of the number of service robots (IFR Statistical Department).

Applications of service robots

The paper is focused on the so called professional service robots, where the issues of personal robots and requirements of the social robotics are only partially included. The possible applications of the service robots (Robotics & Automation Society [3]):

Cleaning & Housekeeping	Edutainment
Humanoids	Humanitarian Demining
Rehabilitation	Inspection
Agriculture & Harvesting	Lawn Mowers
Surveillance	Medical Applications
Mining Applications	Construction
Automatic Refilling	Guides & Office
Fire Fighters	Picking & Palletising
Food Industry	Search & Rescue

The service robots are complex technical devices able to interact with the environment. [4] As a rule their constructions resemble mobile manipulators. The basic task of the service robotics rests in the robot localization and navigation in a space. To fulfil the definition there is a needed to equip the robots with sufficiently complex sensor subsystem and powerful control system. The necessary parts are a mobile platform and a manipulation mechanism. Variability of actual realizations is related to the robot's application area. [10, 11] Regarding the appearance and abilities, the humanoid robots are similar to humans and they are frequently considered to be on the top of service robotics. One of them, the robot Robonaut 2 is prepared for use on orbital station ISS.

The localization and mapping belong to the basic problems of the service robotics. [6, 7, 8, 9, 11] Knowledge of the robot localization and representation of the environment is needed in many applications. The task is solved differently for out-door and in-door robotic applications. Which solution is adopted depends on the environment complexity and the way of data representation. A suitable method is the SLAM method (Simultaneous Localization and Mapping). It uses Kalman filter, Markovian localization and precise GNSS (Global Navigation Satellite System).

For the task of motion control inside the environment the local or global navigation methods may be applied. While the reactive (or reflex) control is mostly considered for local navigation, the global methods allow for planning and optimization. Putting both methods together may set up prerequisites for an optimal operability with respect to the changing environment.

Further widening of possible applications of the service robotics rests in using them in the groups of cooperative robots. Such systems allow for mutual learning and, in such a way, reaching better qualitative behaviours. Such a group then operates on the basis of so called distributed intelligence. Possible inspirations can be taken from biological systems, theory of artificial life and the like.

The operating industrial robots are commonly separated from the people by fences. Contrary to that, the service

robots operate in environments where the presence of people is natural. This calls for need of taking special measures with regard to choice of the sensors, control system, but also the whole structure of robot control.

The Figs. 2 and 3 show the trends in service robotics with respect to the number of installed robots and their technological level.

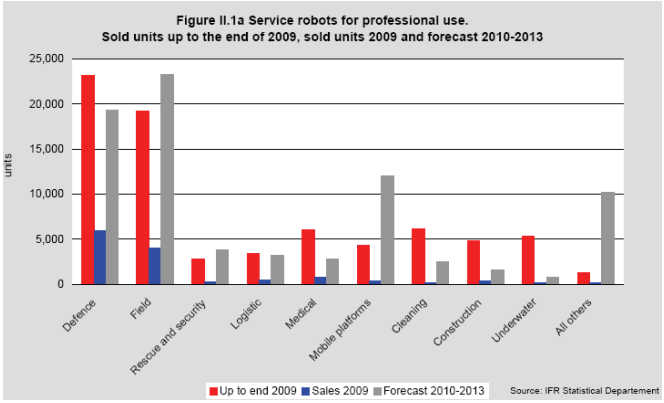


Fig.2 Number of profesional service robots in various areas (IFR Statistical Department [3]).

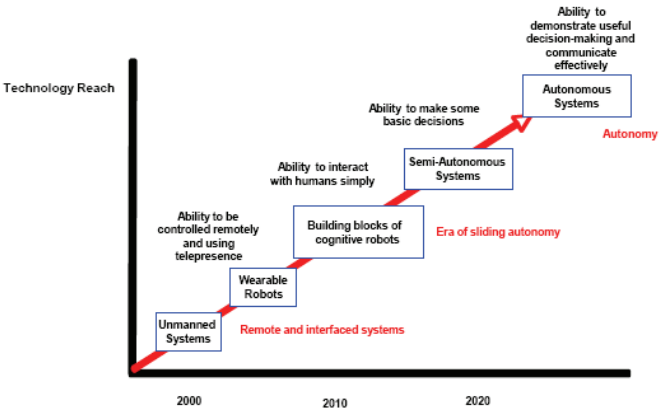


Fig.3 Trend in service robots technology (SRI Consulting Business Intelligence [5]).

Control tasks of a service robot

Considering the control view, the service robots can work either in a partially autonomous or fully autonomous regime, while from the energetic view they commonly work autonomously. The working regime determines complexity of the sensor, control and communication systems, as well as the methods and algorithms used. The fully autonomous systems can solve the most complicated tasks, but their hardware and software may become extraordinary complex and the cost may be high.

A compromise, which is in praxis often absolutely sufficient, rests in using a tele-control with automated routine tasks. The simple systems can work on the basis of local information about the environment. But, for the reasons of comprehensive information required by the operator, the better systems are those which work with global information, namely the global map. The map may be known in advance, but in the most instances the system builds it on the basis of sensor information. Another variation may use landmarks that simplify the robot localization and navigation.

The common sensors allowing environment mapping are the odometry, ultrasound, infrared and visual sensors, laser rangers, accelerometers, GPS measurement systems and gyroscopes. The information fusion also comes into play.

The communication operator – robot can be done by a joystick, keyboard, voice, eye or head movement etc. Also emotionally based communication, like the gestures or the voice intonation may come into consideration. For the system robot-operator and robot-robot also other types of communication may come in handy.

The environment state is commonly visualized on a monitor, stereo glasses and other devices. In case that both the environment map and sensor information is used, the so called mixed reality, aggregating various pieces of information into integrated information is a recommended solution. In spite of it, the robot control requires significant experienced operator. Therefore, from the aspect of the operational safety and certainty, the systems securing an autonomous observation of the robot moving towards the goal position in a cluttered environment are demanded. Such autonomously operating systems make the operator tasks noticeably easy and increase the global system safety.

Regarding the reliable functioning, an important group of sensors belongs to the introceptive sensors delivering information about internal variables, like voltages, currents, temperatures, inclinations and the like.

Description of the service robot MR VK

The service robot is commonly built on a mobile platform. The hardware configuration consists of mobile robot MR VK and the operator station with a powerful PC. No additional computer is installed. The only additional equipment is gyroscope Crosbow IMU330CC, which is used for accurate determination of the robot location. The robot is manipulated from operator's station through the wifi net. The mobile robot together with the installed control computer works as a server to which it is possible to connect and, in accordance to the defined protocol, the operator can deliver various instructions. The video signal is sent in an analogue form into operator's computer where, using a videograber, it is further processed by a higher control level. Higher control level can either visualize the image and sensed data to operator or perform the data processing per se.

The control is divided into two layers. The control system on the low layer has implemented means for measuring odometric data, parsing data form laser scanners, switching on/off the instruments, controlling robot motion, stopping the robot in an automatic regime and routing/switching signals. Within the data parsing it is performed a simple filtration. There are pre-prepared algorithms for measuring odometric data coming from the laser scanner. From among the motion instructions the following ones have been implemented:

- Motion to XY coordinates in accordance with a given coordinate system
- Setting the robot velocity and speed difference between left and right robot side (the robotic platform is controlled by the wheel slip)
- Setting the speed of the left and right side
- Setting the distance and the direction of the motion

On the top of this, the following commands are implemented:

- Zeroing the relative position
- Zeroing the current rotation
- Request for the data
- Instrument calibration
- Switch AUTO/MANUAL
- Governing the camera

The control system of the high level processes and sends signals from the operator or module of autonomous control to the low level. So far the module of autonomous control

has been switching on exclusively by the operator and the robot performed some planned automatic operation (at present only the robot motion initiated by the commands listed above). Various supporting functions are implemented and tested in the operator module. The visualization of all data coming from the instruments as well as the image visualization at the operator's panel (created by a PC equipped with a card for image digitization) is also possible.

The second complex module is a module of automatic control. It allows for controlling the robot function. The operator may or may not track the robot activities. In case of non-appropriate intervention the robot can require a correction from operator or stops functioning. The following variants were realized:

- In case of the transfer based on inner coordinate system the robot control runs within the framework of the own coordination system. The task is implemented into the own sub-module while considering data from other sub-modules. The functions of signalization of the presence of an obstacle, the robot stop in front of the obstacle, the signalization of reaching a specified location and the obstacle avoidance are activated in the autonomous regime. The transfer from the site A to site B w.r.t the defined coordination system was also implemented. The system is prepared for implementation of a state automaton which would be able to respond to certain situations.
- In case of the transfer based on the map the implementation of an appropriate module is conditioned by the successful solution of some problems. These problems have been solved within their own sub-modules, like the module of planning trajectory in a local map, the module of localization in the map based on odometric measurements of several instruments and the localization module based on a visual system. The system utilizes a simple mapping. The endeavour focused on upgrading the mapping algorithms was exerted and the basis of the environment representation module was laid.

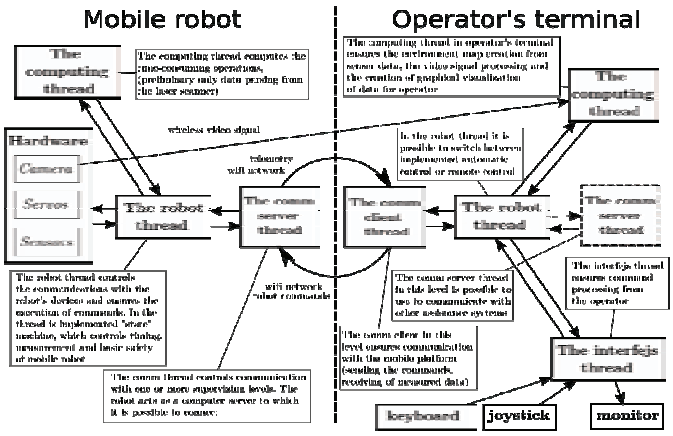


Fig.4 Block scheme of service robot

In fig. 4 there are depicted the following blocks:

The hardware - a block specifying the configuration. Every hardware element in the system of the mobile robot is implemented as an independent object, which, in need can be easily removed from the main programme. The following objects were implemented: the gyroscope Crosbow IMU330CC, the bus RS422 (as a basic form of robot control), laser scanner HOKUYO, and control of camera SONY, GPS (various configurations of the sentences from the sensor are considered), compass OceanServer 5000. The hardware classes are based on the possibilities of communication with hardware. One can say that every single cable represents a class, therefore e.g. within the class RS422 the communication with motion drivers (MotorControlBoard right and left), SensorBoard, MainControlBoard is completely elaborated. The defined classes are parts of the class Robot and their methods are exclusively called from the thread Robot.

The thread Robot of lower level - within which the time measurement, communication with hardware and data distribution into particular levels is solved. In this thread it is solved also the state machine of the mobile robot, which still should be revised from the aspect its possibility to implement elements of artificial intelligence. Various states of the mobile robot were considered and the following three of them have shown to be relevant:

- The initialization state - the hardware is tested whether it communicates or not. If it communicates it is configured with respect to the actual control purposes and subsequently it is tested for delivering expected data.
- The state of automatic/manual control – it is a set of states in which the commands of the high level are expected, namely the motion commands, camera rotation, and on/of switching the instruments. To every activity within which there are solved various problems is assigned a particular state. The current implementation does not allow for the robot to be in various states at the same time. From the aspect of reliability that currently seams to be a passable solution.
- The de-initialization state – it is mostly employed in case of the weakly charged battery. All instruments are switched off and battery should be exchanged.

The thread Comm-server of lower level – the communication protocol for communication with higher level. At present only one connection is possible, but in the future it can be widened to several connections. For instance, through one port it will be possible to send control commands to the mobile platform and through another to control other devices, let say the camera.

The thread Computing of the lower level accepts raw data from the laser scanner which are parsed in the thread. A method for odometry of laser scanner data has been implemented.

The thread Com-client of higher level implements a communication protocol with lower level. Possible communication blackouts are indicated to the operator.

The thread Robot of higher level – similarly to lower level, solves the communication with potential hardware (GPS and supplementary laser scanner). Also the state machine is here designed similarly to lower level.

The thread Interface with blocks joystick, key board, monitor displays pieces of information on the monitor and processes inputs form operator (joystick, key board)

The thread Com-server of higher level is now prepared for communication with other systems, e.g. a group control; supplementary systems of image processing, inputs form the unit of voice processing. The robot is shown in fig.5

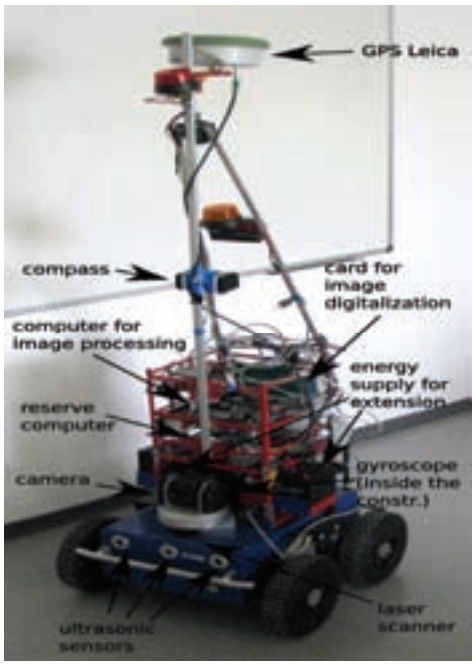


Fig.5 Service robot MRVK on a mobile platform

Conclusion

The described robotic system allows for verification and validation of several variants of the hardware as well as various structures of robot control. Due to the professional robot equipment it also allows for analysis of the behavioural properties and their reproducibility.

Acknowledgements

The research was supported by the grants VMSP-P-0004-09 and VEGA 1/0690/09

References

[1] Robotic Industries Association ISO 8373; www.robotics.org/company-profile-detail

[2] Internal Federation of Robotics; www.ifr.org

[3] Robotics & Automation Society IEEE; www.service-robots.org

[4] SCHRAFT, R.D., SCHMIERER, G.: Service robots. A.K.Peters, Ltd.2000

[5] SRI Consulting Business Intelligence; www.sric-bi.com

[6] JURIŠICA, L., DUCHOŇ, F.: Landmarks Detection with Laser Scanner. In: Metalurgija. Metallurgy. - ISSN 0543-5846. - Roč. 49, č. 2 (2010), s. 305-309

[7] HANZEL, J., JURIŠICA, L.: Comparison of Mapmaking Methods for Mobile Robots.In: Journal of Electrical Engineering. - ISSN 1335-3632. - Vol. 57, No. 5 (2006), s. 276-284

[8] JURIŠICA, L., VITKO, A., DUCHOŇ, F., KAŠTAN, D.: Systém GPS-princíp, prednosti a nedostatky. In: Automa. - ISSN 1210-9592. - Roč. 17, č.1 (2011), s. 50-53

[9] BABINEC, A., VITKO, A.: Histogramové navigačné algoritmy - vývoj a princíp.In: Automa. - ISSN 1210-9592. - Roč. 16, č. 5 (2010), s. 26-29

[10] KELEMEN, M., KELEMENOVÁ, T.: Study model of the snake like robot. In: Recent Advances in Mechatronics 2008-2009. - Berlin : Springer-Verlag, 2009 P. 228-232. - ISBN 978-3-642-05021-3

[11] MIKOVÁ, Ľ., KELEMEN, M., KELEMENOVÁ, T.: Štvorkolesový inšpekčný robot s diferenčným riadením kolies. Acta Mechanica Slovaca. Roč. 12, č. 3-B (2008), s. 548-558. - ISSN 1335-2393

[12] VITKO, A., ŠAVEL, M., JURIŠICA, L., HUBINSKÝ, P.: Data fusion and context awareness in autonomous robotics. In: International Journal of Mechanics and Control. - ISSN 1590-8844. - Vol. 5, No. 2 (2004), s. 29-39

Ing. František Duchoň, PhD.

Slovak University of Technology,
Faculty of Electrical Engineering
and Information Technology,
Institute of Control and Industrial Informatics,
Ilkovičova 3,
812 19 Bratislava,
Slovakia
E-mail: frantisek.duchon@stuba.sk

prof. Ing. Ladislav Jurišica, PhD.

Ing. Marián Kľúčik

doc. Ing. Anton Vitko, PhD.

A conceptual design of the self-reconfigurable mobile robot Wheeking 1

Róbert Surovec, Michal Kelemen, Martina Vacková, Ivan Virgala

Summary
This paper deals with the conceptual design of a self-reconfigurable mobile robot, which can be able to transform its undercarriage from wheeled to walking in dependence on terrain. It describes a development progress from early stage of construction process via necessary calculations to CAD model. These steps are made for creating an exact robot, which fulfill requirements set on self-reconfigurable mobile robots. In early stage of construction process the diagrams for visualization of components and their functions were made. Calculations were divided into analysis of wheel undercarriage and walking undercarriage. Then the CAD model in SolidWorks 2009 was created for a motion analysis. Based on these steps a prototype was designed, which nowadays serves as a didactical tool.

Keywords: Robot, Reconfiguration, Undercarriage, Obstacle

Introduction

Many people think robotics is a modern science, but it is a very old discipline. It has origins in the ancient world and in the present we recording only its advancements. There are many types of robots. Some of them being developed for floating underwater, the other ones have undercarriages for terrestrial locomotion, but some are able to fly. Nowadays the designers are trying to build robots, which could transform one type of undercarriage to another.

One of these self-reconfigurable mobile robots is WHEEKING 1, developed on the Faculty of Mechanical Engineering at Technical University of Košice. It can turn a wheel undercarriage into a walking undercarriage when the obstacle appears in front of it, which is unbeatable by its wheels. During the design process it is necessary to fulfill requirements set on mechanism with two mentioned undercarriages, as the low dead weight, relatively high load capacity, ratio ordered dimensions, good maneuverability, safety crossing of obstacles, resistance to environmental influences, conduct intelligence, single and positive control, power independence for the duration of operation onset and the operation simplicity as well. To achieving these criterions we had to realize a series of operations from early stage of construction process to necessary calculations.

Early stage of construction process

For visualization of robot components and interactions between them the model of a problem area in UML language (Fig.1) was created. In this block diagram we can see that WHEEKING 1 consists of body and undercarriage. In the body there are components like control unit and power supply. The undercarriage composed of upper and lower legs, driven by lumbar- and knee joints drives. On the end of each leg there are wheels.

Interactions between the robot and user is important too, therefore a facilities models in MSC language are needed to make. These diagrams are helpful tool for describing the services for environment and provide analysis of the robot behavior at turn on, turn off, move on, stop and reconfiguring commands settings.

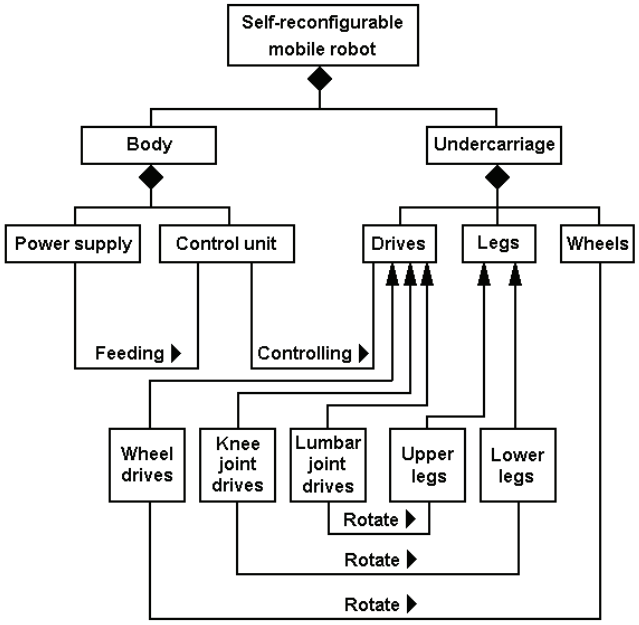


Fig.1 Model of a problem area in UML language

A mechatronics system control is a very important aspect and therefore the system-level description of the robot logical behavior was made by SDL language. It helps to analyze the signals responsible for system behavior in the feedback control circuit. It can be seen that the user programming the control unit through the user interface. This unit regulates the actuators, which positioning are sensing by position sensors. These information are evaluated in the control unit and make corrections of the required value [1].

Within the frame of the principled design three various solutions in kinematic structure form were proposed. Based on the power, geometric, financial criterions and also precision criterion the optimal solution was chosen (Fig.2). In the following steps of the design progress it was focused on this solution.

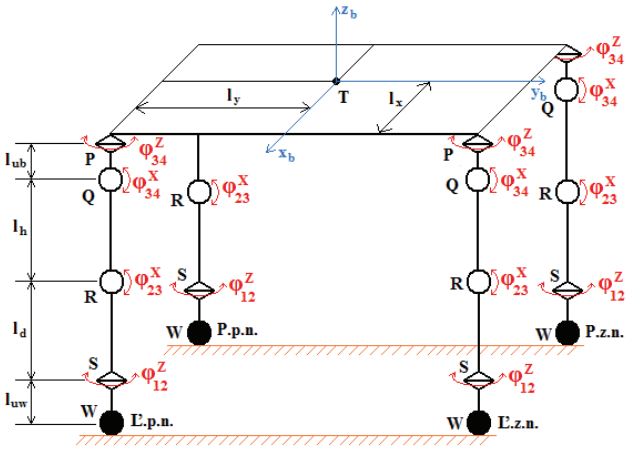


Fig.2 The chosen kinematic structure of Wheeking 1

Necessary calculations

For the gravity forces determination of each component it is necessary to calculate their weights. The Lexan polycarbonate with thickness of 4 mm has been selected as a construction material. Subsequently the calculations of torques in each joint were made for the drive selection. Based on the largest value of torque, which in the lumbar joint is 0,0397 N.m, the Hitec HS-55 Micro Servo Motor with torque 0,11 N.m was selected as an actuator.

Hypothetically the robot mostly would be use the wheel undercarriage and therefore the calculation of propulsive force F_H was realized. This force is defined as a summary of all driving resistances (1), which affect against the direction of motion.

$$F_H = O_f + O_s + O_z + O_p \tag{1}$$

At first the rolling-resistance force O_f was determined:

$$O_f = f.G_{robot} \tag{2}$$

where f is a friction coefficient and G_{robot} representing the gravitational force of the entire robot.

Next the climbing resistance O_s was calculated (Fig.3):

$$O_s = G_{robot}.\sin\alpha \tag{3}$$

where α is a climbing angle.

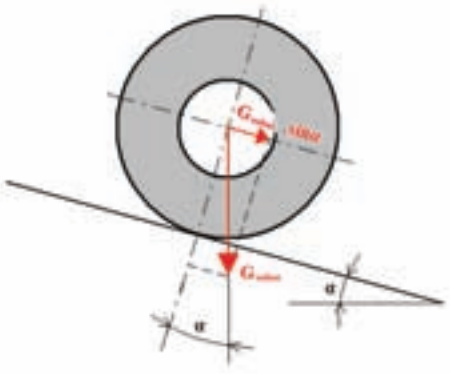


Fig.3 Forces in climbing resistance calculation

Acceleration resistance O_z:

$$O_z = \vartheta.m_{robot}.a_1 \tag{4}$$

where ϑ is an acceleration coefficient, m_{robot} represents the robot weight and a_1 is its acceleration.

An obstacle crossing resistance O_p is very important too (Fig.4):

$$O_p = \frac{G_{robot}.(l_{sf}+u).(\mu_\alpha - f).(f.\sin\xi+\cos\xi)}{(\cos\xi + f.\sin\xi - \mu_\alpha.\sin\xi).[h.(-\mu_\alpha + f)+2.l_{sf}+u]} \tag{5}$$

The l_{sf} is a distance between the front surface of the robot body and centre of gravity, u is distance between frontal wheels and front surface of the body, μ_α represents an adhesion coefficient and ξ is the angle of contact between the wheel and the edge of obstacle.

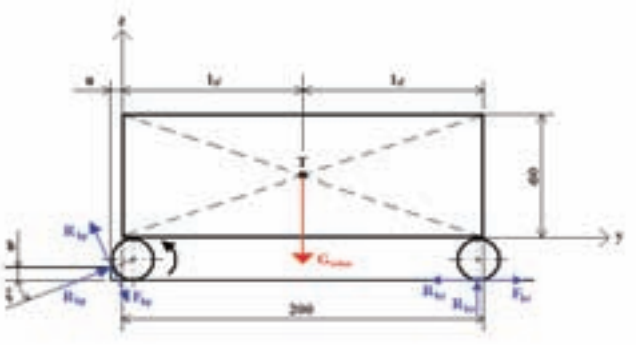


Fig.4 A model of the obstacle crossing resistance

The examination of the robot was planned in the interior to get knowledge of its possibilities [2]. Using the propulsive force a velocity, acceleration and also a time were calculated after the trajectory of 1 m passed by wheels (Fig.5). The velocity was considered as zero in the initial position.

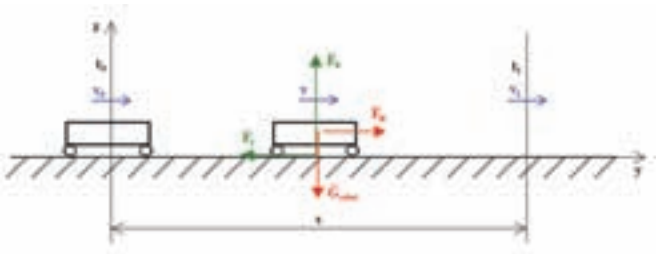


Fig.5 A model for kinematic parameters calculation

Using Newton´s second law of motion firstly the acceleration a_1 was obtained:

$$a_1 = a_y = \frac{dv_y}{dt} = \frac{F_H}{m_{robot}} - f.g \tag{6}$$

By integration of equation (6):

$$\int_{v_0}^{v_1} dv_y = \int_{t_0}^{t_1}\left(\frac{F_H}{m_{robot}} - f.g\right).dt \tag{7}$$

the time t_1 was obtained:

$$t_1 = \frac{v_1}{\frac{F_H}{m_{robot}} - f.g} \tag{8}$$

After extension of equation (6) by convenient unit:

$$a_y = \frac{dv_y}{dt}.\frac{dy}{dy} = \frac{F_H}{m_{robot}} - f.g \tag{9}$$

and by its integration:

$$\int_{v_0}^{v_1} v_y.dv_y = \int_0^s \left(\frac{F_H}{m_{robot}} - f.g \right).dy \tag{10}$$

The robot velocity v_1 was obtained:

$$v_1 = \sqrt{2.s.\left(\frac{F_H}{m_{robot}} - f.g \right)} \tag{11}$$

A kinematics analysis of the walking undercarriage was carried out using the Denavit-Hartenberg method. Because of self-reconfigurable mobile robot symmetry a one quarter part of the model was considered during the calculation (Fig.6).

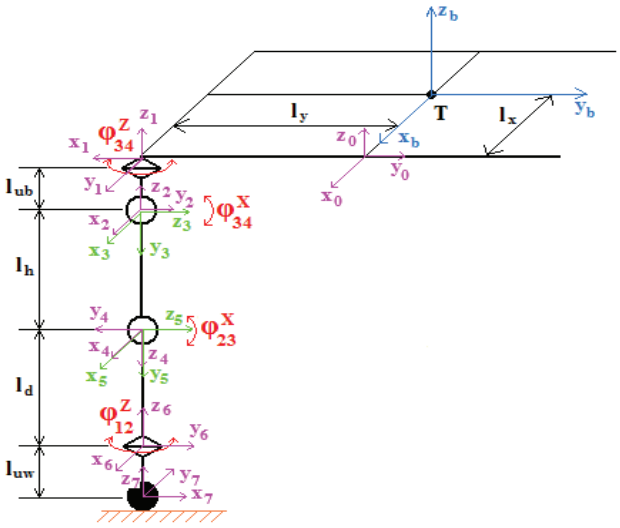


Fig.6 One quarter of the robot model for kinematic analysis

From the Fig,6 the Denavit-Hartenberg matrix T_b was obtained:

$$T_b = \begin{bmatrix} A & D & G & r_x \\ B & E & H & r_y \\ C & F & I & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

which includes the elements of position vector:

$$r_x = l_{uw}.(\cos\varphi^Z_{34}.\cos\varphi^X_{34}.\sin\varphi^X_{23} + \cos\varphi^Z_{34}.\sin\varphi^X_{34}.\cos\varphi^X_{23}) + \\ + l_d.\cos\varphi^Z_{34}.\sin\varphi^X_{34} - l_h.\cos\varphi^Z_{34} + l_x \tag{13}$$

$$r_y = l_{uw}.(\sin\varphi^Z_{34}.\sin\varphi^X_{34}.\cos\varphi^X_{23} + \sin\varphi^Z_{34}.\sin\varphi^X_{34}.\cos\varphi^X_{23}) + \\ + l_d.\sin\varphi^Z_{34}.\sin\varphi^X_{34} - l_h.\sin\varphi^Z_{34} + l_y \tag{14}$$

$$r_z = l_{uw}.(\sin\varphi^Z_{34}.\sin\varphi^X_{23} - \cos\varphi^X_{34}.\cos\varphi^X_{23}) - \\ - l_d.\cos\varphi^X_{34} + l_{ub} \tag{15}$$

By derivation a velocity of the endpoint, i.e. the wheel according to centre of gravity was obtained [3].

Mobility analysis of the CAD model

Another stage of the conceptual design was mobility analysis, which was realized on the CAD model (Fig.7) in SolidWorks 2009 environment. It helps to get a motion simulation. During a wheel undercarriage motion only the wheel drives will be active. A differential control and a lumbar joint inching were considered at the act of direction changing. A reconfiguration was simulated by lift off the legs under the robot body by reeling of lumbar- and knee-joints. It caused the body lifts up from the ground. At the same time the wheels turns over from its treads to side discs. Then the advance reeling of legs to the direction of movement follows.

A walking simulation was inspired by locomotion of big mammals. At the initial position the knees will be bend insomuch that the stability would be secured. First the right (left) front leg lifts up from the ground, i.e. the servomotor´s of the lumbar- and knee-joints turns. Then the right (left) back leg follows. Meanwhile the feet of the left (right) legs touch the ground, the robot body moves smoothly forward along with the left (right) hams. When the steps of right (left) legs are done, it is left (right) legs turn. It happens around about and the robot continuously runs forward with a mild pendulous movement of the body because of keeping the centre of gravity in its area [4].

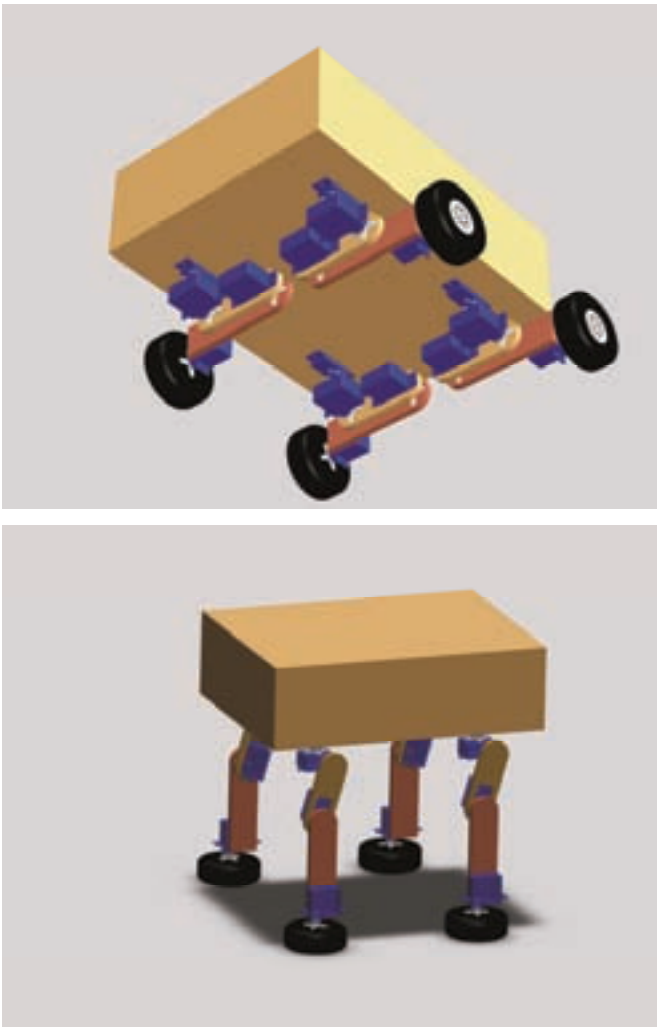
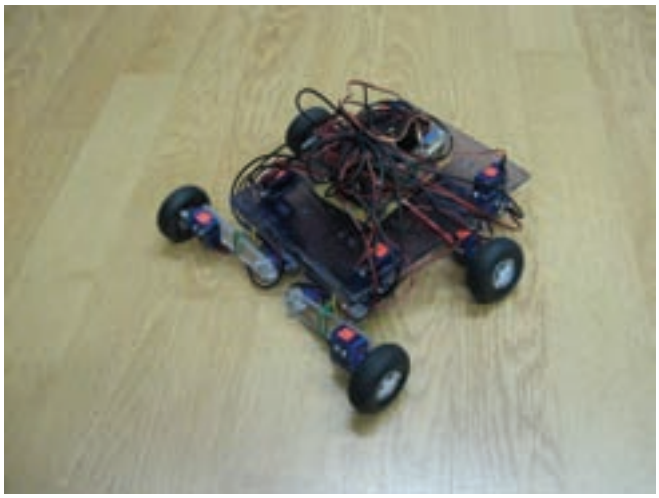


Fig.7 A CAD model of the robot

Conclusion

Robot Wheeking 1 was designed for finding possibility of mentioned reconfiguration type. After the conceptual design a physical model (Fig.8) was made, which nowadays performs only the function of a didactical tool. For its application in out-of-school sectors, it is necessary to realize some improvements like larger drives using or extend the robot by another helping devices. After these modifications the prototype will be able to move on rough terrain without problems.

Thanks to universality of this solution it may be a new contribution in robotics [5, 6, 7, 8].



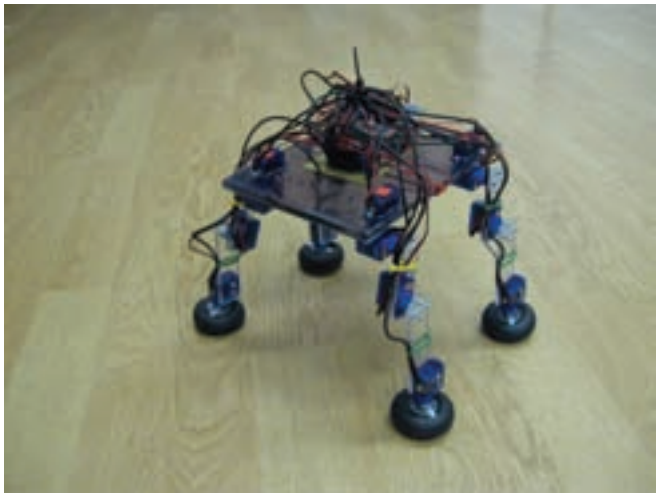


Fig.8 A physical model of the robot

Acknowledgment

The authors would like to thank to Slovak Grant Agency – project VEGA 1/0454/09 "Research on mechatronic systems imitating snake locomotion in confined and variable area". This contribution is also the result of the project implementation: Centre for research of control of technical, environmental and human risks for permanent development of production and products in mechanical engineering (ITMS:26220120060) supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] SHAKERI, A.: A methodology for development of mechatronic systems. *PH.D. thesis*, Norwegian University of Science and Technology 1998. ISBN 82-471-0340-0.

[2] SMRČEK, J., KÁRNÍK, L.: Robotics – Service robots. Designing, construction, solutions. *TU-SjF*, Košice 2008. ISBN 978-80-7165-713-2. (In Slovak)

[3] SUROVEC, R., KELEMEN, M., FABIAN, M.: Transformer – self-reconfigurable mobile robot Wheeking 1. In: *IT CAD*, 2010, vol. 20, no. 4, p. 68-70. (In Czech)

[4] ANTAL, D.: Dynamical modelling of a path controlled vehicle. *XXIV. Micro CAD, International Scientific Conference 2010*, 2010, p. 1-6.

[5] VITKO, A., JURIŠICA, L., KĽÚČIK, M., MURÁR, R., DUCHOŇ, F.: Embedding Intelligence Into a Mobile Robot. In: *AT&P Journal Plus*. 2008, no. 1: Mobilné robotické systémy (2008), p. 42-44. ISSN 1336-5010.

[6] VITKO, A., JURIŠICA, L., KĽÚČIK, M., DUCHOŇ, F.: Context Based Intelligent Behaviour of Mechatronic Systems. In: *Acta Mechanica Slovaca,* 2008*,* vol. 12, no. 3-B: (2008), p. 907-916. ISSN 1335-2393.

[7] OLASZ, A., SZABÓ, T.: Direct and inverse kinematical and dynamical analysis of the Fanuc LR Mate 200IC robot. *XXV. International Scientific Conference microCAD*, March-April 2011, p. 37-42.

[8] LÉNÁRT, J., JAKAB, E.: Machine vision used in robotics. *OGÉT 2010, XVIII. International Conference on Mechanical Engineering*, 2010, p. 272-274. (In Hungarian)

Ing. Róbert Surovec

Technical university of Košice
Faculty of mechanical engineering
Department of applied mechanics and mechatronics
Letná 9/B
042 00 Košice
Tel.: 00421-55-602-2456
E-mail: robert.surovec@tuke.sk

Advanced Control of a Walking Platform

Anton Vitko, Ladislav Jurišica, František Duchoň, Jaroslav Hanzel, Marián Kľúčik, Andrej Babinec, Martin Dekan and Dušan Kaštan

Abstract
The paper presents two important issues of robotics. The first is related to the optimized control of manipulator kinematics and the second addresses a problem of the behaviour based and fuzzy-neural based navigation of a mobile walking platform in an environment cluttered with obstacles.

Key words: walking platform, advanced control, mechatronic systems, navigation, learning

Introduction

Today, both the fix-based and mobile robots, being typical representative of mechatronic systems, have a profound impact on society. New ideas are generated for many application areas that are fundamentally different from their predecessors, which were mainly focused on manufacturing and automobile industries. New application areas are e.g. hazardous environments (nuclear, underwater, skyscrapers) medicine, reconnaissance of unstructured terrains like volcanoes and forests, decommissioning, inspection and maintenance of petrochemical and power plants, space missions, humanitarian demining, service robots, and many others. In such applications the robot must be reliable, robust and self-reliant. Although there exists a great demand for such machines, the technology is not ripe enough to deliver the systems that are needed. Customers are not convinced that robotic solutions are viable and hence progress has been slow. Therefore any improvement is strongly required.

The word "mechatronics" is a Janglish (Japanese-English) word, derived from the liaison of "mechanics" and "electronics". This liaison together with an incorporation of the means of artificial intelligence have significantly strengthen the role of mechatronics as an emerging engineering technology for the 21st century and widened applications areas of mechanical systems which themselves are pushed to the limits of their performance. All those have created prerequisites for appearance of new breed of intelligent machines, like smart road vehicles, haptic feedback manipulating devices, and last but not least autonomously operating mobile machines and industrial robots.

1. Redundant manipulator with minimized joint motion

Redundant manipulators are robotic arms possessing more degrees of freedom (DOFs) than there are necessary for establishing an arbitrary position and orientation of the end effector. They have attracted great attention just due to their operational flexibility. Let us recall the general form of a “direct kinematic model” given by the relation (1) between the Cartesian velocity vector $\dot{x}$ and the joint velocity vector $\dot{q}$.

$$J(q)\dot{q} = \dot{x} \tag{1}$$

where J(q) stands for Jacobian matrix. Since the dim(x) = n and dim (q) = m and (n< m) the Jacobian matrix is a rectangular (n x m) matrix. Therefore, the general solution for unknown vector q· may be given in the form (2)

$$\dot{q} = J^{+}(q)\dot{x} \pm \left[I - J^{+}(q)J(q) \right] \rho \tag{2}$$

where ρ is an arbitrary vector, and J^{+} is a pseudo-inverse matrix of J. The solution $\dot{q}$ consists of the following two parts

$$\dot{q} = \dot{q}_p + \dot{q}_n \tag{3}$$

The first part $\dot{q}_p$ is a particular solution representing a motion in the linear vector subspace spanned by the “n” independent rows of the Jacobian matrix, i.e.

$$\dot{q}_p = \dot{q} \in \left\{ R^n \,|\, J\dot{q} = \dot{x} \text{ for a given } \dot{x} \right\} \tag{4}$$

The second part $\dot{q}_n = \left[I - J^{+}J \right] \rho$ is a homogenous solution from the null space of Jacobian i.e.:

$$\dot{q}_n = \left\{ \dot{q} \in R^n \,|\, J\dot{q} = 0 \right\} \tag{5}$$

For any desired Cartesian velocity $\dot{x}$ the homogenous velocity $\dot{q}_n$ can be chosen so as to meet a specified performance criterion $H(q)$. Further specification depends on the choice of the vector ρ. For reaching an extreme value of $H(q)$, it is natural to place the grad $H(q)$ for the vector ρ. Then (6) will take the form [3,7]

$$\dot{q} = J^{+}\dot{x} \pm K\left(I - J^{+}J \right) grad\ H(q) \tag{6}$$

The theory was verified for the performance criterion securing that the joint angles q_j, j=1,2,…m will be kept close to the

middle of their maximum range $q_{j,min}$ and $q_{j.max}$. This requirements may be fulfilled by the choice of H(q) in accordance with (7)

$$H(q)=\sum_{j=1}^{n}\frac{\left(q_{j\max}-q_{j\min}\right)^2}{\left(q_{j\max}-q_j\right)\left(q_j-q_{j\min}\right)} \qquad (7)$$

where q_j is the j-th joint angle, $q_{j\max}$ and $q_{j\min}$ are the upper and lower limits laid on the joint angle q_j respectively. Clearly, the criterion (7) goes to the infinity at the joint limits and drives the joint angles away from their limits toward the middle of their ranges.

The feasibility of the proposed scheme was verified by simulation. A 3 DOF planar manipulator shown in Fig 1 was required to trace the desired straight line going from an arbitrary initial position to the origin. Fig.1

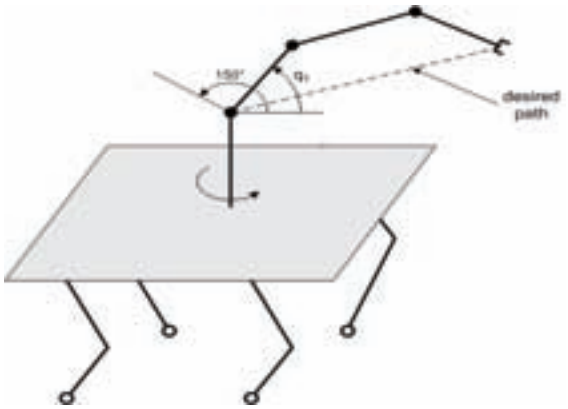


Fig.1 The walking robotic platform

The optimized and non optimized motions are shown in Fig.2 Clearly, in the non-optimized motion the first link's angle q1 approaches the lower limit 0^0 while in the optimized case) the angle q1 is kept symmetrically to the axis of the range $0\text{-}150^0$.

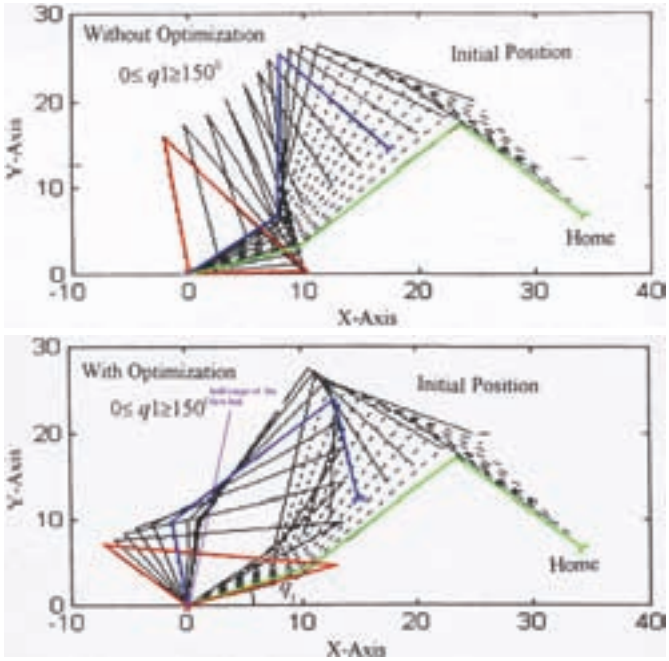


Fig.2 Motion of the end-effector without and with optimization

2. The behavioural based robot navigation

Search for a free and optimal robot path during navigation in an unknown environment cluttered with obstacles belongs to primordial problems of mobile robotics. Here we describe both the working principle and performance of an intelligent navigating algorithm. The robot is supposed to be equipped with ultrasonic range sensors providing information about the distance and azimuthal angle of an obstacle. The navigator's outputs represent the desired robot speed and heading angle. The developed algorithm is in essence a fuzzy expert system able to update fuzzy membership functions when moving among obstacles in an unknown environment. Learning process runs online. After learning the robot is able to run quickly from the same start position to the same goal positions without any additional learning.

The global behaviour of the robot is composed of blended elementary behaviours; namely wall following behaviour and striving towards the target behaviour, both described by an appropriate set of fuzzy rules. The blending process is controlled by a set of metarules through the inhibition or excitation of partial behaviours. For instance the global behaviour of the robot when finding itself in a U-shaped obstacles is shown in fig3 is controlled by metarules of the form:

IF (*number-of- loops is great)* THEN (*weight-of-managing-deadlock-behaviour is great)*

An advantage of the fuzzy subsumtion architecture is that the transit from the one behaviour to another is smooth. The navigation subsumes the wandering and wall-following behaviours. Due to this, the navigation algorithm is clearly more structured and the contribution of particular behaviours can be easily evaluated. Besides, the number of rules is much smaller then in the traditional monolithic structure. Situation is demonstrated in Fig.3. In the first figure it is clearly seen a loop shaped as the number "8" along which the robot wanders. The wandering would be endless without intervenes of the fuzzy meta-rule, which weakens the strength of the rules governing the obstacle avoidance if the wandering lasts too long. Due to this the robot starts to follow the wall in the opposite target direction. These allow leaving the deadlock and finally reach the target

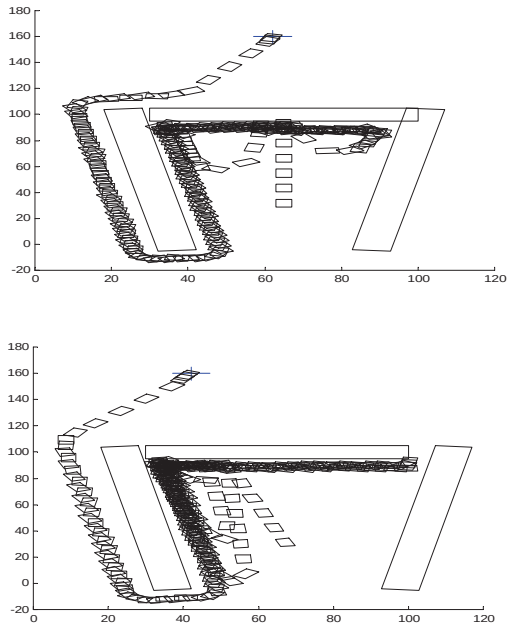


Fig.3 Robot motion navigated by blending of the wander and wall-following behaviours

In the Fig. 4 the process of navigation goes on in the same way though the number “8” shaped path isn’t so clear, what is caused by a slightly shifted target location.

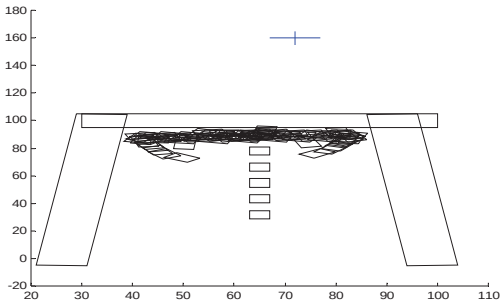


Fig.4 Demonstration of the inability to escape from a deadlock without blending behaviours

Fig.4 demonstrates that the same set of rules if not structured into subsumption architecture is unable to drive the robot from the U-shaped obstacle. That is because the strengths of all rules remain unchanged.

3. The fuzzy-neural based robot navigation

To keep the motion smooth and free of sharp turnings and transversal swings parameters of fuzzy rules are updated on-line. It is done periodically in two steps for each period. Within the first step the tuning of the parameters of the rectangular membership functions (MFs) takes place. To this end, the fuzzy rules are converted into a feed forward neural network and updated by back propagation. In the second step an additional fine tuning of MFs is added. In particular, normally straight walls of rectangular MFs are deformed into appropriate irregular shapes.

The cost function has been chosen to keep the actual turning radius “r” close to the desired one “r^d” so as to protect the robot from possible overthrowing due to a high speed when moving along a too bended path. The cost function E given below and illustrated in Fig.5 is minimized during the run of back propagation algorithm.

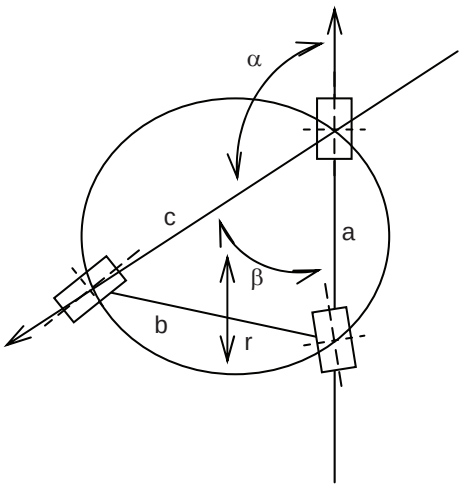


Fig.5 The geometry behind the cost function

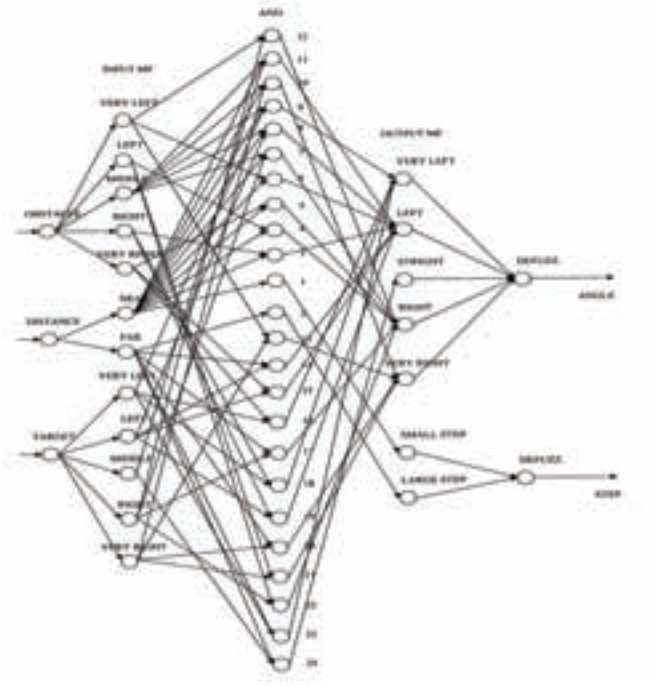


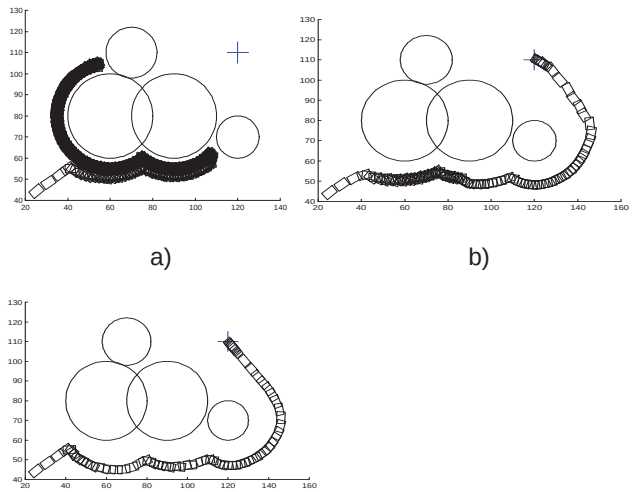
Fig.6 The neural network

The cost function E is was chosen as follows

$$E = (r^d - r)^2 = \left(r^d - \frac{\sqrt{a^2 + c^2 + 2ac\cos\alpha}}{2\sin\alpha} \right)^2 \quad (8)$$

The cost function was minimized with respect to the parameter “α” – the turning angle. The neural network used for learning navigation is shown in Fig. 6. Input and output neurons represent premise and consequent MFs respectively while fuzzy composition runs in hidden neurons,.

The net has three inputs: Obstacle- the azimuthal angle to the obstacle, Distance – the distance to the obstacle and Target - the azimuthal angle to the target. Outputs are: Angle- the direction of the nearest robot movement and Step- the size of the step the robot should perform next. The learning process is demonstrated in Fig 7a,b,c. The robot was required to move from the origin to the target denoted by a cross



a) b)

c)

Fig.7 Stages of the learning process

The Fig.7a shows navigation without learning. The robot motion is full of transversal swings caused by repulsive force of the obstacle and attractive force of target. The

Fig.7b shows the motion after first stage learning within which only parameters of MFs were updated. Clearly, the motion is much smoother, free of intensive swinging. Finally, the Fig.7c depicts motion after second stage learning, within which the walls of MFs were modified. The robot motion is totally free of the swing and robot quickly and smoothly moves toward the target.

Both the simulation and real experiments indicate that the chosen navigation philosophy is feasible. Besides fulfilling the primary requirement, namely to imbue the robot with learning abilities, the robot can quickly follow the learned path without being jeopardized by overthrowing caused by too high speed movement along sharp turnings.

Acknowledgement

Support of the Grants VEGA 1/0177/11 and VEGA 1/0690/09 is duly acknowledged.

References

[1] FUJIMORI, A., NIKIFORUK, P.N., GUPTA, M.M.: Adaptive Navigation of Mobile Robots with Obstacle Avoidance

[2] KHATIB, O.: Real-Time Obstacle Avoidance for Manipulator and mobile robots. Int. J. Rob. Research 5 (1986), No1, 90-98

[3] JANG, J.S.R., SUN, C.T., MIZUTANI, E.: Neuro-Fuzzy and Soft Computing. Prentice Hall, Inc., Upper Saddle River, 1997

[4] NAUCK, D., KLAWON, F., KRUSE, R.: Foundations of Neuro-Fuzzy Systems, John Willey & Sons Ltd., 1996

[5] SAFFIOTI, A.: The Uses of Fuzzy Logic in Autonomous Robot Navigation: a catalogue raisonné, Technical Report TR/IRIDIA/97-6, Univ Libre de Bruxelles, Belgium

doc. Ing Anton Vitko, PhD.

Slovak University of Technology,
Faculty of Electrical Engineering
and Information Technology,
Institute of Control and Industrial Informatics,
Ilkovičova 3,
812 19 Bratislava,
Slovakia
E-mail: anton.vitko@stuba.sk

prof. Ing. Ladislav Jurišica, PhD.

Ing. František Duchoň, PhD.

Ing. Jaroslav Hanzel, PhD.

Ing. Marián Kľúčik

Ing. Andrej Babinec

Ing. Martin Dekan

Ing. Dušan Kaštan

Global metric map for mobile robotics

Martin Dekan, František Duchoň, Anton Vitko

Abstract
In this paper two modification of environment mapping algorithms based on local map processing are proposed and tested. The first one is polygonal approximation of obstacles and second one is line approximation of obstacles. In the paper are described algorithms for local map processing and algorithms for map building itself. The proposed methods are tested in simulations

Keywords: global map, local map, image processing, ultrasonic range finder, obstacle representation

Introduction

Creating a global map in the field of mobile robotics is closely associated with the path planning, localization and environment mapping for other purposes. Currently, topological maps and metric maps are mainly used to represent the environment. The article in the following section deals with metric maps.

Representing a global map should satisfy these criteria

1. The accuracy of the map has to match the accuracy which is needed to perform the task of the mobile robot.

2. The accuracy of maps and kept features of the environment has to correspond with accuracy and features which is mobile robot capable of capturing with its sensors

3. The complexity of the map directly affects on the computational complexity in mapping, navigation and localization of mobile robot.

There are three groups of metric maps
- Road maps [2]
- Grid maps [2]
- Continuous maps [2]

1. Type of the metric map

Global map is based on local map and sensors proposed in [1]. Local map is represented as occupancy grid with a fixed size of 10x10cm and used sensors are ultrasonic[4] and infrared range finders.

Because of the principle of storing information in this local map would be not appropriate to use such environment representation, which seeks to store information about the constraints exactly.

For our solution, we proposed a modification of Polygonal representation of the environment. It is a continuous representation of the environment which substitutes obstacles with their polygonal approximation. This representation has the advantage over other representations in memory consumption, because complex real objects are in memory represented by a small number of data. Special case of this representation is a Line representation, where the edges of obstacles are approximated by a finite number of lines. [2]

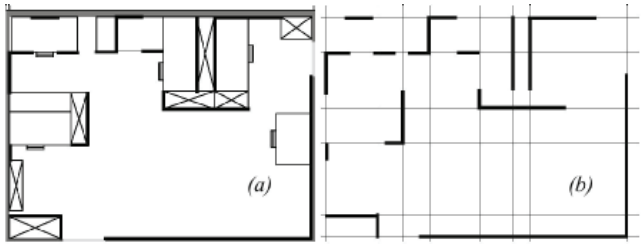


Fig. 1. Continuous representation of the environment[2]

Disadvantage of polygonal and hence the line representation is possible loss of narrow passages due to the approximation of the data which are used to create maps.

In the next section the following two modifications are shown.
- Polygonal approximation
- Line approximation

2. Local map preprocessing

Before the local map can be used to create the map of the environment, the data in the local map have to be modified. Because the local map is represented by a matrix, the data preprocessing is analogues to image preprocessing.

The first step to the local map preprocessing is smoothing filter. Purpose of such operation is to eliminate low value in the matrix in the high values surroundings and high values in the low. This type of noise arises in the local map because of rounding data or because of blind spots between the sensors. [3][7] For this operation we use median filter.

The next step in local map preprocessing is thresholding. Thresholding is a function that modifies the data by rule:

$$f(c)=\begin{cases} A & if \quad c < threshold \\ B & if \quad c \geq threshold \end{cases} \tag{1}$$

where c is the input value, f (c) the resulting value is threshold value, A and B are new values for the input value c above and below the threshold. In our case, the value A = 0, B = 1, where a value of 1 represents that the place is occupied f and value of 0 represents the free space. [3]

3. Edge detector

As for the edge detector, Canny identified three criteria for optimal edge detector, which must be met:[3][6]

- Detection criterion = detector must not forget an important edge and for one edge there must be only one response.
- Localization criterion = difference between actual and found edge should be minimal.
- One response criterion.

Canny detector uses a two-dimensional convolution with a Gaussian derivative in gradient direction. The output of the Canny detector is the size and direction of edges. [3]

$$G_n = \frac{\partial G}{\partial n} = n\nabla G \tag{2}$$

The G in the formula above is the two-dimensional Gaussian. Gn is the first derivative of the gradient G and n is the direction of the gradient :

$$n = \frac{\nabla(G * f)}{\left|\nabla(G * f)\right|} \tag{3}$$

The edge is at the point where the function Gn f * reaches a local maximum, and the second derivative is zero.

$$\frac{\partial^2}{\partial n^2} G * f = 0 \tag{4}$$

As for the edge strength the following applies:

$$\left|G_n * f\right| = \left|\nabla(G * f)\right| \tag{5}$$

4. Hough transformation

Hough transformation is used to identify incomplete or damaged objects such as line, circle and ellipse in the image. Usually, the Hough transformation is applied to image after image processing, edge detector.

The simplest case of the Hough transformation is the detection of lines. The idea of transformation is an expression of the line y = mx + b by not using the image points x and y, but with parameters m and b.[3][5]

For computing purposes, instead of using m and b parameters Θ r are used [3]

The r represents the distance from the start of grid system and Θ the angle between the perpendicular to the line y = mx + b passing through the beginning of grid axes.

The line will then be expressed as

$$y = (-\frac{\cos\theta}{\sin\theta})x + (\frac{r}{\sin\theta}) \tag{6}$$

Where $\theta = [0, \pi), r \in R$ or $\theta = [0, 2\pi), r \geq 0$

In the space (r, Θ), thus representing all of you True Sine Wave line passing through point (x, y).

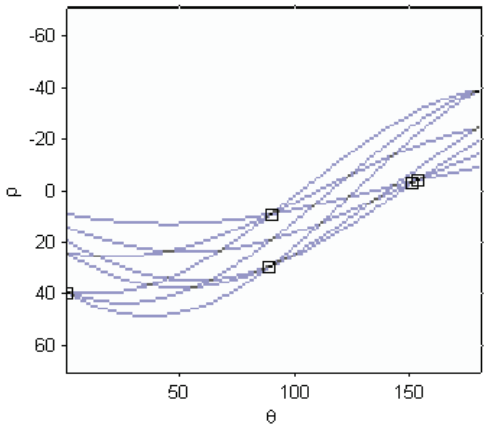


Fig. 2 Principle of hough transformation

After applying the Hough transformation to all the points we get the space in which a place with large number of intersections represent the parameters (r, Θ) of lines identified in the image.

5. Map of the environment using Polygonal approximation

Map of the environment is created according to the local maps by identifying polygons that represent obstacles in the environment.

Due to the corruption of data by impulse noise the median filter is applied. The thresholds function with the threshold 0.5 is used in the occupancy grid. On the adjusted data Canny edge detector and Hough transform are applied.

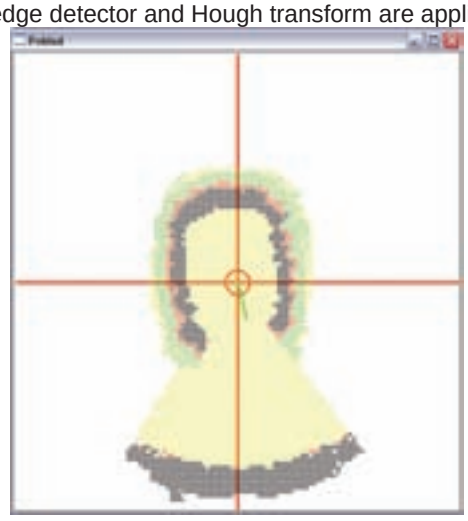


Fig. 3. Local map

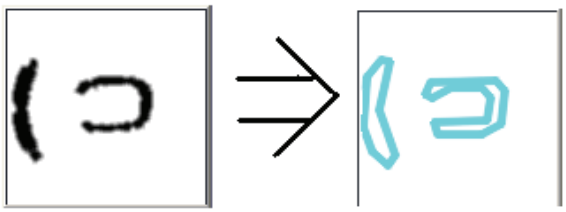


Fig.4. Polygonal approximation of local map.

The figures(Fig. 3.and Fig 4.) shows local map, its filtered and thresholded modification and polygonal approximation of the obstacles.

As it is shown in the figure, the obstacle that we have constructed in this way is actually a circuit of objects created according to the local maps.

Global map of the environment will be made from the data sub-environment maps.

Analogous to the formation of partial maps, map will be created as a circuit of an object that was made from sub-maps. (Fig. 5)

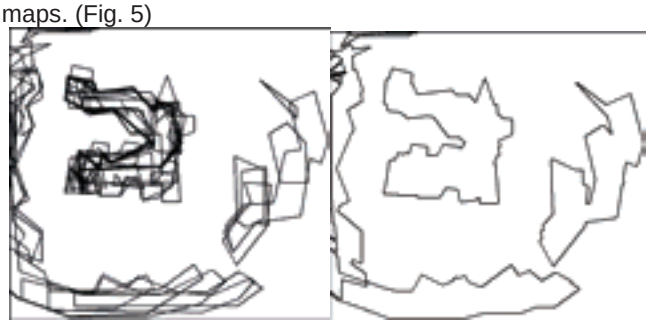


Fig.5. Creating a global map (right) of sub-maps (left)

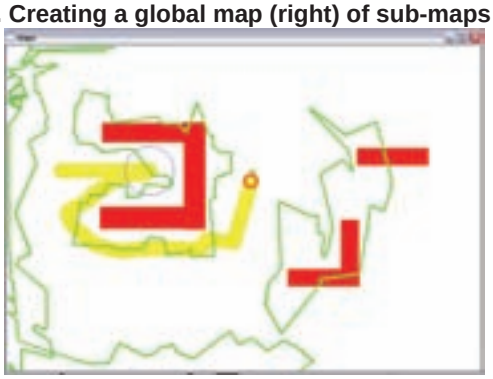


Fig.6. Comparison of the global map with the actual environment

As shown in Figure(Fig. 6), such a global map is very inaccurate. This uncertainty mainly results from the size of local map that we use to produce maps of the environment. Thus such generated map cannot be considered sufficiently accurate, because use of such map could cause the loss of the narrow passages between the two barriers. To eliminate this disadvantage we adjust the size of local map, which is taken to map generation, proportional to the shortest measured distance.

Min distance>1,3 m size of local map=+-3 m

0,75< Min distance <1,3 m size of local map =+-2,2m

0,3< Min distance <0,75 m size of local map =+-1,2 m

Min distance <0,3 m size of local map =+-0,6m

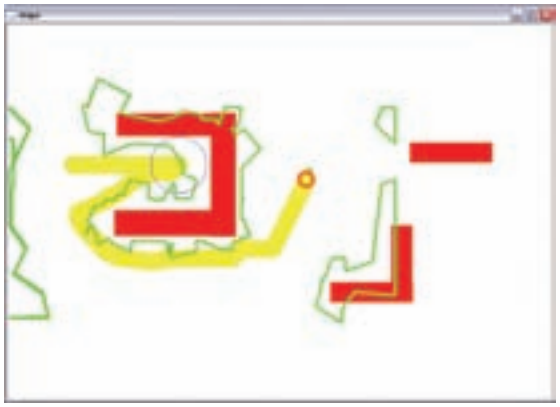


Fig.7. Map of the environment

The result is shown on the (fig.7). As can be seen, the area in the surroundings of mobile robot trajectory remains very similar, but more distant object are represented with fewer polygons. With this approach obstacles are included in the map only if they are sufficiently mapped.

6. Map of the environment created by Line approximation

This method of map building is based on the previous method, but the algorithm takes only those edges that are directly visible from the robot position into account and thus directly measured distances.

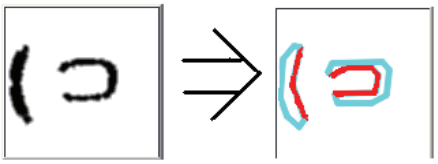


Fig.8. Illustration of the taken edges (red) in local map

The results are show on (fig.9).

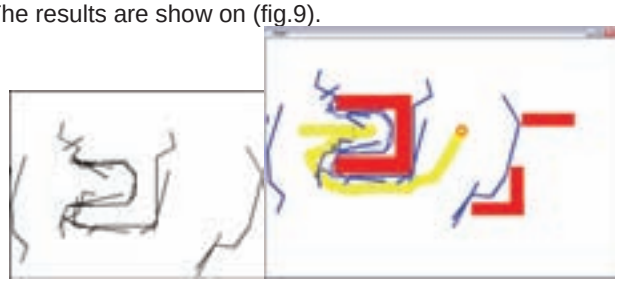


Fig.9. Map of the environment using line approximation

As can be seen in the figure, this method creates map which corresponds with the environment significantly better, but further improvements are needed. Based on the assumption of diffuse reflections when ranging with ultrasonic sensors, we can claim that the measured distance reflects real distance of the obstacle. Using the principle explained in the figure, we can reduce the number and length of the lines

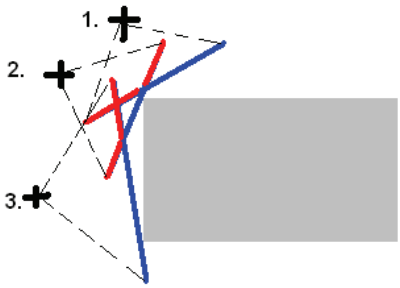


Fig.10.The principle of removing unnecessary parts of the processed data

Fig.11 the resulting map created using the line approximation method.

The result of such modification is shown on the (fig.11). As can be seen, this modification generates the most accurate map of the proposed methods.

7. Conclusion

We have developed a mobile robot environment mapping algorithm based on continuous representation of the environment. In this paper two modification of this algorithm were presented.The first one is polygonal approximation which is computational less expensive and the second is line approximation which describes better the environment. Results of the test indicate the usefulness of such algorithm. This article describes results of mapping by ultrasonic sensors, but it is possible to use this algorithm using better sensors such as laser rangefinders.

ACKNOWLEDGMENT

This work was supported by MS SR under the contract VEGA 1/0177/11 and VMSP-P-0004-09.

References

[1] Dekan Martin , Duchoň František: Navigation of Autonomous Mobile Robot Using Ultrasonic and Infrared Sensors. In: Robotics in Education 2010 : Proceedings of the 1st International Conference. Bratislava, 16.-17. 9. 2010. - Bratislava : Slovak University of Technology in Bratislava, 2010. - ISBN 978-80-227-3353-3. - S. 193-196

[2] Siegwart Roland, Nourbakhsh Illah R. Intorduction to Atuonomous Mobile Robots. [s.l.] : MIT Press, 2004. ISBN 978-0-262-19502-7.

[3] Bartek, Tomáš. 2010 Detection of the target object by the visual system, determination of the relative position of the object. :Diploma thesis,. Bratislava: STU FEI, 2010. 76 p.

[4] Hanzel Jaroslav, Jurišica Ladislav. Experimental identification of parameters of ultrasonic rangefinder. AT&P Journal Plus, vol. 2/2006. ISSN 1336-5010

[5]Pásztó Peter , Hubinský Peter: Application of a Visual System for Mobile Robot Navigation (OpenCV). In: AT&P Journal Plus. - ISSN 1336-5010. - Č. 1: Systémy automatického riadenia (2010), s. 62-65

[6] Kľúčik Marian , Jurišica Ladislav: Visual Systems and Line Detection.In: Selected Topics in Modelling and Control. Vol. 6. - Bratislava : Slovak University of Technology in Bratislava, 2010. - ISBN 978-80-227-3318-2. - S. 135-143

[7] Rodina Jozef , Hubinský Peter: Stability Control Design of SegwayTM Like Differential Drive by Using MEMS Sensors.In: Metalurgija. Metallurgy. - ISSN 0543-5846. - Roč. 49, č. 2 (2010), s. 483-487

Ing. Martin Dekan

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail:martin.dekan@stuba.sk

Ing. František Duchoň, Phd.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail:frantisek.duchon@stuba.sk

doc. Ing. Anton Vitko, Phd.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering
and Information Technology
Institute of Control and Industrial Informatics
Department of Robotics and Artificial Intelligence
Ilkovičova 3
812 19 Bratislava
E-mail:anton.vitko@stuba.sk

Dynamic obstacle avoidance in mobile robotics

Ladislav Jurišica, František Duchoň, Martin Dekan

Abstract
This paper briefly describes two methods, which can navigate robot in the environment with moving obstacles. Basically it is not hard to avoid moving obstacle, but it is more important to detect these dynamic obstacles. In paper are presented avoidance methods only. First method deals with obstacle, which has detectable position and velocity. Second method is well-known potential field method enhanced by velocity of the obstacles and the velocity of goal.

Key words: moving obstacle, velocity obstacle method, potential field method

Introduction

Mobile robots exist in our world. Most of the navigational functions of mobile robots consider only static obstacles. But in real world, there are a lot of moving obstacles. They should be classified in three categories:

- Obstacles greatly slower than the robot.
- Obstacles of which speed is approximately equal to the speed of robot.
- Obstacles greatly faster than the robot.

Obstacles involved in the first group are too slow and they can be considered as static. Obstacles included in the third group are too fast and even if robot can register them with his sensors, robot often fails to avoid them. Methods of dynamical obstacles avoidance settle with the second groups of obstacles.

There are many solutions how to avoid dynamic obstacle:

1. Its ignore, assuming that proposed navigational function can avoid such obstacle.
2. Addition of the time dimension into navigational function, assuming known trajectories of the obstacles.
3. Decomposition to two navigational functions. First function is a classic navigational function among static obstacles. Second function is looking for the set of velocities, which can lead to dynamic obstacles avoidance. Second function is proposed on basis of paths from first function.
4. Extension of navigational function by the velocities of dynamic obstacles (e.g. addition of another repulsive force into the method of artificial potential field).

Most difficult task for all of the solutions is how to detect dynamic obstacle. Despite of limited sensorial equipment, there are some methods able to estimate direction and speed of the dynamic obstacle [6].

1. Velocity obstacle method [1] [2]

This method uses two sets of velocities. First set represents the velocities that lead to collision with moving obstacle on basis of its position and movement. Second set involves the velocities that enable avoiding the collision. It is clear that velocity obstacle method works with coordinate system defined on the velocities. Consequently, geometrical shape of the obstacles must be transformed into this system. The principle of collision avoidance consists in the heuristic searching of maneuver tree (translational and rotational velocity).

Let the robot is marked as A and obstacle is marked as B. Couple of objects A and B is transformed into configuration system of the robot, so robot A becomes $\hat{A}$ and obstacle becomes $\hat{B}$ (Fig. 1). Because of secure obstacle avoidance obstacle $\hat{B}$ is enlarged by the radius of the robot. Each object has own velocity vector with beginning in the centre of gravity. These vectors are defined in given coordinate system. Assuming that both objects will keep their velocity, collision will happen in some time $t_1 > t_0$, if line $\lambda_{A,B}$ coming out from relative velocity $v_{A,B}$ intersects obstacle $\hat{B}$. Thus there exists cone in velocity coordinate system (Fig. 1), which defines collision between two objects (so-called collision cone):

$$CC_{A,B} = \left\{ v_{A,B} \mid \lambda_{A,B} \cap \hat{B} \neq 0 \right\}, \qquad (1)$$

where $v_{A,B}$ is relative velocity of $\hat{A}$ in respect of $\hat{B}$ and $\lambda_{A,B}$ is perpendicular coming out from $v_{A,B}$ in the direction of $\hat{B}$. Obstacle $\hat{B}$ is bounded by forward λ_f and reverse λ_r relative velocity. Any relative velocity $v_{A,B}$ contained in collision cone $CC_{A,B}$ leads to collision between robot and obstacle. Usually commands for robots does not uses rela-

tive velocity between robot and obstacle. For this reason, collision cone is shifted by velocity v_B:

$$VO = CC_{A,B} \oplus v_B, \quad (2)$$

where $\oplus$ is Minkowski's summary operator of vectors (i.e. $(a,b)+(c,d)=(a+c,b+d)$.

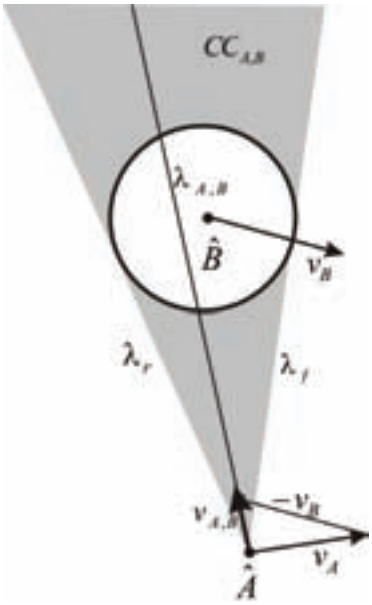


Fig.1 Relative velocity $v_{A,B}$ and collision cone. [2]

This shifted cone (Fig. 2) is representing obstacle with velocity. With this shifted collision cone there are three sets of robot velocities:

1. Velocities inside the shifted cone are leading to collision.
2. Velocities on boundaries of shifted cone are leading to gentle touch.
3. Velocities outside the shifted cone are leading to avoidance.

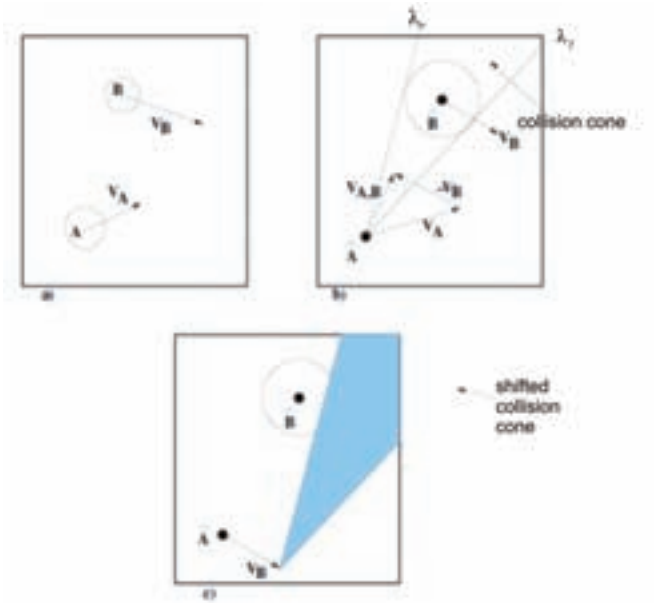


Fig.2 Velocity obstacle principle: a) Robot A with dynamic obstacle B. b) Relative velocity $v_{A,B}$ with collision cone $CC_{A,B}$. c) Obstacle with velocity - shifted collision cone. [1]

For avoiding the multiple dynamic obstacles (Fig. 3) it is needed to combine collision cones into the one:

$$VO = \cup_{i=1}^{m} VO_i, \quad (3)$$

where m is number of obstacles.

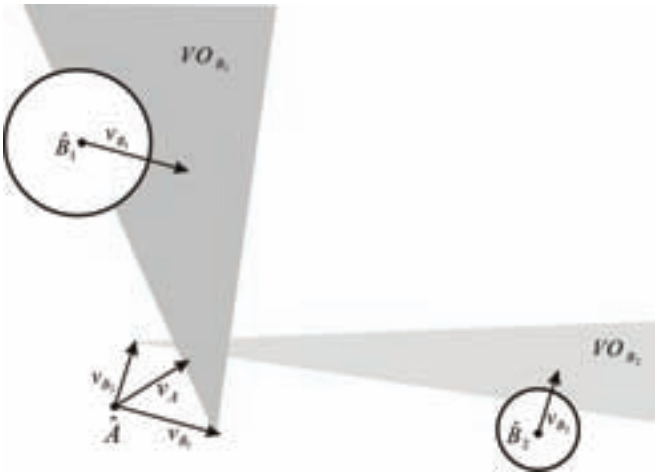


Fig.3 Collision cones and obstacles with velocities $\hat{B}_1$ and $\hat{B}_2$. [2]

Velocities generated by velocity obstacle method guarantee collision free movement of the robot independently of expected time of collision between robot and obstacle. For this reason, velocities that can lead to collision in long period are not excluded. However these situations can be solved when robot is near the collision with obstacles in terms of velocity obstacle method.

Velocity obstacle method also involves dynamic restrictions from robot. This is done by definition of the set of reachable velocities V_r. In terms of velocity obstacle method this set must be reduced to velocities which are leading to obstacle avoidance. Set V_r contains three not overlaying subsets S_f, S_r and S_d. S_f are foreran velocities, S_r are reversing velocities and S_d are deflecting velocities (Fig. 4).

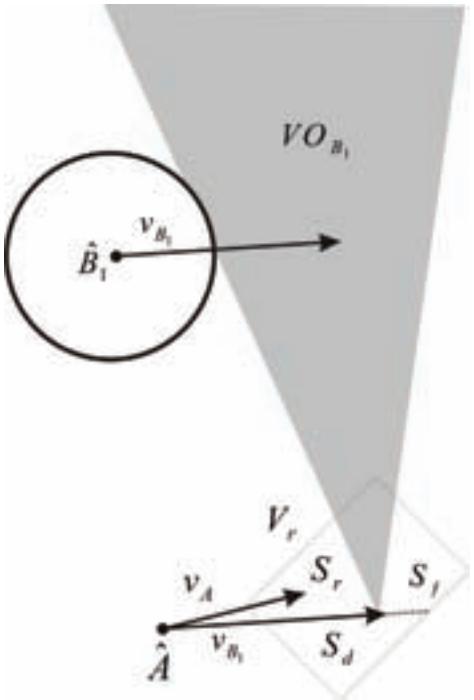


Fig.4 Set of reachable velocities and its subsets. [2]

Minimally $3.o$ foreran, reversing and deflecting velocities exist for robot A and o dynamic obstacles $(i:1,...,o)$ (Fig. 5).

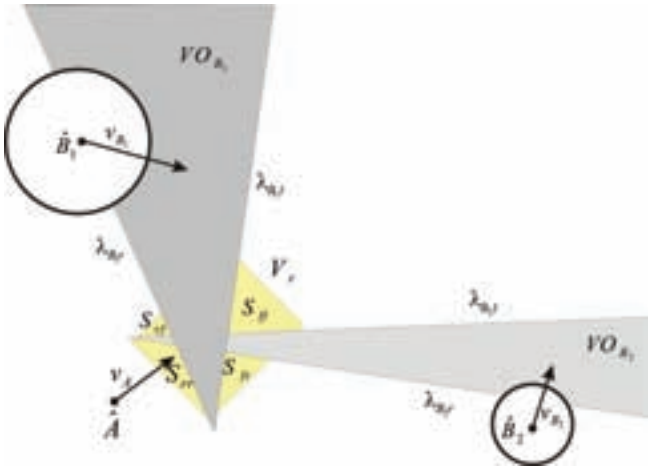


Fig.5 Set of reachable velocities and its subsets with two dynamic obstacles. [2]

Generally the biggest problem is to how to differentiate moving obstacle from the static one. One acceptable solution is to use two laser rangefinders. From their data can be derived which obstacle is moving and which not.

2. Artificial potential field and moving obstacles [1] [3]

This method has three basic assumptions:

1. Shape, position p and velocity v of the robot are known.
2. Position of goal p_{ciel} and velocity of goal v_{ciel} are known, whereby holds $|v_{ciel}| < v_{\max}$.
3. Obstacles are convex polygons those shape, positions $p_{prek,i}$ and velocities $v_{prek,i}$ $(i:1,...,o)$ are known and measurable; o is number of the obstacles.

Standard potential field method defines attractive and repulsive forces on basis of robot's and goal's position. However artificial potential field dealing with moving obstacles includes also velocities of the obstacles and velocity of the goal. New attractive potential is defined:

$$U_{prit}(p,v) = \alpha_p.\|p_{ciel} - p\|^m + \alpha_v.\|v_{ciel} - v\|^n, \qquad (4)$$

where $\|p_{ciel} - p\|$ is Euclidean distance between goal and robot, $\|v_{ciel} - v\|$ is magnitude of relative velocity between goal and robot, α_p and α_v are scalar positive parameters, m and n are positive constants. Attractive potential is null only when distance and relative velocity between robot and goal are null.

With new attractive potential defined is related new attractive force:

$$F_{prit}(p,v) = -\nabla U_{prit}(p,v) = -\nabla_p U_{prit}(p,v) - \nabla_v U_{prit}(p,v) \qquad (5)$$

where:

$$\nabla_p U_{prit}(p,v) = \frac{\partial U_{prit}(p,v)}{\partial p}$$
$$\nabla_v U_{prit}(p,v) = \frac{\partial U_{prit}(p,v)}{\partial v} \qquad (6)$$

It is clear, that potential $U_{prit}(p,v)$ is not different in regard to p, if $p = p_{ciel}$ and $0 < m \le 1$. Potential $U_{prit}(p,v)$ is also not different in regard to v, if $v = v_{ciel}$ and $0 < n \le 1$. If $p \ne p_{ciel}$ and $v \ne v_{ciel}$, from equations (4) and (5) can be derived:

$$F_{prit}(p,v) = F_{prit1}(p) + F_{prit2}(v), \qquad (7)$$

where:

$$F_{prit1}(p) = m.\alpha_p \|p_{ciel} - p\|^{m-1}.n_{RC}$$
$$F_{prit2}(v) = n.\alpha_v \|v_{ciel} - v\|^{n-1}.n_{VRC} \qquad (8)$$

where n_{RC} is a base vector heading from the robot to the goal and n_{VRC} is a base vector denoting the direction of relative velocity of the goal in regard to the robot (Fig. 6).

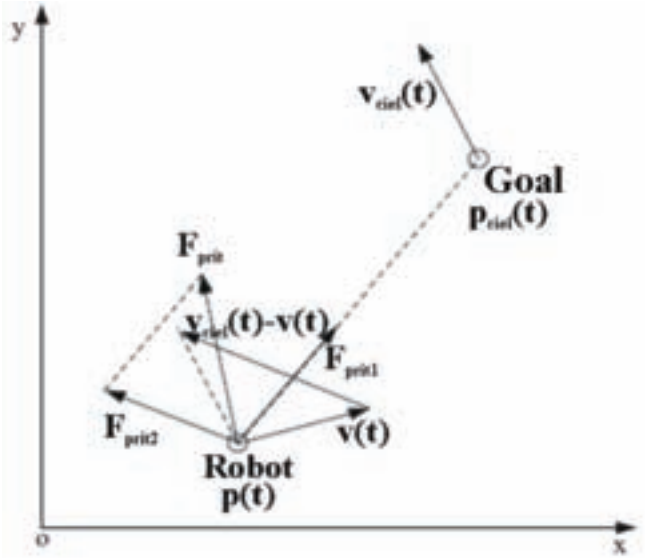


Fig.6 Definition of attractive force in two-dimensional space. [3]

Attractive force consists of two components. $F_{prit1}(p)$ pushes robot towards the goal and reduces the distance between the robot and the goal. $F_{prit2}(v)$ forces robot to move as fast as the goal. For $m > 1$ and $n > 1$, if robot reaches the goal, $F_{prit1}(p)$ is null, and if robot is moving as fast as the goal, $F_{prit2}(v)$ is null. This means that if robot reaches the goal and its velocity is equal to velocity of the goal, $F_{prit}(p,v)$ is null. Therefore robot caught up the goal and it has the same velocity as the goal.

New repulsive force is defined on basis of the distance to the nearest obstacle and on basis of its velocity. Application of new repulsive force requires convex shape of the ob-

stacles and spot representation of the robot. Function for repulsive force can be enhanced by dimensions and shape of the robot. Relative velocity between robot and obstacle in the direction from robot is defined:

$$v_{RP} = \left[v - v_{prek}\right]^T .n_{RP}\,, \tag{9}$$

where n_{RP} is a base vector heading from robot to the obstacle. If $v_{RP} \le 0$, then robot is moving outwards the obstacle. If $v_{RP} > 0$, then robot is moving towards the obstacle and robot must avoid it. Let the smallest distance between robot and obstacle is denoted as $\rho_n\left(p, p_{prek}\right)$. Let the maximal deceleration of the robot is denoted as $a_{\max}$, then distance, which robot passes before v_{RP} will be equal to zero:

$$\rho_m\left(v_{RP}\right) = \frac{v_{RP}^2}{2.a_{\max}}\,. \tag{10}$$

New repulsive potential is defined:

$$U_{odp}(p,v) = \begin{cases} 0 & if\ \left|\rho_n(p,p_{prek}) - \rho_m(v_{RP})\right| \ge \rho_0 \vee v_{RP} \le 0 \\ \eta\left(\dfrac{1}{\rho_n(p,p_{prek}) - \rho_m(v_{RP})} - \dfrac{1}{\rho_0}\right) & if\ 0 < \left|\rho_n(p,p_{prek}) - \rho_m(v_{RP})\right| < \rho_0 \wedge v_{RP} > 0 \\ nedef. & if\ v_{RP} > 0 \wedge \rho_n(p,p_{prek}) < \rho_m(v_{RP}) \end{cases} \tag{11}$$

where U_{odp} is repulsive potential generated by obstacle, ρ_0 is positive constant describing influence of the distance to obstacle and η is positive constant. If $\rho_n\left(p, p_{prek}\right) < \rho_m\left(v_{RP}\right)$, repulsive potential is not defined, because there is no solution how to avoid the obstacle. If robot is far away from obstacle, $\left[\rho_n\left(p, p_{prek}\right) - \rho_m\left(v_{RP}\right)\right] \ge \rho_0$, movement of the robot is not influenced by the obstacle. If robot reaches the area of obstacle, $\rho_n\left(p, p_{prek}\right) \cong \rho_m\left(v_{RP}\right)$, repulsive potential rises to the infinity. With increasing of the relative velocity v_{RP} repulsive potential is also increased.

New repulsive force is defined:

$$F_{odp}(p,v) = -\nabla U_{odp}(p,v) = -\nabla_p U_{odp}(p,v) - \nabla_v U_{odp}(p,v) \tag{12}$$

Relative velocity of the robot in regard to the obstacle is defined:

$$v_{RP} = \left[v - v_{prek}\right]^T .n_{RP} = \frac{\left[v - v_{prek}\right]^T .\left(p_{prek} - p\right)}{\left\|p_{prek} - p\right\|}\,. \tag{13}$$

Gradients of relative velocity of the robot are defined:

$$\nabla_v v_{RP} = n_{RP}$$
$$\nabla_p v_{RP} = \frac{1}{\left\|p - p_{prek}\right\|}\left[v_{RP} n_{RP} - \left(v - v_{prek}\right)\right], \tag{14}$$

where $v_{RP} n_{RP}$ is element of the velocity $v - v_{prek}$ in direction from robot to the obstacle.

Let $v_{RP\perp} n_{RP\perp}$ is perpendicular element to $v_{RP} n_{RP}$:

$$v_{RP\perp} n_{RP\perp} = v - v_{prek} - v_{RP} n_{RP}\,, \tag{15}$$

where:

$$v_{RP\perp} = \sqrt{\left\|v - v_{prek}\right\|^2 - v_{RP}^2}\,. \qquad n_{RP\perp}^T .n_{RP} = 1 \tag{16}$$

Equation (14) can be transferred:

$$\nabla_v v_{RP} = n_{RP}$$
$$\nabla_p v_{RP} = \frac{1}{\left\|p - p_{prek}\right\|} v_{RP\perp} n_{RP\perp}\,. \tag{17}$$

Relations between mentioned vectors can be seen on Fig. 7.

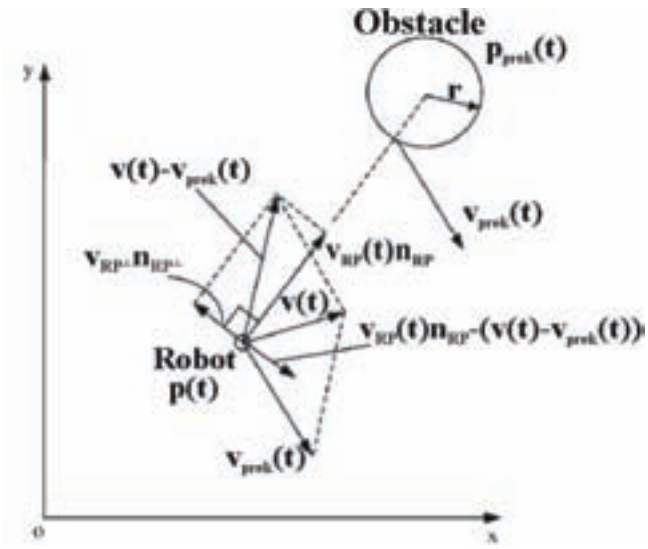


Fig.7 Relations between vectors used in definition of repulsive potential. [3]

Repulsive force (Fig. 8) generated by the obstacle is defined:

$$F_{odp}(p,v) = \begin{cases} 0 & if\ \left|\rho_n(p,p_{prek}) - \rho_m(v_{RP})\right| \ge \rho_0 \vee v_{RP} \le 0 \\ F_{odp1} + F_{odp2} & if\ 0 < \left|\rho_n(p,p_{prek}) - \rho_m(v_{RP})\right| < \rho_0 \wedge v_{RP} > 0 \\ nedef. & if\ v_{RP} > 0 \wedge \rho_n(p,p_{prek}) < \rho_m(v_{RP}) \end{cases} \tag{18}$$

where:

$$F_{odp1} = \frac{-\eta}{\left[\rho_n\left(p, p_{prek}\right) - \rho_m\left(v_{RP}\right)\right]^2}\left(1 + \frac{v_{RP}}{a_{\max}}\right) n_{RP}$$
$$F_{odp2} = \frac{-\eta . v_{RP} . v_{RP\perp}}{\rho_n\left(p, p_{prek}\right) . a_{\max} . \left[\rho_n\left(p, p_{prek}\right) - \rho_m\left(v_{RP}\right)\right]^2} n_{RP\perp} \tag{19}$$

F_{odp1} pushes robot away from the obstacle and F_{odp2} is control force within obstacle avoidance and its direction is in $v_{RP\perp} n_{RP\perp}$. In real applications dimensions of the robot must be considered. Let the diameter of the robot is denoted as ρ_{rob}^* and $\rho_{rob} > \rho_{rob}^*$ is security zone around the robot. Then repulsive potential is defined:

$$U_{odp}(p,v)=\begin{cases}0 & if\left[\rho_n(p,p_{prek})-\rho_{rob}-\rho_m(v_{RP})\right]\geq\rho_0 \vee v_{RP}\leq 0\\ \eta\left(\dfrac{1}{\rho_n(p,p_{prek})-\rho_{rob}-\rho_m(v_{RP})}-\dfrac{1}{\rho_0}\right) & if\ 0<\left[\rho_n(p,p_{prek})-\rho_{rob}-\rho_m(v_{RP})\right]<\rho_0 \wedge v_{RP}>0\\ nedef. & if\ v_{RP}>0 \wedge \left[\rho_n(p,p_{prek})-\rho_{rob}\right]<\rho_m(v_{RP})\end{cases} \quad (20)$$

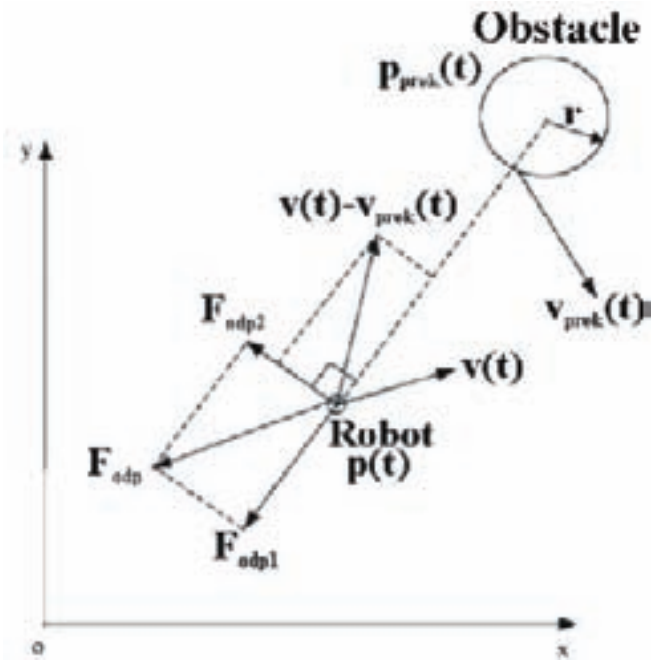


Fig.8 Repulsive force and its elements. [3]

ρ_{rob} represents security zone around the robot, but it can also represent uncertainty in sensors measurements. Generally ρ_{rob} is higher when robot has inaccurate sensors or large security zone is needed within navigation. On the contrary ρ_{rob} is lower when robot has accurate sensors as laser rangefinder or when gentle touches between robot and obstacles are permitted.

Resulting force effecting on robot is:

$$F = F_{prit} + F_{odp}\,.$$

If there are more obstacles, then:

$$F_{odp} = \sum_{i=1}^{o} F_{odp,i}\ ,$$

where o is number of the obstacles and $F_{odp,i}$ is repulsive force generated by obstacle number i.

Acknowledgments

The work presented in this contribution has been supported by the grant VMSP-P-0004-09 and KEGA3/7307/09.

References

[1] GIESBRECHT J.: Local Navigation for Unmanned Ground Vehicles, A Survey. Defence R&D Canada – Suffield, Technical Memorandum, DRDC Suffield TM 2005-038, December 2005.

[2] FIORINI P., SHILLER Z.: Motion planning in dynamic environments using velocity obstacles. International Journal of Robotics Research, 17(7), July 1998, p. 760-772.

[3] GE S.S., CUI Y.J.: Dynamic Motion Planning for Mobile Robots Using Potential Field Method. Autonomous Robots 13, 2002, p. 207–222.

[4] BABINEC A., VITKO A.: Histogram navigational algorithms - development and principles (In Slovak: Histogramové navigačné algoritmy - vývoj a princíp). Automa 5/2010, p. 26-29. ISSN 1210-9592.

[5] KÖNIG R., KELEMEN M.: Inspection robot GTR2006 (In Slovak: Inšpekčný robot GTR2006). Strojárstvo 10/2006, p. 60-61. ISSN 1335-2938.

[6] KONIAR D., HARGAŠ L., HRIANKA M., BOBEK V., DRGOŇA P., FIBICH P.: Kinematics analysis of biomechanical systems using image analysis. In: Metalurgija - Vol. 49, br. 2 (2010), p. 350-355.

prof. Ing. Ladislav Jurišica, PhD.,

Ing. František Duchoň, PhD.

Ing. Martin Dekan

Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3
812 19 Bratislava
ladislav.jurisica@stuba.sk, frantisek.duchon@stuba.sk

Programming of mobile robot with RoboRealm

Ladislav Jurišica, František Duchoň, Juraj Tóth

Abstract
Research, development and education in the mobile robotics requires verification of properties of different systems and verification of various control algorithms. It is useful if these activities can be performed effectively, with minimal costs and with minimal time. In this paper are analyzed possibilities of the programming system RoboRealm, which simplifies several tasks in research, development and education of the mobile robots.

Key words: mobile robot, robot programming, computer vision

Introduction

Mobile robotics as realm of robotics requires solution of many tasks. It is proper that some tasks can be verified as simulation, without large costs and in short time. For that, verification in area of programming of simple tasks in mobile robotics, RoboRealm seems as proper programming system.

1. Programming system

RoboRealm is application used for control of mobile robot systems, which mainly uses algorithms of computer vision. By simple clicking a program can be made up. With this simple programming, robot can obtain an "ability to see", ability to recognize objects etc., whereby result is shown in pseudo-real time. RoboRealm contains over 240 of programming modules. In addition rich documentation and many examples are situated on the website of RoboRealm [DOC].

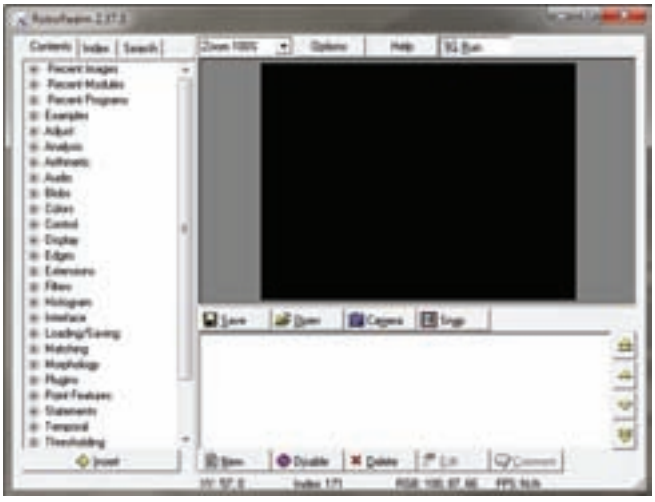


Fig.1 GUI of RoboRealm.

RoboRealm has quite simple GUI (Fig. 1). Left window contains every available module. Modules are logically separated into bigger thematic groups, so limpidity and simplicity is guaranteed. In the middle part, there is an output of the camera. In the lower part modules can be added into programming sequence. This is the part, where the main program is formulated.

For the programming a system of pipes is used. This system is sorted out from UNIX. In other words, output of one module is the input of next module. Program consists of sequences created from filters, scripts, transformations etc. Programs are saved in *.robo format, which is just an XML file.

Support of several cameras as Linksys, Creative, firewire cameras, etc. is directly supported in RoboRealm. Recently was added support for Microsoft Kinect. Among supported robots, commercially robots as Surveyor, Rovio, Spykee, Lego NXT, BoeBot etc. are available. At this place the possibilities of RoboRealm are not closed. RoboRealm can work even with different types of servos, motors, gadgets etc. (e.g. Striker USB Missile Launcher, see [DEV] for more).

RoboRealm provides different conventional modules such as edge detection, color adjustment, morphology, transformations, number of objects, size detection, orientation detection, arithmetic operations etc. As more sophisticated modules, object detection, bar code detection, wall, ground and sky retrieval, GPS module, movement detection, etc. should be mentioned.

Every object created in RoboRealm is accessible in the programming system. They can be internally processed by scripting languages as Visual Basic, Python and picoC. If a communication with other application is needed, RoboRealm provides an API server, where communication by sockets and XML messages is ensured. This is solved by API interface in the source code of application, which can be downloaded from web page of RoboRealm [API]. Downloaded file contains this API processed in several programming languages as C++, Python, Lisp, Java, etc.

Simple example of communication between application and RoboRealm for reading variable cogX can be showed:

1. In source code application reads the value:

```
String cogX = rr.getVariable("COG_X");
```

2. A request is send to RoboRealm:

```
<request>
  <get_variable>COG_X</get_variable>
</request>
```

3. RoboRealm will send an answer:
```
<response>
  <COG_X>10</COG_X>
</response>
```
4. To the string cogX value of COG_X is saved, which is actually equal to 10.

Minimal requirements of RoboRealm on hardware are:

- Memory - 64 MB
- HDD - 100MB (installation has over 8 MB)
- Graphic card of True color type - 1024x768 px
- Keyboard, mouse
- Webcam with VFW or DirectX support - resolution 320x240 px, 30fps, USB1.0
- CPU with 2 GHz or more
- OS Windows 98 or newer (works with Windows 7 too)

RoboRealm doesn't support other OS such as Linux, because RoboRealm is dependent on library Video for Windows.

2. Applications

One of the solved tasks in mobile robotics is in advance known landmarks searching [KÖNIG]. This task can be solved in RoboRealm too. Main part of the solution is module Matching → Fiducials. At first it is needed to train this module onto landmarks identification. Fig. 2 shows default landmarks used with RoboRealm. Own landmarks can be used too.

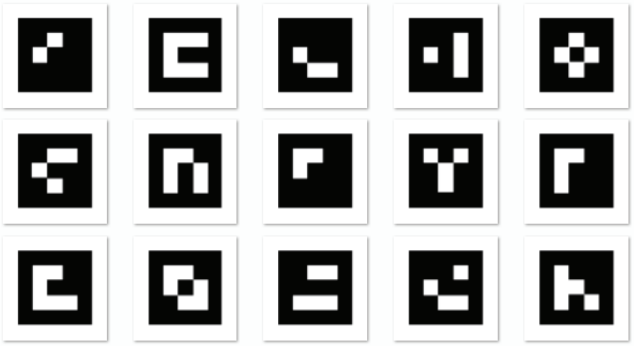


Fig.2 Package of default landmarks used in RoboRealm.

Movement of autonomous robot on color line is shown on other simple example. Spykee [SPYKEE] robot was used or more precisely the module Controls → Robots → Erector_Spykee. This robot contains simple color camera with resolution 320x240 px, differentially driven conveyor chassis and communication module based on Wi-Fi.

Fig.3 Basic view on the scene.

Fig. 3 shows basic view on the scene. Besides the blue line, on which robot has to move, there are also three other lines with different colors. These lines must be filtered out. Therefore at the beginning of the image processing, filter Colors → RGB_Filter is applied (Fig. 4). With appropriate parameters, this filter passes blue color only. The module of this filter works on principle of normalization, or more precisely if the blue color is selected, then for particular color channels stand:

$$R = 0, G = 0, B = \left(\left(B - R\right) + \left(B - G\right)\right)$$

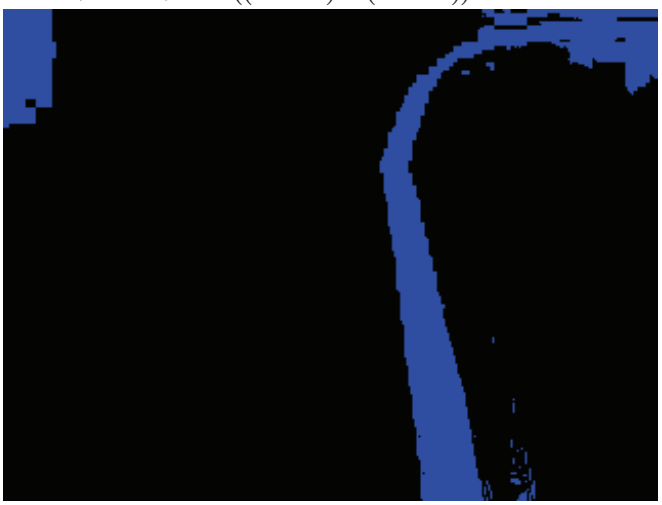


Fig.4 Application of RGB_Filter.

From the original image, only blue color line remains, but there is also noise from the surroundings. Assumption, that blue color line will occupy the biggest area in the image is used. Therefore for the noise removal [KONIAR] module Blobs → Blob_Size is used. In our case, parameter Limit to was set at a value of 1. With this step it is achieved, that only one area with blue color remains. This area defines sought line.

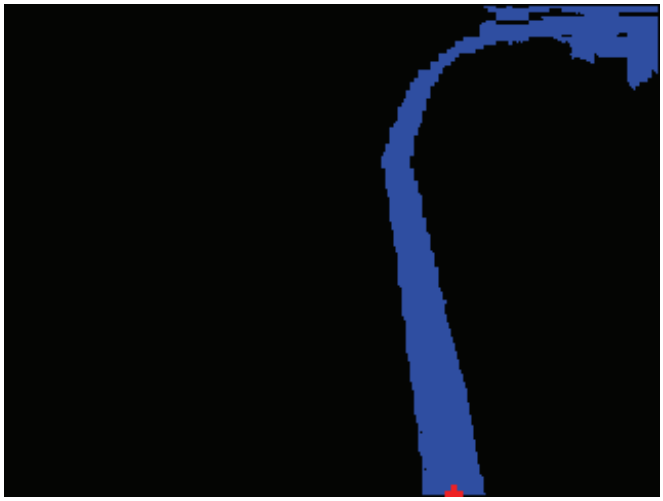


Fig.5 Application of Blob_Size and Point Location.

Finally the coordinates of centre of the lowest blue line must be acquired. This can be done with module Point Features → Point Location, in which Lowest (Middle) is selected. With that parameter module returns two coordinates LOWEST_MIDDLE_X and LOWEST_MIDDLE_Y (range depends on used camera).

Behavior of the robot itself can be defined with module Extensions → VBScript_Program (for Visual Basic). For complex operation of the Spykee robot, in module Erector_Spykee variables LEFT_MOTOR and RIGHT_MOTOR must be set up (with range 0-255, value 128 stands for stop).

```
lowestMiddleX = GetVariable("LOWEST_MIDDLE_X")

if lowestMiddleX < 0 then           'rotate when no blue line
  leftMotor = 128-5
  rightMotor = 128+5
elseif lowestMiddleX < 70 then      'go left
  leftMotor = 128-2
  rightMotor = 128+5
elseif lowestMiddleX > 250 then     'go right
  leftMotor = 128+5
  rightMotor = 128-2
else                                'go straight
  leftMotor = 128+5
  rightMotor = 128+5
end if

SetVariable("LEFT_MOTOR"), leftMotor
SetVariable("RIGHT_MOTOR"), rightMotor
SetVariable("LOWEST_MIDDLE_X"), -1
```

Fig.6 Definition of robot's behavior.

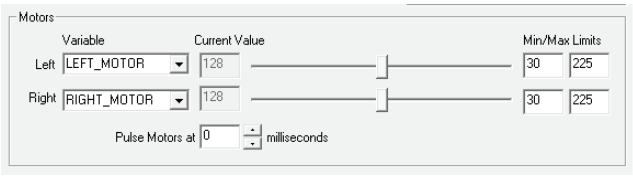


Fig.7 Setting of variables in module Erector_Spykee.

After initialization of program, robot begins to turn around its own axis. After a blue color line is found, robot begins to follow this line. If coordinate lowestMiddleX is far away from the middle of image, robot stops and turns so that this coordinate will be in the middle of image. In so far that appropriate control intervention on both motors is performed. If robot reaches end of the line, it will turn around its own axis again until it will find the blue color line.

Conclusion

Programming environment RoboRealm simply and effectively provides verification of methods used in computer vision. And moreover, with RoboRealm can be verified control algorithms of mobile robotic systems too. Accordingly RoboRealm can be used as powerful tool in mobile robotics research and education.

Acknowledgments

This work was supported by grants VMSP-P-0004-09 and KEGA 3/7307/09.

Literature

[DOC] http://www.roborealm.com/help/index.php

[DEV] http://www.roborealm.com/help/index.php#Control

[API] http://www.roborealm.com/help/API.php

[SPYKEE] http://www.spykeeworld.com/spykee/UK/index.html

[KONIAR] KONIAR D., HARGAŠ L., HRIANKA M., BOBEK V., DRGOŇA P., FIBICH P.: Kinematics analysis of biomechanical systems using image analysis. In: Metalurgija - Vol. 49, br. 2 (2010), p. 350-355.

[KÖNIG] KÖNIG R., KELEMEN M.: Inspection robot GTR2006 (In Slovak: Inšpekčný robot GTR2006). Strojárstvo 10/2006, p. 60-61. ISSN 1335-2938.

prof. Ing. Ladislav Jurišica, PhD.

Ing. František Duchoň,PhD.

Bc. Juraj Tóth

Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3
812 19 Bratislava
ladislav.jurisica@stuba.sk, frantisek.duchon@stuba.sk, iuryno@gmail.com

Ultrasonic mapmaking system

Jaroslav Hanzel, Ladislav Jurišica, Marian Kľúčik, Anton Vitko

Abstract
The problem of building a map of an unknown environment by mobile robot's control system is considered. The sensory system is equipped with an ultrasonic range finder used to sense the working environment. A grid-based representation of the operating space is computed. The goal is determination of occupied and empty areas in environment. In order to obtain the more abstract environment representation, the occupancy grid is processed to a hybrid representation consisted of metric and topological part. This hybrid map is suitable for robot navigation in the broad environments.

Keywords: ultrasonic rangefinder, occupancy grid, mathematical morphology, hybrid map

Introduction

Fully autonomous performance of given tasks by mobile robots is the main goal of recent investigation in the field of mobile robotics [1], [5]. The interest is focused on the problem of generating a working environment representation for the navigation tasks. The robot motion over a trajectory is considered. At the same time the robot collects data from its sensory system. The most common sensors [1], such as sonars and laser-scanners, detect the distance to an obstacle within range of the sensor by sending out a signal and measuring the time until the echo of the signal returns. In the majority of occasions the signal will bounce against the nearest obstacle in the direction of the sensor and therefore the measured distance will be the distance to the nearest obstacle. Certainly, there are some reasons for measurement failures, and thus the measured distance is generally erroneous. The acquired distance can be used by many ways. One of them is the creation of the environment representation, which is in principle a map of the environment.

1. Environment representations

The actual research in the mobile robotics produced two fundamental approaches for environment modelling: the metric and the topological paradigm. These two concepts are essentially different at the qualitative as well as quantitative levels. Metric approaches are characterized by relatively large amount of rather accurate data about the spatial structure of the environment. The amount of the data is strongly dependent on the dimensions of the modelled environment and therefore these methods are especially suitable for the construction of smaller local maps with higher accuracy. On the other hand the topological approaches express the spatial structure of the environment by more abstract form. This feature allows an effective representation of wide-range environment and is applicable for global maps of the environment.

The metric maps represent spatial structure of the environment by basic geometric elements. These elements can be in principle rigid or adaptive. A typical example of environment representations by rigid geometric elements are in robotics widely used occupancy grids, first introduced by Moravec and Elfes, [10], [2]. The occupancy grids represent the environment by its discretization into the regular tessellation. The basic elements of a grid are called cells. The cells represent the properties of the environment by numerical values. The advantage of the occupancy grids is relatively accurate model of the environment. The disadvantage is mainly the amount of data processed and needed for accurate model of larger environment.

The second class of modelling methods is based on topological properties of a given environment. These approaches abstract some essential attributes of the environment and determine significant relations between them [6], [7], [8], [9]. The environment is mostly modelled by graphs. The edges in the graph reflect space or functional relation between nodes. The nodes correspond to distinct situations, locations or landmarks. If a direct path exists between nodes, these are connected with edges in the graph. The main advantage of topological representation is its compactness and the applicability for fast path planning. The main disadvantage of topological representation is relatively small amount of information about nearest neighbourhood of the robot.

2. Sensory device

Nowadays, the ultrasonic range finders (sonars) are widely used for measuring relative distances in many technical domains [1]. They have a very broad possibility of exploitation due to their simplicity of operation, low cost and modest realisation. In mobile robotics the sonars are used for measuring distances to the obstacles in the robot neighbourhood. The ultrasonic range finders work according to a simple principle: a packet of ultrasonic waves is generated and the resulting echo is detected. The time elapsed between a transmission and a reception is assumed to be proportional to the distance of the sensed obstacle. Thereafter it will be referred to as Polaroid ultrasonic ranging system widely used in mobile robotics [11].

In the process of determination of an object presence and its relative distance with the ultrasonic sensors, there are basically three main sources of measurement uncertainty. First, the measured distance r is affected by an error. In the case of the Polaroid range finder which can detect distances from 0.15 to 10.7 m, the error of measurement is about +/- 1% of the measured distance over the entire range [11]. This uncertainty is caused by the characteristics of air such as its temperature, humidity, turbulence and pressure [14]. The second uncertainty results from the propagation of the ultrasonic signal to the space in the form of a cone with an

axis in the scanning direction. So the exact angular position of the object reflecting the echo might not be determined. The angle of radiation can be in fact fairly wide. For practical purposes a diffusion of ultrasonic waves over a radiation cone of 25° width is considered. The third source of uncertainty is a phenomenon of multiple reflections, which may occur in the case that the incidence angle of signal to the obstacle is larger than a so-called *critical angle* which is strongly dependent on the surface characteristics. In this case the reflection of the signal is mainly specular and the sensor may receive the ultrasonic signal after multiple reflections, which is called a *long reading,* or it may even disappear. Therefore, in order to return a significant range reading, the angle of incidence on the object surface has to be smaller than the critical angle.

3. Occupancy grids

The occupancy grids provide an effective framework for the data fusion from multiple sensors and sensing positions. When a reading from a sensor is taken, the sensor model is overlaid on the grid, and each cell is updated. Occupancy grids are mostly used for 2-D representation of the floor plan of a scene. Each cell of the grid represents some area and contains information about the space that it represents.

A two-dimensional environment U is discretized into a bitmap structure of finite number of square elements with size of edge δ called cells. The set U can be formally written as $U = \{c_1, \ldots, c_M\}$ for $j = 1, \ldots, M$. A set of range readings $R = \{r_1, \ldots, r_n\}$, collected at known locations $L = \{l_1, \ldots, l_k\}$ is assumed. In principle, the task for a map generating system is to process the range data in order to determine, which cells are (even partially) occupied by obstacles, and which are (completely) empty. In this manner, the set U is divided into the set of empty cells E and the set of occupied cells O. It is possible due to the presence of uncertainty to provide only an estimate, of which cells belong to the set E and which ones belong to the set O. A compact way to represent this estimate is through navigation map N, in which to each cell c_j from U a real number $v(c_j) \in <0, 1>$ is assigned, which incorporates the information about c_j gathered from R. The interpretation as well as the computation of N depends on the particular method chosen for managing uncertainty. The higher values in the navigation map N denote the higher probability that the given cell is occupied.

Methods that make use of Bayesian estimation are the most common data fusion methods in occupancy grid creation [2], [10]. With this approach each cell c_j of the grid stores the probability of being occupied by the obstacle in the form of a *state variable* $s(c_j)$. The goal is the computation of that probability on the basis of measurements R. The cell states are exclusive and exhaustive and satisfy the condition

$$P[s(cj) = O] + P[s(cj) = E] = 1 \quad (1)$$

for all cells from U.

The sequential updating formulation of Bayes' theorem allows the incremental composition of sensory information. For a current estimate $P(s(c_j) = O|r_1, \ldots, r_{n-1})$of the state of a cell c_j based on the observations $R = \{r_1, \ldots, r_{n-1}\}$, and a new observation r_n, the improved estimate is given by

$$P(s(c_j) = O \mid r_1,\ldots,r_n) = \frac{p(r_n \mid s(c_j) = O)P(s(c_j) = O \mid r_1,\ldots,r_{n-1})}{\sum_{X\in\{E,O\}} p(r_n \mid s(c_j) = X)P(s(c_j) = X \mid r_1,\ldots,r_{n-1})} \quad (2)$$

and $X \in \{E,O\}$. $P(s(c_j) = X|r_1, \ldots, r_{n-1})$ is the estimate of the cell state on the basis of measurements $r_1, \ldots, r_{n-1}$ and $p(r|s(c_j) = X)$ is determined from the sensor model. For the initial cell state the probability estimates are used with the maximum entropy $P(s(c_j) = E) = P(s(c_j) = O) = 0.5$ [2]. With these initial values the state of the cell is set to "unknown". The navigation map N is obtained assigning $v(c_j) = P(s(c_j) = O|r_1, \ldots, r_n)$.

4. Experimental data acquisition

The experimental data gathering was performed with a movable platform equipped with the Polaroid sensor device on the top. The experimental environment was a long hallway with the length about 20 *m* and width approximately 1.8 *m*. The sensor device was able to rotate in full, to an angle of 360°. The measuring locations were evenly distributed with an interval of 0.5 m to obtain a good coverage of the environment. Along the hallway 39 measuring locations were placed. At each location a data set composed of the 400 measurements was collected by the sensor rotation with a fixed step 0.9°. The universal set U for the hallway had 300×100 cells of the size $\delta = 0.1m$. The values of the parameters were $p_O = 0.6$, $\rho_v = 1.2$ *m* and $\Delta r = 0.15$ *m*.

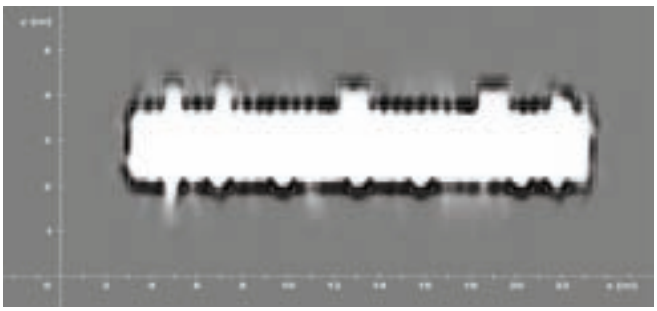


Fig.1 Resultant navigational map.

The algorithms were implemented in C language on PC equipped with 2500 *MHz* Athlon processor running under Linux. The resultant navigation map is depicted as grey scale images in the Fig. 1. The colour in the image indicates the level of confidence in the proposition, that the corresponded environment area is occupied with the obstacles. The lighter areas $v(c_j) \rightarrow 0$ represent the empty cells and the darker areas where $v(c_j) \rightarrow 1$ represent occupied cells. The boundaries of the walls are represented with the white lines. The measuring locations are depicted as small crosses. The grey background colour represents entire ignorance about the state of the cells in the map.

5. Hybrid maps based on methods of mathematical morphology

The ideal representation should be able to support the higher level tasks such as strategic planning or situation estimation as well as the lower level tasks such as obstacle avoidance, localization or path planning. That property is possible to gain with combination of appropriate features of the metric and the topological representations into one representation, which is sometimes called a hybrid map [13]. The hybrid maps represent information about the space in two different levels of abstraction, in the geometric and the topological level. These maps consist essentially of the local metric maps connected into the topological net. By the navigation in the environment the robot uses topological map for planning the sequencies of sectors across which it must move to reach the target. Metric map is used for navigation inside the sectors and for transition between them. The construction of such representation is not a simple task. It can be divided to two basic tasks. The first is the collection of the measurements with appropriate data processing to the local metric map. The second is extraction of useful

features from local map and their combination in the topological map.

The input data set for the hybrid map construction method is the navigation map *N*, which is the output of the methods for metric map construction. It has the form of a matrix of real numbers from interval $<0, 1>$. The levels of grey scale can be assigned to the values in the grid and though the map can be treated like a grey scale image. In this manner it is possible to employ the image processing methods in development of the solution. One interesting and effective class of image manipulation techniques are methods based on mathematical morphology.

The term "morphological analysis" describes a range of nonlinear image processing techniques that deal with the shape or morphology of features in an image. Most morphological analysis techniques operate on binary images. The morphological operations use a small shape or template known as a *structuring element* (*SE*). The structuring element is positioned at all possible locations in the image and is compared to the corresponding neighbourhood of pixels. The structuring element applied to a binary image can be represented as a small matrix of pixels, each with a value of 1 or 0. The dimensions of the matrix determine the size of the *SE*, and its shape is determined by the pattern of ones and zeros. The structuring element has an origin, which allows the positioning of the *SE* at a given pixel. The simplest morphological operations are erosion and dilation [4]. These operations form the basic ingredients of mathematical morphology. Morphological operators discard information. In general, it is not possible to recover the original image after dilation or erosion. If one erodes an image *X* with a structuring element *A* and subsequently dilates it with the same *SE*, one ends up with an image which is smaller than the original one. Such combined operation is called the (structural) opening by *A*. Dually, first dilating, then eroding, yields a set which is larger than *X*. The operator is called the (structural) closing of *X* by *A*. These are the basic operators utilized by the mathematical morphology, although there is a huge set of other more complicated operators.

6. Map computation

The input data set is the grey scale image of experimental environment. The size of the image is equal to the size of the occupancy grid. Thus one pixel in the image with coordinates (*x*, *y*) corresponds to one cell in the grid with same coordinates. Likewise the grey-scale values *f*(*x*, *y*) of the image function in the input image correspond to the values stored in the grid of the form of the navigation map *N*. The background colour in the image is shade of grey assigned to initial conditions of probabilistic mapmaking algorithm ($P(s(c_j) = E) = P(s(c_j) = O) = 0.5$ in map *N*, and $f(x, y) = 128$ in the 8 bit grey scale image). The occupied cells are depicted with darker and empty cells with brighter shades of grey as the unknown cells.

There is a need to divide the pixels in the image to three sets: the set of the occupied cells denoted as *OCC*, the set of empty cells denoted as *EMP*, and the set of unknown cells *UNK*. Altogether the separation of the cells consists of finding out the two threshold values of grey levels. These thresholds divide the values of image scale from <0, 255> to three intervals defining the assignment of the pixels into the sets *OCC*, *EMP* and *UNK*. The effective source of the information needed for this task is a histogram of the image. After this manner there are acquired three binary images representing the sets *OCC*, *UNK* and *EMP*, in which the value of 1 represent membership of given cell to the set and 0 is background colour.

Determination of the object boundaries which hold the metric information about occupied space is now necessary. Before the processing of data, it is necessary somehow accommodate the image to minimize division of object boundaries. It can be accomplished by morphologic operator of closing. The boundaries of the objects in the map can be obtained by computation of object contours in the image. The contours can be computed by application of erosion operator at image and by subtraction of eroded image from original image, and the set of object boundaries is obtained. In the following computations the set of cells representing the skeletons of the objects is needed due to their useful features. The skeletons can be computed by sequential thinning with sequence of operators *L* of Golay alphabet [4].

The desired metric information in hybrid map is the visible part of object boundaries. Therefore determination of cells of object boundary which were really detected by sensor is needed. The fact that the visible boundary of any object occurs at the transition between empty and occupied space is employed. For determination of this transition, the set *EMP* is repeatedly dilated under following condition. The image pixel is going to belong to dilated image if it is not member of set of object skeletons. This condition allows to pixels from set *EMP* to reach after few iterations the skeletons of objects and to determine the visible side of the objects. The skeletons are lines in image with thickness of 1 pixel, and this feature allows us to define the ending condition for the iterations. The empty space is dilated until all visible objects of the measured range are reached, and this maximal empty area is stored. Then the visible boundaries are computed like conjunction of dilated empty space and set of object boundaries. The metrical part of hybrid representation is computed by this manner and it is visible in the Fig. 2.

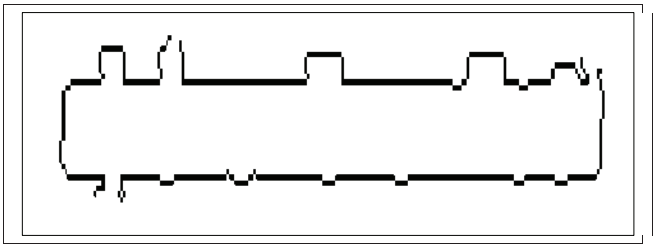


Fig. 2 The final metric representation of the measured environment.

The topological part is computed from the empty cells which absolutely fill the space between the objects defined by their boundaries. The first step is the repeated erosion of dilated empty set until nonzero conjunction with the boundary set is there. Then this eroded empty set is once again conditionally dilated (with condition defined by boundary set) conditionally till it reaches the maximal empty area stored before. Now whole reachable empty space between the object boundaries is defined by this set. From this set the skeleton is extracted, which can be used directly as the topological representation of empty space. The topological part of hybrid map is depicted in Fig. 3. The skeleton of empty space composes of a set of 1 pixel-thick lines, which approximately follow the mean outlines of the empty space. The crossings of the lines of the skeleton can be interpreted as the nodes in the topological graph of the environment and the edges of graph are the lines of the skeleton.

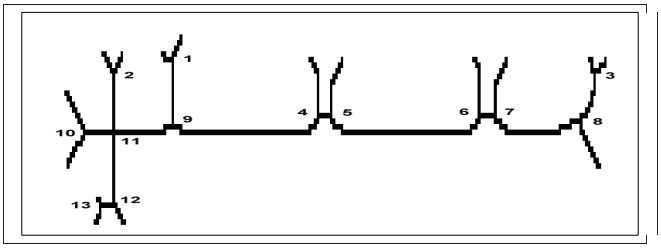


Fig. 3 The final topological representation of the measured environment.

Conclusion

The methodology for construction of hybrid representation of the operating environment for mobile robots is proposed. The sensing of space is done by ultrasonic range finder. The measured data can be processed by well-known algorithms for occupancy grids creation based on the theory of probability. The visible boundaries of the measured objects in the form of line segments are extracted from the occupancy grid and they are stored as the metrical representation of the environment. Then a set of reachable empty cells is extracted and its skeleton is stored as the topological representation of the environment. In this manner, the requirements of the minimal storage demands and maximal informational content of the hybrid representation are accomplished. These two complementary parts of a hybrid map allow to robot's control system successfully accomplish the given tasks. The obtained results give the good starting point for the future research.

Acknowledgements

The work has been supported by VEGA grant 1/0690/09 and grant VMSP-P-0004-09. This support is very gratefully acknowledged.

References

[1] BORENSTEIN, J., EVERETT, B., FENG, L.: Navigating Mobile Robots: Systems and Techniques, A. K. Peters, Ltd., Wellesley, MA, 1996, ISBN 1-56881-058-X.

[2] A. ELFES: "Using occupancy grids for mobile robot perception and navigation", *Computer Magazine*, June, 1989, pp. 46-57.

[3] GAMBINO, F., ORIOLO, G., ULIVI, G.: "A comparison of three uncertainty calculus techniques for ultrasonic map building", In.:*SPIE 1996 Int. Symp. on Aerospace-Defence Sensing and Control*, Orlando, Fl, 1996, pp.249-260.

[4] HLAVÁČ, V., ŠONKA, M.: *Počítačové vidění*. Praha: Grada, 1992.

[5] KORTENKAMP D., BONASSO, P. R., MURPHY, R.: "Artificial Intelligence and Mobile Robots, Case Studies of Successful Robot Systems", AAAI Press/The MIT Press, 1998.

[6] KUIPERS, B., J.: "Modeling spatial knowledge", *Cognitive Science*, vol. 2, 1978, pp. 129-153.

[7] KUIPERS, B., J.: "The Spatial Semantic Hierarchy", *Artificial Intelligence*, vol. 119, 2000, pp. 191-233.

[8] KUIPERS, B., J., BYUN, Y., T.: "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", *Journal of Robotics and Autonomous Systems*, vol. 8, 1991, pp. 47-63.

[9] MATARIČ, M., J.: "A distributed model for mobile robot environment learning and navigation", MIT EECS Master's Thesis, Jan 1990, MIT AI Lab Tech Report AITR-1228, May 1990.

[10] MORAVEC, H. P.: "Certainty Grids for Mobile Robots", *Proceedings of the NASA/JPL Space Telerobotics Workshop*, Vol. 1.,1987 pp. 307-312.

[11] Polaroid, "Polaroid ultrasonic ranging system handbook – application note/technical papers", 1992.

[12] RIBO, M., PINZ, A.: "A comparison of three uncertainty calculi for building sonar-based occupancy grids", *Robotics and Autonomous Systems*, Vol. 35. No. 3-4., 2001, pp. 201-209.

[13] THRUN, S.: "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence* 99, 1998, pp. 21-71.

[14] TOMAN, M.: "Ultrazvuk pre priestorové merania." *AT&P JOURNAL plus2*, roc. 7, 2001, s. 66-75.

Ing. Jaroslav Hanzel, PhD.
Prof.Ing.Ladislav Jurišica,PhD.
Ing.Marian Kľúčik
Doc.Ing.Anton Vitko,PhD.

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Ilkovičova 3
812 19, Bratislava 1
Slovakia
+421 2 602 91 864
jaroslav.hanzel@stuba.sk

Archív

Archive

2001 AT&P journal PLUS 1: Adaptívne a nelineárne riadenie systémov (tlačená verzia)
Adaptive and nonlinear control systems (printed version)

AT&P journal PLUS 2: Robotika, mechatronika, diskrétne výrobné systémy (tlačená verzia)
Robotics, mechatronics, discrete manufacturing systems (printed version)

2002 AT&P journal PLUS 3: Robustné systémy riadenia (tlačená verzia)
Robust control systems (printed version)

2003 AT&P journal PLUS 4: Samonastavujúce sa systémy v riadení procesov (tlačená verzia)
Selftuning systems in process control (printed version)

2004 AT&P journal PLUS 5: Robotické systémy (elektronická – CD verzia)
Robotics systems (electronic – CD version)

2005 AT&P journal PLUS 6: Mechatronika (elektronická – CD verzia)
Mechatronics (electronic – CD version)

AT&P journal PLUS 7: Umelá inteligencia v praxi (elektronická – CD verzia)
Artifical intelligence in Practise (electronic – CD version)

2006 AT&P journal PLUS 1: Mechatronické systémy (elektronická – CD verzia)
Mechatronic systems (electronic – CD version)

AT&P journal PLUS 2: Inteligentné meracie systémy (elektronická – CD verzia)
Intelligent measurement systems (electronic – CD version)

2007 AT&P journal PLUS 1: MMaMS‘2007 (elektronická – CD verzia)
MMaMS‘2007 (electronic – CD version)

AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process Control (electronic – CD version)

2008 AT&P journal PLUS 1: Mobilné robotické systémy (elektronická – CD verzia)
Mobile robotic systems (electronic – CD version)

AT&P journal PLUS 2: Riadenie v energetike (elektronická – CD verzia)
Control of Power Systems (electronic – CD version)

2009 AT&P journal PLUS 1: Inteligentné pohybové systémy (elektronická – CD verzia)
Intelligent motion control systems (electronic – CD version)

AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process control (electronic – CD version)

2010 AT&P journal PLUS 1: Systémy automatického riadenia (elektronická – CD verzia)
Systems of automatic control (electronic – CD version)

AT&P journal PLUS 2: Robotika vo vzdelávaní (elektronická – CD verzia)
Robotics in education (electronic – CD version)