ISSN 1562 - 2703

INTERNATIONAL JOURNAL AUTOMATION AUSTRIA

Heft 2 im Jahrgang 20 (2012)

Inhalt *Seite*

SCOPE

"International Journal Automation Austria" (IJAA) publishes top quality, peer reviewed papers in all areas of automatic control concerning continuous and discrete processes and production systems.

Only original papers will be considered. No paper published previously in another journal, transaction or book will be accepted. Material published in workshops or symposia proceedings will be considered. In such a case the author is responsible for obtaining the necessary copyright releases. In all cases, the author must obtain the necessary copyright releases before preparing a submission. Papers are solicited in both theory and applications

Before preparing submissions, please visit our instructions to authors (see back cover) or web page.

International Journal Automation Austria also is the official executive authority for publications of IFAC-Beirat Österreich.

Imprint: Propagation of Automatic Control in Theory and Practice.

Frequency: Aperiodically, usually twice a year.

Publisher: IFAC-Beirat Österreich, Peter Kopacek, Alexander Weinmann

Editors in Chief: Alexander Weinmann, Peter Kopacek

Address: 1) Institut für Automatisierungs- und Regelungstechnik (E376), TU Wien, A-1040 Wien, Gußhausstrasse 27-29, Austria
Phone: +43 1 58801 37677, FAX: +43 1 58801 37699
email: danzinger@acin.tuwien.ac.at
Homepage: http://www.acin.tuwien.ac.at/publikationen/ijaa/

2) Intelligente Handhabungs- und Robotertechnik (E325/A6), TU Wien, A-1040 Wien, Favoritenstrasse 9-11, Austria
email: e318@ihrt.tuwien.ac.at

Layout: Rainer Danzinger
Printing: Grafisches Zentrum an der TU Wien

Dummy injury value reduction via crash pulse design

G. Stettinger, M. Horn, W. Leitgeb*, B. Fellner♢
Institute of Smart System-Technologies:
Control and Measurement Systems
Alpen Adria University Klagenfurt, Austria
*Virtual Vehicle Competence Center (ViF) Graz, Austria
♢Magna Steyr Fahrzeugtechnik Graz, Austria

Abstract:
The goal of the presented work is to reduce injury on vehicle occupants during frontal impacts by decelerating their vehicle in an optimized way. To achieve this, a model of the relation between occupant and vehicle is created in order to mathematically model the driver's motion during the frontal impact of the vehicle sufficiently detailed. This model forms the basis for the nonlinear controller design, which generates an optimized crash pulse out of the original one. This optimized crash scenario, characterized by the crash-pulse, leads to a reduction of the probability of suffering head injuries during a frontal impact.

Keywords: Vehicle safety, crash pulse, crash test dummy modelling, frontal impact, sled test, input-output linearization, compensated plant, Head Injury Criterion (HIC)

1 Introduction

An important feature of a safe vehicle is to protect its occupants during an accident. To achieve this, predefined crash scenarios are used to evaluate the vehicle's safety level. These scenarios originate either from regulations (ECE for Europe, FMVSS for USA) or well known consumer tests (Euro NCAP, IIHS, US NCAP) with the goal of improving general vehicle safety. In this work, because of data availability and simplicty of the test setup, the US-NCAP (US-New Car Assessment Program) crash test configuration with focus on the driver is used, see Fig. 1. In this configuration a Hybrid III 50th percentile Male dummy is used to model the driver. This dummy represents an average person, this means that 50% of the people are greater and 50% are smaller than this dummy [8]. According to the US-NCAP test the car crashes into a rigid barrier at a speed of 56km/h.

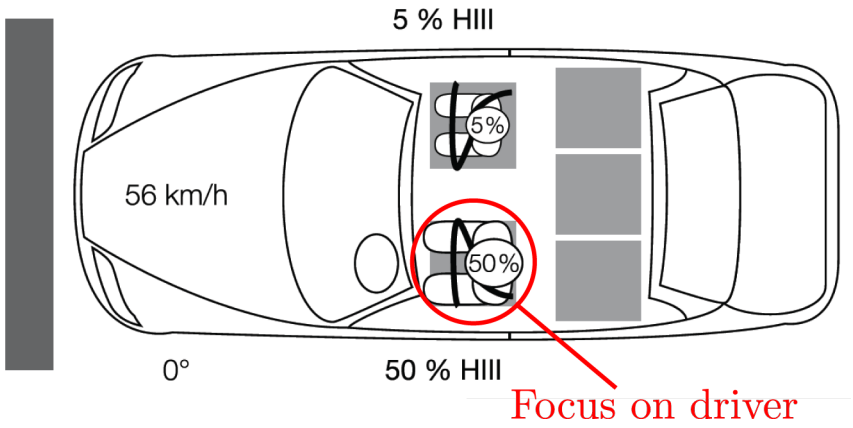

Figure 1: US-NCAP crash configuration [2]

The aim of this work is to reduce injury on vehicle occupants during frontal impacts by decelerating their vehicle in a modified way. The vehicle deceleration is given by the crash pulse. The crash pulse is a characteristic deceleration signal, a vehicle endures, during a defined frontal crash. Every car has its characteristic crash pulse which depends on the chassis construction, vehicle mass, centre of gravity and other vehicle specific parameters. For illustration two crash pulses from two different cars are shown in Fig. 2.

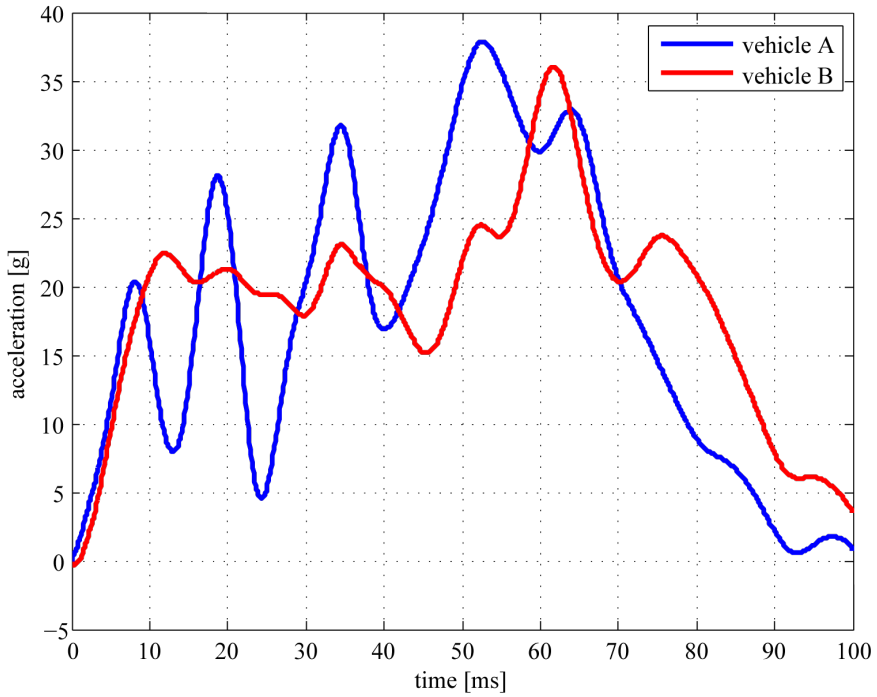

Figure 2: US-NCAP example acceleration signals

State of the art is to influence the effect of the crash pulse on the occupant by means of restraint systems such that the car passes the crash tests. All devices which hold back the occupant and that way lessen the probability of injury during an accident are called

restraint systems. The most common systems are the airbag and the safety belt. These restraint systems are limited by maximum forces and deployment time constraints. To push these limits further and to improve safety, complexity and price increases are inevitable. In this work a different way is chosen to reduce the probability of occupant injury. Here the focus concentrates on developing an improved, theoretical crash pulse to reduce one relevant dummy injury value but leaving the restraint systems identical. The observed dummy value in this case is called Head Injury Criterion (HIC), it represents the probability of suffering head injuries [8], [5], [9].

The proposed strategy modifies the crash pulse with the help of a controller such that the car gets safer during frontal impacts. The output of the controller is added to the original crash pulse and so an improved, i.e. safer crash pulse results. The so generated optimized crash forms the target pulse the car designer aims for to construct a safe vehicle by improving the structure of the car in the early design phase where it is still possible to implement structural changes at relatively low cost.

Due to high costs full scale crash tests are often substituted by sled tests. This is basically a full vehicle cockpit mounted on an acceleration sled, see Fig. 3, including the occupant represented by a crash test dummy on the driver seat. In one such test the sled with the occupant is accelerated to a desired speed and is decelerated according to the crash pulse. This procedure emulates the real crash test with the advantage that no car gets crashed. That allows layout and fine-tuning of restraint systems to be performed faster and more cost effective. Such a substitute system forms the basis for this work.

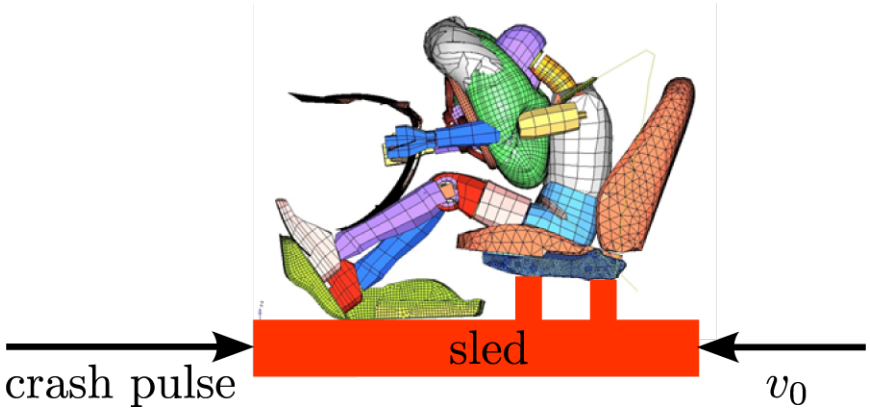

Figure 3: Sled test configuration as a substitute for the full scale crash test

For verification purposes the sled test is implemented as a Finite-Element-Model (FE-Model). This FE-Model includes the sled, the cockpit, the seat, the steering wheel with the airbag, the safety belt and the FE dummy itself. The FE-Model was validated on a full scale vehicle and is able to describe the dynamics and effects of the dummy during the crash test accurately. This FE-Model consists of about 200000 deformable finite elements and requires the crash pulse as an input. The FE-Dummy Model generates mechanical injury data, amongst others the head acceleration which is necessary to calculate the mentioned HIC. So the effects of the modified crash pulse can be analyzed [8], [1].

In Fig. 4 the crash pulse design procedure is shown. In the initial situation the original crash pulse is given which has to be improved. After the modification with the help of the controller, the generated optimized crash pulse is validated using the FE-Model before it is implemented in the vehicle.

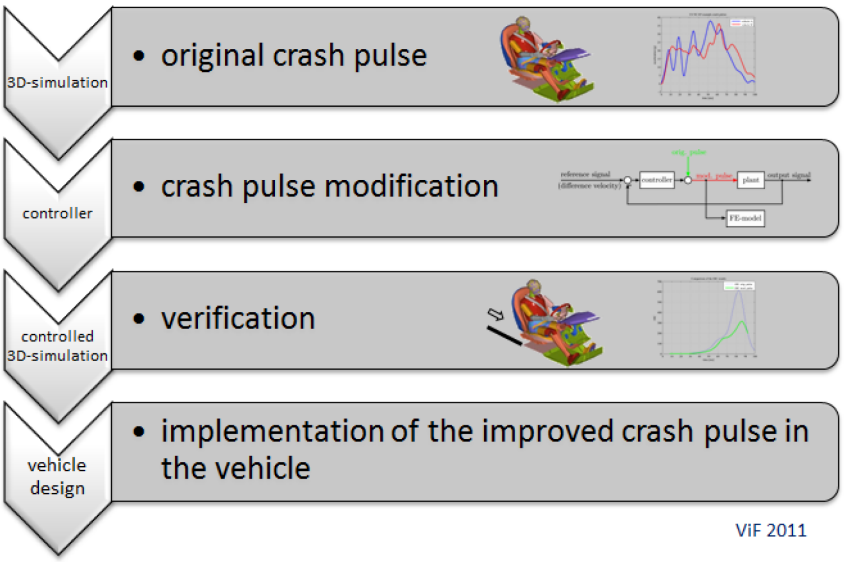

Figure 4: crash pulse design procedure

2 Modelling

2.1 Motivation

The aim of the controller is to generate a modified crash pulse which makes the car safer during frontal impacts. This modified crash pulse must be realizable by the car designers in order to make the car safer. So if the controller manages to generate an implementable, modified crash pulse out of the original one, complex restraint systems can be omitted to achieve a desired safety standard. To make use of all the advantages of the systematic controller design a mathematical model of the frontal crash is required. The FE-model is not suitable for the controller design as no mathematical description of the crash can be derived. The mathematical model for the controller design should collect the essential dynamics of the occupant during the frontal crash.

2.2 Mathematical model

The design model consists of three masses, where the first mass describes the sled, the second the body of the crash dummy and the last the head of the dummy[1]. These three mass elements are coupled via spring/damper elements as shown in Fig. 5. The

[1]the three masses are modeled as particles so the rectangular drawing of the masses in Fig. 5 is only for better graphical representation

first spring/damper element (characterized by the spring constant k_1 and the damper constant β_1) between the sled and the body describes the safety belt and the motion of the dummy along the seat. The second spring/damper element (spring constant k_2 and damper constant β_2), located between the head and the body, models the damped rotary motion of the head during the frontal impact. The crash pulse which is the input to the plant acts directly on the sled mass. The initial velocity (v_0) of the sled is set to 56km/h.

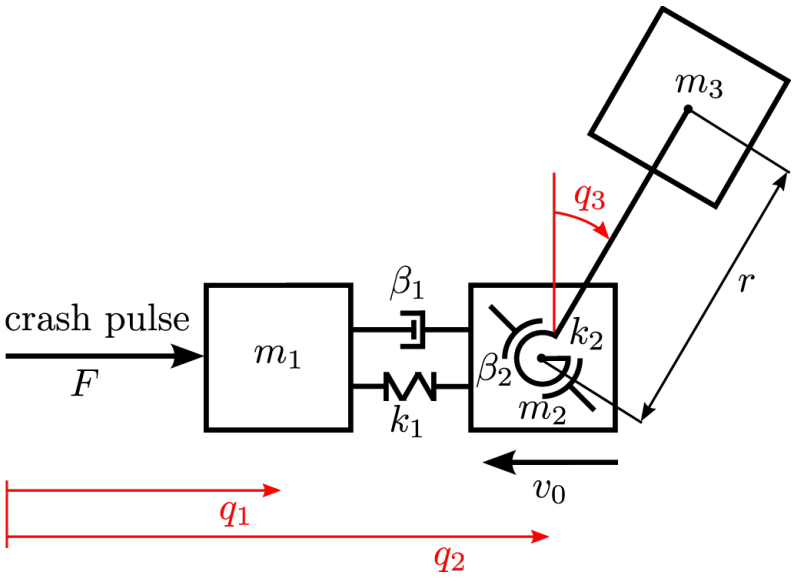

Figure 5: structure of the physical model with generalized coordinates q_1, q_2, q_3

This physical multi body model therefore is characterized by 8 parameters:

- sled mass m_1, mass of the dummy without the head m_2 and the mass of the dummy head m_3
- spring constants k_1 and k_2
- damper constants β_1 and β_2
- the distance r, between the mass m_2 and the mass m_3

The values of the masses m_2 and m_3 are calculated from the datasheet of the frontal impact Hybrid III 50th Percentile Male dummy of the company DENTON ATD, INC [4]. The mass of the sled m_1 is assumed to be very high compared to the dummy masses m_2 and m_3, e.g. $m_1 = 100000kg$. The distance r is chosen to be 16cm. So only the two damper- and the two spring-constants are unknown.

To capture the dynamics of this multi body system the equations of motion are determined via the formalism of Lagrange [7], i.e.:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) + \frac{\partial L}{\partial q_j} = Q_j - \frac{\partial P}{\partial \dot{q}_j} \qquad \text{for} \quad j = 1, ..., n \tag{1}$$

In these equations L denotes the Lagrange function, P is the dissipation function, q_j are the generalized coordinates, Q_j represent the generalized forces and n is the number of

generalized coordinates.

The following generalized coordinates are chosen (see Fig. 5):

- Position of the centre of gravity of the sled mass q_1
- Position of the centre of gravity of the dummy mass without the head q_2
- Angle between the body and the head of the dummy q_3

Due to the circular motion of the head, the distance r between the head and the body can be regarded as constant.

The position vectors of the three masses $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}$ can be computed as:

$$\mathbf{r_1} = \begin{pmatrix} q_1 \\ 0 \end{pmatrix}, \qquad \mathbf{r_2} = \begin{pmatrix} q_2 \\ 0 \end{pmatrix}, \qquad \mathbf{r_3} = \begin{pmatrix} q_2 + r\sin q_3 \\ r\cos q_3 \end{pmatrix}. \tag{2}$$

The kinetic energy T is given as:

$$T = \frac{1}{2} m_1 \dot{\mathbf{r}}_\mathbf{1}^2 + \frac{1}{2} m_2 \dot{\mathbf{r}}_\mathbf{2}^2 + \frac{1}{2} m_3 \dot{\mathbf{r}}_\mathbf{3}^2. \tag{3}$$

The potential function V reads as:

$$V = \frac{1}{2} k_1 \left(q_2 - q_1\right)^2 + \frac{1}{2} k_2 q_3^2. \tag{4}$$

With the help of (3) and (4) the Lagrange-function L

$$L = T - V \tag{5}$$

can be computed.

The dissipation function P is given by:

$$P = \frac{1}{2} \beta_1 \left(\dot{q}_2 - \dot{q}_1\right)^2 + \frac{1}{2} \beta_2 \dot{q}_3^2. \tag{6}$$

Finally the generalized forces Q_j are required:

$$Q_1 = F = u, \quad Q_2 = 0, \quad Q_3 = 0. \tag{7}$$

The modeling process leads to a nonlinear system of sixth order

$$\dot{\mathbf{x}} = f\left(\mathbf{x}, u, t\right) \tag{8}$$

where the state vector $\mathbf{x}$ is given as $\mathbf{x} = \begin{bmatrix} q_1 & q_2 & q_3 & \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T =$
$= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T$.

2.3 Output of the plant

The output y of the plant should be related to the HIC. With respect to this request the output is defined as the absolute value of the difference velocity of head and sled, i.e.

$$y = |\dot{\mathbf{r}}_\mathbf{3} - \dot{\mathbf{r}}_\mathbf{1}| = \left| \begin{array}{c} x_5 + r\cos(x_3)\,x_6 - x_4 \\ -r\sin(x_3)\,x_6 \end{array} \right| = \\ = \sqrt{x_5^2 + 2x_5 r\cos(x_3)\,x_6 - 2x_5 x_4 - 2r\cos(x_3)\,x_6 x_4 + x_4^2 + r^2 x_6^2}. \quad (9)$$

The FE-Simulation confirmed the choice of the output as the HIC gets reduced for lowered outputs. This means that a reduced difference velocity y leads to a reduced HIC, (see Fig. 6).

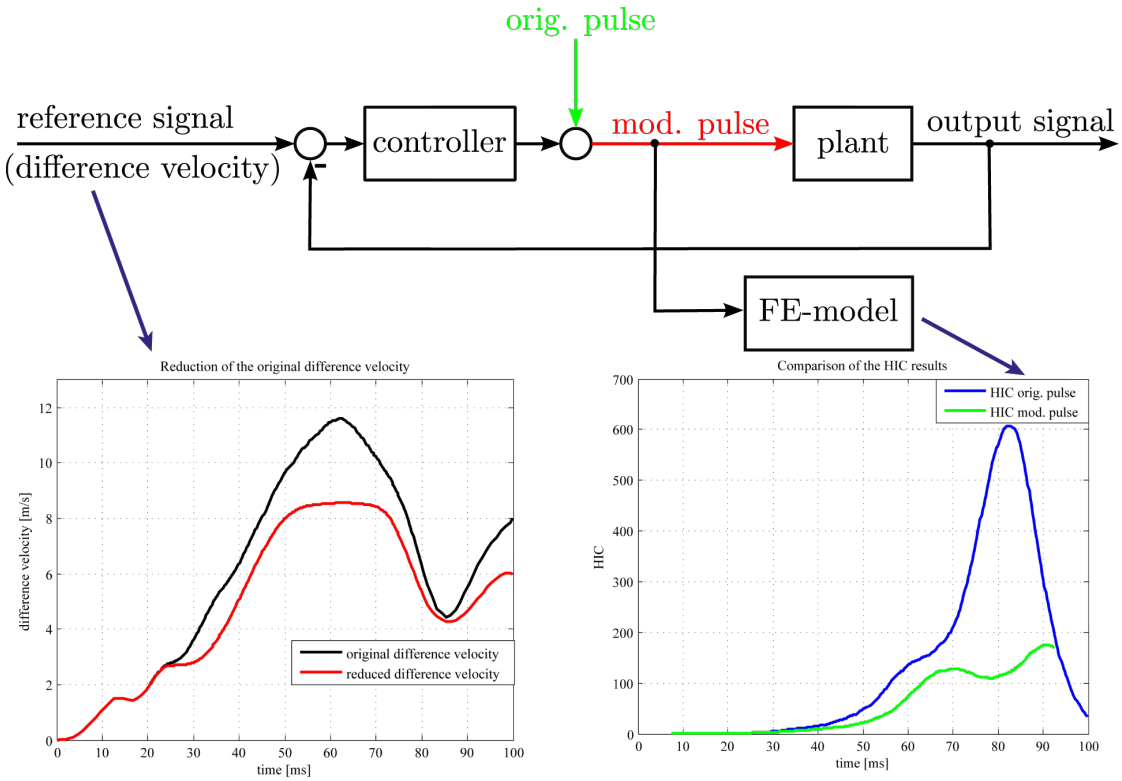

Figure 6: Reduction of the HIC due to a reduced difference velocity

2.4 Parameter identification

The parameters of the plant are identified in that way that the reaction on the original crash-pulse of the plant and the FE-simulation are very close to each other. The parameter identification yields the constants:

$$k_1 = 135000\frac{N}{m}, \quad k_2 = 300\frac{N}{m}, \quad \beta_1 = 140\frac{Ns}{m}, \quad \beta_2 = 1\frac{Ns}{m}$$

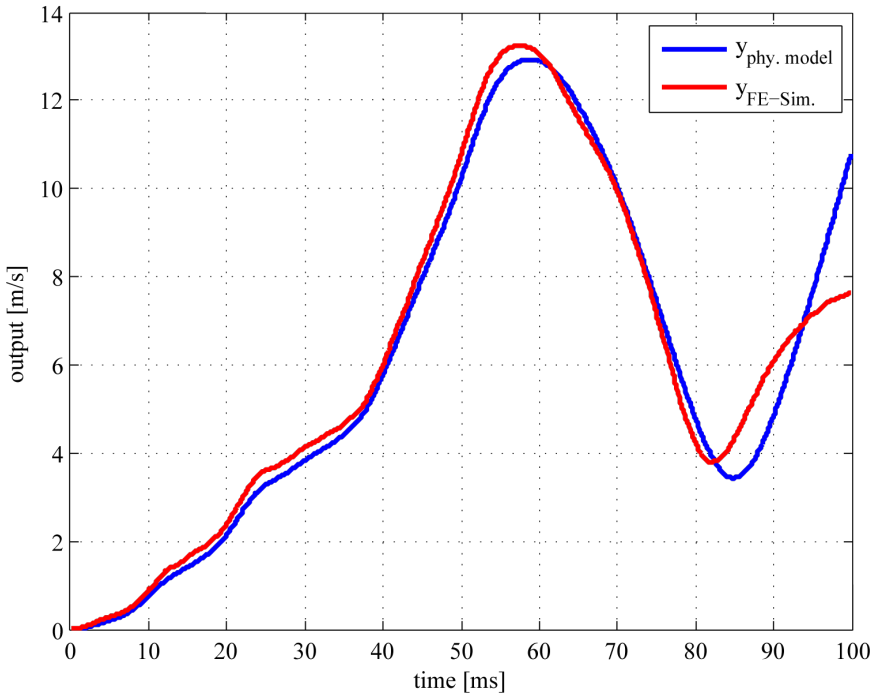

Figure 7: result of the parameter identification

Fig. 7 shows that model (8) is able to describe the behaviour of the FE-model very good in the first 80ms. After $t = 80ms$ the model reacts in a different way because this simple model is not able to describe the complex rebound procedure of the dummy. This model-mismatch is of minor interest because after 80ms a pulse-modification is not reasonable as the dummy has already crashed into the airbag.

2.5 Validation

In order to validate the mathematical model (8) another original crash pulse from a different car is applied to the model using the same set of parameters.

Fig. 8 shows the corresponding reaction of the model. Obviously the reaction is also very close to that of the FE-model. So it can be concluded that model (8) of the frontal impact is able to reproduce the essential dynamics of the crash process and therefore is not linked to one special crash pulse.

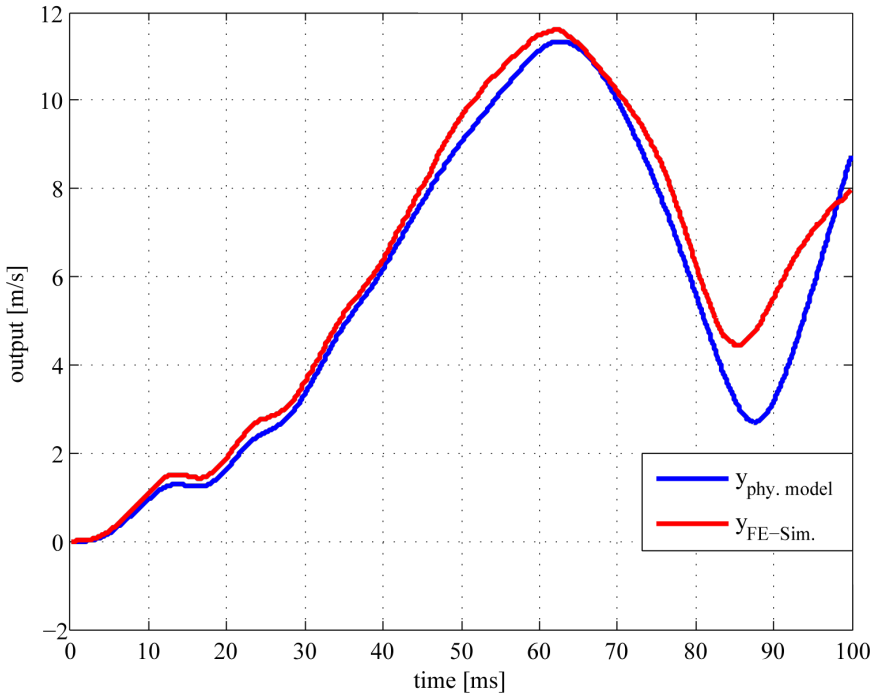

Figure 8: model validation

3 Control strategy

3.1 Structure of the closed loop system

The structure of the closed loop system is illustrated in Fig. 9. The output of the controller is added to the original crash pulse to generate the desired modified crash pulse. This modified crash pulse is the input of the nonlinear plant. The aim of the controller is to make the output of the nonlinear plant y track the reference signal r with sufficient accuracy.

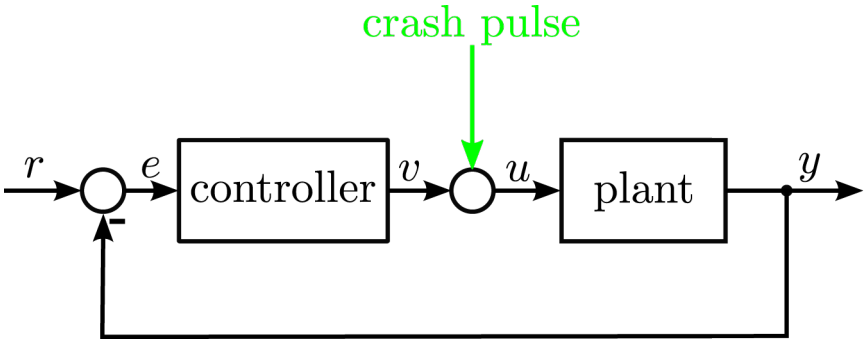

Figure 9: closed loop system

3.2 Input-output linearisation

To control this nonlinear plant of sixth order the concept of input-output linearization is used. With this method the nonlinearity of the plant gets compensated such that a linear input-output behaviour results. This compensated linear plant is equivalent to an integrator chain, where the number of integrators is defined by the relative degree δ of the nonlinear plant. This integrator chain can be controlled with the help of linear control design techniques. The relative degree is defined as the number of times one has to differentiate the output y with respect to time in order to have the input u explicitly appearing. When the relative degree is less than the system order n an internal dynamic with the order $n - \delta$ exists. The state variables of the internal dynamic are rendered unobserveable by the nonlinear compensation. The internal dynamic doesn't influence the compensated linear dynamic but is important for the stability of the overall system. The overall system is stable iff the internal dynamics are. If the internal dynamics are not stable the method of input-output linearization is not realizable due to the instability of the overall system [6], [10].

3.2.1 Relative Degree

In the first derivative of the output y with respect to time the input u appears explicitly. So the relative degree δ is 1 in this case:

$$\frac{dy}{dt} = \dot{y} = g\left(\mathbf{x}, u\right) \rightarrow \delta = 1. \tag{10}$$

A relative degree $\delta = 1$ leads to an internal dynamics of order 5 (order of the internal dynamics: $n - \delta$). As discussed before this internal dynamics must show stable behaviour to use the method of input-output linearization. No mathematical stability proof of the internal dynamics or rather the zero-dynamics is performed here due to the complexity of the calculation of the internal dynamics. As a huge number of numerical simulations showed stable behaviour of the internal dynamics, it is assumed to be stable.

3.2.2 Compensation

To determine the compensation, the δ^{th} derivation of the output y with respect to time is set to the new input v:

$$y^{(\delta)} = \dot{y} = g\left(\mathbf{x}, u_1\right) = v. \tag{11}$$

From (11) we get the compensation u_1 as a function of the state vector $\mathbf{x}$ and the new input v:

$$u_1 = h\left(\mathbf{x}, v\right). \tag{12}$$

With this compensation the closed loop system gets modified (see Fig. 10).

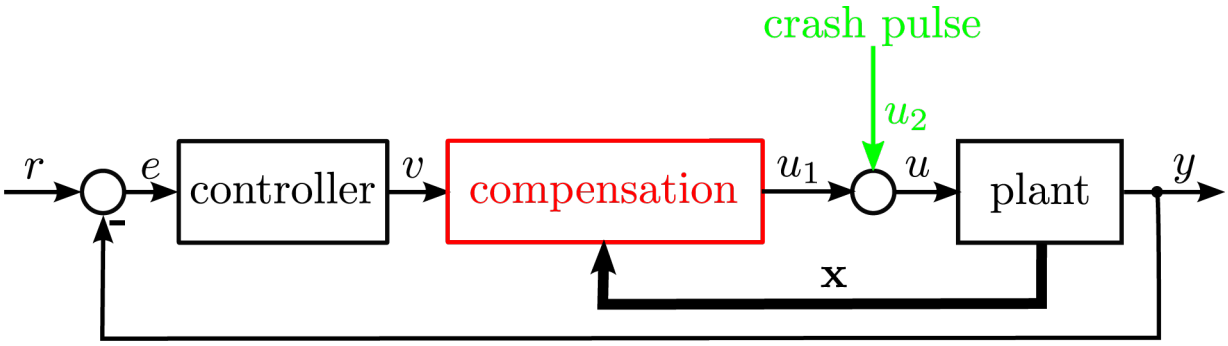

Figure 10: closed loop system with input-output linearization

With the help of the input-output linearization a linear dynamics between the new input v and the output y is achieved. This linear system represents one integrator because of the relative degree $\delta = 1$.

3.3 Controller design for tracking

The controller design can now be done on basis of the linear compensated plant. The unity feedback loop can be interpreted with the integrator as the plant:

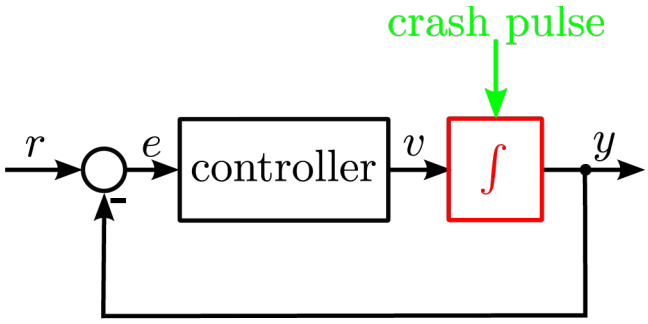

Figure 11: closed loop system for the controller design

The controller should guarantee that the output y tracks the reference signal r in a satisfactory way. To achieve this, a controller based on the error dynamics was designed.

$$e = r - y. \tag{13}$$

The derivative of the tracking error e with respect to time is given by

$$\dot{e} = \frac{d}{dt}(r - y) = \dot{r} - \dot{y}. \tag{14}$$

As discussed before the derivative of the output y with respect to time is equal to the new input v. So the new input v can be expressed as:

$$v = \dot{r} - \dot{e}. \tag{15}$$

In the next step the error dynamics is defined:

$$\dot{e} + \alpha e = 0. \tag{16}$$

This error dynamics leads to the following characteristic polynomial for the error:

$$\Delta_e(s) = s + \alpha, \quad \alpha > 0 \tag{17}$$

Relations (13), (15) and (16) lead to the control-law:

$$v = \dot{r} + \alpha r - \alpha y. \tag{18}$$

This controller ensures that the tracking error converges to zero exponentially.

3.3.1 Controller deactivation

When no pulse modification is meaningful the controller gets deactivated. This is the case if

- the simulation time $t \geq 80ms$ because of the model mismatch,
- the energy of the modified pulse is equal to the energy of the original pulse,
- the reference signal r is the original difference velocity itself.

3.4 Choice of the reference signal

The reference signal r must be chosen such that the HIC gets reduced. To achieve this, the original difference velocity as a reaction of the original crash pulse is reduced. The reduction of the original difference velocity can be done by bounding or by new definition.

4 Simulation

The model for the simulation includes the closed loop system itself and also the coupling with the FE-simulation [3]. This coupling is needed to analyse the effects of the new modified pulse in the FE-Simulation. The FE-simulation is a realistic model of the sled test where all elements are deformable. Also the safety belt and the airbag are exactly modelled. The configuration of the simulation is based on the US-NCAP frontal crash. This means that the car crashes with $v_0 = 56km/h$ into a rigid barrier. A scheme of the model is shown in Fig. 12.

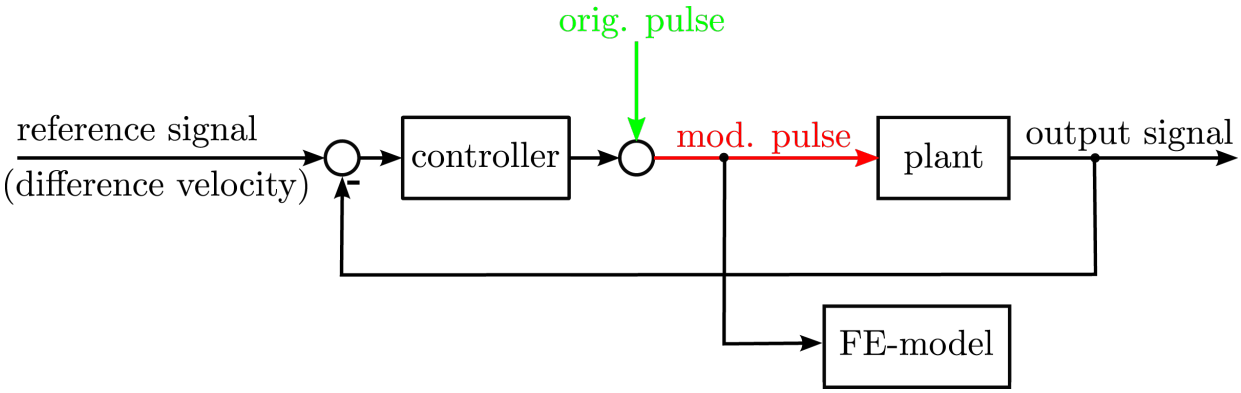

Figure 12: Structure of the closed loop systen with the coupling to the FE-simulation

The reference signal r was chosen such that the original difference velocity gets nearly constant reduced by about $0.5m/s$. The resulting reference signal r (Fig. 13, black curve) was implemented via spline functions. In Fig. 13 all difference velocities are displayed.

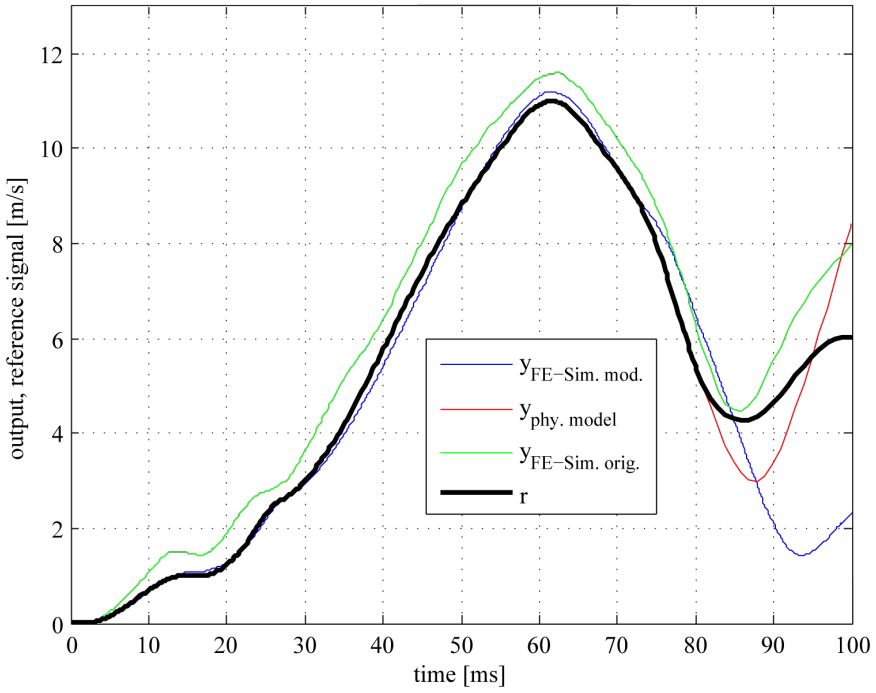

Figure 13: comparison of the different output signals y with the reference signal r

As Fig. 13 illustrates, the modified pulse generates a difference velocity ($y_{FE-Sim.mod}$, blue curve) in the FE-simulation, which tracks the reference signal r satisfyingly until the controller gets deactivated at $t = 80ms$. The output of the physical model ($y_{phy.model}$, red curve) tracks the reference signal r perfectly until the controller gets inactive. The modification of the crash pulse is not highly distinctive due to the slight modification of the difference velocity. The original difference velocity was mainly changed in the amplitude and not in shape. In Fig. 14 the original and the modified crash pulse are compared:

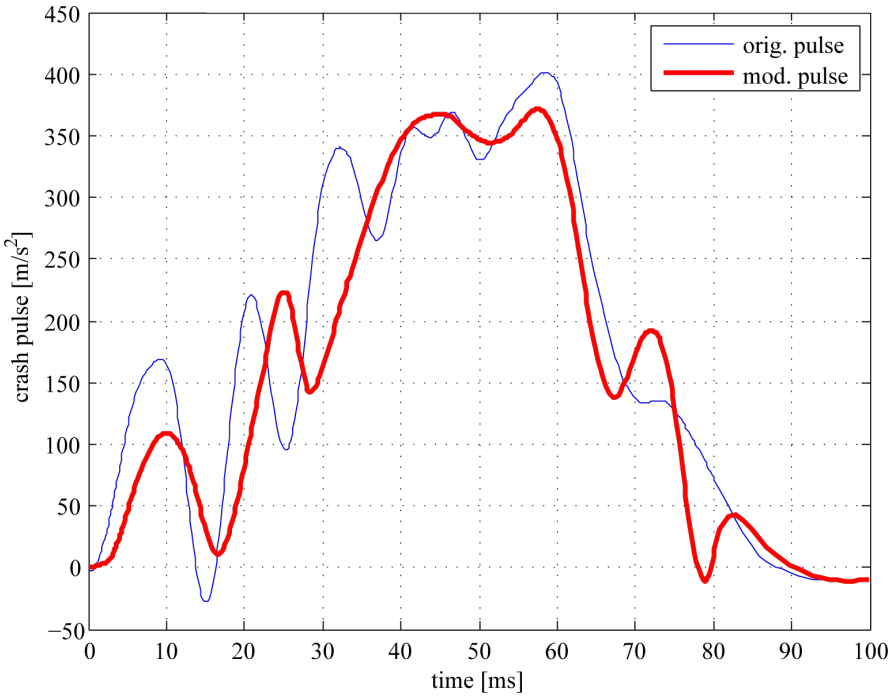

Figure 14: Comparison of the crash pulses

At the beginning the modified crash pulse (red curve) shows a time shift compared to the original one. From $t = 80ms$ the controller is deactivated and the modified pulse tends to the original one. The effect of the modified pulse on the HIC is shown in Fig. 15.

As we can conclude from Fig. 15 the maximum value of the HIC in this case gets reduced by nearly 50 percent although the difference velocity wasn't reduced that much. This surprising result can be explained by the constant reduction of the original difference velocity. The generated reference signal leads to a modified crash pulse whose characteristic hasn't changed significantly. The important result is that this slight modification of the crash pulse causes a reduction of the head injury probability of nearly 50 percent! The modified pulse effects that the crash energy experiences a time delay. Another interesting effect is that the whole HIC-curve (see Fig. 15) shows a time delay. Therefore the maximum value also occurs later than in the original situation.

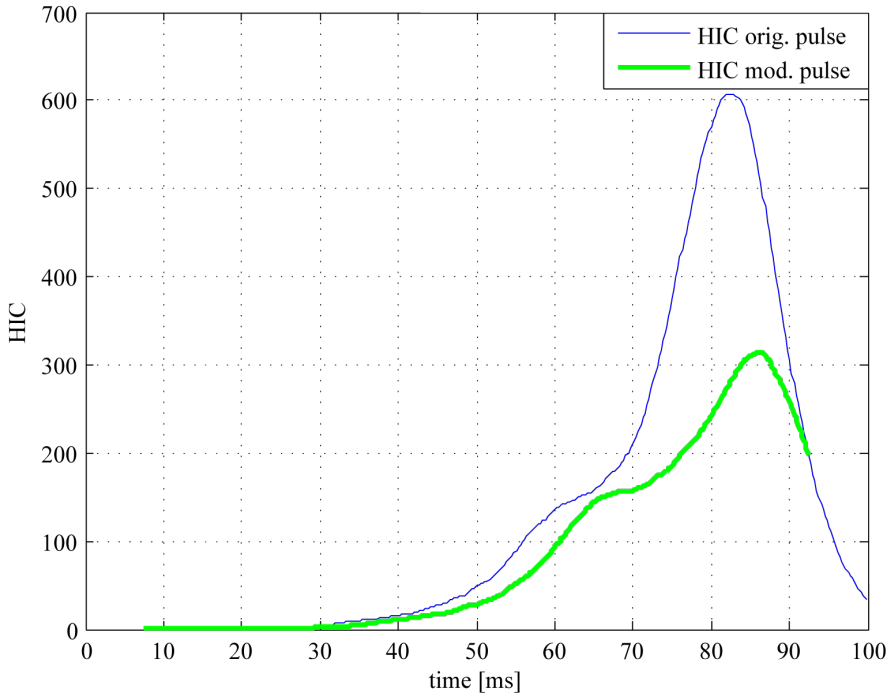

Figure 15: Effect of the modified crash pulse on the HIC

5 Conclusion and Outlook

In the future the goal should be a crash pulse design with respect to all dummy injury values. This is important due to the fact that every car is different and so has different failings in different dummy injury values which have to be improved. For research aspects it will be very interesting if a crash pulse designed with respect to the HIC has similarities with a crash pulse designed with respect to another dummy injury value. This analysis will help to understand the dynamics of a frontal impact. Another fact is the assembly of two or more modified crash pulses. This always occurs when a car has failings in two or more dummy injury values. Adapting a car design to a better crash pulse will be the key challenge, but has the potential to save costs for advanced restraint systems.

Acknowledgements

The authors would like to acknowledge the financial support of the "COMET K2 - Competence Centres for Excellent Technologies Programme" of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), the Austrian Federal Ministry of Economy, Family and Youth (BMWFJ), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG).

We would furthermore like to express our thanks to our supporting industrial and scientific project partners, namely "Magna Steyr Fahrzeugtechnik" and to the Alpen-Adria-University Klagenfurt.

References

[1] Carhs.gmbh: *Safety Companion 2010*

[2] Carhs.gmbh: *Safety Wissen: NCAP Tests* <http://www.carhs.de/de/training/safetywissen/_img/ncap_tabelle_1.gif>

[3] Cresnik R., Rieser A., Schluder H.: *Dynamic Simulation of Mechatronic Systems*, 7th European LS-DYNA Conference, 2009

[4] DENTON ATD, INC: *Datenblatt des Hybrid III 50th Percentile Dummy* <http://www.dentonatd.com/dentonatd/pdf/HIII50M.PDF>

[5] Henn, H. W.: *Crash Tests and the Head Injury Criterion*, Teaching Mathematics and its Applications, Volume 17, No. 4, 1998

[6] Isidori, A.: *Nonlinear Control Systems - 3rd Edition*, Springer-Verlag, London 1995

[7] Janschek, K.: *Systementwurf mechatronischer Systeme: Methoden - Modelle - Konzepte,* Springer-Verlag, Berlin Heidelberg 2010

[8] Kramer, F.: *Passive Sicherheit von Kraftfahrzeugen,* 3. Auflage Vieweg + Teubner — GWV Fachverlage GmbH, Wiesbaden 2009

[9] McHenry B. G.: *Head Injury Criterion and the ATB*, ATB Users' Group, McHenry Software, Inc. 2004

[10] Vidyasagar, M.: *Nonlinear Systems Analysis - 2nd Edition*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993

Optimal Efficiency for Linear Control Systems, a Design Approach Based on Generalized Eigenvalues

Alexander Weinmann, OVE, Senior Member IEEE
Vienna University of Technology, Institute of Automation and Control
Gusshausstrasse 27-29/376, A-1040 Vienna / Austria
Phone: +43 1 58801 37611, Fax: +43 1 58801 37699
email: weinmann@acin.tuwien.ac.at

Manuscript received June 19, 2012

Abstract

A per-unit measure, termed efficiency, is reduced in order to optimize the dynamic quality of a control system. The design is focussed on the maximum closed-loop eigenvalue (closest to the imaginary axis or to the unit circle). The remaining eigenvalues, which are more to the left or in the interior, are not taken into consideration. Another issue is to analyze the opportunities to change the controller coefficient matrix in order to maximize the most critical eigenvalue.

A gradient algorithm is presented to design the controller with respect to the optimal efficiency. Output quality and actuating effort can be easily balanced to each other.

Keywords: Rayleigh factor, per-unit performance weighting, controller gain gradient, state-variable-independent general dynamic state

1 Introduction

The Lyapunov function, usually termed V, is an artificial measure of the general state energy of a dynamic system. Engineering aspects often influence the coefficients of the

weighting matrices generating the Lyapunov function. Of course, the real physical energy can also be used as a Lyapunov function.

Irrespective of the arbitrary weights in the appropriate matrices, a per-unit weighting of the derivative with respect to time, i.e., $\dot{V}$ with respect to V, seems advantageous; effort should be put on providing a maximum decrease of $\dot{V}$ in relation to V.

If the state variable $\mathbf{x}$ is the reaction to some forcing input or the response of some initial condition, the relation $r \stackrel{\triangle}{=} \dot{V}/V$ is not influenced (see, e.g., Fig. 6) and provides a good assessment of the internal dynamics of the system as good as the range of generalized eigenvalues of $\mathbf{P}$ in the Lyapunov function V in Eq.(1).

Evaluating optimal systems by following some square index of performance for $\mathbf{x}$, $\mathbf{u}$ with the help of Riccati equation, the result is essentially influenced by the weighting matrices. Opposite to this, the efficiciency-based design is a setting closer to some natural assumption.

We utilize the common notation for control system matrices and signals $\mathbf{x} \in \mathcal{R}^n$; $\mathbf{y} \in \mathcal{R}^r$; $\mathbf{u} \in \mathcal{R}^m$; $\mathbf{A}, \mathbf{\Phi} \in \mathcal{R}^{n \times n}$; $\mathbf{B}$, $\mathbf{\Psi} \in \mathcal{R}^{n \times m}$; $\mathbf{C} \in \mathcal{R}^{r \times n}$; $\mathbf{K} \in \mathcal{R}^{m \times r}$.

Reviewing some Lyapunov stability properties for linear systems, suppose an autonomous system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ and $\mathbf{P} = \mathbf{P}^T > 0$, and define a Lyapunov function

$$V = V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x} \ . \tag{1}$$

Then, the derivative with respect to time is

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T (\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A}) \mathbf{x} \stackrel{\triangle}{=} -\mathbf{x}^T \mathbf{Q} \mathbf{x} \tag{2}$$

with the algebraic Lyapunov equation

$$\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} = \mathbf{0} \ . \tag{3}$$

Assuming a positive (generalized) energy V, one is interested in a decrease of V, i.e., in a negative $\dot{V}$ with respect to time, in order to guarantee stability. This is a well-known fact for decades of automatic control research. We follow the sign definition of $\mathbf{Q}$ in Eq.(3). Then, both $\mathbf{P}$ and $\mathbf{Q}$ being positive definite symmetric is a necessary and sufficient condition for $\mathbf{A}$ beeing stable (*Kalman, R.E., and Bertram, J.E., 1960; Slotine, J.J., and Weiping, Li, 1991*).

Most investigations in automatic control are content with the guarantee of positive definiteness, irrespective of the norm amount of $\mathbf{P}$ and $\mathbf{Q}$.

2 Lyapunov-Based Efficiency for Continuous-Time Systems

We define a per-unit measure of $\dot{V}$ by means of the relation $\dot{V}/V$. This measure is denoted efficiency r when there was only a single solution

$$r \triangleq \frac{\dot{V}}{V} = -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} \ . \tag{4}$$

The efficiency should be as low as possible in the space of negative numbers, where $\mathbf{P}$ and $\mathbf{Q}$ are positive definite matrices for stability reason in Eq.(3) . Using MATLAB, $\mathbf{P} = \texttt{lyap}(\mathbf{A}^T, \mathbf{Q})$.

If there are many solutions referring to the order n of the system, the maximum within the negative numbers is the most critical one, i.e., the worst efficiency, since it is closest to the stability boundary. Owing to high dynamic quality, $\dot{V}$ should be negative with big modulus; large $\mathbf{x}^T\mathbf{P}\mathbf{x}$ is the instantaneous (generalized) energy. The efficiency r is a per-unit factor valuating the energy dynamics versus the system energy. The optimum within the space of $\mathbf{x}$ is given at

$$\frac{\partial r}{\partial \mathbf{x}} = -\frac{2\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} + \frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{(\mathbf{x}^T\mathbf{P}\mathbf{x})^2}2\mathbf{P}\mathbf{x} = \mathbf{0} \tag{5}$$

$$= -\mathbf{Q}\mathbf{x}(\mathbf{x}^T\mathbf{P}\mathbf{x}) + [\mathbf{P}\mathbf{x}](\mathbf{x}^T\mathbf{Q}\mathbf{x}) = \mathbf{0} \ . \tag{6}$$

Inserting $\mathbf{x} = \mathbf{p}$, where $\mathbf{p}$ is one of the generalized eigenvectors, resulting from the relation $\mathbf{P}\mathbf{p}_i = \lambda_i\mathbf{H}\mathbf{p}_i$, and $\mathbf{H} = \mathbf{Q}$,

$$\frac{\partial r}{\partial \mathbf{p}} = -\mathbf{Q}\mathbf{p}(\mathbf{p}^T\mathbf{P}\mathbf{p}) + [\mathbf{P}\mathbf{p}]\mathbf{p}^T\mathbf{Q}\mathbf{p} \stackrel{?}{=} \mathbf{0} \quad \Big| \ \times\lambda \tag{7}$$

$$\frac{\partial r}{\partial \mathbf{p}} = -(\lambda\mathbf{Q}\mathbf{p} - \mathbf{P}\mathbf{p})\mathbf{p}^T\mathbf{P}\mathbf{p} = \mathbf{0} \ . \tag{8}$$

The generalized eigenvector is the optimal solution. Then, the efficiency factor results as the most critical solution of

$$-\frac{\mathbf{p}^T\mathbf{Q}\mathbf{p}}{\mathbf{p}^T\mathbf{P}\mathbf{p}} = -\frac{1}{\lambda_Q[\mathbf{P}]} \ . \tag{9}$$

For each eigenvalue i and λ_i we get an eigenvector entailing extreme values of r. During any motion $\mathbf{x}(t) = \exp(\mathbf{A}\,t)$, an area of $r(\mathbf{x})$ is passed through when t varies. Maximum r is worst, minimum r is best in dynamics, the instantaneous efficiency is located in between. The worst r should be reduced during the design process. The chain of gradients is a follows: Reducing $\max(r)$ within negative numbers corresponds to increasing $1/\max(r)$

or decreasing $-1/\max(r)$. Hence, $\max_i \lambda_{Qi}[\mathbf{P}]$ should be reduced, reduced via change of $\mathbf{K}$ (or $\mathbf{Q}$). Then and finally,

$$r \triangleq \max\ r_i = \max_i\{\frac{-1}{\lambda_{Qi}[\mathbf{P}]}\} = -\min_i\{\frac{1}{\lambda_{Qi}[\mathbf{P}]}\} = -1/\max_i \lambda_{Qi}[\mathbf{P}]\ . \tag{10}$$

Among the negative values, the maximum is the most critical one. The eigenvalue with smallest modulus yields the worst efficiency factor. The reduction of that one closest to the stability boundary $\lambda_i = 0$ is the design target.

Remark: Choosing V the output-based energy, i.e., $V = \mathbf{y}^T\mathbf{P}\mathbf{y} = \mathbf{x}^T\mathbf{C}^T\mathbf{P}\mathbf{C}\mathbf{A}\mathbf{x}$, then

$$\dot{V} = \mathbf{x}^T(\mathbf{A}^T\mathbf{C}^T\mathbf{P}\mathbf{C} + \mathbf{C}^T\mathbf{P}\mathbf{C})\mathbf{x} \triangleq -\mathbf{x}^T\mathbf{Q}\mathbf{x}\ . \tag{11}$$

Replacing $\mathbf{P}$ by $\mathbf{C}^T\mathbf{P}\mathbf{C}$, the results remain unchanged

$$r = \frac{\dot{V}}{V} = \frac{\mathbf{x}^T(\mathbf{A}^T\mathbf{C}^T\mathbf{P}\mathbf{C} + \mathbf{C}^T\mathbf{P}\mathbf{C})\mathbf{x}}{\mathbf{x}^T\mathbf{C}^T\mathbf{P}\mathbf{C}\mathbf{A}\mathbf{x}}\ . \tag{12}$$

As an alternative, selecting $V = \mathbf{u}^T\mathbf{R}\mathbf{u}$, the replacement $\mathbf{K}^T\mathbf{P}\mathbf{K}$ instead of $\mathbf{P}$ had to be introduced.

Example 1. Second-Order System: Consider $\mathbf{A} = \begin{pmatrix} -2 & 1 \\ 0 & -4 \end{pmatrix}$, $\mathbf{Q} = \mathbf{I}_2$, $\mathbf{b} = (0\ \ 1)^T$, $\mathbf{c} = (0\ \ 1)$. For the second-order system, a graphical display is given for illustration. In the regular eigenvalue case, $\mathbf{x} \equiv \mathbf{k}_s \triangleq (k_{s1}\ \ k_{s2})^T \triangleq R(\cos\varphi\ \ \sin\varphi)^T$, we get r independent of R

$$r = \frac{1}{(\cos\varphi\ \ \sin\varphi)\mathbf{P}(\cos\varphi\ \ \sin\varphi)^T} = \frac{1}{P_{11}\cos^2\varphi + P_{12}\sin(2\varphi) + P_{22}\sin^2\varphi}\ , \tag{13}$$

and the optimum $\varphi = 0.5\ \mathrm{atan}[2\ P_{12}/(P_{11} - P_{22})]$.

The entire function r versus the $\mathbf{k}_s$-plane is given in Fig. 1.

The efficiency r is located in the region given by the Rayleigh condition

$$-\frac{\lambda_{\max}[\mathbf{Q}]}{\lambda_{\min}[\mathbf{P}]}\ \le\ \frac{\dot{V}(\mathbf{x})}{V(\mathbf{x})} = -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} = -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}/\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}/\mathbf{x}^T\mathbf{x}}\ \le -\frac{\lambda_{\min}[\mathbf{Q}]}{\lambda_{\max}[\mathbf{P}]}\ . \tag{14}$$

The search process for the extrema in the $\mathbf{k}_s$-plane is illustrated in Fig. 2. We find $-\lambda_{\max}[\mathbf{Q}]/\lambda_{\min}[\mathbf{P}] = -8.20$ and $-\lambda_{\min}[\mathbf{Q}]/\lambda_{\max}[\mathbf{P}] = -3.79$. The heaps of solution points o and $*$, lowest and highest, coincide with the direction given by the eigenvectors. (Due to Eq.(3), for symmetric $\mathbf{Q}$ also $\mathbf{P}$ is symmetric.) The upper and the lower bound of r are constant, and independent of the amount of $\mathbf{k}_s$.

Numerator and denominator of r are separately displayed in Fig. 3. The states and the accompanying energy expressions are plotted in Fig. 4. In Fig. 4 one has $\max r = -1.034; \min r = -10.94$ in correspondence with

$$\mathrm{eig}(\mathbf{Q},\mathbf{P}) = \lambda_P[\mathbf{Q}] = 1./\mathrm{eig}(\mathbf{P},\mathbf{Q}) = 1./\lambda_Q[\mathbf{P}] \ = \ 10.965;\ \ 1.034 \tag{15}$$

$$\text{and}\ \ \mathrm{eig}[\mathbf{P},\mathbf{Q}] \ = \ 0.0912;\ \ 0.9668\ . \tag{16}$$

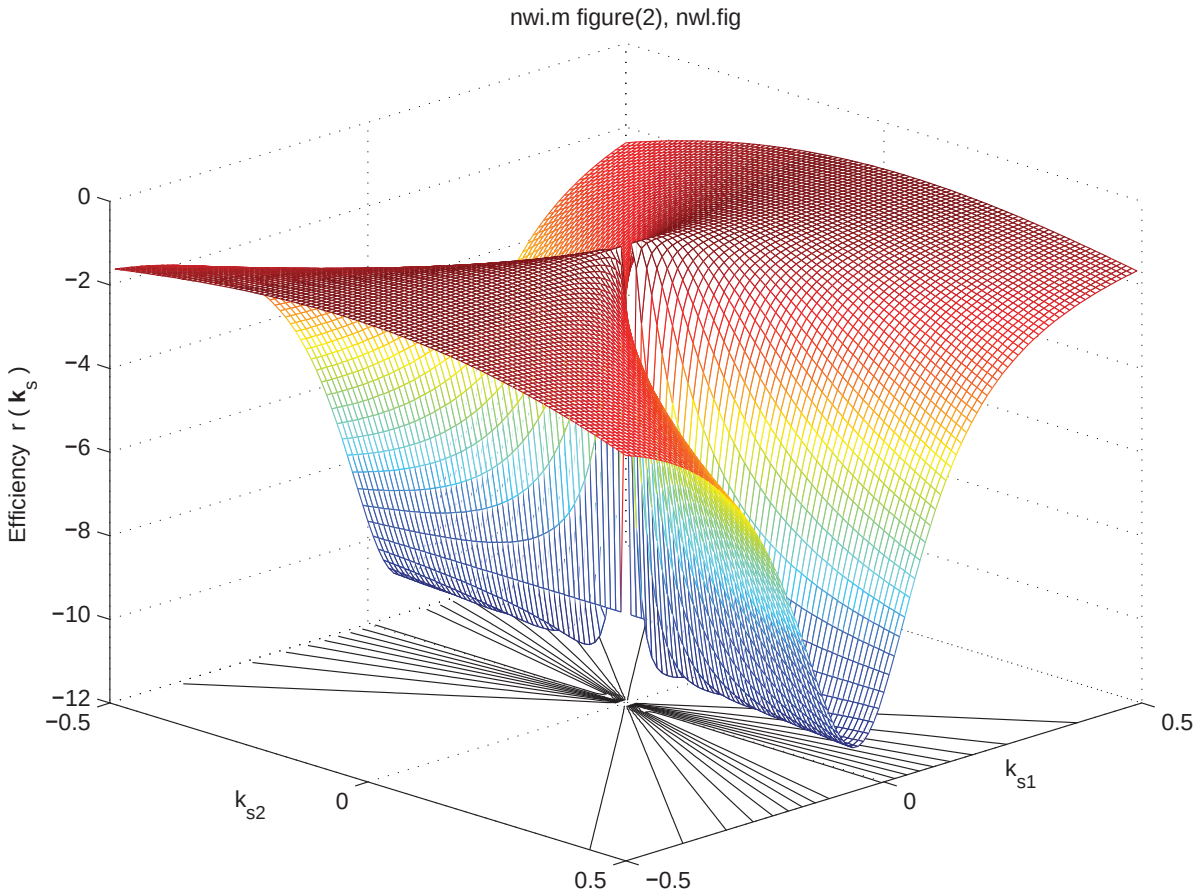

Figure 1: r versus $\mathbf{k}_s$ for Example 1

The motion of the efficiency versus time, $r(t)$, is given in Fig. 5, in addition the classic phase plane plot is depicted dotted at the height of zero. For better emphasizing the dynamics, in Fig. 5 $A(2,1)$ is changed from 0 to -15.

In Fig. 6, the efficiency $r(t)$ is presented for an example with some excitation dying out slowly. The extreme values of r can be registered to a far extent. □

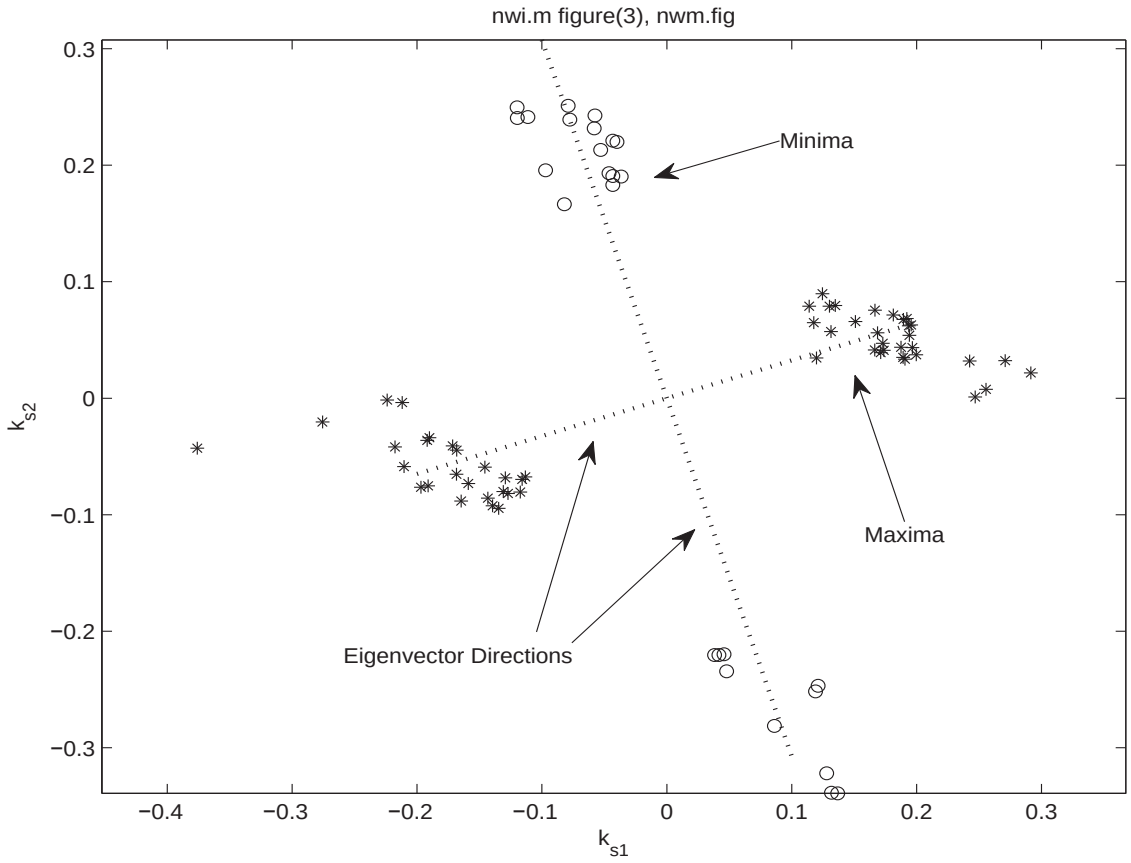

Figure 2: Optimum search $\mathbf{k}_s$ in the $\mathbf{k}_s$-plane for $A(2,1) = 0$ for Example 1

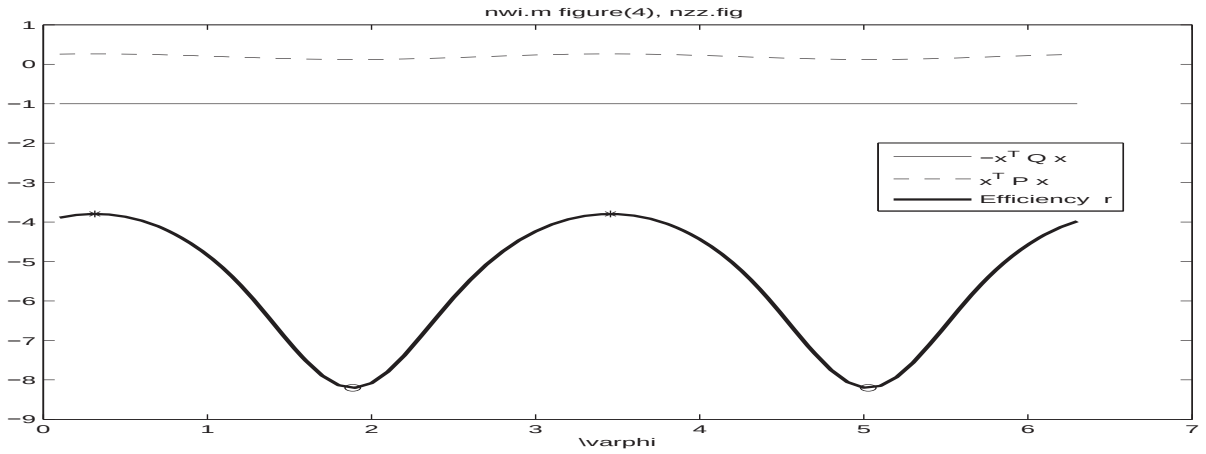

Figure 3: Components of the efficiency r for $A(2,1) = 0$ for Example 1

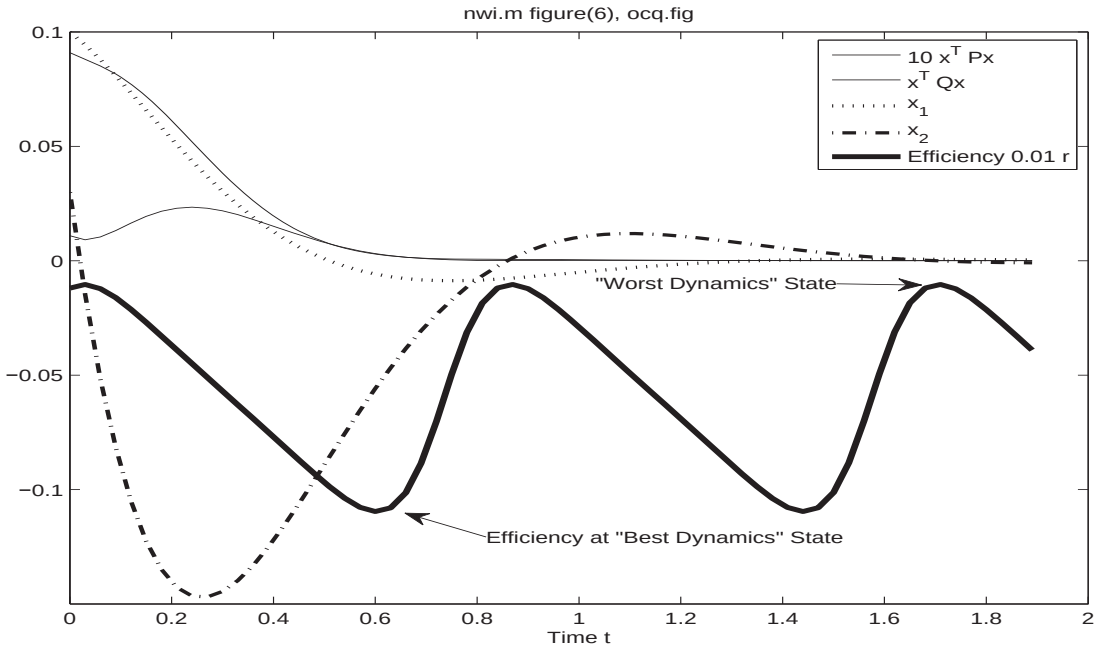

Figure 4: State variables and efficiency r versus time for $A(2,1) = -15$ for Example 1

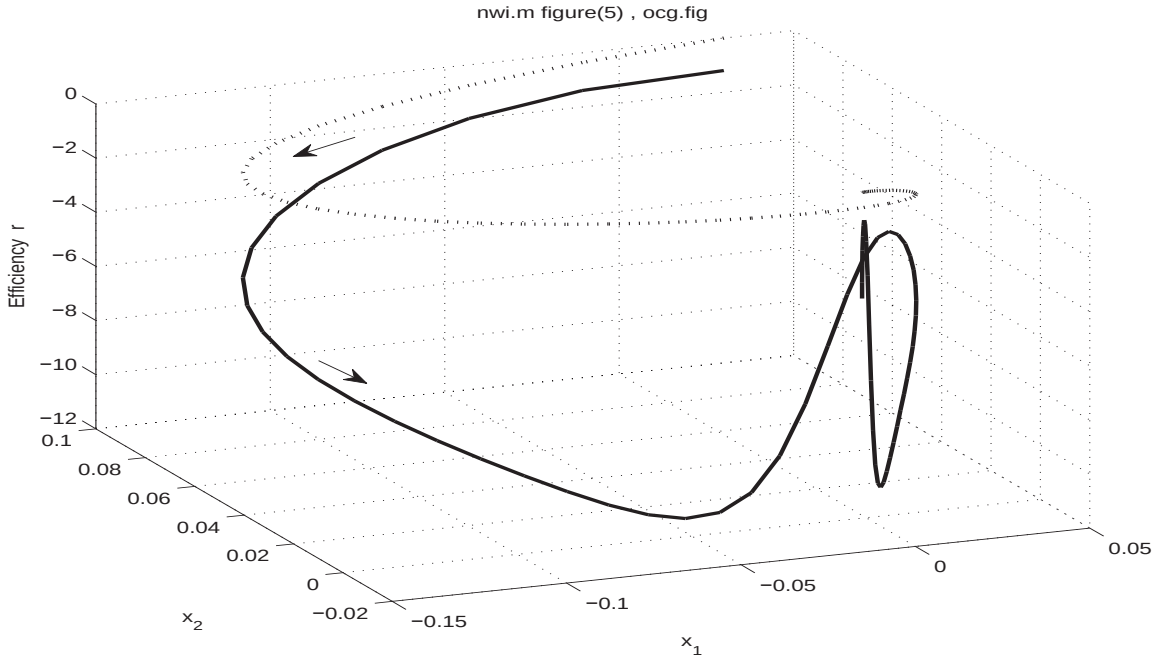

Figure 5: Efficiency plot and classical phase plot for $A(2,1) = -15$ for Example 1

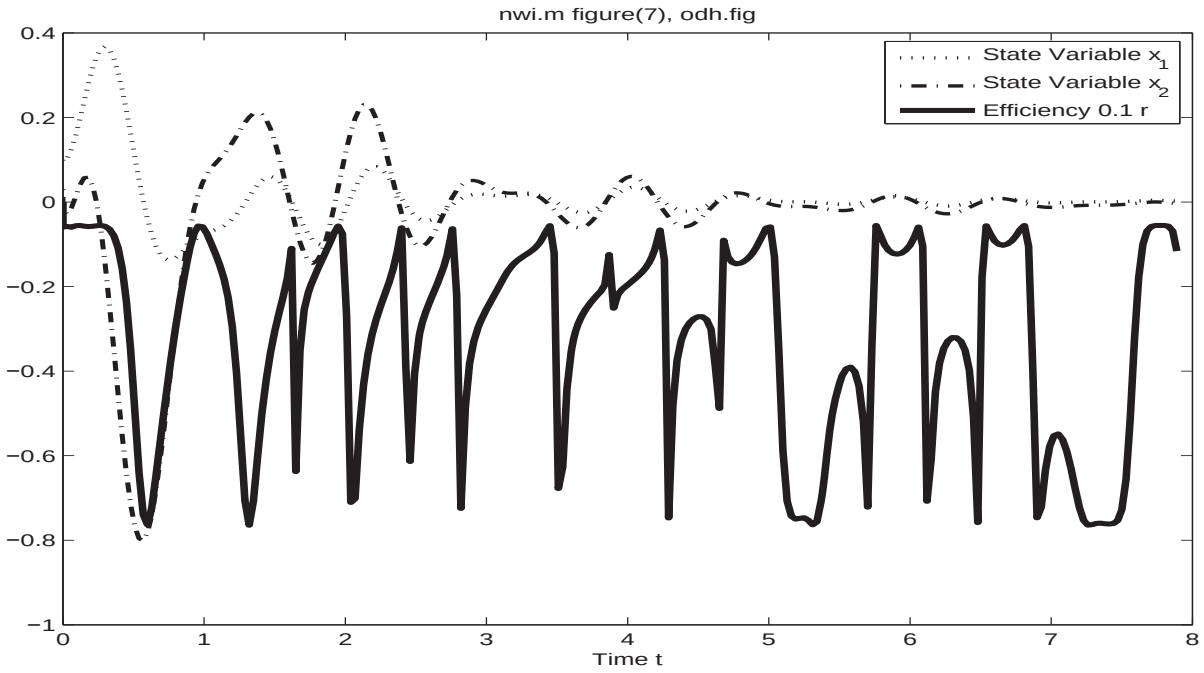

Figure 6: Plot of the efficiency $r(t)$ for a slowly decreasing excitation in Example 1

3 Efficiency Gradients for the Controller Gain

The issue is: How to change the controller gain $\mathbf{K}$ by means of $\Delta\mathbf{K}$ to get better dynamics, i.e., to get less (negative values) r. There are two equivalent derivations:

Method 1: From the setup

$$\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{x}^T\Delta\mathbf{P}\ \mathbf{x}} \triangleq -(r+\Delta r) \tag{17}$$

one finds

$$\Delta r = -r_o(\mathbf{x}^T\otimes\mathbf{x}^T)\mathrm{col}\Delta\mathbf{P}, \quad \text{where} \quad r_o \triangleq -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{(\mathbf{x}^T\mathbf{P}\mathbf{x})^2}\ . \tag{18}$$

From the Lyapunov equation, we find by evaluating with small increments $\mathbf{P}+\Delta\mathbf{P}$ caused by $\mathbf{K}+\Delta\mathbf{K}$

$$(\mathbf{A}+\mathbf{B}\mathbf{K}\mathbf{C})^T\mathbf{P}+\mathbf{P}(\mathbf{A}+\mathbf{B}\mathbf{K}\mathbf{C})+\mathbf{Q}=\mathbf{0} \tag{19}$$

$$\mathrm{col}\Delta\mathbf{P} = -\mathbf{Y}_3^{-1}\mathbf{Y}_4\mathrm{col}\Delta\mathbf{K}\ , \tag{20}$$

where

$$\mathbf{Y}_3 \triangleq \mathbf{I}_n\otimes(\mathbf{A}+\mathbf{B}\mathbf{K}\mathbf{C})^T+(\mathbf{A}+\mathbf{B}\mathbf{K}\mathbf{C})^T\otimes\mathbf{I}_n \tag{21}$$

$$\mathbf{Y}_4 \triangleq [(\mathbf{P}\mathbf{B})\otimes\mathbf{C}^T)\mathbf{U}_{1,n}+\mathbf{C}^T\otimes(\mathbf{P}\mathbf{B})\ . \tag{22}$$

This is the result for an output state controller; for a state controller select $\mathbf{C}=\mathbf{I}$ and adapt the dimensions and replace $\mathbf{Y}_3,\ \mathbf{Y}_4$ by $\mathbf{Y}_1,\ \mathbf{Y}_2$.

Combining Eq.(18) and Eq.(20), we get the gradient $\mathbf{g}$ or its transpose

$$\Delta r = \mathbf{g}_1^T\mathrm{col}\Delta\mathbf{K} \quad \text{where} \quad \mathbf{g}_1^T = r_o(\mathbf{x}^T\otimes\mathbf{x}^T)\mathbf{Y}_3^{-1}\mathbf{Y}_4\ . \tag{23}$$

Preferably, for $\mathbf{x}$ the generalized eigenvector, resulting from MATLAB `eig`$(\mathbf{P},\mathbf{Q})$, is inserted for $\max\lambda_{Q,i}[\mathbf{P}]$, $\lambda>0,\quad r<0$.

Method 2: We need $\frac{\partial\mathbf{P}}{\partial\mathbf{K}}$, but only $\mathrm{col}\Delta\mathbf{P}=-\mathbf{Y}_3^{-1}\mathbf{Y}_4\mathrm{col}\Delta\mathbf{K}$ is available in Eq.(20). Using Eq.(107) and Eq.(108), (*Weinmann, A., 2012*) the vector $c=[\mathbf{I}_{mn}\otimes(-\mathbf{Y}_3^{-1}\mathbf{Y}_4)]\ \bar{\mathbf{U}}_{mn,1}$ is obtained. Afterwards block-decolumnization is applied.

Replace $\mathbf{F}$ by $\mathbf{P}=\mathrm{lyap}[(\mathbf{A}+\mathbf{B}\mathbf{K})^T,\mathbf{Q}]$ in the following equation

$$\frac{\partial\lambda_i[\mathbf{F}]}{\partial\mathbf{K}} = (\mathbf{f}^H\mathbf{Q}\mathbf{f})^{-1}(\mathbf{I}_m\otimes\mathbf{f}_i^{\triangleleft H})\frac{\partial\mathbf{F}}{\partial\mathbf{K}}(\mathbf{I}_n\otimes\mathbf{f}_i)\ . \tag{24}$$

Then, the gradient is achieved as

$$\frac{\partial r}{\partial\mathbf{K}} = \frac{\partial(-\frac{1}{\lambda})}{\partial\mathbf{K}} = \frac{1}{\lambda^2}\frac{\partial\lambda}{\partial\mathbf{K}}\ , \qquad \lambda=\lambda_Q[\mathbf{P}] \tag{25}$$

$$\boxed{\frac{\partial r}{\partial\mathbf{K}^{[1\times n]}} = \frac{1}{\lambda^2}(\mathbf{f}^H\mathbf{Q}\mathbf{f})^{-1}(\mathbf{I}_m\otimes\mathbf{f}_i^{\triangleleft H})\frac{\partial\mathbf{F}}{\partial\mathbf{K}}(\mathbf{I}_n\otimes\mathbf{f}_i)=\mathbf{g}_2^T} \tag{26}$$

and

$$\mathbf{g}_2 = \frac{\partial r}{\partial \mathrm{col}\mathbf{K}} = \mathrm{col}\frac{\partial r}{\partial \mathbf{K}} \quad \rightsquigarrow \quad \mathbf{g}_2^T = \frac{\partial r}{\partial \mathbf{K}^{[1\times n]}} \ . \tag{27}$$

Remember that the controller $\mathbf{K}$ has limited influence on the closed-loop dynamics. Considering, e.g., $\mathbf{A}_{cl} = \mathbf{A} + \mathbf{BKC} \in \mathcal{R}^{3\times 3}$ and $\mathbf{B} = (0 \quad 0 \quad 1)^T$, (i) for $\mathbf{C} = \mathbf{I}$ only the last row of $\mathbf{A}_{cl}$ can be controlled, and (ii) for $\mathbf{C} = (1 \quad 0 \quad 0)^T$ only the element $A_{cl} = A_{3,1}$ can be influenced.

Example 2. Third-Order System. Optimal Controller: For very different initial eigenvalues of $\mathbf{A}$, i.e., $\mathbf{A} = [-0.001 \quad 0 \quad 0; 0 \quad -2 \quad 0; 0 \quad 0 \quad -4]$; $\mathbf{B} = [0.3 \quad 0.2 \quad 1]^T$ and $\mathbf{Q} = \mathrm{diag}([1 \quad 4 \quad 1])$ the controller gain changes from $\mathbf{K} = -[1 \quad 2 \quad 3]$ to $\mathbf{K} = -[6.1234 \quad 1.4943 \quad 1.6038]$ during twenty optimization steps. The result is shown in Fig. 7. □

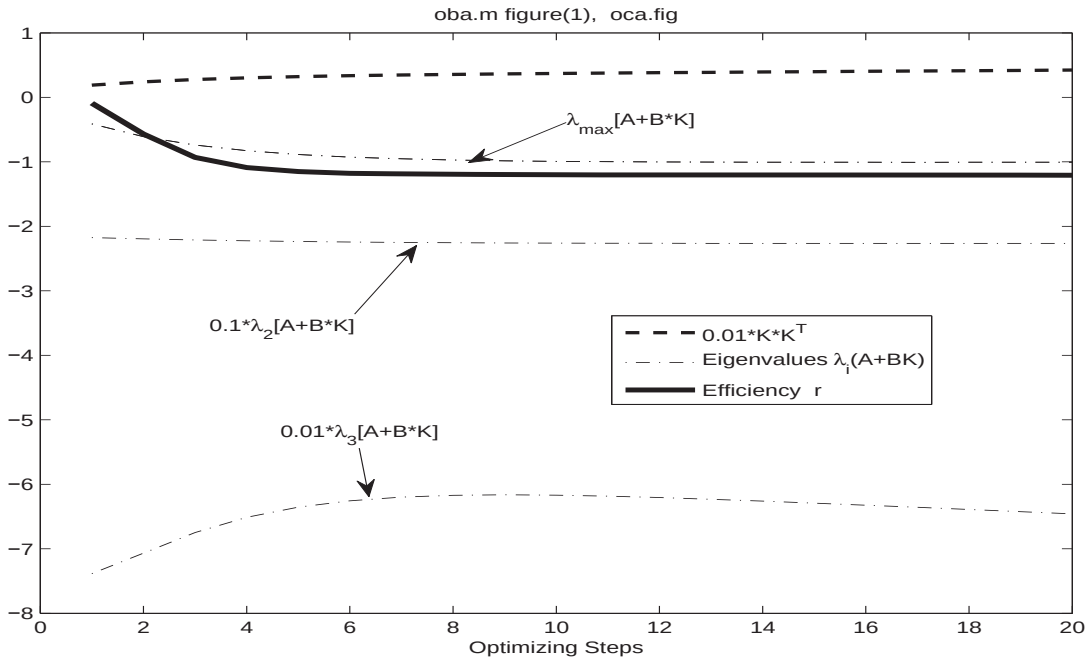

Figure 7: Optimizing the efficiency r in Example 2

4 The Optimum Matrix Q and its Efficiency Gradient

Recall the generalized eigenvalue $\mathtt{eig}(\mathbf{P},\mathbf{Q}) \triangleq \lambda_Q[\mathbf{P}]$. For $\mathbf{Q}=\mathbf{I}$ one has the ordinary eigenvalue $\lambda[\mathbf{P}]$. Inserting $\lambda_P[\mathbf{Q}] = 1/\lambda_Q[\mathbf{P}]$ in

$$\det(\mathbf{P}\ \lambda_P[\mathbf{Q}] - \mathbf{Q}) \;=\; 0 \tag{28}$$

$$\det(\mathbf{P}\ \cdot\ 1/\lambda_Q[\mathbf{P}] - \mathbf{Q}) \;=\; 0 \tag{29}$$

$$\det(\mathbf{P}\ -\mathbf{Q}\cdot\lambda_Q[\mathbf{P}]) \;=\; 0 \tag{30}$$

yields the simple proof. Including MATLAB notation in parts,

$$[\mathbf{V},\mathbf{D}] = \mathtt{eig}(\mathbf{Q},\mathbf{P}) \;\rightsquigarrow\; \lambda_P[\mathbf{Q}] = 1/\lambda_Q[\mathbf{P}],\ \ \det(\mathbf{Q}\cdot\lambda_Q[\mathbf{P}]-\mathbf{P}) = 0,\ \ \mathbf{PV} = \mathbf{QVD}. \tag{31}$$

For which $\mathbf{Q}$ is r of Eq.(9) maximum (within the negative numbers) in order to detect the worst efficiency point? One has to follow

$$r = \max(-\frac{1}{\lambda_Q[\mathbf{P}]}) = \max_i(-\lambda_{P,i}[\mathbf{Q}])\ , \tag{32}$$

where $\mathbf{P}$ results from Eq.(3).

To reduce r as much as possible, $\mathbf{Q}=\mathbf{I}$ could be selected; but it is not an optimum solution although the identity matrix has a long tradition with Lyapunov equations *Patel, R.V., and Toda, M., 1980*. The tabular equations in what follows summarize several interrelations

$$\mathbf{Q}=\mathbf{I} \quad\vdots\quad \mathbf{Q}\neq\mathbf{I} \tag{33}$$

$$\min_i \lambda_i[\mathbf{P}] \;>\; \min_i \lambda_{Qi}[\mathbf{P}] \tag{34}$$

$$-\min_i \lambda_i[\mathbf{P}] \;<\; -\min_i \lambda_{Qi}[\mathbf{P}]\ . \tag{35}$$

On the other hand, there is no interrelation as follows

$$\max_i \lambda_i[\mathbf{P}] \;\not<\not> \max_i \lambda_{Qi}[\mathbf{P}] \tag{36}$$

$$\min_i \frac{1}{\lambda_i[\mathbf{P}]} \;\not<\not> \min_i \frac{1}{\lambda_{Qi}[\mathbf{P}]} \tag{37}$$

$$\max_i \frac{-1}{\lambda_i[\mathbf{P}]} \;\not<\not> \max_i \frac{-1}{\lambda_{Qi}[\mathbf{P}]}\ . \tag{38}$$

Eq.(38) yields no relation. If Eq.(38) was true for $<$, this would be helpful and $\mathbf{Q}=\mathbf{I}$ would be the best choice. However it does not hold. Hence, variations of $\mathbf{Q}$ are reasonable; since $\max(-1/\lambda_{Qi}[\mathbf{P}])$ is that value closest to the boundary of positive realness it should be made as small as possible. That is, find

$$\min_{\mathbf{Q}} \max_i \{\frac{-1}{\lambda_{Qi}[\mathbf{P}]}\ \}\ . \tag{39}$$

Eq.(34) is guaranteed but not useful for design. Note that the true r and not $\max_i\{\frac{-1}{\lambda_i[\mathbf{P}]|_{\mathbf{Q}=\mathbf{I}}}\}$, which is usually bigger, should be obtained.

In other words, find

$$\max_i\{-\lambda_{Pi}[\mathbf{Q}]\} = \max_i\{-1/\lambda_{Qi}[\mathbf{P}]\} = -\min_i\{1/\lambda_{Qi}[\mathbf{P}]\} \not<\not> \max_i\{-1/\lambda_i[\mathbf{P}]\} \ . \tag{40}$$

Referring to these results, starting at a random $\mathbf{Q} > 0$, find a gradient of $\mathbf{Q}$ such that the efficiency r is reduced, and $\mathbf{Q}$ such that r becomes a minimum; considering $\mathbf{P}$ resulting from the appropriate Lyapunov equation

$$\mathbf{A}_{cl}^T\mathbf{P} + \mathbf{P}\mathbf{A}_{cl} + \mathbf{Q} = \mathbf{0} \ . \tag{41}$$

Then, from

$$\mathbf{A}_{cl}^T\Delta\mathbf{P} + \Delta\mathbf{P}\cdot\mathbf{A}_{cl} + \Delta\mathbf{Q} = \mathbf{0} \quad \rightsquigarrow \quad \mathrm{col}\Delta\mathbf{P} \overset{(54)}{=} -\mathbf{H}^{-T}\mathrm{col}\Delta\mathbf{Q} \ . \tag{42}$$

From the setup Eq.(4) and Eq.(9), and using $\mathbf{x},\ \mathbf{y}$ as defined in the special case of Eq.(53)

$$\begin{aligned}
r + \Delta r &= -\frac{\mathbf{x}^T(\mathbf{Q}+\Delta\mathbf{Q})\mathbf{x}}{\mathbf{x}^T(\mathbf{P}+\Delta\mathbf{P})\mathbf{x}} && (43)\\
\Delta r &= -\frac{(\mathbf{x}^T\Delta\mathbf{Q}\ \mathbf{x})(\mathbf{x}^T\mathbf{P}\mathbf{x}) - (\mathbf{x}^T\Delta\mathbf{P}\ \mathbf{x})(\mathbf{x}^T\mathbf{Q}\mathbf{x})}{(\mathbf{x}^T\mathbf{P}\mathbf{x})^2} && (44)\\
\Delta r &= -\frac{(\mathbf{y}^T\Delta\mathbf{q})(\mathbf{x}^T\mathbf{P}\mathbf{x}) - (\mathbf{y}^T\mathrm{col}\Delta\mathbf{P})(\mathbf{y}^T\mathrm{col}\mathbf{Q})}{(\mathbf{x}^T\mathbf{P}\mathbf{x})^2} && (45)\\
\Delta r &= \frac{-1}{[\mathbf{y}^T(-\mathbf{H}^T)\mathrm{col}\mathbf{Q}]^2}[\mathbf{y}^T\Delta\mathbf{q}\cdot\mathbf{y}^T(-\mathbf{H}^{-T})\mathrm{col}\mathbf{Q} - \mathbf{y}^T(-\mathbf{H}^{-T})\Delta\mathbf{q}\cdot\mathbf{y}^T\mathbf{q}] && (46)\\
\Delta r &= \frac{-1}{[\mathbf{y}^T\mathbf{H}^{-T}\mathrm{col}\mathbf{Q}]^2}\mathbf{y}^T[-\Delta\mathbf{q}\cdot\mathbf{y}^T\mathbf{H}^{-T} + \mathbf{H}^{-T}\Delta\mathbf{q}\cdot\mathbf{y}^T]\mathbf{q} && (47)\\
\Delta r &= \frac{-1}{[\mathbf{q}^T\mathbf{H}^{-1}\mathbf{y}]^2}\mathbf{y}^T[\mathbf{H}^{-T}\Delta\mathbf{q}\cdot\mathbf{y}^T - \Delta\mathbf{q}\cdot\mathbf{y}^T\mathbf{H}^{-T}]\mathbf{q} && (48)\\
\Delta r &= \frac{-1}{[\mathbf{q}^T\mathbf{H}^{-1}\mathbf{y}]^2}[\mathbf{y}^T\mathbf{q}\mathbf{y}^T\mathbf{H}^{-T}\Delta\mathbf{q} - \mathbf{y}^T\mathbf{H}^{-T}\mathbf{q}\mathbf{y}^T\Delta\mathbf{q} && (49)\\
\Delta r &= \frac{1}{[\mathbf{q}^T\mathbf{H}^{-1}\mathbf{y}]^2}\mathbf{y}^T[\mathbf{q}\mathbf{y}^T\mathbf{H}^{-T} - \mathbf{H}^{-T}\mathbf{q}\mathbf{y}^T]\Delta\mathbf{q} && (50)
\end{aligned}$$

$$\boxed{\frac{\partial r}{\partial\mathbf{q}} = \frac{1}{(\mathbf{q}^T\mathbf{H}^{-1}\mathbf{y})^2}[\mathbf{H}^{-1}\mathbf{y}\mathbf{q}^T - \mathbf{y}\mathbf{q}^T\mathbf{H}^{-1}]\mathbf{y}\ ,} \tag{51}$$

where $\frac{\partial\mathbf{b}^T\mathbf{k}}{\partial\mathbf{k}} = \mathbf{b}$ and the following abbreviations have been used

$$\mathbf{q} \overset{\triangle}{=} \mathrm{col}\mathbf{Q}, \quad \Delta\mathbf{q} \overset{\triangle}{=} \mathrm{col}\Delta\mathbf{Q} \tag{52}$$

$$\mathbf{y} \overset{\triangle}{=} \mathbf{x}\otimes\mathbf{x} \ \text{ for } \ \mathbf{x} = \mathbf{v}_{i_M} \text{ and } \ [\mathbf{V},\mathbf{D}] = \mathrm{eig}[\mathbf{P},\mathbf{Q}],\ \arg_i\max D(i) = i_M \tag{53}$$

$$\mathbf{H} \overset{\triangle}{=} \mathbf{I}_n\otimes\mathbf{A}_{cl} + \mathbf{A}_{cl}\otimes\mathbf{I}_n \ . \tag{54}$$

A step for the improvement of r is given by $\Delta\mathbf{q} = (\frac{\partial r}{\partial \mathbf{q}})^T \Delta r_{arb}$.

Having arrived at $\frac{\partial r}{\partial \mathbf{q}} = \mathbf{0}$ in Eq.(51), the result is termed $\mathbf{Q}^\star$ and $\mathbf{P}^\star$. There is a huge multitude of solutions. Irrespective of this multitude it turns out astonishingly that the resulting r is the same and given by Eq.(66); the same value as resulting from a diagonal $\mathbf{A}$. Anticipating the result $\lambda_Q[\mathbf{P}] = \lambda_{Qi}[\mathbf{P}] \stackrel{\triangle}{=} -0.5/\lambda_{i\max}[\mathbf{A}]$ even for general $\mathbf{A}$, $\mathbf{P}$, $\mathbf{Q}$,

$$\begin{aligned} \det\{\lambda_Q[\mathbf{P}] \cdot \mathbf{Q} - \mathbf{P}\} &= 0 && (55)\\ \det\{\lambda_Q[\mathbf{P}] \cdot (\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A}) + \mathbf{P}\} &= 0 && (56)\\ \det\{\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - 2\,\mathbf{P}\lambda_{i\max}[\mathbf{A}]\} &= 0 && (57)\\ \det\{(\mathbf{A}^T - \lambda[\mathbf{A}] \cdot \mathbf{I}_n)\mathbf{P}^\star + \mathbf{P}^\star(\mathbf{A} - \lambda[\mathbf{A}] \cdot \mathbf{I}_n)\} &= 0\ . && (58) \end{aligned}$$

The matrix $\mathbf{P}$ is termed reflected on the matrix $\mathbf{A}$ if $\mathbf{P}$ obeys

$$\det\{(\mathbf{A}^T - \lambda[\mathbf{A}] \cdot \mathbf{I}_n)\mathbf{P} + \mathbf{P}(\mathbf{A} - \lambda[\mathbf{A}] \cdot \mathbf{I}_n)\} = 0\ . \qquad (59)$$

The multitude of solutions is illustrated in the Appendix.

This result coincides with the problem: Investigate $\lambda_{Qi}[\mathbf{P}]$ of

$$\frac{\partial \lambda_{Qi}[\mathbf{P}]}{\partial \mathbf{Q}} = \mathbf{0} \quad \text{s.t.} \quad \det(\lambda_{Qi}[\mathbf{P}] \cdot \mathbf{Q} - \mathbf{P}) = \mathbf{0}, \qquad \mathbf{Q} = -\mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{A}\ , \qquad (60)$$

where the solution $\lambda_{Qi}[\mathbf{P}] = -0.5/\lambda_i[\mathbf{A}]$ holds. The huge magnitude of solutions is due to the only one condition $\det(\lambda_{Qi}[\mathbf{P}] \cdot \mathbf{Q} - \mathbf{P}) = 0$ in Eq.(60) for the n^2 elements of $\mathbf{Q}$.

Example 3. Third-Order System. Optimum Q: The gradient is computed for $\mathbf{A} = \mathrm{diag}[-1,\ -2,\ -4]$. The norm of the gradient of Eq.(51) for a general and symmetric $\mathbf{Q}$ is plotted in Fig. 8, when only two of the six parameters are changed, i.e., $Q(2,1)$ and $Q(3,1)$.

The resulting r is given in Fig. 9, starting with $\mathbf{Q} = \mathbf{I}_3 + 1.3 \cdot \mathtt{ones}(3,3)$ and ending with

$$\mathbf{Q}^\star = \begin{pmatrix} 2.9210 & 0.4316 & 0.2495 \\ 0.4316 & 2.5025 & 1.4988 \\ 0.2495 & 1.4988 & 2.5364 \end{pmatrix}\ . \qquad \square \qquad (61)$$

Summarizing the results yields

1. Based on Eq.(3), for $\mathbf{Q} = \mathbf{I}$ we get $r = \max\{-\frac{1}{\lambda_i[\mathbf{P}]}\}$.
2. For some general $\mathbf{Q} \neq \mathbf{I}$ we have $r = \max\{-\frac{1}{\lambda_{Qi}[\mathbf{P}]}\}$.

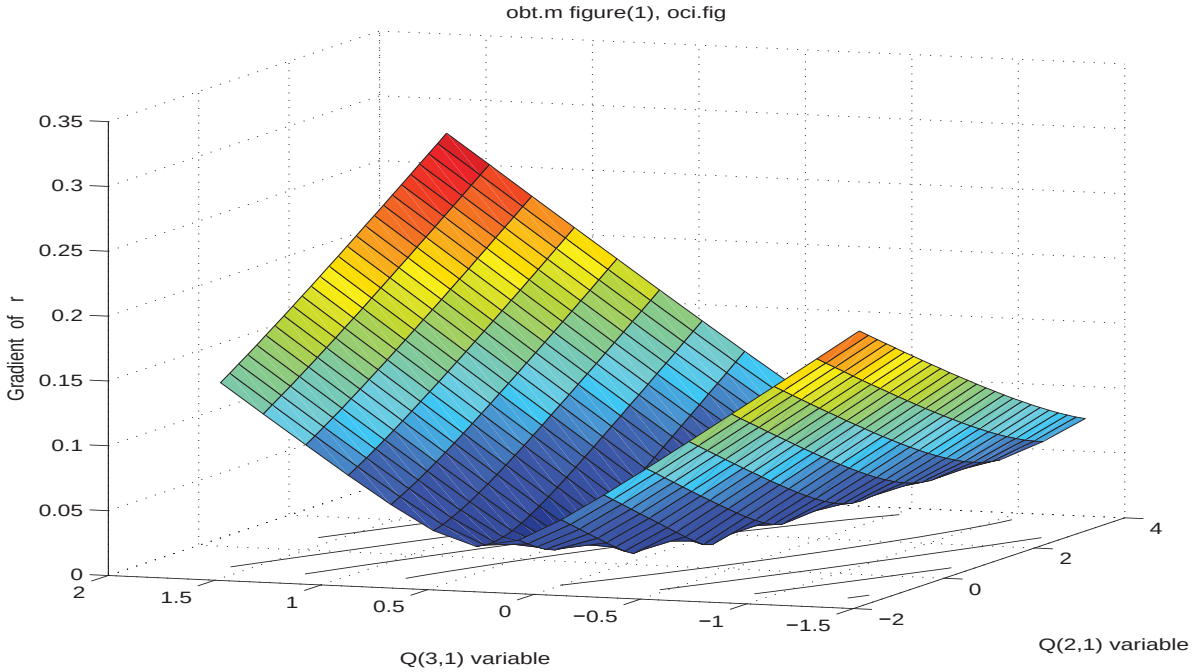

Figure 8: Gradient in the neighborhood of the gradient zero in Example 3

3. For diagonal $\mathbf{A} = \mathrm{diag}[a_i]$ only, all a_i real, and the result of the Lyapunov equation simply turns out $\mathbf{P} = -0.5\mathbf{A}^{-1}\mathbf{Q}$ and

$$\det(\lambda_Q[\mathbf{P}] \cdot \mathbf{Q} - \mathbf{P}) = 0 \tag{62}$$

$$\det\{\lambda_Q[\mathbf{P}] \cdot \mathbf{Q} + 0.5\mathbf{A}^{-1}\mathbf{Q}\} = 0 \tag{63}$$

$$\det\{(\lambda_Q[\mathbf{P}] \cdot \mathbf{I}_n + 0.5\mathbf{A}^{-1})\} \det \mathbf{Q} = 0 \tag{64}$$

$$\text{and without influence of } \mathbf{Q}: \quad \lambda_{Qi}[\mathbf{P}] \overset{(87)}{=} -\frac{0.5}{a_i} = -\frac{0.5}{\lambda_i[\mathbf{A}]} \tag{65}$$

$$r \triangleq \max_i \frac{-1}{\lambda_{Qi}[\mathbf{P}]} = -\min_i \frac{1}{\lambda_{Qi}[\mathbf{P}]} \overset{(65)}{=} \min_i(-\frac{\lambda_i[\mathbf{A}]}{0.5}) = 2\max_i \lambda_i[\mathbf{A}] \ . \tag{66}$$

4. $r = \frac{-1}{\lambda_{Qi}[\mathbf{P}]}$ is true only for the specific result $\mathbf{Q}$ which is the final result of Eq.(51). For several different initial $\mathbf{Q}$, the results differ as well, but the resulting efficiency r converges to the same value r of Eq.(66), astonishingly. It does not matter, if $\mathbf{Q}$ is symmetric or not, or $\mathbf{A}$ in nonsymmetric or nondiagonal. Even if Eq.(51) was not solved and $\mathbf{Q}$ remained unknown, one has $r = 2\max_i \lambda_i[\mathbf{A}]$ in general. There is no need to know $\mathbf{P}$ and $\mathbf{Q}$ in detail, since the optimum choice always presents $2\lambda_{\max}[\mathbf{A}]$. Selecting an arbitrary $\mathbf{Q}$, which does not follow Eq.(51), there results some r which is worse.

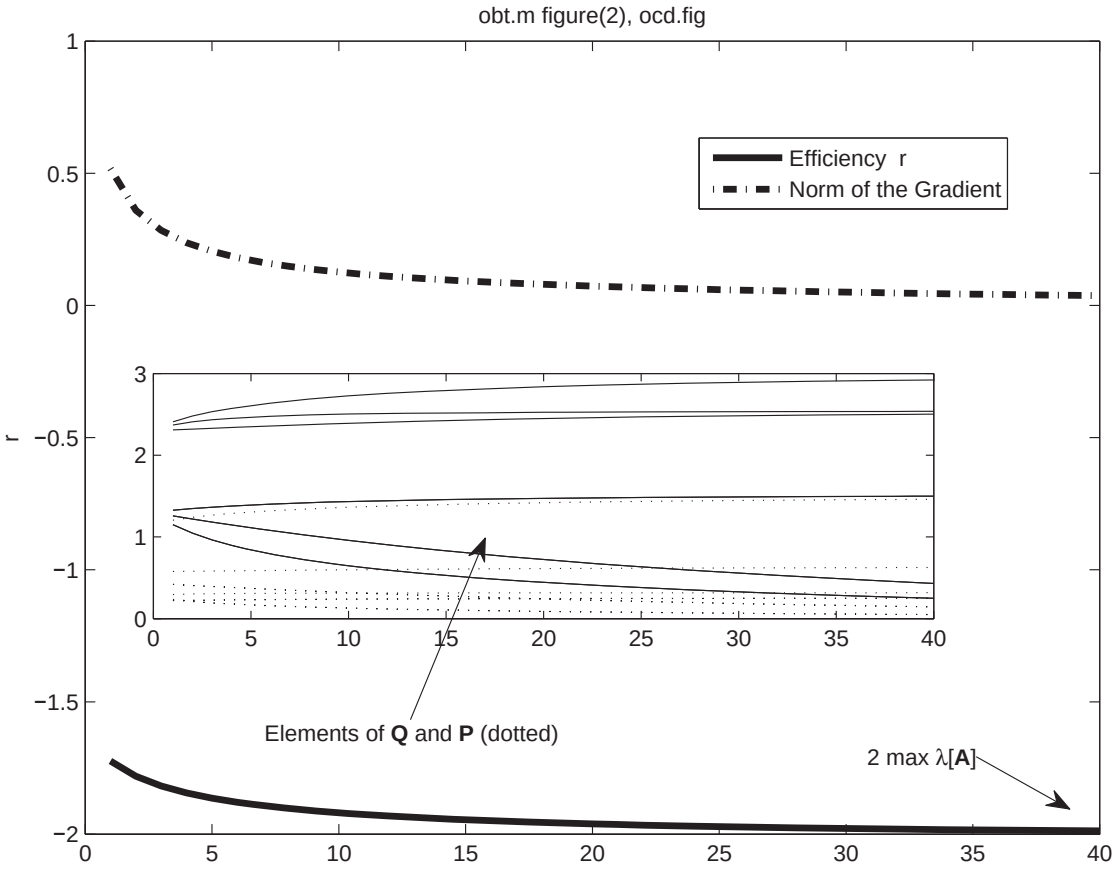

Figure 9: Improvement of the efficiency $r = -\mathbf{x}^T\mathbf{Q}\mathbf{x}/\mathbf{x}^T\mathbf{P}\mathbf{x}$ when following the maximum of the negative generalized eigenvalue $\lambda_Q[\mathbf{P}]$ in Example 3

Example 4. Several initial symmetric Q: With the step factor 0.5 of Δcol **Q** in Eq.(51), applied for 20 steps, we find

```
  A = -0.8000   -0.5000   -0.8000
       0.1000   -2.0000    0
      -2.0000   -1.0000   -4.0000

       initial Q                final Q                  final P
0.9413  0.4116  0.1065     0.9804  0.0246  0.1716     1.0766 -0.1672 -0.1939
0.4116  0.7586  0.2737     0.0246  0.8838  0.4055    -0.1672  0.2201  0.0852
0.1065  0.2737  0.7120     0.1716  0.4055  0.6882    -0.1939  0.0852  0.1248

final efficiency r = -0.7515;    2 times max eig(A) = -0.7515
```

Example 5. Several initial non-symmetric Q: With the same **A** as in Example 4, the results are presented in Fig. 10 and as follows

```
       initial Q                final Q                  final P
1.3991  0.6255  0.1899     1.4751  0.2791  0.2581     1.6333 -0.1371 -0.2866
0.3876  0.9660  0.4247     0.0411  1.0304  0.5737    -0.2489  0.2550  0.1244
0.1215  0.2353  0.9587     0.1897  0.3843  0.8875    -0.3018  0.0792  0.1698

final efficiency r = -0.7506;    2 times max eig(A) = -0.7515
```

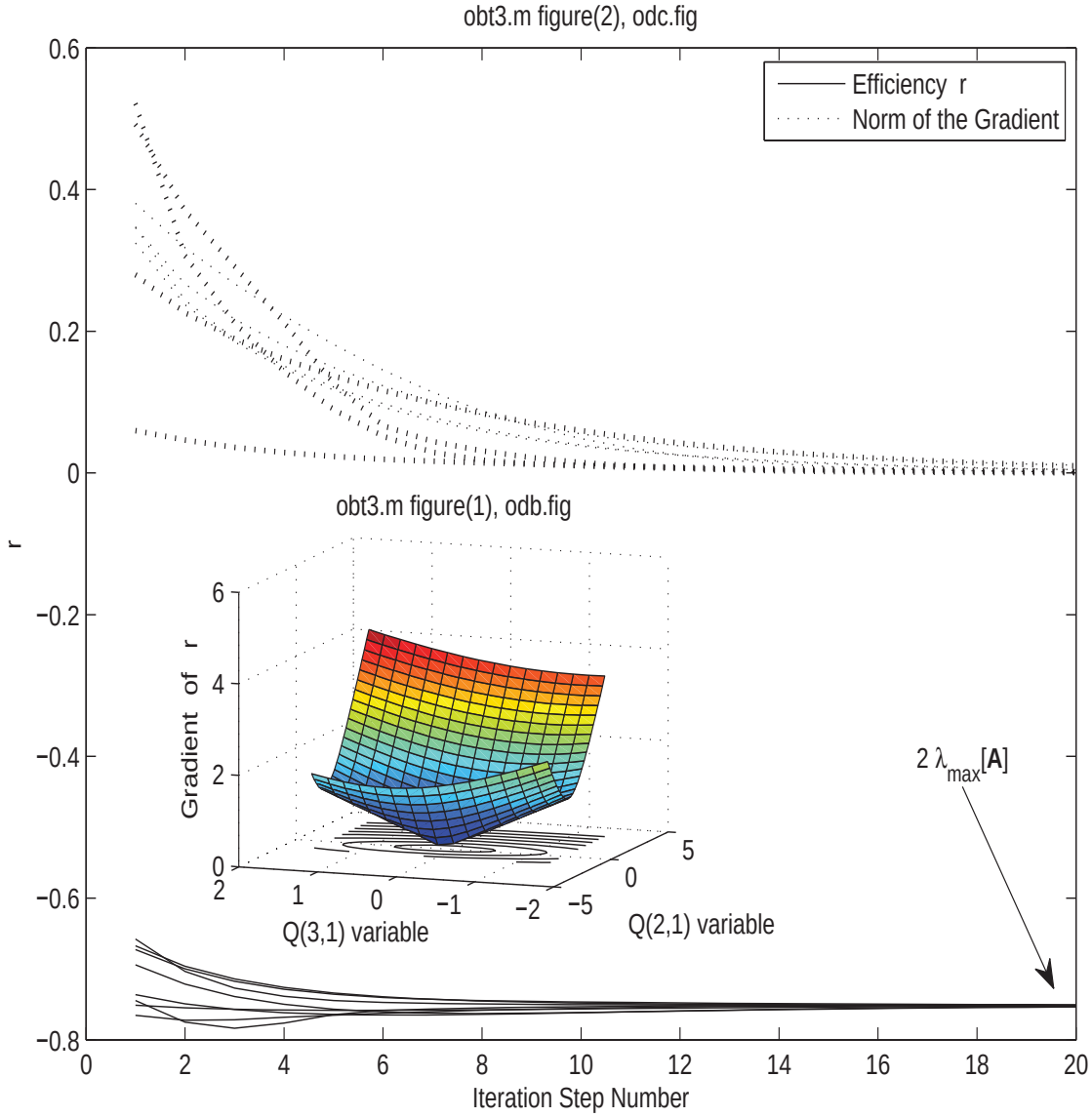

Figure 10: Efficiency and norm of the gradient and typical efficiency surface for Example 5 for a couple of assumptions $\mathbf{Q}$

5 Lyapunov-Based Efficiency for Discrete-Time Systems

Consider a discrete-time system of the type $\mathbf{\Phi}(t) = e^{\mathbf{A}t}$. In what follows, always $\mathbf{\Phi} = \mathbf{\Phi}(T_s)$

$$\mathbf{x}_{i+1} = \mathbf{\Phi}\mathbf{x}_i \ . \tag{67}$$

Assume the Lyapunov function $V_i = \mathbf{x}_i^T\mathbf{P}\mathbf{x}_i$ with $\mathbf{P} = \mathbf{P}^T > 0$. Then, the increment of the Lyapunov function is

$$\Delta V = V_{i+1} - V_i = \mathbf{x}_i^T(\mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{P})\mathbf{x}_i \ . \tag{68}$$

Postulating a negative increment requires positive definiteness of

$$\mathbf{Q} \stackrel{\triangle}{=} \mathbf{P} - \mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} > 0 \ . \tag{69}$$

Then, $\mathbf{Q} = \mathbf{Q}^T$. Stability is guaranteed if $\mathbf{P}$ is positive definite, having solved the Lyapunov equation, e.g., via

$$\mathrm{col}\,\mathbf{P} = (\mathbf{I}_n - \mathbf{\Phi}^T \otimes \mathbf{\Phi}^T)^{-1}\mathrm{col}\mathbf{Q} \ , \tag{70}$$

or via MATLAB $\mathbf{P} = \texttt{dlyap}(\mathbf{\Phi}^T, \mathbf{Q})$.

We now leave the necessary stability conditions based on pure definiteness. We consider the numerical values of $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ and $\mathbf{x}^T\mathbf{P}\mathbf{x}$ in the entire $\mathbf{x}$-space. We have in mind to analyze nonlinear systems where the transition matrix $\mathbf{\Phi}$ is a function of the state variable and differs from sampling instant to sampling instant. In order to utilize a state dependent stability efficiency r we define the efficiency in discrete time as in Eq.(10)

$$r \stackrel{\triangle}{=} \max\ r_i = \max_i\{-\frac{\Delta V}{V}\} = \max_i\{\frac{-1}{\lambda_{Qi}[\mathbf{P}]}\} \stackrel{\triangle}{=} -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} \qquad \mathbf{x} \neq \mathbf{0} \ . \tag{71}$$

For $\mathbf{P} = \mathbf{I}_n$, r results from the inverse Rayleigh quotient. Inside a given region $\mathcal{X}$ of the state space, $r < 0$ is required. If $\mathbf{\Phi}$ can be influenced by a state controller $\mathbf{K} \in \mathcal{R}^{m\times n}$, that is,

$$\mathbf{\Phi} = \mathbf{\Phi}_0 + \mathbf{\Psi}\mathbf{K} \ , \tag{72}$$

we intend to decrease r by modifying $\mathbf{K}$, e.g., in a gradient-based algorithm. In order to obtain best dynamics, that is, to obtain steepest $\dot{V}$ (best decrease ΔV) for a given $\mathbf{P}$, r should be minimum, as carried out in Eq.(26).

(Another option with given positive definite matrix $\mathbf{P}$ would be

$$r_P \stackrel{\triangle}{=} -\frac{\mathbf{x}^T\mathbf{Q}\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} = -\frac{\mathbf{x}^T[\mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{P}]\mathbf{x}}{\mathbf{x}^T\mathbf{P}\mathbf{x}} \ . \tag{73}$$

This alternative is no more considered in what follows.)

For linear discrete-time systems, the results of continuous-time systems can be transferred with the only difference in the matrix $\mathbf{H}$

$$\begin{aligned} \boldsymbol{\Phi}^T\mathbf{P}\boldsymbol{\Phi} - \mathbf{P} + \mathbf{Q} &= \mathbf{0} && (74)\\ (\boldsymbol{\Phi}^T \otimes \boldsymbol{\Phi}^T)\mathrm{col}\ \mathbf{P} - \mathrm{col}\ \mathbf{P} + \mathrm{col}\ \mathbf{Q} &= \mathbf{0} && (75)\\ \text{and, using } \mathbf{H} \stackrel{\triangle}{=} (\mathbf{I} - \boldsymbol{\Phi} \otimes \boldsymbol{\Phi}),\ \text{ finally } \ \mathrm{col}\ \mathbf{P} &= \mathbf{H}^{-T}\mathrm{col}\ \mathbf{Q}\ . && (76) \end{aligned}$$

For diagonal $\mathbf{Q},\ \mathbf{P},\ \boldsymbol{\Phi}$, one finds

$$(\boldsymbol{\Phi}^2 - \mathbf{I}_n)\mathbf{P} + \mathbf{Q} \stackrel{(69)}{=} \mathbf{0} \ \rightsquigarrow\ \ \mathbf{P} = (\mathbf{I}_n - \boldsymbol{\Phi}^2)^{-1}\mathbf{Q}\ . \qquad (77)$$

Referring to the correspondences Eq.(87), one has

$$\begin{aligned} \det\{\lambda_{Qi}[\mathbf{P}] \cdot \mathbf{Q} - \mathbf{P}\} &= 0 && (78)\\ \det\{\lambda_{Qi}[\mathbf{P}] \cdot \mathbf{Q} - (\mathbf{I}_n - \boldsymbol{\Phi}^2)^{-1}\mathbf{Q}\} &= 0 && (79)\\ \det\{\lambda_{Qi}[\mathbf{P}] \cdot \mathbf{I}_n - (\mathbf{I}_n - \boldsymbol{\Phi}^2)^{-1}\}\det\mathbf{Q} &= 0 && (80)\\ \lambda_{Qi}[\mathbf{P}] &= \lambda_i[(\mathbf{I}_n - \boldsymbol{\Phi}^2)^{-1}] && (81)\\ 1/\lambda_{Qi}[\mathbf{P}] &= \lambda_i[\mathbf{I}_n - \boldsymbol{\Phi}^2] = 1 - \lambda_i[\boldsymbol{\Phi}^2] = 1 - \lambda_i[e^{2\,\mathbf{A}\,T_s}] && (82) \end{aligned}$$

$$\begin{aligned} r &= \max_i \frac{-1}{\lambda_{Qi}[\mathbf{P}]} && (83)\\ r &= -1 + \max_i\ \lambda_i[e^{2\,\mathbf{A}\,T_s}] = -1 + \max_i e^{2\,\lambda_i[\mathbf{A}]\,T_s} = -1 + e^{2\,(\max\lambda_i[\mathbf{A}])\,T_s}\ . && (84) \end{aligned}$$

Example 6. For the setup of Example 4 or 5, we find the results in what follows and in Figure 11

```
     initial Q                 final Q                   final P
 1.2506  0.5648  0.2294    1.2537  0.0675  0.1915    14.4314  -1.9240  -2.3535
 0.5648  0.5649  0.2347    0.0675  0.8113  0.4587    -1.9240   2.4653   1.2535
 0.2294  0.2347  0.6899    0.1915  0.4587  0.7552    -2.3535   1.2535   1.8692

final r = -0.0723;  -1+e^{2 max eig(A) T_s} = -0.0724; max(-1./eig(P,Q)) = -0.0723
```

6 Conclusion

The generalized eigenvalues of Lyapunov equation solution $\mathbf{P}$ with respect to the weighting matrix of $\dot{V}$ provides a per-unit measure of the general dyanamic position of a system

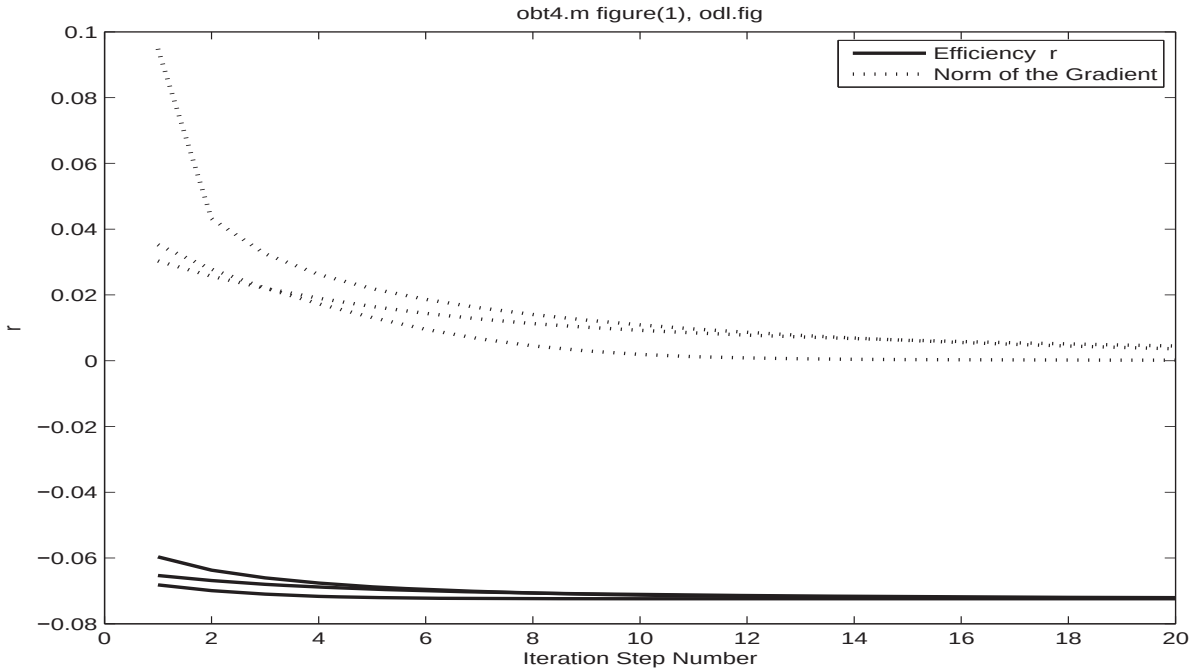

Figure 11: Efficiency r and norm of the gradient for Example 6

without being influenced by any details of the system motion. Up to some numerical bound, the amount of system variables is completely irrelevant.

The result of the efficiency states equal to the double distance to the imaginary axis or to the unit circle, see Eqs.(66) and (84).

If a large sample of states $\mathbf{x}$ was available at a specific time instant before and after a change of the plant A, the efficiency quotient could be used for compensating the change by a change in the controller to obtain the identical closed-loop performance. However, this is not availabe and thus identification applications must be left out of consideration.

References

Ludyk, G., 1990, CAE von dynamischen Systemen. Berlin: Springer-Verlag

Patel, R.V., and Toda, M., 1980, Quantitative measures of robustness for multivariable systems, *Proceedings of Joint Automatic Control Conference San Francisco* Paper TP 8-A

Weinmann, A., 2012, Performance gradients in automatic control analysis. A catalogue of correspondences. *Int. J. Automation Austria* **20**, pp. 1-100. See http://www.acin.tuwien.ac.at/de/Publikationen/Zeitschriften/IJAA

Appendix

A Correspondences

$$\mathrm{col}(\mathbf{x}^T\mathbf{M}) = \mathbf{M}\mathbf{x}\ , \quad \mathrm{col}(\mathbf{x}^T) = \mathbf{x} \tag{85}$$

$$\frac{\partial}{\partial \mathbf{q}}\ \frac{1}{\mathbf{a}^T\mathbf{q}} = \frac{-1}{(\mathbf{a}^T\mathbf{q})^2}\cdot\mathbf{a} = \frac{-\mathbf{a}}{(\mathbf{a}^T\mathbf{q})^2} \tag{86}$$

$$\lambda[\mu\mathbf{I}+\mathbf{A}] = \mu + \lambda[\mathbf{A}],\quad \lambda[\mathbf{A}^{-1}] = \lambda^{-1}[\mathbf{A}],\quad \lambda_i[e^{\mathbf{A}\ t}] = e^{\lambda_i[\mathbf{A}]\ t} \tag{87}$$

$$\det[\mathbf{R}\mathbf{S}] = (\det\mathbf{R})(\det\mathbf{S})\ \text{ for }\ \mathbf{R},\ \mathbf{S}\ \text{ square}\ . \tag{88}$$

Replacing a scalar expression intentively as follows

$$\mathbf{x}^T\mathbf{P}\mathbf{x} \overset{(93)}{=} (\mathbf{x}^T\otimes\mathbf{x}^T)\mathrm{col}\,\mathbf{P} = (\mathbf{x}\otimes\mathbf{x})^T\mathrm{col}\,\mathbf{P}\ . \tag{89}$$

The matrix $\mathbf{U}_{r,m}$ is the permutation matrix in Kronecker matrix sense; entries zero except one solitary digit one in *each* row and column. $\mathbf{E}_{ij}$ is the Kronecker matrix with entries zero except a *solitary* digit one at position i,j

$$\mathbf{U}_{k,l} \overset{\triangle}{=} \mathbf{U}_{k,l}^{(kl\times kl)} \overset{\triangle}{=} \sum_{i=1}^{k}\sum_{j=1}^{l}\mathbf{E}_{ij}^{(k\times l)}\otimes\mathbf{E}_{ji}^{(l\times k)} = \sum_{i}^{k}\sum_{j}^{l}\mathbf{E}_{ij}^{(k\times l)}\otimes\left(\mathbf{E}_{ij}^{(k\times l)}\right)^T. \tag{90}$$

$$\frac{\partial\mathbf{M}^{(k\times l)}}{\partial\mathbf{M}} = \bar{\mathbf{U}}_{k,l} = \bar{\mathbf{U}}_{k,l}^{(k^2\times l^2)} \overset{\triangle}{=} \sum_{i=1}^{k}\sum_{j=1}^{l}\mathbf{E}_{ij}^{(k\times l)}\otimes\mathbf{E}_{ij}^{(k\times l)}\ , \tag{91}$$

e.g.,

$$\bar{\mathbf{U}}_{2,1}^{(4\times 1)} = \begin{pmatrix}1\\0\\0\\1\end{pmatrix}\ ,\qquad \bar{\mathbf{U}}_{2,3}^{(4\times 9)} = \begin{pmatrix}1&0&0&0&1&0&0&0&1\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0&1\end{pmatrix}\ . \tag{92}$$

$$\mathrm{col}(\mathbf{A}\mathbf{C}\mathbf{B}) \ = \ (\mathbf{B}^T\otimes\mathbf{A})\mathrm{col}\mathbf{C}\ . \tag{93}$$

$$\text{Replacing } \mathbf{C} = \mathbf{B} = \mathbf{I}_n \text{ and } \mathbf{A} = \mathbf{x}^T\mathbf{G}^T\mathbf{G} \tag{94}$$

$$\mathrm{col}\mathbf{A} = \mathrm{col}(\mathbf{x}^T\mathbf{G}^T\mathbf{G}) \ = \ (\mathbf{I}_n\otimes(\mathbf{x}^T\mathbf{G}^T\mathbf{G})\mathrm{col}\,\mathbf{I}_n \overset{(93)}{=} \mathbf{G}^T\mathbf{G}\mathbf{x}\ . \tag{95}$$

For the Frobenius norm, one has

$$\|\mathbf{K}\|_F \ = \ \mathrm{tr}[\mathbf{K}^T\mathbf{K}] \tag{96}$$

$$\Delta\mathrm{tr}[\mathbf{K}^T\mathbf{K}] \ = \ \mathrm{tr}[(\mathbf{K}+\Delta\mathbf{K})^T(\mathbf{K}+\Delta\mathbf{K})] - \mathrm{tr}[\mathbf{K}^T\mathbf{K}] \tag{97}$$

$$= \ 2\mathrm{tr}[\Delta K^T\mathbf{K}] = 2\mathrm{col}\Delta\mathbf{K}^T\cdot\mathrm{col}\mathbf{K} \tag{98}$$

$$\frac{\partial\|\mathbf{K}\|_F}{\partial\mathrm{col}\mathbf{K}} \ = \ 2\mathrm{col}\mathbf{K} \tag{99}$$

$$\frac{\partial}{\partial \mathbf{M}}\text{tr}\,[\mathbf{AMBM}^T] = \frac{\partial}{\partial \mathbf{M}}\text{tr}\,[\mathbf{M}^T\mathbf{AMB}] = \mathbf{A}^T\mathbf{MB}^T + \mathbf{AMB}, \tag{100}$$

$$\mathbf{A} = \mathbf{I}_n, \quad \mathbf{B} = \mathbf{x}\mathbf{x}^T, \quad \mathbf{M} = \mathbf{K} \;\rightsquigarrow\; \frac{\partial \text{tr}[\mathbf{K}^T\mathbf{K}\mathbf{x}\mathbf{x}^T]}{\partial \mathbf{K}} \overset{(100)}{=} 2\mathbf{K}\mathbf{x}\mathbf{x}^T \tag{101}$$

$$\begin{aligned} \mathbf{x}^T\mathbf{K}^T\mathbf{K}\mathbf{x} &= \text{tr}[\mathbf{x}^T\mathbf{K}^T\mathbf{K}\mathbf{x}] = (\text{col}\mathbf{K}\mathbf{x})^T\text{col}\mathbf{K}\mathbf{x} && (102)\\ &= (\text{col}\mathbf{K})^T(\mathbf{x}\otimes\mathbf{I}_n)(\mathbf{x}^T\otimes\mathbf{I}_n)(\text{col}\mathbf{K}) = (\text{col}\mathbf{K})^T[(\mathbf{x}\mathbf{x}^T)\otimes\mathbf{I}_n](\text{col}\mathbf{K}) && (103)\\ \frac{\partial \mathbf{x}^T\mathbf{K}^T\mathbf{K}\mathbf{x}}{\partial \text{col}\mathbf{K}} &= 2[(\mathbf{x}\mathbf{x}^T)\otimes\mathbf{I}_n]\text{col}\mathbf{K} && (104)\end{aligned}$$

A.1 Matrix Product Differential Operation Details

The function to be analyzed is $\mathbf{B}$, operating as a simple matrix-valued factor of $\mathbf{K}$.

$$\mathbf{T}^{[n\times n]} = \mathbf{B}^{[n\times m]}\mathbf{K}^{[m\times n]} \tag{105}$$

$$\frac{\partial \mathbf{BK}}{\partial \mathbf{K}} \overset{(110)}{=} (\mathbf{I}_m\otimes\mathbf{B})\frac{\partial \mathbf{K}}{\partial \mathbf{K}} \overset{(91)}{=} (\mathbf{I}_m\otimes\mathbf{B})\bar{\mathbf{U}}_{m,n}^{[m^2\times n^2]} \tag{106}$$

$$(\text{col}\;\Delta\mathbf{T})^{[n^2\times 1]} \overset{(93)}{=} \underbrace{(\mathbf{I}_n\otimes\mathbf{B})}_{\triangleq\mathbf{M}^{[n^2\times mn]}}\left(\text{col}\;\Delta\mathbf{K}^{[m\times n]}\right)^{[mn\times 1]} \tag{107}$$

$$\begin{aligned} \frac{\partial\,(\text{col}\;\mathbf{T})^{[n^2\times 1]}}{\partial\,(\text{col}\;\mathbf{K})^{[mn\times 1]}} &\overset{(110)}{=} (\mathbf{I}_{mn}\otimes\mathbf{M})\bar{\mathbf{U}}_{mn,1}^{[(mn)^2\times 1]} = (\mathbf{I}_{mn}\otimes\mathbf{I}_n\otimes\mathbf{B})\bar{\mathbf{U}}_{mn,1}^{[(mn)^2\times 1]} && (108)\\ \left(\frac{\partial \text{col}\;\mathbf{T}}{\partial \text{col}\;\mathbf{K}}\right)^{[mn^3\times 1]} &= (\mathbf{I}_{mn^2}\otimes\mathbf{B})^{[mn^3\times(mn)^2]}\,\bar{\mathbf{U}}_{mn,1}^{[(mn)^2\times 1]}\;, && (109)\end{aligned}$$

having used $\mathbf{I}_{mn}\otimes\mathbf{I}_n = \mathbf{I}_{mn^2}$ and

$$\frac{\partial(\mathbf{A}+\mathbf{BK})}{\partial \mathbf{K}} = \frac{\partial \mathbf{BK}}{\partial \mathbf{K}} = (\mathbf{I}_m\otimes\mathbf{B})\frac{\partial \mathbf{K}}{\partial \mathbf{K}} = (\mathbf{I}_m\otimes\mathbf{B})\bar{\mathbf{U}}_{mn}\;. \tag{110}$$

B Reflected Matrix

For the sake of illustration, two cases are shown:

- For the two-dimensional matrices

$$\mathbf{A} = \begin{pmatrix} a & 0 \\ 1 & b \end{pmatrix}\;, \qquad \mathbf{P} = \begin{pmatrix} 1 & p \\ p & q \end{pmatrix}\;, \tag{111}$$

  one finds

$$-[q + p(a-b)]^2 \overset{(59)}{=} 0 \quad\rightsquigarrow\quad q = (b-a)/p. \tag{112}$$

- For a three-dimensional matrix $\mathbf{A}$ and $\mathbf{P}$ one finds in detail the result of Eq.(59)

```
      [ a, 0, 0]          [ p, 0, 0]       [ 2*a*p,     q,     r]
  A = [ 1, b, 0]       P= [ 0, q, 0]    Q=[     q, 2*b*q,     r]
      [ 1, 1, c]          [ 0, 0, r]       [     r,     r, 2*c*r]
Equation (59) yielding p, q, r   for  eig(A)=a, b, or c
         2*q*r*(q*(a - c) + r*(a - b + 1)) =0
       2*r*((b - c)*q^2 + r*q - p*r*(a - b))=0
         -2*r^2*(p*(a - c) - q*(c - b + 1)) =0 .
```

The multitude is recognizable from $r = q(c-a)/(a-b+1)$.

Mixed Reality Server and Remote Interface Communication for ROS Based Robotic System

Nayden Chivarov, Svetlin Penkov, George Angelov,
Daniel Radev, Nedko Shivarov, and Vladimir Vladimirov

Service Robotics Group of the
Institute of Systems Engineering and Robotics,
Bloc II Acad. Bonchev str., Sofia 1113 Bulgaria

e-mails: nshivarov@code.bg, svetlin_penkov@abv.bg, george@ebpw.com,
djradev@gmail.com, nedko@code.bg, and vl_vladimirov@yahoo.com

Abstract
This article describes the design and the implementation of a Mixed Reality Server (MRS) and gives communication design concept for the Care-o-Bot robot user interface using the rosbridge stack and the open-source meta-operating system/platform ROS. The open source Robot Operating System (ROS) becomes more popular both within the academic and the service robotics community with the ability for rapid integration in different types of robotic platforms, existence of easy configurable navigational stacks and wide range of robotic hardware support. The server is developed for the Multi-Role Shadow Robotic System for Independent Living project whose goal is to develop a service robot capable of assisting elderly people in their homes. The purpose of the Mixed Reality Server is to improve the human-robot interaction by mixing the video streams from the robot's cameras with the virtual objects that the robot has perceived. It also augments the map of the robot's environment with the objects present that are detected by the robot. The implementation is based on standard TCP/IP communication. The server waits for HTTP requests, reads the video stream from the appropriate camera, draws the virtual objects on every frame of the camera stream and responds to the request with a MJPEG stream of the resulting enhanced video. The HTTP based communication allows multi-platform utilization of the MRS capabilities.

1 Introduction

Very important part of any robotic system is the user interface, which allows the two-way interaction between the operator and the robot. The requirements for such a control component can be very complex and demanding, especially in the area of service robotics where interaction between untrained persons and the robotic system should be human-safe, flexible, near real-time and easy to use. Such an interface requires communication medium, which allows transparent command and data transfer between the UI and the robotic system. The primary goal is to provide an intuitive, user friendly, safe and network bandwidth optimized interface access to the Care-o-Bot robot system which can be used from any remote site. Controlling such a complex service robot as the Care-O-bot could be quite a difficult task

which only an expert is capable of performing. That is why different methods for human-robot interaction are becoming more and more extensively investigated. One of these methods is augmenting real video feed with virtual objects that are of interest. This allows the operator of the robot to comprehend the environment of the robot much easier and send appropriate commands. Mixed Reality Server (MRS) which provides that functionality implemented in the environment of the Robot Operating System (ROS) is described in the paper. ROS is a meta-operating system that aims to provide a uniform software solution for different types of robots. It has libraries which implement algorithms for solving problems in the field of robotics such as the forward and inverse kinematic problem, navigation, sensor fusion, object recognition etc. Utilizing all these packages drastically reduces the development time and improves the success rate of the project. Currently, ROS is best supported for the Ubuntu Linux operating system, which means that the server machine on which the MRS is deployed should run on Ubuntu Linux.
The basic structural and functional unit of the ROS is the node. Every node performs at least one function and can listen to or publish messages on one or many topics. The communication between the nodes relies on the TCP/IP protocol. A single executable binary can work with many nodes and several binaries related to a single problem can be combined in a stack that could be thought as a library for solving the problem.

2 System information and communication requirements of the Care-O-Bot system

Figure 1: The Care-o-Bot Robot

The Care-o-Bot robot (fig.1) is a service robot designed as an elderly people care helper, developed by the Fraunhofer-IPA institute.

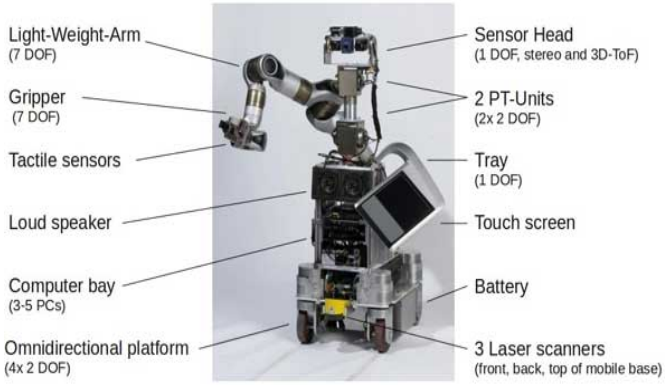

Figure 2: Anatomy of the Care-o-Bot system

The system consists of omnidirectional drive 4-wheel mobile base, foldable tray, arm with 7 degrees of freedom, tactile sensitive gripper, head with 3 cameras, laser scanners mounted on the base, other sensors, control PCs and Li-ion battery power.
The control system is mounted on the robot as carried servers, based on the ROS (Robot Operating System)[1] running on Ubuntu Linux [2] [3]. The system can have one or more connected remote interfaces. The whole communication in ROS is based on the standard TCP/IP network stack. The user interface (UI) and corresponding communication layers are using 5 main data types:

2.1 Mapping and Navigation data

Enables UI to visualize the working environment map created by the laser scans and other sensors data and also marks the robot footprint position.

2.2. Object Detection Data

Objects detected by the robot using cameras and laser scans (for human leg detector).
This includes elements as furniture and household objects, human and pet presence.

2.3. Robot Feedback

Provides full information in real-time about current robot status such as:
- power data
- health & diagnostic information
- status and results from the completion of the user invoked tasks

2.4. Robot Actions Control

Enables the operator to execute common tasks as:
- direct robot control (manual movements)
- navigation aided move to desired map position
- grasp objects
- detect or learn new objects
- execute more complex tasks – for example: bring an object on the table
- define new tasks based on available basic actions as: move, grasp, detect object and etc.

2.5. Video Transport

Visualize robot camera feeds and recognized objects boundaries 3D models and point cloud data visualization (reserved for professional interface).

3 Requirements for the MRS

The MRS has to perform a collection of tasks that are relatively trivial, but combined together they form a complex software module. That demands for detailed requirements analysis and careful design of the system.

3.1 Functional Requirements

The mixed reality server should be able to communicate with the robot and more specifically retrieve video frames from the cameras mounted on the robot. Secondly, the server should be able to fetch information from the objects database. The contours of the detected objects must be drawn on the video frames. Then the resulting video must be encoded and sent to the client. The user on the client interface should be able to manipulate the virtual objects (e.g. click on them) which means that the MRS should provide a way of identifying the objects. The actions of augmenting video frames from the robot's cameras and augmenting the map of the environment are identical so the MRS should support that functionality as well. Moreover, many users should be able to use the MRS at the same time.

3.2 Non-functional Requirements

There will be several interfaces to control the robot for different types of users, so the architecture of the MRS must be as portable as possible in case the different interfaces run on different operating systems. Moreover, the MRS must not induce significant delay in the system, keeping it as real-time as possible. The communication with the server must be reliable and secure.

4 Design of the MRS

The non-functional requirements for the system dictated most of the decisions made for the design of the system. Figure 3 depicts the general design of the Mixed Reality Server.

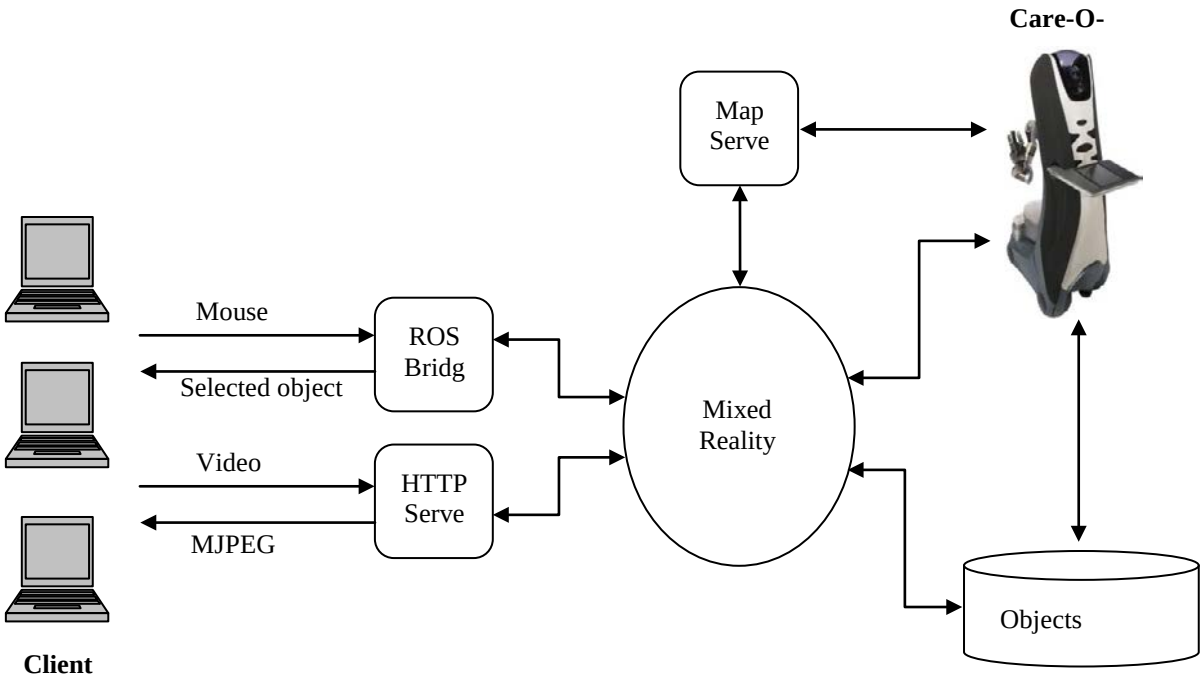

Figure 3: The general architecture of the Mixed Reality Server

The Mixed Reality Server is deployed on the physical server machine that is in the house where the robot operates and it runs in ROS. There are two main ways in which clients can

communicate with the MRS – Websockets through ROS Bridge [4] and HTTP requests. The communication being entirely web based allows for any kind of operating system to communicate with the MRS.

4.1 Streaming augmented video

The implemented video streamer was based on the MJPEG Server in ROS [5]. When the client needs to display augmented video to the user it has to send an HTTP request to the Mixed Reality Server. A compact HTTP server receives it, parses the arguments and the MRS takes the appropriate actions. The URL sent for the request has the following format:

```
http://serveraddress:port/stream?topic=/camera_topic?width
  =640?height=480?quality=90?invert=none
```

The value for "/topic=" should be the name of the topic which publishes the video form the desired camera on the robot. The parameters after that are optional and can set the resolution of the resulting MJPEG stream, the compression quality and if the video should be flipped horizontally. The MRS can also provide just a single augmented frame from the video on the following URL:

```
http://serveraddress:port/snapshot?topic=/camera_topic?wid
  th=640?height=480?quality=90?invert=none
```

If the value for topic is "/map" or "/mappos" then the server responds with video stream or a single image of the augmented map of the environment with or without the position of the robot noted.

4.2 Augmenting the video with virtual objects

Retrieving frames from the camera relies on the Image Transport stack in ROS [6]. From image messages used by the Image Transport stack the frames are converted to IplImage structures which is the native image structure of OpenCV [7]. Then information for the objects that has to be drawn on the video is fetched from the database. The database stores information about the position, orientation and the contour of the object. Using functions provided by OpenCV the objects' contours and labels are drawn on every frame of the video stream.

If a request for the map is made, where the position of the robot should be noted, then the Mixed Reality Server calculates what is the position of the robot in pixels using the "/tf" node in ROS.

The "/tf" node supports transformations of coordinates between different frames of the robot. Thus the position of the robot is transformed with respect to the frame of the map and then scaled according to the resolution of the digital map.

4.3 Identifying manipulated objects

Just highlighting the objects of interest on the video stream would increase the quality of the experience of the user, but does not improve the human-robot interaction. That is why a method for manipulating the highlighted objects is needed. When the user clicks anywhere on the displayed video relative coordinates of the mouse click are sent to the MRS through the

ROS Bridge. The ROS Bridge listens to web message and retranslates as a ROS message on the specified topic. The Mixed Reality Server is subscribed to the topic and receives the message. It performs a hit test on every drawn virtual object. If the mouse click is inside an object its ID is sent back to the client. The interface then displays suitable actions for that object. This mechanism makes the robot control much simpler and easier to be understood by elderly people who usually do not follow the latest technological breakthroughs.

5 Implementation of the MRS

ROS provides development stacks for C++ and Python. The Mixed Reality Server is implemented entirely in C++ and in total it consists of several nodes combined in two binaries encapsulated in a single stack.
The first executable binary is called “ControlMRS” which communicates with the database and sends appropriate messages to the drawing module called “MRS”. For every user that connects to the server a thread is started by each of the binaries that are responsible for augmenting and streaming the video. Figure 4 shows how the two separate binaries should be invoked in order to start the MRS to work.

```
srg@ros-server2:~$ roscd MixedRealityServer/
srg@ros-server2:~/git/care-o-bot/srs/MixedRealityServer$ rosrun MixedRealityServer MRS
[ INFO] [1316343449.417057789]: Starting mjpeg server
[ INFO] [1316343449.417277629]: Bind(8080) succeeded
[ INFO] [1316343449.417313652]: waiting for clients to connect
```

```
srg@ros-server2:~$ roscd MixedRealityServer/
srg@ros-server2:~/git/care-o-bot/srs/MixedRealityServer$ rosrun MixedRealityServer ControlMRS
```

6 Results for MRS

The Mixed Reality Server has been tested using a simulation tool called gazebo. Figure 5 shows a picture of the simulating environment.

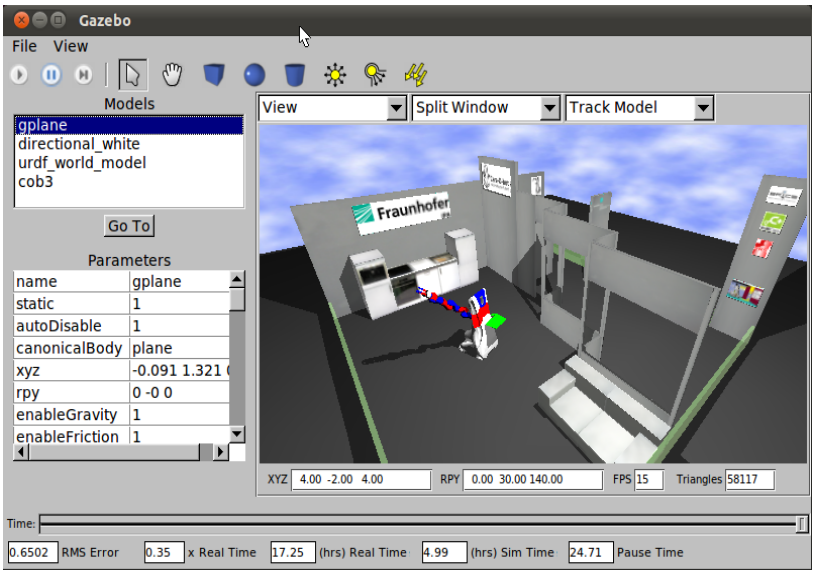

Figure 5: The simulation environment

The next two figures show the achieved result with the integrated components.

Figure 6: Snapshot from the augmented video with a highlighted recognized object

Figure 7: Snapshot from the augmented video with a highlighted recognized object that has been identified and the respective menu has been displayed

The achieved results in the simulation were satisfactory and multiple types of user interfaces running on different platforms (e.g. iOS, Linux, web based) were able to connect to the Mixed Reality Server.

7 The Rosbridge stack

The Rosbridge [8] [9] stack allows unified socket and web-socket [10] based access to the ROS via JSON [11] formatted messages. It uses single TCP [12] port (9090) for the communication which is very convenient for remote access as it can be easily forwarded via network firewalls.
The websockets support is an emerging feature in the new HTML5 [13] browsers which is also a plus for a future web-based version of the interface. Websockets are providing low latency communication path which is very important in such an application where real time response is critical.

The current implementation of the Rosbridge supports the following features:

- creating topics in ROS
- publishing messages to a topic
- subscribing and listening to a topic
- calling a ROS service
- authentication and security features
- launching a ROS launch file (experimental)

From the practical side it allows presence and interaction of the remote UI as a native ROS component using a single TCP port communication.
The data is encapsulated in JSON format which allows flexible ROS message structure representation and relatively low processor requirements to parse and generate.
From the ROS side this is done transparently by the Rosbridge. For the UI_PRI a custom parser was implemented.

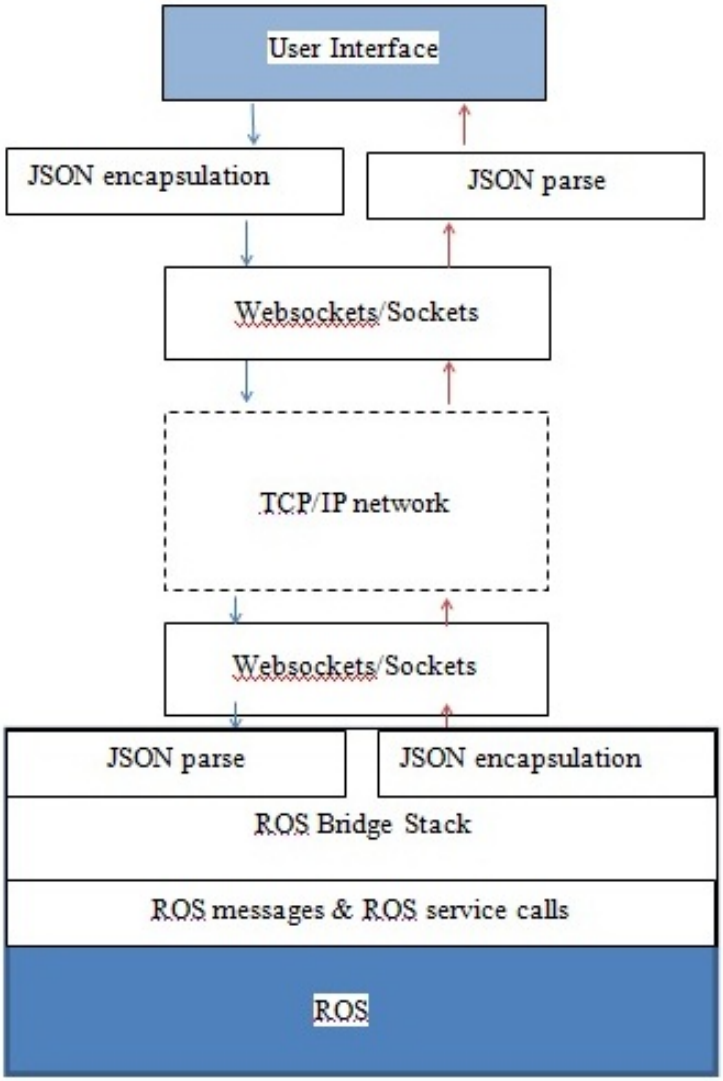

Figure 8: Communication data flow

On fig.8 are presented the dataflows between the remote interface and ROS via the Rosbridge (blue arrows) and backwards (red arrows):

8 Results of the practical realization of the Care-O-Bot to UI communication

8.1 General Structure

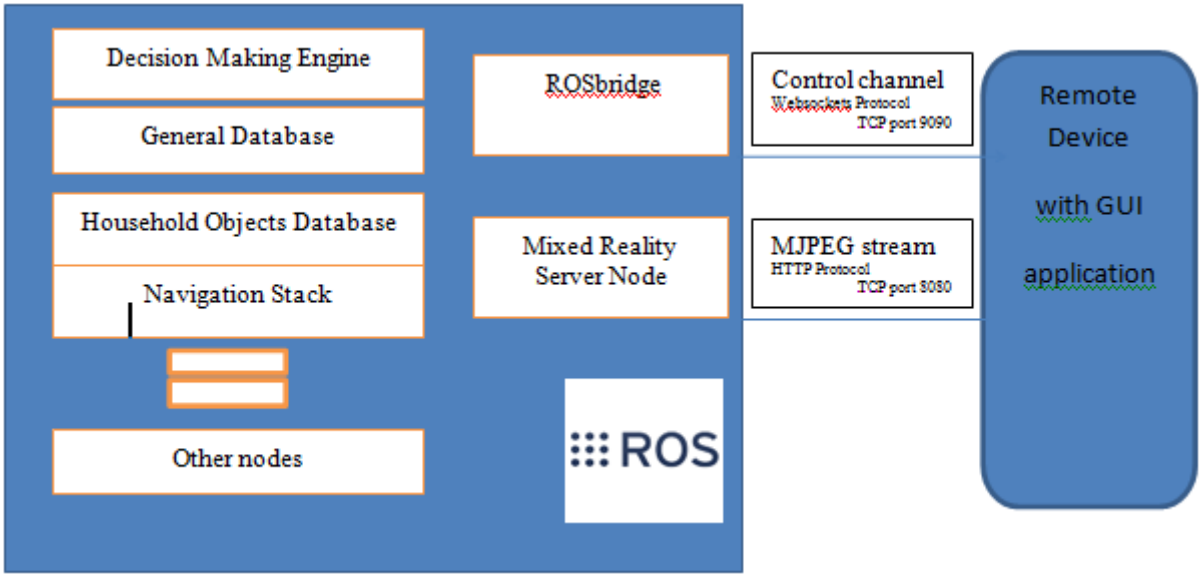

Figure 9: Interconnections between ROS nodes and user interface in the Care-O-Bot

Figure 9 represents relation between ROS nodes and the user interface via the rosbridge and the Mixed Reality Server.

8.2 Mapping and navigation data

Information about current map is fetched by the MRS from the map server node in topics:

/map

/map_metadata

and the position of the robot read from the (base_link) in /tf

8.3 Object information

Current objects data in range of the robot is a union between the information fetched from the General Database and Household Object Database.

The object data is read by the Control MRS component and then represented on the same map. The resulted frames are streamed to the User Interface by the Mixed Reality Server via standard HTTP protocol. The multithreaded approach allows simultaneous connection to any other image topic via the MRS, which will be discussed in the video feed section.

On figure 10 is presented an example of a real resulting map frame generated from the MRS. Merging of the data flows into one stream eliminates the synchronization problems. This approach offloads the portable device from complex calculation and processing of the tf.

The usage from various mobile devices is possible and also enables browser based multiplatform access. Also object list is fetched via Rosbridge in order to display the object page in the UI.

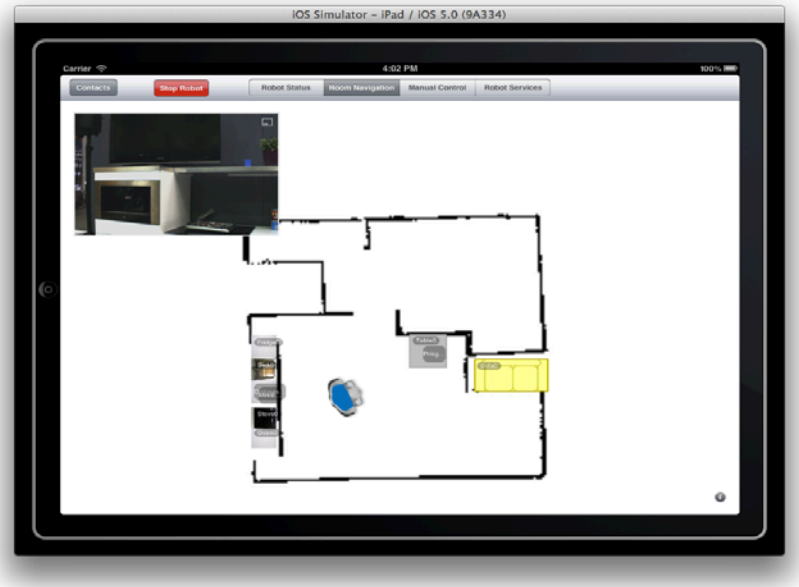

Figure 10: Example aggregated frame representing the map, the robot position and the detected objects

8.4 Robot Action Control and Feedback

The Care-O-Bot [14] is designed to execute tasks requested via a special node which acts as a Decision Making. The communication is realized as a native ROS topic communication via the Rosbridge stack. Task execution from the UI works via publishing to the topic /control_task_server/goal then the Decision making engine (DM) executes the desired task and writes a feedback result on /control_task_server/result.

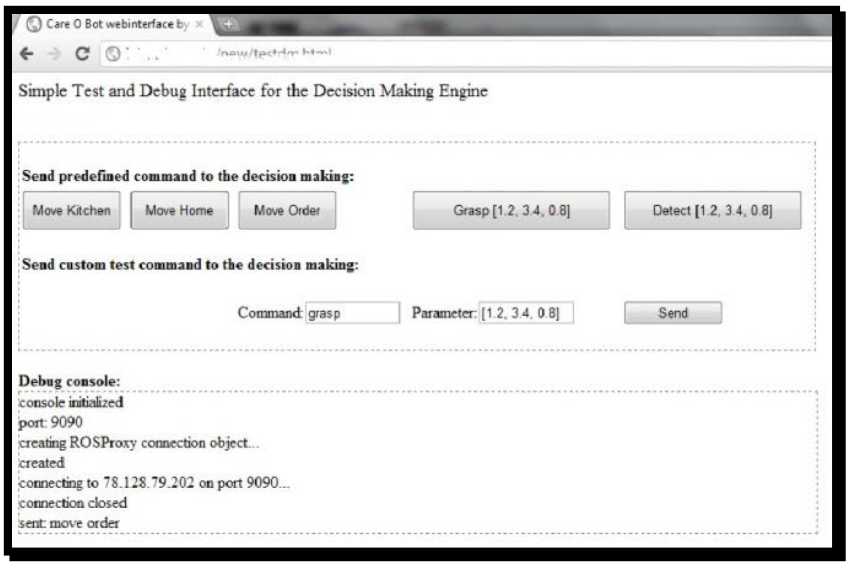

Figure 11: HTML5 Test interface for the communication and DM

In case of exception (e.g. the robot meets unavoidable obstacle) the DM can request user interaction by writing to /DM_UI/interface_cmd.
Also current task status can be monitored by the UI via /control_task_server/status topic.
The communication path and MRS were tested with native application on iPad – UI_PRI and also with a specially developed browser application – fig.11.

8.5 Other robot feedback sources

Using the Rosbridge the user interface monitors various other robot parameters as battery level, emergency statuses and etc. The data is fetch as a standard ROS topic communication.

8.6 Robot cameras Video transfer

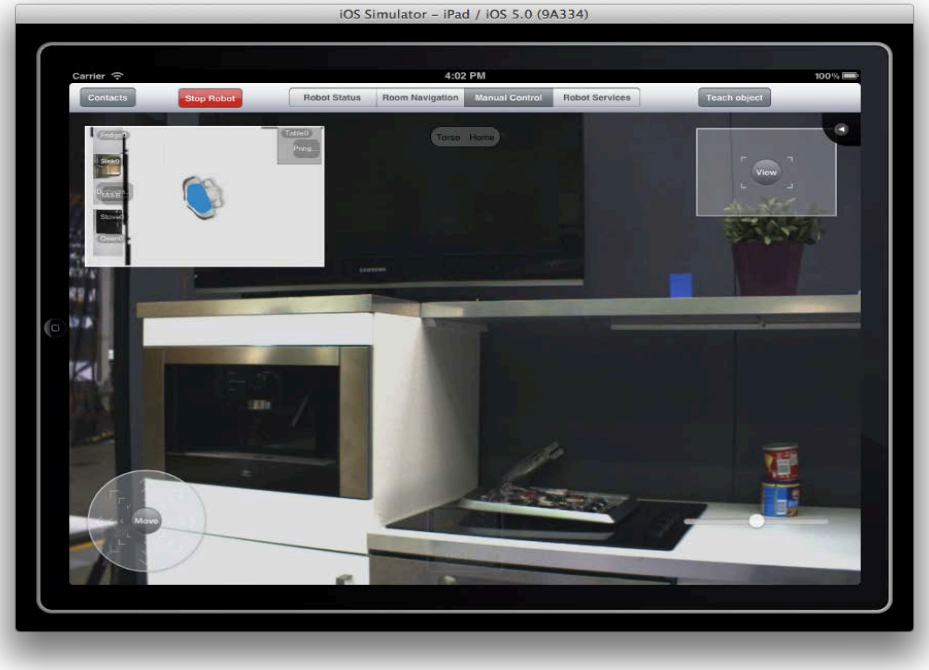

Figure 12: Video feed view in the UI streamed by the Mixed Reality Server

The important video feedback for the operator is streamed via the Mixed Reality Server – fig.12. The communication protocol is standard HTTP on port 8080. Any available image topic in the ROS can be compressed and streamed to the user interface on request by specifying of the source topic.
The MRS also generates an augmented reality stream which includes an overlay with the recognized objects.
Currently the format of output stream is JPEG/MJPEG which is supported by many systems. This format is more suitable by most of MPEG compressions because creates a sequence of consistent images and the image quality does not degrade by the motion.

9 Conclusion

The article describes the MRS and communication concept for a remote user interface for the Care-O-Bot service robot. The proposed architecture uses the Rosbridge stack and Mixed

Reality Server via standard TCP/IP networks. The process of augmenting video streams and identifying virtual objects was also explained. Results achieved with the real robot and using the simulation were discussed in relation to the positives and drawbacks of all used components and protocols. The future development of the system will be in implementation of the security user access authentication protocol, performance/latency improvements and 3D data transfer from the ROS servers to the user interface. Similarly, possible future development of the system is to perform object recognition on the video frames and automatically find the objects thus reducing the number of database queries. Another problem of interest is to visualize 3D objects and reconstruct the scene in 3D, which would allow the user to navigate in the 3D environment of the robot.

Acknowledgement

This research was carried out as part of the research project “Multi-Role Shadow Robotic System for Independent Living” (SRS), funded by the European Commission under the 7th Framework Programme, grant agreement no. 247772.

References:

[1] Robot Operating System
http://www.ros.org/wiki/
[2] Ubuntu Operating System
http://www.ubunto.com
[3] Linux Operating System
http://www.linux.org
[4] ROS Bridge
http://www.ros.org/wiki/rosbridge
[5] ROS MJPEG Server
http://www.ros.org/wiki/mjpeg_server
[6] ROS Image Transport
http://www.ros.org/wiki/image_transport
[7] OpenCV Reference
http://opencv.willowgarage.com/wiki/
[8] ROS Bridge, Brown University
http://www.ros.org/wiki/rosbridge
[9] ROS Bridge modified version, Matthew Donoughe
https://github.com/mdonoughe/rosbridge/wiki
[10] Fette, I, Melnikov, A., The WebSocket protocol, 2011, HyBi Working Group
http://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-17
[11] Crockford, D., The application/json Media Type for JavaScript Object Notation (JSON), 2006, IETF
http://www.ietf.org/rfc/rfc4627.txt
[12] Blank, A, TCP/IP Foundations, 2004,
[13] Lubbers,P., Pro HTML 5 Programming, 2010, Apress
[14] Robots Using ROS: Care-O-bot 3
http://www.ros.org/news/2010/03/robots-using-ros-care-o-bot-3-fraunhofer-ipa.html

ROBCO 11 - Intelligent Modular Service Mobile Robot for Elderly Care

Nayden Chivarov, Yasen Paunski, Georgi Angelov, Daniel Radev, Svetlin Penkov, Vladimir Vladimirov, Roman Zahariev, Maya Dimitrova Orlin Dimitrov, Vania Ivanova, Nedko Shivarov and Peter Kopacek*

Service Robotics Group of the
Institute of Systems Engineering and Robotics,
of the Bulgarian Academy of Sciences
Bloc II Acad. Bonchev str., Sofia 1113 Bulgaria
e-mail: nshivarov@code.bg
* Intelligent Handling and Robotics, Vienna University of Technology
Favoritenstrasse 9-11/E325A6, Vienna, Austria

Abstract:
The article focuses on the development, research and prototyping of the "Intelligent Modular Service Mobile Robot which will work in Elderly home for providing Elderly Care". Prognoses of the European Commission show that the tendency in Europe and especially in Bulgaria is of a continuously growing ageing of the population. Most of the elderly people want to live in their own houses for as long as possible and the proposed Intelligent Modular Service Mobile Robot can help them with the tasks such as "stand up" or "seat down", preparing or warming food, serve and clear the table, bringing water, books, medicines etc., it can fetch and carry difficult and heavy objects, video and audio programs selection contact with physician or with family members, day and night monitoring and fall preventing.

Keywords: *Service Robots, Mobile Robots, Modular Robots, Intelligent Robots*

1. Introduction

The key concept of the proposed Intelligent Modular Service Mobile Robot (Fig.1) is its easy adaptability in order to achieve services for a wide range of Elderly needs by performing different tasks for supporting Elderly care.

Fig.1 ROBCO 11 - Intelligent Modular Service Mobile Robot for Elderly Care

The Intelligent Modular Service Mobile Robot for Elderly Care consists of the following components:

- Intelligent onboard multilevel control system and ROS based software for the control of the different modules of the Intelligent Modular Service Mobile Robot;
- Modular remote control User Interfaces Teleoperation UI – Joystick based; Tactile UI – Touch screen based; Voice UI – Speech recognition based; Gesture UI – Gesture recognizing device; Brain control UI - Brain control devices;
- Electro-actuating systems, batteries and automatic recharging systems (docking stations) allowing 24 hours Service;
- Mobile Robot Platform - 4 wheels/2 powered wheels Intelligent fully autonomously vehicle with all its systems built on a modular principle able to carry peripheral systems or tools;
- Manipulating Articulated Robot Arm System including a proper gripper, which will allow different tasks to be performed like fetch an carry, bringing and manipulating difficult objects etc.;
- Sensor Systems – including Tactile sensors, Infrared sensors, Ultrasound Sensors, Artificial vision system and Voice generation and recognition systems, which help performing all functions related to environment and user interactions – monitoring of spatial location and orientation of the robot, maintaining proper course, obstacles detection, safety of the people and robot itself, object visualization and recognition, robot control and communication between a family member or physician and the Elderly for supporting the Robo-Care.

Movement of the robot in the range of service is one of the basic functions which determine the operational characteristics and abilities. Parameters as carrying capacity, positioning accuracy, speed, range of operation depends on area of applications of the robotic system.
A typical mobile platform consists of mechanic base, motors, gear, wheels, motor controller and power stage.
The control subsystem is very important for the operation and integration in the robot. The presented concept of mobile base control introduces an IP based approach, which allows easy integration in many environments including ROS (Robot Operating System).
The control board is based on modern 32-bit MCU equipped with many ready to use features in hardware and allows high level programming in C or C++.

2. Components

2.1. Drive systems

There are many construction designs for wheel based drive. Every of them has own positives and drawbacks so the choice should be based on application area of the system.
Here is presented a brief look of some of the used drive systems in robotics:

Differential (tank) drive – fig.2

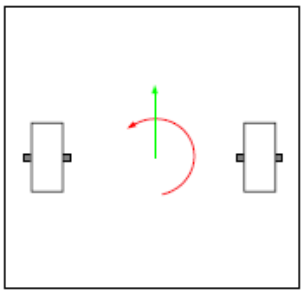

Fig. 2

Omnidirectional drive -fig.3

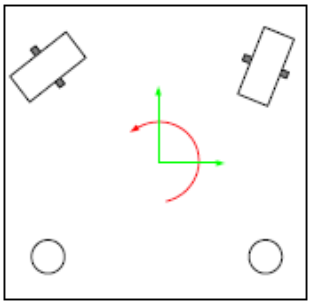

Fig.3

Mecanum Drive (Sweddish wheel) – fig.4

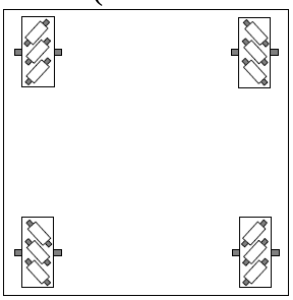

Fig.4

The presented control concept can be used in all types of the drive systems but for experimental purposes we have chosen a differential drive mobile base.

2.2. Motors

The choice of the driving motors depends on the construction and power requirements of the mobile platform.
The most commonly used are:

- Brushed DC motors
- Brushless motors
- AC induction motors
- Steeping motors
- Direct drive motors.

In our construction we use a brushed DC motors, which have good power characteristics and linear correlation between the applied current and motor torque, which makes the control more straight-forward and deterministic.

2.3. The control board

The control system – Pixeye NetIO (fig.5) is based on the modern 32 bit MCU.

Fig.5: The Net IO control board

The board features 6 PWM Outputs 10 Digital I/O and 4 ADC, 2 quadrature encoder inputs and a 100Mbps Ethernet interface.
The internal block diagram of the TI Stellaris LM3S6965 is presented on fig.6.

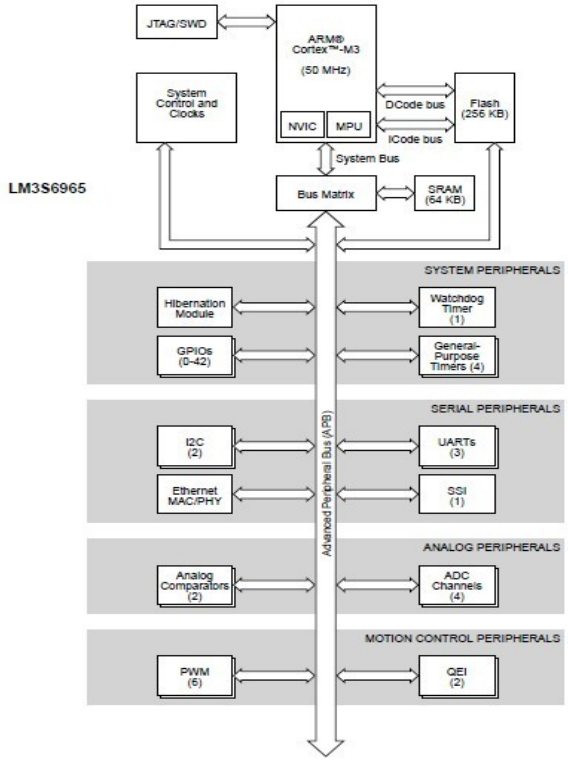

Fig.6: Internal structure of LM3S6965

For the current mobile platform drive two of the available PWM outputs are used and two of digital outputs as a motor direction switching.
All of the inputs and outputs of the board is accessible via the Ethernet interface from any IP based network.
The protocol is HTTP based on the API of the board, provided by the manufacturer.
The typical PWM output of the board is presented on fig.7**.**

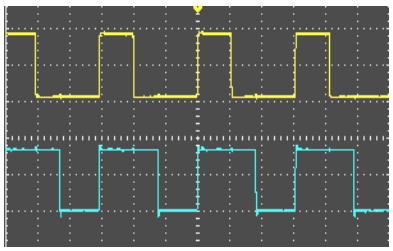

Fig.7 Typical PWM output

The power stage and the control of the DC motors is discussed in details in the next chapter. Two types of feedback are implemented using integrated features of the board:

- Motor current, which gives direct information of the torque sampled via the ADC of the board. The current is measured by current sensors integrated on the H-bridge boards.
- Motor coupled encoders, sampled by the quadrature inputs provided by the board. Readout of the encoders gives full details for the velocity and position.

The leftover I/O pins and ADC inputs can be used for various sensors.

2.3. The power stage

A very common circuit in robotics for driving DC motors is used between the control board and the motors. It's called H-bridge because looks like the capital letter 'H' when viewed on a discrete schematic. The H-Bridge is the link between digital circuitry and mechanical action.
The great ability of an H-bridge circuit is that the motor can be driven forward or backward at any speed, optionally using a completely independent power source.
On figures 8, 9 and 10 respectively are presented the current flows in a typical H-bridge when the motor is driven in forward and reverse direction.

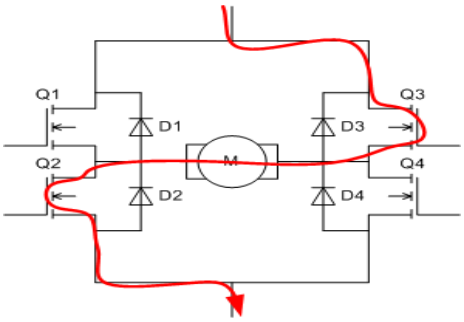

Fig.8: Current flow in forward direction

The switches are turned on in pairs, either high left and lower right, or lower left and high right, but never both switches on the same "side" of the bridge.
The speed control is done via changing of the duty cycle of the PWM from the control board.

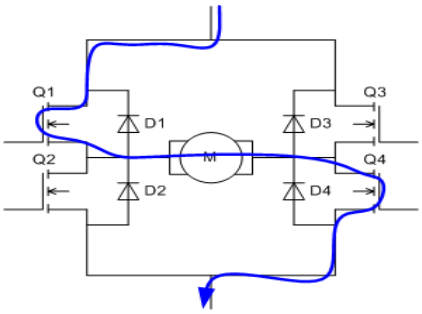

Fig. 9: Current flow in reverse direction

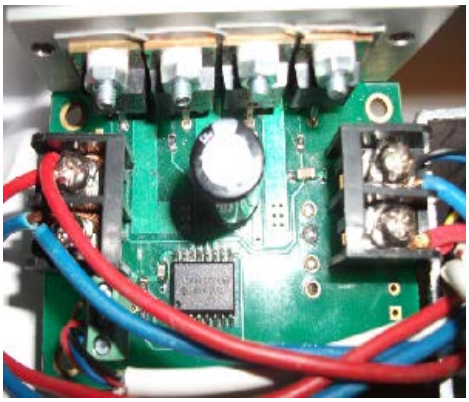

Fig.10. H-Bridge used in the system

2.4. System Structure

On figure 11 is presented the full structure of the control system. It consists of the control board, dual H-bridges, DC brushed motors, current sensors and optical relative encoders for feedback.

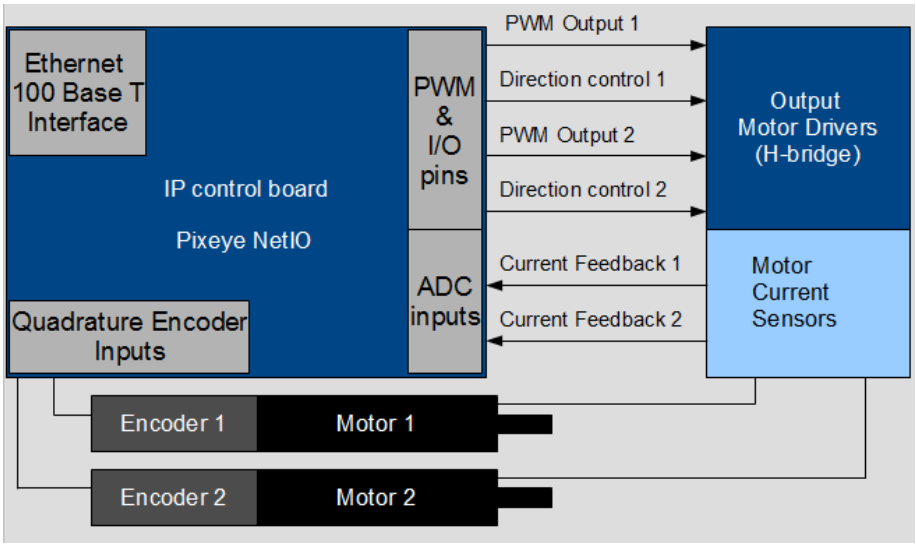

Fig.11: Control system for differential drive mobile platform

3. Intelligent Modular Service Mobile Robot Software Systems

Robot Operating System (ROS) is an open-source meta-operating system. It is a platform for applications in the field of robotics. For the moment Linux distribution Ubuntu is the operating system that supports fully ROS. That distribution has a great future related to porting and integration into embedded world. That gives ROS an excellent perspective for development and use of robot-related applications in systems varying greatly in scale and complexity. The following graphic is generated using the *rxgraph* tool of the ROS system. It displays the node architecture of the application (Figure 12):

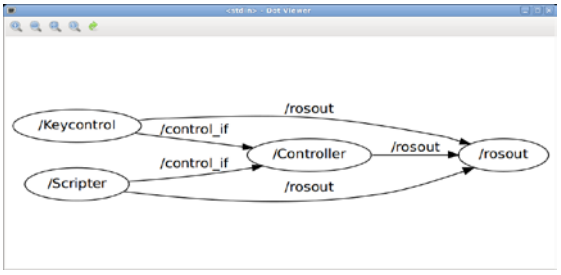

Figure 12. Node architecture of the application

A basic software is called *roscore* and it is a part of every standard ROS application. It can be started from a shell console using command *roscore*. The node *rosout* is a standard node for almost every application under ROS and it implements visualization of various system messages – information and error logs and other changes of the state of application. It takes messages from all of the rest of nodes that belong to the application and displays them in the console. The application itself consists of the nodes:

- *Controller*
- *Keycontrol*
- *Scripter.*

The *Controller* node (takes commands from the other two modules and implements their validation, formatting and transmitting through the serial link to the embedded controller of the robot. It is accomplished by registering for receiving messages on the topic *control_if.*
The other two nodes transmit commands for the manual and automatic mode of operation correspondingly.
The *Keycontrol* node reads commands from user input (terminal console), generates appropriate commands, formats them into messages and publishes them on *control_if* topic.
The *Scripter* node reads commands from a given script file (a text file, containing commands to the robot embedded controller), filters and formats them and publishes them on the topic *control_if.*
In this application the flexibility of the ROS platform is clearly visible. Such a system becomes very scalable and easily extendable. New nodes can easily be added, for example a node for receiving commands from a remote terminal and many others. All these nodes can publish on the same topic *control_if* without changing the Controller node, as there is no need for any node to be aware of the number and functions of the others that publish on the same topic.

4. Intelligent Modular Service Mobile Robot Sensor Systems

Sensor systems are essential part of every intelligent mobile robot. They help performing all functions related to the environment and user interactions – monitoring of spatial location and orientation of the robot, maintaining proper course, obstacles detection, safety of people and the robot itself. Depending on its primary function sensors can be divided into several groups:

- Tactile sensors

These sensors react on a physical collision with hard obstacles. They are usually attached to the mechanical bumpers surrounding the mobile robot. The most common implementation of these anti-collision sensors is by mechanical micro-switches, reacting on pressure. Other possible implementations will be considered, such as using materials for which the electrical resistance depends on mechanical pressure, as well as electromagnetic-based sensors.

- Proximity sensors

They react on presence of an object that is closely located but not colliding with the robot. Their implementation can be based on infrared transmitters/receivers, ultrasonic echolocators, laser distance meters, etc. In all cases the proximity of the object is evaluated by parameters of the echoed signal. The effectiveness of different proximity sensor types will be evaluated and compared with respect to detecting different types of objects, possibilities of precise distance measurement, limitations of range and viewing angle and other parameters.

- Acceleration sensors

They can be used both for tracking velocity changes of the mobile robot and for detecting the direction of the earth gravity (when the robot is not moving) thus detecting the spatial orientation of the robot itself. *Different types of accelerometers will be examined* and their parameters will be evaluated – resolution, number of axes etc.

- Vision system

The growing demands on improved interactivity of the robot as well as performing different types of specific tasks lead to the necessity of introducing robot vision capabilities. The possibilities of application of commonly used cameras today will be considered, as well as usage of special cameras with higher speed and/or resolution and other relevant parameters. Today's trends in the field of robot vision will be analyzed such as *color* spots discovery, contour extraction, object recognition using different types of classifiers (i.e. ANN - Artificial Neural Networks and others).

- Voice interaction system

The need for verbal interaction with the mobile robot is none less necessary than Robot Vision for improving interactivity. This feature consists of two different types of functions – speech recognition and speech synthesis. The task for speech synthesis has always been basically simpler but the challenge today is synthesis of natural human speech with proper intonation, pauses, and other parameters of natural human speech, that leads to significantly improved interaction between the robot and users. For the task of speech recognition different advanced systems will be considered and their parameters – evaluated.

5. Intelligent Modular Service Mobile Robot Control System

The Intelligent Modular Service Mobile Robot for Elderly care has three main modes:

- Manual Mode for manual control of the robot - Depending from the necessity of the Elderly, the Intelligent Modular Service Mobile Robot will be controlled either by Joystick – Teleoperation UI; Touch screen - Tactile UI; Speech recognition - Voice UI; Gesture recognition- Gesture UI; Brain control - Brain control UI.;
- Semi-Autonomous mode use Control System by executing high level tasks like warm food or bring glass of water;
- Autonomous mode use Control System for Elderly day and night monitoring, fall preventing and for emergency cases.

The Intelligent Modular Service Mobile Robot will have the ability to learn from its own experience (already executed tasks) as well as to interpret human behaviors

Since the Intelligent Modular Service Mobile Robot will stay in the Elderly home and will perform a real interaction with the Elderly, the robot motion must ensure Elderly and robot's safety.

6. Conclusion

Prognoses of the European Commission show that the tendency in Europe and especially in Bulgaria is continuously growing ageing of the population. Most of the elderly people want to live in their own houses for as long as possible and the proposed Intelligent Modular Service Mobile Robot for Elderly Care can help them with tasks such as stand up or seat down assistance, preparing or warming food, serve and clear table, bring water, book, medicine etc., fetch and carry difficult and heavy objects, video and audio contact with medical physician or with family members, day and night monitoring and fall preventing.

Since Bulgaria is on the one of the first places in Europe for the number of disabled persons (According to the Bulgarian Government Statistics), the proposed Intelligent Modular Service Mobile Robot will be able even to provide Care for disabled persons as well.

ROBCO 11 can „live" at the home of the disabled person and helps him throughout the day. Robot will be able to remind him to take medications, it will serve food and drinks, will turn on electronic devices, will alert when his health is getting worse and will connect with his physician, relatives or in emergency ambulance.

References:

N. Chivarov, I. Genchev, N. Shivarov, R. Zahariev, D. Radev, V. Vladimirov, Remotely Controlled Articulated Robot "ROBCO" under ROS/ UBUNTU, Proc. 20th Int. Conf. on Robotics and Mechatronics "SRS" Invited Session, 06-09 October 2010, Varna Bulgaria, pp. 7 – 12.

V. Vladimirov, N. Chivarov, D. Radev, I. Genchev, N. Shivarov, Technology Research on Implementation Scenarios for the Remote User Interface of the Multi-Role Shadow Robotic System for Independent Living, Proc. 20th Int. Conf. on Robotics and Mechatronics, "SRS" Invited Session, 06-09 October 2010, Varna Bulgaria pp. 13-19, ISSN 1310-3946

Dragoiceaand M. and Shivarov N., Assistive Mobile Robot Technology for Real-Time Task Implementation, RAAD 2009, May 25-27, 2009, Brasov, Romania, ISBN 978-606-521-315-9

Graf, Birgit: Dependability of Mobile Robots in Direct Interaction with Humans. In: Prassler, Erwin (Ed.) u.a.: Advances in Human-Robot Interaction. Berlin u.a.: Springer, 2005, S. 223-239 (Springer Tracts in Advanced Robotics - STAR 14).

Forlizzi, J. (2005). Robotic Products to assist the aging population. Interactions. March & April, 16-18.

Forlizzi, J., DiSalvo, C., and Gemperle, F. (2004). Assistive Robotics and an Ecology of Elders Living Independently in Their Homes. Human-Computer Interaction, 19, pp. 25-59

Graf, Birgit; Hans, Matthias; Schraft, Rolf Dieter: Care-O-bot II - Development of a Next Generation Robotic Home Assistant. In: Autonomous Robots 16 (2004), Nr. 2, S. 193-205

Frank G Miskelly “Assistive technology in elderly care” Age and Ageing 2001,30:455-458

Buchbesprechung

Fundamentals of Robotics
Linking Perception to Action

Ming Xie

World Scientific Publishing (UK)
692 Seiten, 2003, Reprint 2008
ISBN: 13-978-981-238-335-8

Das vorliegende, in der Serie „ Machine Perception and Artificial Intelligence“ erschienene Buch gibt eine erste Einführung in die Robotertechnik. Es beginnt mit einer Beschreibung von Robotern einschließlich möglicher Anwendungen in der Fertigungsautomatisierung sowie einen Ausblick über den Inhalt.

Ausgehend von der Robotermechanik beginnend mit der Festkörperdynamik über die Roboterkinematik bis hin zur diskreten Kinematik werden die Grundlagen erläutert. Basierend auf diesen erfolgt eine Beschreibung der elektrischen Roboterantriebe und der Roboterdynamik in klassischer Weise mittels der Gleichungen von Newton-Euler sowie Euler-Lagrange. Die Regelung der Roboter umfasst nach einer allgemeinen Einführung eine Zusammenstellung der erforderlichen Baugruppen (Antriebe, Sensoren, …). Der Abschnitt über den Reglerentwurf ist eher klassisch abgefasst. Ein weiteres Kapitel beschäftigt sich mit Hard- und Software von Robotern, wobei relativ ausführlich die Hardware Beachtung findet.

Den Hauptteil des Buches bilden die folgenden Kapitel in denen ausführlich auf visuelle Sensoren und das Wahrnehmungssystem (Perception System) als Basis für „Roboterintelligenz“ eingegangen wird. Den Abschluss bildet ein Kapitel über Entscheidungsfindung (Decision Making) bei Robotern.

Dieses Lehrbuch beschreibt die elementaren Grundlagen der Robotertechnik, wobei am Ende jedes Kapitels neben einer Zusammenfassung auch Übungsbeispiele sowie Literaturhinweise angegeben sind. Es kann daher den Studierenden sowie auch Praktikern als erste Einführung in die Robotertechnik empfohlen werden.

P.Kopacek

Instruction to authors – presented as a pattern paper (18 pt)

A. Maier, F. Huber (12 pt)
Department ....., Vienna, Austria

Received April 8, 1999

Abstract
This paper shows …… (italics, 12 pt)

1 General (14 pt)

Authors should prepare their manuscript camera ready, format A 4, 12 typeface and must present their manuscript in good quality. At the left/right edge 2.5 cm, at the to/bottom edge 3 cm. Authors are invited to use papers of this journal as a sample. Please do not use an eraser or erasing fluid. Footnotes should be avoided if possible. Authors are expected to submit their paper to one of the publishers (O.Univ.Prof.Dr. Peter Kopacek, Intelligent Handling Devices and Robotics, Vienna University of Technology, Favoritenstrasse 9-11, A-1040 Vienna, Austria, Fax: +43 1 58801-31899 or O.Univ.Prof.Dr. Alexander Weinmann, Institute of Automation and Control, Vienna University of Technology, Gusshausstr. 27-29, A-1040 Vienna, Austria, Fax: +43 1 58801 37699).

Email address for submitting pdf-manuscripts: weinmann@acin.tuwien.ac.at

2 References (14 pt)

Within the paper references should appear in the following form:
(Mayer, H., 1990) or (*Mayer, H., 1990)* (12 pt);
Mayer, H., 1990, discovered that....

3 Figures and Tables (14 pt)

Figures and Tables should be integrated in the paper and have to be referred to as Fig. 4.1 or Tab. 5.2.

4 References

References are to be listed alphabetically according to first author. (11 pt)

5 Word Processing System/Editor

Microsoft Word 2000 or higher; TeX or LaTeX.

Wenn unzustellbar, retour an:

IFAC-Beirat Österreich (E318 / E376)
Favoritenstraße 9-11, A-1040 Wien /
Gußhausstraße 27-29, A-1040 Wien