DESIGN AND IMPLEMENTATION OF ROBUST CONTROL LAWS

P. Petkov, J. Kralev and Ts. Slavov
Department of Systems and Control
Technical University of Sofia
1756, Sofia, Bulgaria
Email: php@tu-sofia.bg

KEYWORDS

Robust control; Real time control; Embedded systems; Digital Signal Processors

ABSTRACT

This paper is devoted to various issues related to the design and practical implementation of high order robust control laws. We consider derivation of plant uncertainty models using analytical or identification procedures, implementation of different schemes for μ-synthesis, choice of weighting filters and controller order reduction. Additional important problems arising in the framework of embedded control systems, like removing the sensor drifts, generation of control code from Simulink® and effect of single precision arithmetic on the controller stability, are discussed in some details. As a case study we present the robust control of two-wheeled robot using μ-controller of order 30. The experimental results confirm that the closed-loop system achieves both robust stability and robust performance in respect to the uncertainties related to the identification of robot model.

INTRODUCTION

The Robust Control Theory involves powerful methods for analysis and design of control systems in presence of signal and parameter uncertainties (Sánchez-Peña and Sznaier 1998; Skogestad and Postlethwaite 2005; Zhou et al. 1996). The most frequently used techniques for robust control design are the $\mathscr{H}_\infty$ design and the μ–synthesis (Skogestad and Postlethwaite 2005). The $\mathscr{H}_\infty$ optimization is usually preferred in robust design because it produces controllers of smaller order which facilitates their implementation. The common disadvantage of all $\mathscr{H}_\infty$ design methods is that they are suitable for plants with unstructured uncertainties but can not ensure robust stability and robust performance in the general case of unstructured and structured (parametric) uncertainties. In contrast, the μ–synthesis which aims at minimization of the structured singular value (Zhou et al. 1996) may ensure robust stability and robust performance in the presence of exogenous disturbances, noises and different type of uncertainties. The high order of the controller obtained is usually pointed out as a disadvantage of μ–synthesis. However, with the appearing of powerful and cheap processors in the recent years this peculiarity of the μ–synthesis does not pose a significant difficulty.

In contrast with the theoretical achievements, the practical implementation of robust control laws is still in its beginning. There is a few real life applications of high order robust control laws reported in the literature (see for instance (Bautista-Quintero and Pont 2008; Howlader et al. 2013; Raafat et al. 2012). The main obstacle of robust control laws implementation is the difficulties related to the development, testing and verification of the necessary real-time software which is highly dependent on the type of digital controller platform used. These difficulties are reduced significantly using the recent technologies for automatic code generation and embedding implemented in MATLAB®/Simulink®program environment (Simulink Coder 2013).

This paper is devoted to various issues related to the design and practical implementation of high order robust control laws. We consider derivation of plant uncertainty models using analytical or identification procedures, implementation of different schemes for μ-synthesis, choice of weighting filters and controller order reduction. Additional important problems arising in the framework of embedded control systems, like removing the sensor drifts, generation of control code from Simulink® and effect of single precision arithmetic on the controller stability, are discussed in some details. As a case study we present the robust control of two-wheeled robot using μ-controllers of order up to 30.

The paper is organized as follows. First, we consider the derivation of uncertain plant models using analytical models or identification procedures. We present the difficulties related to the identification of uncertain models by an example. In the next Section we give a brief overview of Robust Control Theory based on the usage of structured singular value μ. Several implementation aspects of μ-synthesis are discussed including weighting filters selection and D-K-iterations convergence. The third Section is devoted to the implementation of robust controllers in embedded control systems. The issues discussed involve removing the sensor drifts and effect of single precision arithmetic on the controller stability. In the final Section we present as a case study the design and implementation of robust controller intended for stabilization of two-wheeled self-balancing robot. The experimental results

Proceedings 29th European Conference on Modelling and Simulation ©ECMS Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova (Editors)
ISBN: 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD)

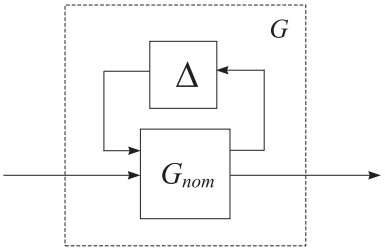

Fig. 1. Uncertain plant

confirm that the closed-loop system achieves both robust stability and robust performance in respect to the uncertainties related to the identification of robot model.

DERIVATION OF UNCERTAIN PLANT MODELS

As noted in (Bittanti and Garatti 2012), control science is basically a model based discipline and the performance of control is determined by the accuracy of the model representing data. Building uncertain system models is an important step in the design of robust control systems. Unfortunately, the derivation of uncertain plant models may be much more difficult in comparison with modeling of plants with negligible uncertainties.

Implementation of analytical models

In practice, it is preferable to have an analytical nonlinear plant dynamics model which is used in the derivation of linearized uncertainty model. The analytical models allow to obtain easily both structured (parametric) and unstructured (complex) uncertainty models. Also, the analytical description may be used to obtain the so-called *parameter-dependent model* (Gahinet et al. 95, Ch. 2) which depends on parameters that may undergo large variations along the time.

Implementing the nonlinear model in Simulink®, it is possible to use the corresponding functions of Robust Control Toolbox®3 to linearize the plant and extract an uncertainty model represented in Figure 1. The linearized uncertain plant model G is in the form of Linear Fractional Transformation (LFT), where G_{nom} is the transfer function matrix of the nominal model and Δ is a block-diagonal uncertainty matrix. The Robust Control Toolbox®3 functions also allow to obtain the time and frequency response characteristics of the uncertainty plant models.

Implementation of identification procedures

Frequently, a reliable analytical model of the plant dynamics is not available and the system designer is faced with the necessity to identify the model from experimental data. As it is well known (Isermann and Münchhof 2011; Landau and Zito 2006; Ljung 1987) identification from noisy data may represent a difficult task especially in case of unstable plant when the measurements are obtained in closed-loop.

The following example illustrates the difficulties in deriving uncertainty model of an unstable plant using the functions of System Identification Toolbox®3 .

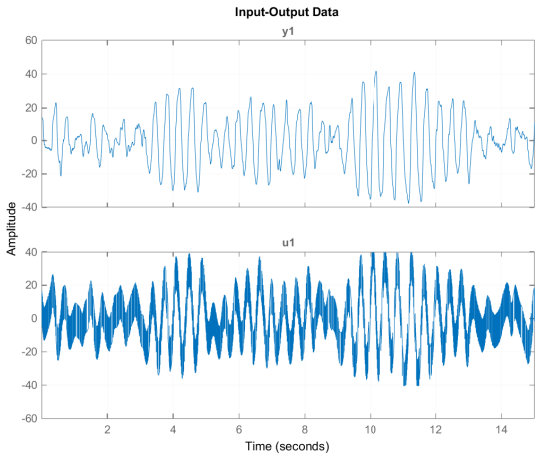

Fig. 2. Input-output data used in the identification

The input-output data obtained by closed-loop experiment for a single-input, single-output system is shown in Figure 2. It consists of 3000 measurements with sampling period of 0.005 s, additional 200 data samples being used for model validation. Based on the three criteria - model loss function, Akaike information index and Rissanen index - a fourth order Box-Jenkins model is chosen for further identification. It is done by the function `bj` using data sets of 1000, 2000 and 3000 data samples. In all three cases the models obtained pass through the test of autocorrelation and crosscorrelation function of model residuals, which indicates unbiased parameter estimates. The latter condition guarantees that the exact parameter values are contained in the confidence intervals of estimates with probability close to 1. Based on confidence intervals, maximum relative deviations from nominal models are obtained, which may be used to derive models with unstructured uncertainty. In Figures 3 - 8 we show bounds on the relative uncertainties derived for the three models along with the Bode plots of the corresponding uncertain plant models.

It is seen from the Figures that the picks of the relative uncertainties derived vary between 1.3 for the case of 2000 samples and 4.3 for the case of 1000 samples. Clearly, the uncertainty model obtained for 1000 data samples is not appropriate for controller design. So different results for the uncertainty bounds show that additional optimization is necessary in order to obtain a model with minimum of uncertainty bound. The alternative is to use special identification methods for uncertainty models, see for instance (Gugercin et al. 2003; Van den Hof 2001; Venkatesh 2003).

ROBUST CONTROL DESIGN

Consider a control problem in the Linear Fractional Transformation shown in Figure 9.

The system denoted by P is the open-loop connection and represents all known elements including the nominal system model and the performance weighting functions, as well as the

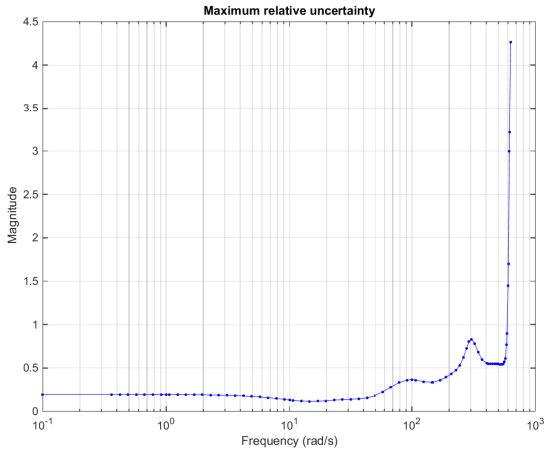

Fig. 3. Relative uncertainty for 1000 samples

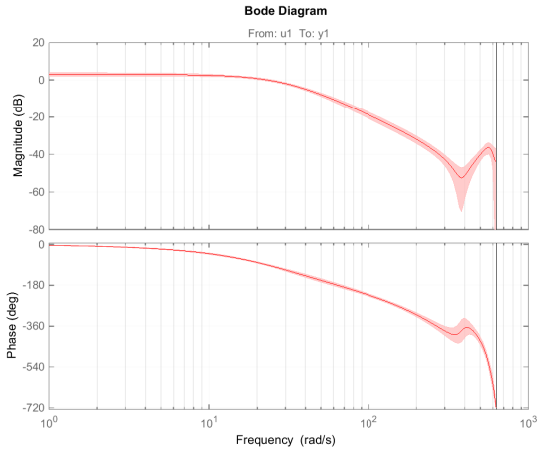

Fig. 6. Uncertain model for 2000 samples

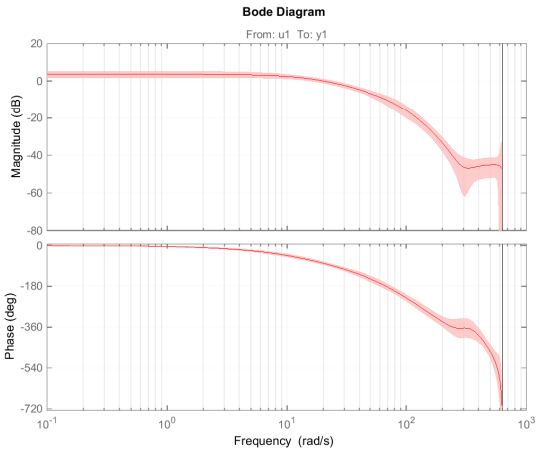

Fig. 4. Uncertain model for 1000 samples

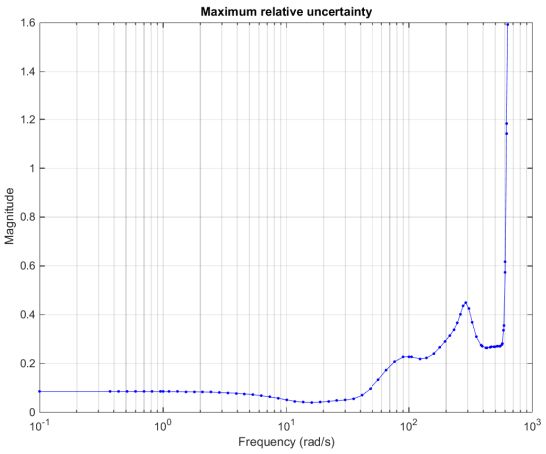

Fig. 7. Relative uncertainty for 3000 samples

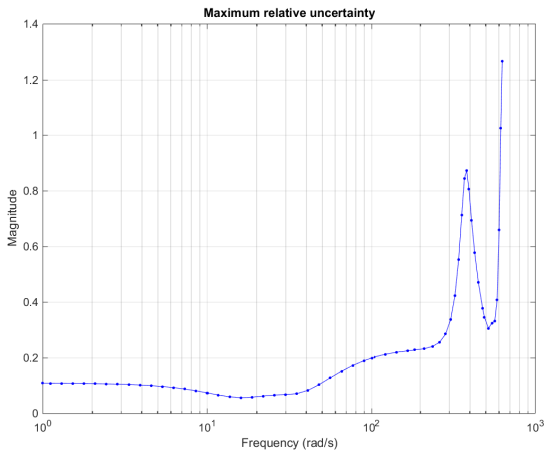

Fig. 5. Relative uncertainty for 2000 samples

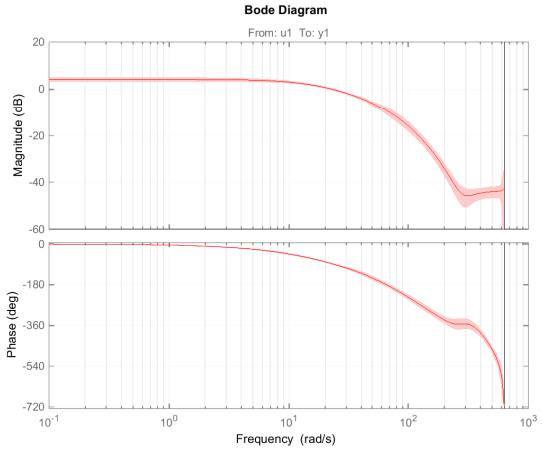

Fig. 8. Uncertain model for 3000 samples

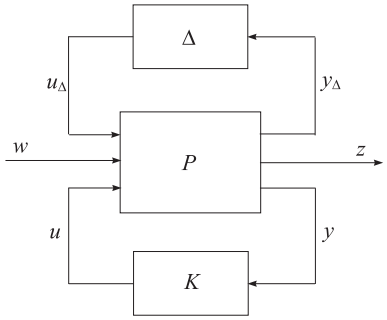

Fig. 9. Uncertain closed-loop system

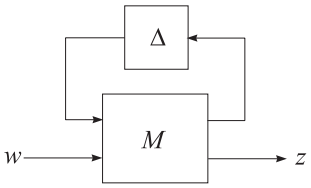

Fig. 10. $M-\Delta$ loop for robust stability analysis

uncertainty weighting functions. The block Δ parameterizes all supposed model uncertainty. The controller is denoted by K. Inputs to P are three sets of signals: inputs u_Δ due to the uncertainty, references and disturbances w and controls u. Three sets of outputs are generated: outputs y_Δ due to the uncertainty, errors or controlled outputs z and measurements y.

For the aim of robust stability analysis, it is convenient to represent the uncertain control system by the $M-\Delta$ loop, shown in Figure 10. In this loop the transfer function of the nominal part (denoted by M) is separated from uncertain part (denoted by Δ). We have that $M = F_\ell(P,K)$ where $F_\ell(P,K)$ denotes the transfer function matrix of the lower Linear Fractional Transformation of P and K. It is possible to prove that the system shown in Figure 10 will achieve robust stability for all Δ if and only if

$$\mu_\Delta[M(j\omega)] < 1 \tag{1}$$

where $\mu_\Delta(F_\ell(P,K))$ is the *structured singular value* of the closed-loop system (Zhou et al., 1996). The design goal is to determine a controller K, stabilizing the nominal system as well as for all $\Delta, \max_\omega \bar{\sigma}[\Delta(j\omega)] \leq 1$. The closed-loop system achieves robust performance if the system is stable and satisfies the performance criterion

$$\|F_U[M,\Delta]\|_\infty < 1 \tag{2}$$

where $F_U(M,\Delta)$ is the transfer function matrix of the upper LFT of M and Δ. For given arbitrary K this criterion may be tested by using the robust performance test on the nominal part $M = F_\ell(P,K)$. The robust performance test should be performed in respect to the extended uncertain structure

$$\Delta_P \stackrel{\text{def}}{=} \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta_F \in \mathbb{C}^{n_w \times n_z} \right\}, \tag{3}$$

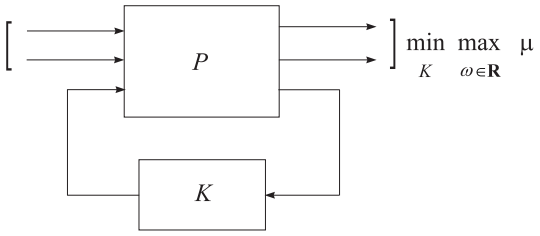

Fig. 11. Block-diagram of the closed-loop system

where Δ_F is a fictitious complex (unstructured) uncertainty. The system with controller K achieves robust performance if and only if

$$\mu_{\Delta_P}(F_\ell(P,K)(j\omega)) < 1 \tag{4}$$

where $\mu_{\Delta_P}(F_\ell(P,K)$ is structured singular value determined with respect to the extended uncertainty Δ_P.

In the case of $\mathscr{H}_\infty$-design we look for a controller which ensures

$$\|M(s)\|_\infty < \gamma \tag{5}$$

for a small positive number γ. Note that the system uncertainty is not taken into account in the $\mathscr{H}_\infty$-design.

The main disadvantage of the $\mathscr{H}_\infty$-design is that it does not allways ensure robust stability and robust performance of the closed-loop system in presence of uncertainty in the plant G.

μ-synthesis problem

The μ–synthesis is one of the most important techniques in Robust Control Design. For properly chosen weighting functions this design method usually produces a controller that ensures both robust stability and robust performance of the closed-loop system.

Consider a control problem shown in Figure 11.

The aim of the *μ–synthesis* is to minimize the peak value of the structured singular value $\mu_{\Delta_P}(.)$ of the closed-loop transfer function matrix $F_\ell(P,K)$ over the set of all stabilizing controllers K. This is written as

$$\min_{\substack{K \\ \text{stabilizing}}} \max_\omega \mu_{\Delta_P}(F_\ell(P,K)(j\omega)). \tag{6}$$

The μ-synthesis has the following features.

1) Good disturbance attenuation and noise suppression.
2) Achieves stability and performance robustness.
3) Doesn't require accurate plant model.
4) Difficult tuning of the weighting functions.
5) The controller order is higher than the order of the $\mathscr{H}_\infty$ controller.

Consider how to set the performance requirements in the μ-synthesis.

The design goal is to achieve robust stability and robust performance of the closed-loop system shown in Figure 12 in the presence of plant uncertainty and output disturbances. The matrix transfer functions W_p and W_u are frequency dependent weighting functions (filters) that will be called weighting

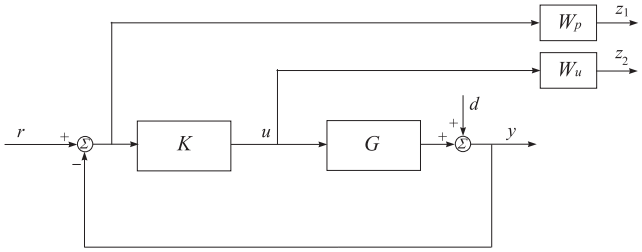

Fig. 12. Block-diagram of the closed-loop system

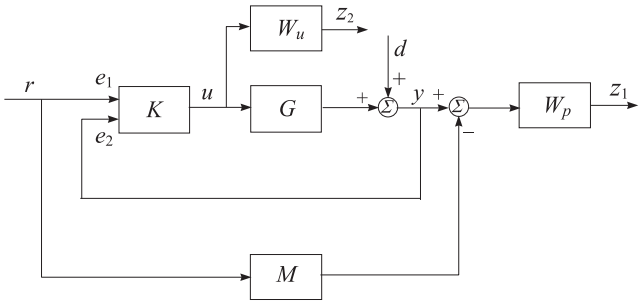

Fig. 13. Design of two-degree-of-freedom controller

performance function and weighting control function, respectively. The role of the function W_p is to "penalize" the difference between system output y and reference r in desired low frequency range and the role of the function W_u is to limit the magnitude of control action u.

The closed-loop system is described by

$$z = T_{zw} w, \tag{7}$$

where

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \; w = \begin{bmatrix} r \\ d \end{bmatrix},$$

$$T_{zw} = \begin{bmatrix} W_p S_o & -W_p S_o \\ W_u K S_o & -W_u K S_o \end{bmatrix},$$

and

$$S_o = (I - GK)^{-1}$$

is the output sensitivity transfer function matrix.

To achieve closed-loop robust performance, i.e.

$$\mu_{\Delta_P}(T_{zw}(j\omega)) < 1, \tag{8}$$

means that the condition

$$\|T_{zw}\|_\infty < 1 \tag{9}$$

will be fulfilled for each possible plant uncertainty. This, in turn, guarantees fulfillment of the conditions

$$\|W_p S_o\|_\infty < 1, \;\; \|W_u K S_o\|_\infty < 1. \tag{10}$$

Figure 13 illustrates the implementation of the so called *two-degree-of-freedom controller*. The goal in this case is to achieve closeness of the closed-loop system behavior to the model for all possible plant uncertainties. The usage of two-degree-of-freedom controller usually lead to better design results in comparison with the usage of one-degree-of-freedom controller.

Choice of weighting filters

The performance weighting transfer functions W_p are chosen as low pass filters to suppress the output disturbance d, and the control weighting transfer functions W_u are chosen as high pass filters with appropriate bandwidth in order to impose constraints on the high frequency spectrum of the control actions [11]. This is based on the assumption that the disturbance spectrum is in the low-frequency range and the noise spectrum is in the high frequency range. In some cases it is necessary to modify the weighting functions many (hundred and more) times in order to achieve the desired closed-loop behavior which is one of the few disadvantages of the μ-synthesis. The choice of weighting functions may be done by implementing optimization procedures.

Implementation of D-K iterations

The μ synthesis is done via the so called *D-K* iterations (Zhou et al. 1996) which represent an approximate method for finding the minimum of pick value of μ. These iterations are realized in Robust Control Toolbox®3 by the function `dksyn`(Balas et al. 2013). The convergence of *D-K* iterations is not guaranteed and in some cases one may observe very poor iteration behavior or even divergence. In such cases it is possible to make slight changes in the performance weighting functions or in the model parameters in order to improve the convergence. Also, the inclusion of small noises in some system inputs may have very good impact on the *D-K* iterations.

Below we give some recommendations about how to do in practice the μ-synthesis.

1) Due to the efforts necessary to derive the uncertain plant model and the almost unavoidable complication of the controller designed it is appropriate to begin with simplified uncertainty description in order to see wether the performance requirements can be met. Only in the case when these requirements are satisfied, it is appropriate to consider more complicated uncertainty descriptions including, for instance, parametric uncertainties to "sharpen" the design with more accurate uncertainty model.
2) The usage of μ means worst-case analysis so one should be cautious when introducing many sources of uncertainties, disturbances and noises. In such a case it becomes less and less possible for the worst case to appear and the analysis and design performed may become unnecessarily conservative.
3) There is always uncertainty in respect to inputs and outputs so that it is reasonable in the general case to include diagonal input and output uncertainties. The relative (multiplicative) uncertainty is very appropriate for this aim.

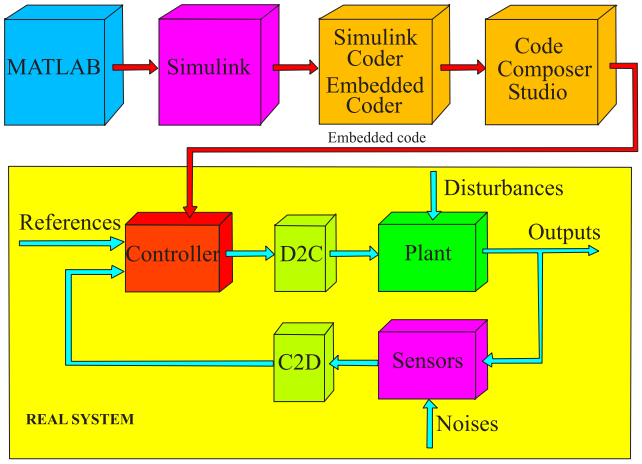

Fig. 14. Block-diagram of an embedded control system

4) In the practical design it is customary to obtain values of μ that exceed one. This may result from very high requirements to the closed-loop performance which are impossible to satisfy for the given plant. In such a case it is necessary to loosen the requirements setting other performance and/or control action weighting functions. Finding a controller that ensures value of μ less than one means that the requirements set are possible to achieve. In other cases it may be necessary to design controller with different structure, for instance two-degree-of-freedom controller, in order to fulfil the requirements posed.
5) In case of a discrete-time system it is appropriate to perform the controller design initially in the continuous-time case. This is due to the fact that the best possible performance may be obtained in the continuous-time case which can then be considered as a limit for discrete-time designs. Also, in the continuous-time case it is easier to find appropriate performance weighting functions that again may be implemented in the discrete-time design.

CONTROLLER IMPLEMENTATION

Embedded control systems

Implementation of robust controllers in real-time is done by *embedded control systems* which are characterized by the fact that the control software is stored in read-only memory (Hristu-Varsakelis and Levine 2005, Part III). The process of embedding the control software by using MATLAB®and Simulink®is illustrated in Figure 14. The robust control law is designed in MATLAB®and its Simulink®model is translated to the embedded processor using the automatic generation code tools Simulink Coder®(Simulink Coder 2013) and Embedded Coder®(Embedded Coder 2013). The embedded systems used for motion control usually implement Micro-electromechanical systems (MEMS) sensors.

Integration of hardware and software in embedded control systems proceeds in three phases: software-in-the-loop, processor-in-the-loop and hardware-in-the-loop. In the first phase the hardware is represented entirely by Simulink®models. The control code is generated automatically from Simulink®model and is validated off-line by simulation of the closed-loop system. At this stage it is possible to use Monte Carlo simulation which can take into account parametric variations and nonlinear effects. In the second phase the control code is tested in the real-time embedded processor using hardware simulated by Simulink®models. Finally, in the third phase the control code is tested with the prototyped hardware in order to verify the integrated functional and operational performance in real environment (strong disturbances, noises and parametric variations). This technique makes possible to examine rapidly different control laws reducing in the same time the danger of accidents during real experiments.

The embedded control systems have the following distinguishing features.

1) Signals are presented as fixed-point integers or single precision numbers. Floating-point arithmetic may be emulated or implemented by integrated hardware floating-point unit.
2) The controller model should be modified to include I/O interfaces with the external devices.
3) The MEMS sensor noises may be intensive in some frequency ranges.

Analysis of embedded control systems requires the usage of methods pertaining to the theory of hybrid control systems (Lunze and Lamnabhi-Lagarrigue 2012).

Removing the sensor drift

The MEMS sensor errors consist of deterministic and stochastic parts. The deterministic part includes constant biases, scale factors, axis misalignment and so on, which are removed from row measurements by the corresponding calibration techniques. The stochastic part contains random errors (noises) which cannot be removed from the measurements and should be modeled as stochastic processes.

As an example, consider the MEMS gyroscope used in motion control. Its noise typically consists of the following terms:

- *Bias instability*. This is a stationary stochastic process which may be considered as a low-order zero-mean Gauss-Markov process.
- *Angular random walk*. This is an angular error process which is due to white noise in angular rate.
- *Rate random walk*. This is a rate error due to white noise in angular acceleration.
- *Discretization error*. This is an error representing the quantization noise.

Several other noise terms are described in detail in (IEEEStd 952-1997).

Different techniques for building models of MEMS sensor noises are presented in (El-Sheimy et al. 2008; Quinchia et

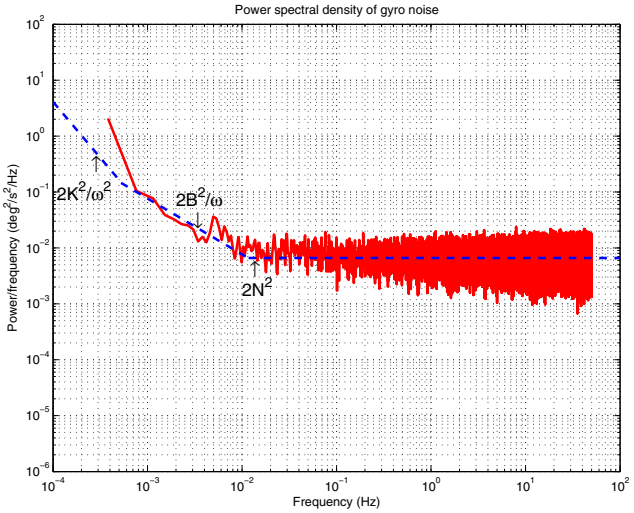

Fig. 15. Power spectral density of gyro noise

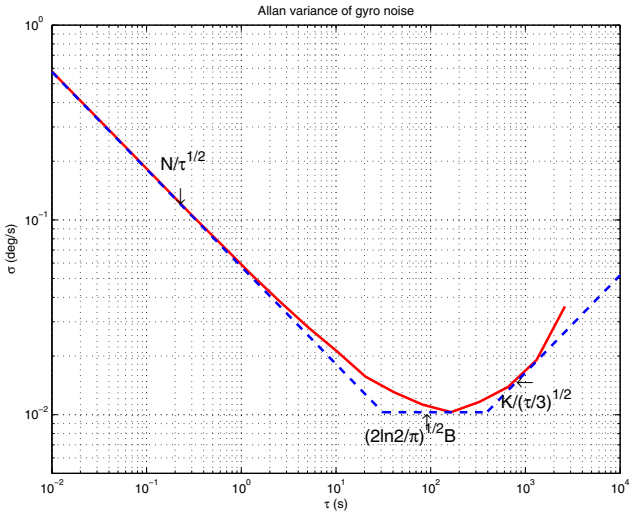

Fig. 16. Allan variance of gyro noise

al. 2012; Petkov and Slavov 2010), to name a few. Usually, they exploit the autocorrelation function of the noise in order to obtain 1st order Gauss-Markov or higher order Auto-Regressive models. Note that it is desirable to keep the model order as low as possible since the model is frequently used in the design of Kalman filter to determine optimal estimates based on the sensor measurements.

Stochastic discrete-time models of MEMS gyro are obtained on the basis of frequency-domain and time-domain characteristics of the sensors noises. For this aim it is possible to use the power spectral density of gyro noise (see Figure 15) and the so called *Alan variance* (Allan 1966) shown in Figure 16. By using nonlinear least-squares minimization it is possible to determine the numbers B, N and K which characterize bias instability, angular random walk and rate random walk, respectively.

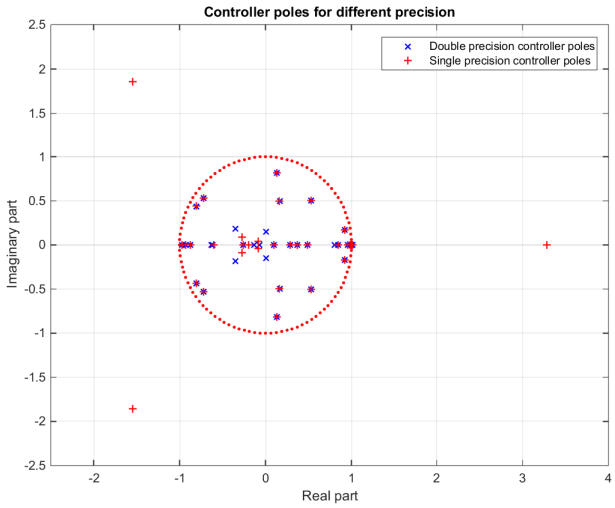

Fig. 17. Influence of the precision used on controller poles

Influence of the working precision

The robust control laws are designed in MATLAB®using double precision (64 bits) floating point arithmetic. In the best case, these control laws are implemented in the embedded processor by using single precision (32 bits) arithmetic. This circumstance may affect significantly the behavior of the discrete-time closed-loop system. As an example we show in Figure 17 the poles of a discrete-time robust controller of order 40 implemented in double and single precision. As it is seen from the Figure, the controller implemented in single precision becomes unstable which is undesirable in practice. This shows that controller stability in single precision should be checked after the design in MATLAB®.

A CASE STUDY: ROBUST CONTROL OF TWO-WHEELED ROBOT

Two-wheeled robots have several applications which make them interesting from theoretical an practical point of view. The most popular commercial product, built on the idea of self balancing two-wheeled robot is the Segway®Personal Transporter (PT), produced by Segway Inc., USA (Segway 2012). Some of the Segway®PTs have maximum speed of 20 km/h and can travel as far as 38 km on a single battery charge. The self-balancing two-wheeled robot NXTway-GS (Yamamoto 2015), built on the basis of the LEGO®Mindstorms NXT developer kit, is widely used in education. Also, the telepresence and video conferencing two-wheeled robot Double®(Double 2014) is very popular recently.

The two-wheeled robots have dynamics which is similar to the inverted pendulum dynamics so that they are inherently unstable and should be stabilized around the vertical position using a control system. Linear-quadratic or proportional-integral-derivative (PID) control laws are usually implemented in order to achieve vertical stabilization and desired position in the horizontal plane.

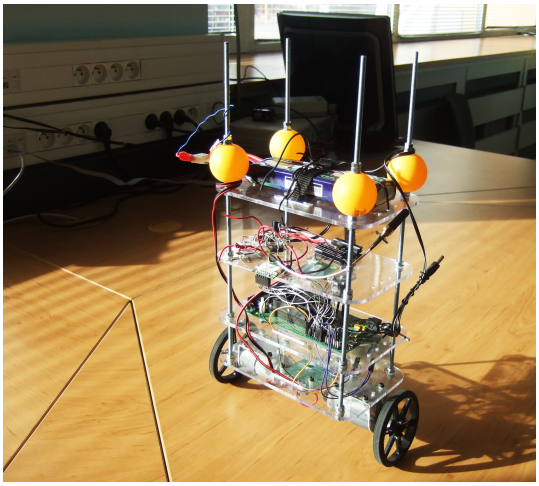

Fig. 18. Two-wheeled robot in self-balancing mode

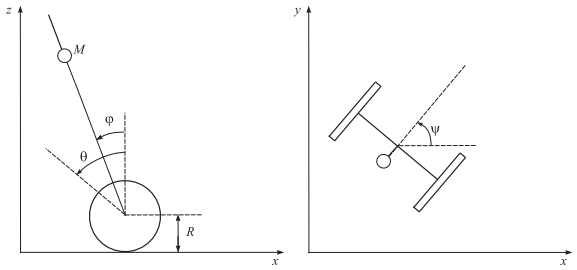

Fig. 19. Robot motion in vertical and horizontal planes

In this section we present the design and experimentation of a robust control system of two-wheeled robot which implements a μ-regulator for vertical stabilization and a proportional-integral controller of the robot rotation around the vertical axis. Due to the lack of accurate analytical robot model, the control system design is done by using a model built with the aid of an identification procedure. A discrete-time Kalman filter of 2nd order is implemented to estimate the plant dynamics around the vertical axis. A software in MATLAB®/Simulink®environment is developed for generation of control code which is embedded in the Texas Instruments Digital Signal Controller TMS320F28335. Results from the simulation of the closed-loop system as well as experimental results obtained during the real-time implementation of the controller designed are given.

Robot description

The robot is equipped with two servo drives for actuation, MEMS inertial sensor ADIS16350 for measuring the angular velocities $\dot{\phi}$ and $\dot{\psi}$ of robot body in the vertical plane and around the vertical axis, respectively, quadrature encoders for measuring the position of the wheels and a digital signal controller Texas Instruments TMS320F28335 implementing a discrete real-time stabilization algorithm with sampling period $T_0 = 0.005$ s. The robot balancing is achieved by rotating the

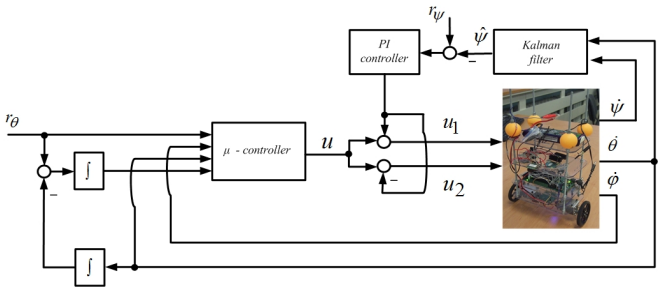

Fig. 20. Block-diagram of the control system

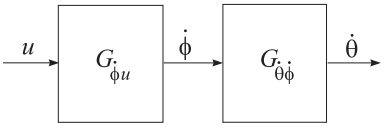

Fig. 21. Representation of robot dynamics

wheels in appropriate direction. The computation of control actions to both DC brushed drive motors is realized in single precision on the basis of signals from the gyro sensor measuring the angular rate (and, after integration, the tilt angle ϕ) and signals from rotary encoders measuring the wheels rotation angles. The control of the DC motors is executed by Pulse Width Modulated (PWM) signals. A 2nd order Kalman filter is used to estimate the yaw angle ψ.

Model identification

The block-diagram of the two-wheeled robot control system is shown in Figure 20.

In order to simplify the robot dynamics it is assumed that the motions in vertical and horizontal planes are independent. This allows to represent the robot dynamics by a single-input single-output plant as shown in Figure 21.

In respect to the stabilization in upper equilibrium state and to the control of forward-backward motion, the robot is described by Auto Regressive Moving Average with eXternal input (ARMAX) and Box-Jenkins (BJ) discrete-time models, respectively. For this aim we use the corresponding functions from System Identification Toolbox®3 (Ljung 2013). The ARMAX model of 7th order with structure parameters $na = 7, nb = 7, nc = 7, nk = 3$ describes the link between the control signal u and the rate $\dot{\phi}$, while the BJ model of 3rd order with structure parameters $nb = 3, nf = 3, nc = 3, nd = 3, nk = 1$ gives the link between the rates $\dot{\phi}$ and $\dot{\theta} = (\dot{\theta}_L + \dot{\theta}_R)/2$ where $\dot{\theta}_L$ and $\dot{\theta}_R$ are the angular velocities of left and right wheels, respectively. ARMAX and BJ models, resulting from identification, are assumed as nominal and have the form

$$\dot{\phi}(z) = G_{\dot{\phi}u,nom}(z)u(z) + v_{\dot{\phi}}(z), \tag{11}$$

$$\dot{\theta}(z) = G_{\dot{\theta}\dot{\phi},nom}(z) + v_{\dot{\theta}}(z) \tag{12}$$

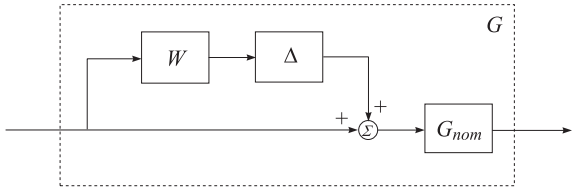

Fig. 22. Input multiplicative uncertainty

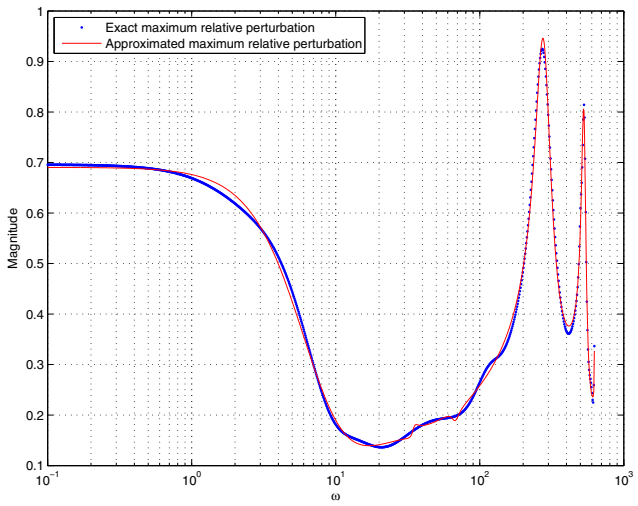

Fig. 23. Approximation of the relative uncertainty in $G_{\dot{\phi},u}$

where $G_{\dot{\phi}u,nom}(z)$ is the nominal transfer function from u to $\dot{\phi}$, $G_{\dot{\theta}\dot{\phi},nom}(z)$ is the nominal transfer function from $\dot{\phi}$ to $\dot{\theta}$, the noises $v_{\dot{\phi}}, v_{\dot{\theta}}$ are obtained during the identification procedure and reflect the uncertainty in the model found. These noises are represented as models with input multiplicative uncertainties which are used in the design of robust controller.

Condition of unbiased parameter estimates guarantees that the exact parameter values are contained in the confidence intervals of estimates with probability close to 1. This allows derivation of multiplicative uncertainty models of the transfer functions $G_{\dot{\phi}u}$ and $G_{\dot{\theta}\dot{\phi}}$ in the form of LFT shown in Figure 22. Based on confidence intervals, maximum relative deviations from nominal models are obtained. These deviations are approximated, through optimization procedure, with shaping filters which are represented by transfer function of 9th and 5th order, respectively.

The maximum relative uncertainties in $G_{\dot{\phi},u}$ and $G_{\dot{\theta},\dot{\phi}}$ along with their approximations are represented in Figures 23 and 24, respectively.

Resulting uncertain models for the robot vertical and longitudinal motion are

$$G_{\dot{\phi}u}(z) = G_{\dot{\phi}u,nom}(z)(1+W_{\dot{\phi}u}(z)\Delta_{\dot{\phi}}), \quad (13)$$

$$G_{\dot{\theta}\dot{\phi}}(z) = G_{\dot{\theta}\dot{\phi},nom}(z)(1+W_{\dot{\theta}\dot{\phi}}(z)\Delta_{\dot{\theta}}) \quad (14)$$

where $W_{\dot{\phi}u}(z), W_{\dot{\theta}\dot{\phi}}(z)$ are the corresponding shaping filters obtained by approximations of the relative magnitude deviations

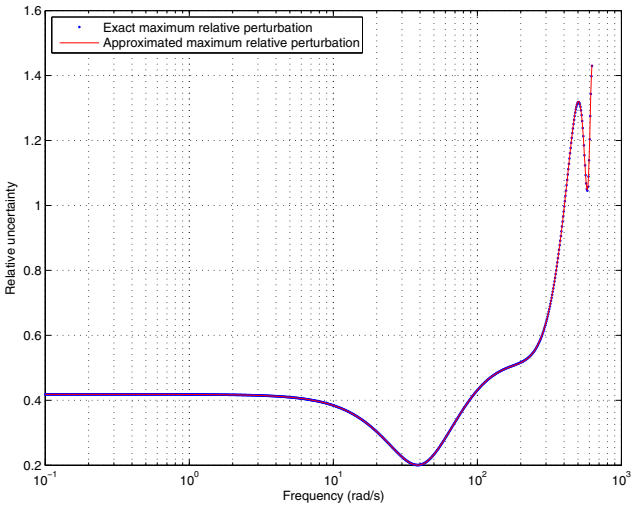

Fig. 24. Approximation of the relative uncertainty in $G_{\dot{\theta},\dot{\phi}}$

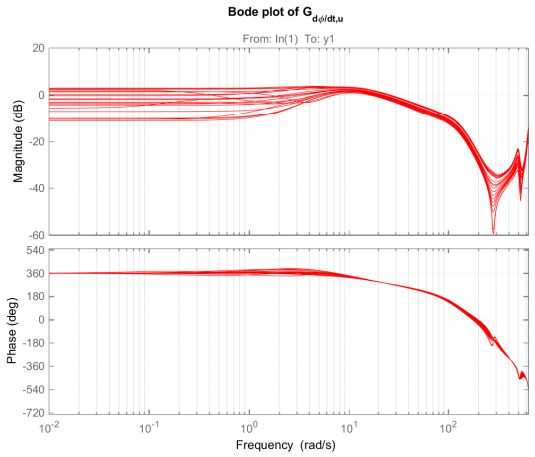

Fig. 25. Bode plot of $G_{\dot{\phi},u}$

and $\Delta_{\dot{\phi}}, \Delta_{\dot{\theta}}$ are uncertainties which satisfy

$$|\Delta_{\dot{\phi}}| < 1, \Delta_{\dot{\theta}} < 1.$$

The Bode plots of the uncertain models $G_{\dot{\phi},u}$ and $G_{\dot{\theta},\dot{\phi}}$ are shown in Figures 25 and 26, respectively.

Controller design

The closed-loop system block-diagram corresponding to the μ-synthesis problem is shown in Figure 27. In order to obtain better position accuracy a feedback from the integral of the tracking error $r_\theta - \theta$ to the controller is introduced (Goodwin et al. 2001). The control actions to the plant are realized by a Digital Signal Controller in real time with sampling frequency $f_s = 200$ Hz. For this reason the μ-synthesis is implemented to design a discrete-time controller at this sampling frequency.

To obtain good performance of the closed-loop system we shall implement a two-degree-of-freedom controller (Gu et

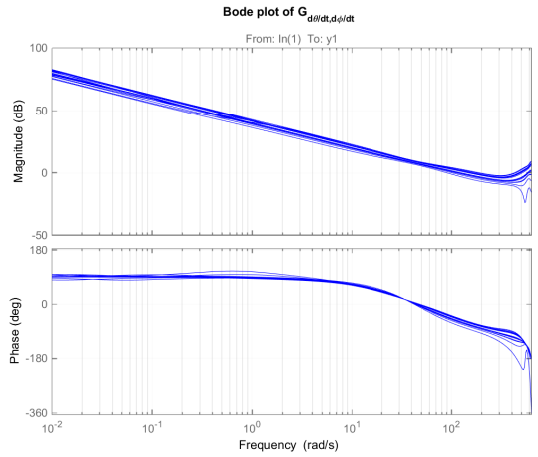

Fig. 26. Bode plot of $G_{\dot{\theta},\dot{\phi}}$

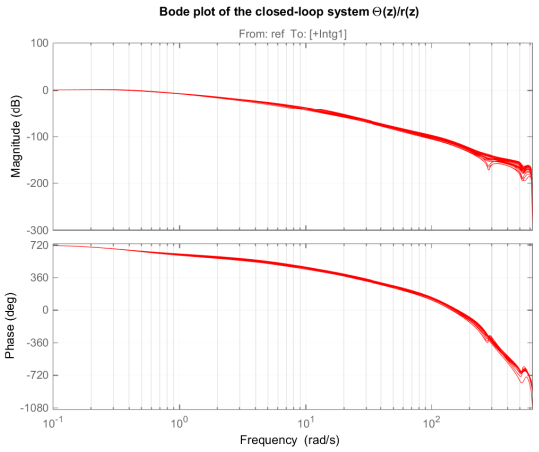

Fig. 28. Bode plot of the closed-loop system

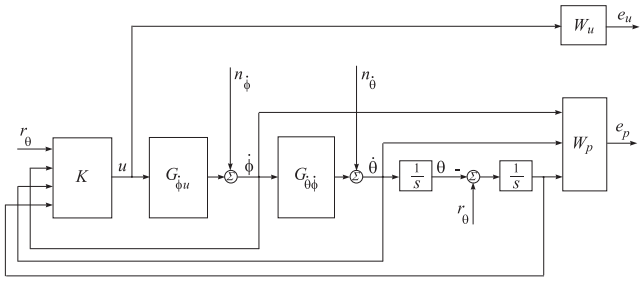

Fig. 27. Block-diagram of the closed-loop system

al. 2013). The control actions are generated according to the expression

$$u_c = [K_r \; K_y] \begin{bmatrix} r_\theta \\ y_c \end{bmatrix} = K_r r_\theta + K_y y_c, \tag{15}$$

where r_θ is the wheels reference angle and

$$y_c = \left[\dot{\phi}, \dot{\theta}, \int (r_\theta - \theta)\right]^T$$

is the output feedback vector, K_y is the output feedback transfer function matrix and K_r is the pre-filter transfer function.

The weighted closed-loop system outputs e_p and e_u satisfy the equation

$$\begin{bmatrix} e_p \\ e_u \end{bmatrix} = \begin{bmatrix} W_p S_o G K_r \\ W_u S_i K_r \end{bmatrix} r_\theta \tag{16}$$

where the matrix $S_i = (I - K_y G)^{-1}$ is the input sensitivity transfer function matrix, $S_o = (I - G K_y)^{-1}$ is the output sensitivity transfer function matrix and $G = G_{\dot{\phi}u} G_{\dot{\theta}\dot{\phi}}$.

The performance criterion requires the transfer function matrix from the exogenous input signal r_θ to the output signals e_p and e_u to be small in the sense of $\|\cdot\|_\infty$, for all possible uncertain plant models G. This leads to small weighted signals $\dot{\phi}$, $\dot{\theta}$ and $\int(r_\theta - \theta)$ and small control action. The transfer function matrices W_p and W_u are used to reflect the relative importance of the different frequency ranges for which the performance requirements should be fulfilled.

The μ-synthesis is done for several performance weighting functions that ensure a good balance between system performance and robustness. On the basis of the experimental results, we choose the performance weighting function (in the continuous-time case) as

$$W_p(s) = \begin{bmatrix} 52.2\frac{0.005s+1}{s+1} & 0 & 0 \\ 0 & 0.029\frac{0.6s+1}{0.007s+1} & 0 \\ 0 & 0 & 4.7\frac{4s+1}{125s+1} \end{bmatrix}$$

and the control weighting function as

$$W_u(s) = \frac{1}{30000} \frac{\frac{1}{0.007}s+1}{\frac{1}{200}s+1}.$$

The performance weighting functions are chosen as low pass filters, and the control weighting function is chosen as high pass filter with appropriate bandwidth in order to impose constraints on the spectrum of the control actions. These functions are discretized for the sampling frequency of 200 Hz and included in the open-loop system model.

The μ-synthesis is performed by using the MATLAB®function `dksyn` (Balas et al. 2012). In order to obtain better convergence of the *D-K* iterations two small noises $n_{\dot{\phi}}$ and $n_{\dot{\theta}}$ with intensity 10^{-4} are added to the angular velocities $\dot{\phi}$ and $\dot{\theta}$, respectively. Three iterations are performed that decrease the maximum value of μ to 0.693. The final controller obtained is of 50th order.

The Bode plot of the closed-loop system is shown in Figure 28. It is seen that the closed-loop bandwidth is approximately 1 rad/s.

The magnitude plot of the closed-loop in respect to the tracking error $e_\theta = r_\theta - \theta$ is shown in Figure 29.

In Figures 30 and 31 we show the influence of the noises in $\dot{\phi}$ and $\dot{\theta}$, respectively, to the control action u. It is seen that the effect of noise in $\dot{\phi}$ is much stronger than the effect of noise in $\dot{\theta}$.

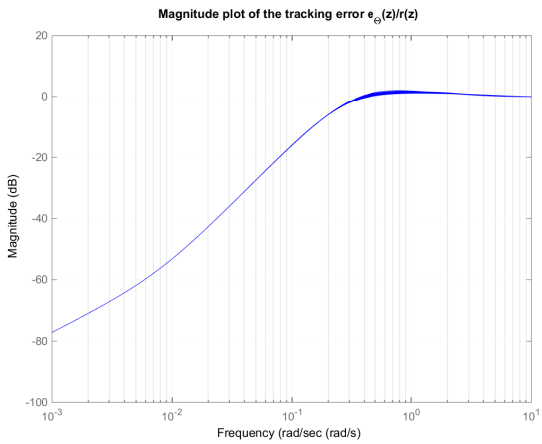

Fig. 29. Magnitude plot of the tracking error

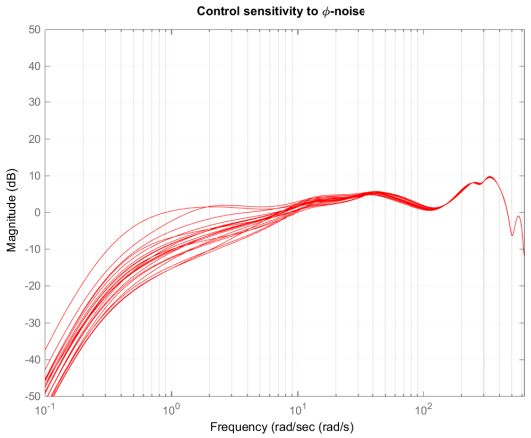

Fig. 30. Sensitivity of control to noise in $\dot{\phi}$

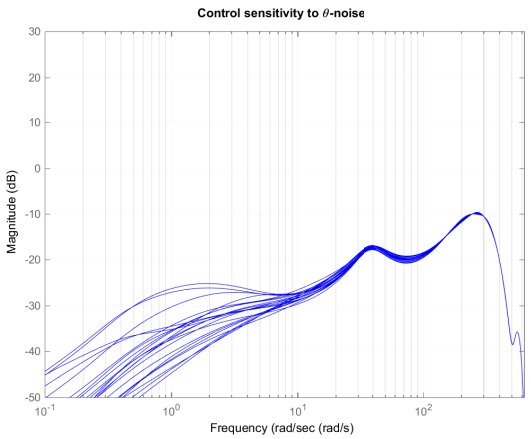

Fig. 31. Sensitivity of control to noise in $\dot{\theta}$

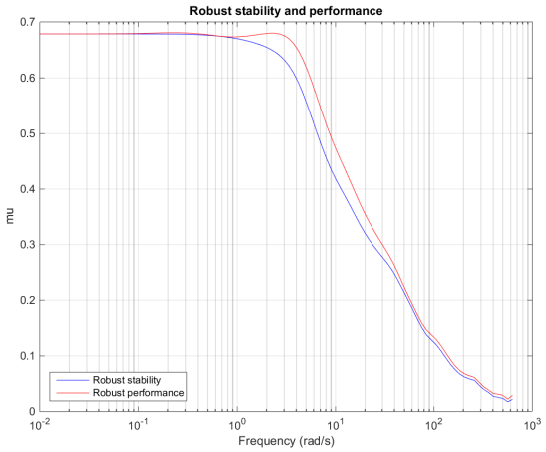

Fig. 32. Robust stability and performance of the closed-loop system

Finally, in Figure 32 we show the plots of the structured singular values corresponding to the robust stability and robust performance of the closed-loop system. The system achieves both robust stability and robust performance in respect to the uncertainties corresponding to the identification of robot model.

A PI controller of the angular motion around the vertical axis is also designed. The yaw angular velocity $\dot{\psi}$ is measured by a gyroscope of the same type as the gyro used to measure the tilth rate. This gyroscope contains a significant noise ψ_g which is modeled by the additional equation

$$\dot{\psi}_g(k+1) = \dot{\psi}_g(k) + J_g \nu_g(k) \tag{17}$$

where ν_g is a white gaussian noise with unit variance and the coefficient J_g is determined experimentally to obtain a good estimate of $\dot{\psi}$ as $J_g = 0.0001$. A second order Kalman filter is designed to produce sufficiently accurate estimate $\hat{\psi}$ of the yaw angle.

Experimental results

A simulation scheme of the control system and a specialized software in MATLAB®/Simulink®environment is developed to implement the control code. To simplify the controller, its order is reduced from 50 to 30. With the aid of Simulink Coder®[27] and Code Composer Studio®, a code is generated from this software which is embedded in the Texas Instruments Digital Signal Controller TMS320F28335 [26].

Several experiments with the controller designed are performed and comparison with the simulation results is done. The experimental results obtained during the real-time robot control are given in Figures 33 - 36.

It is seen form the Figures33 and 34 that the wheels track accurately the reference and the robots keeps well its vertical position. The measured angle ϕ contains a significant bias which is removed by using off-line a 17th order Kalman filter. It is interesting to note that this bias does not affect the vertical stabilization of the robot which is due to the good filtration

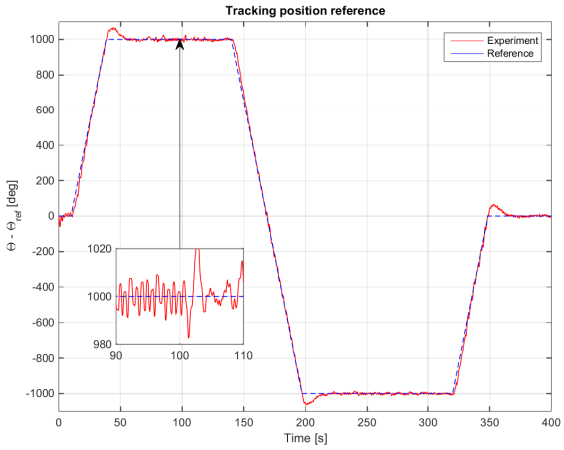

Fig. 33. Tracking position reference

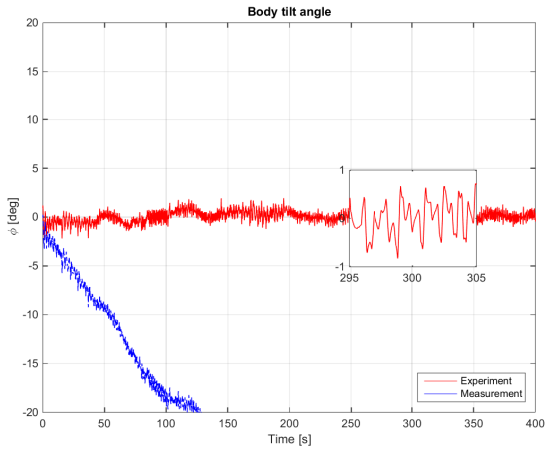

Fig. 34. Body angle variation

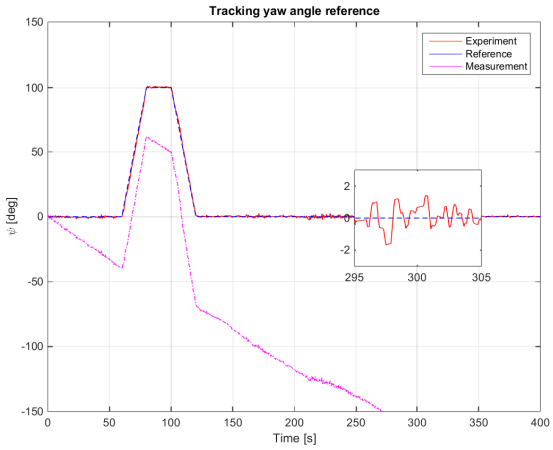

Fig. 35. Tracking yaw angle reference

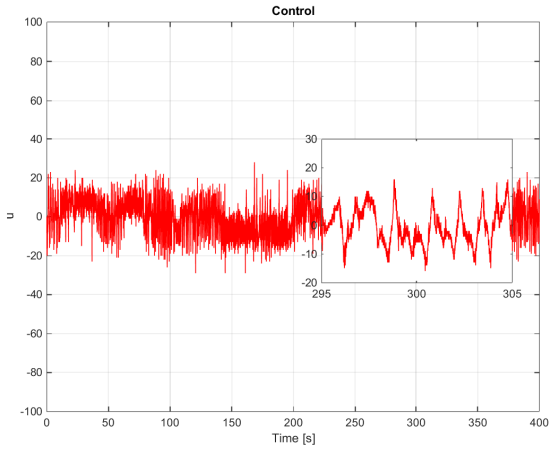

Fig. 36. Control action

properties of the system with μ-controller. The usage of the estimate $\hat{\psi}(k)$, obtained by a 2nd order Kalman filter, instead of $\psi(k)$ ensures exact rotation of the robot around the vertical axis, while there is a increasing with the time bias in the measured value of $\psi(k)$ due to the integration of the gyro noise (Figure 35). The control action does not exceed 40 units with maximum allowable value equal to 50 (Figure 36).

CONCLUSIONS

The paper presents a brief survey of several issues arising in the design and implementation of robust control laws in embedded control systems. Some difficulties and unsolved problems that require further research are pointed out. A case study of robot control design is presented which illustrates the progress in the implementation of high order robust controllers. Several references are included which may help the reader to find more detailed information about the topics discussed.

Acknowledgement

The research presented in this paper is done under the project 142PD0008-08 funded by the Scientific and Research Unit of the Technical University of Sofia.

REFERENCES

[1] Allan, D.W. 1966. Statistics of atomic frequency standards. *Proceedings of the IEEE, vol. 54, 221-230*. DOI: 10.1109/PROC.1966.4634

[2] Balas, G., R. Chiang, A. Packard, M. Safonov 2013. Robust Control Toolbox®3 , User's Guide, The MathWorks, Inc., Natick, MA. *Available from* `http://www.mathworks.com/help/pdf_doc/robust/robust_ug.pdf`

[3] Bautista-Quintero, R., M.J. Pont 2008. Implementation of H-infinity control algorithms for sensor-constrained mechatronic systems using low-cost microcontrollers. *IEEE Transactions on Industrial Informatics, vol. 4, 175-184*. DOI: 10.1109/TII.2008.2002703

[4] Bittanti, S., S. Garatti 2012. System Identification and Control: a fruitful cooperation over half a century, and more. *Proceedings of the 31st Chinese Control Conference, July 25-27, 2012, Hefei, China. IEEE, 6-15*. ISBN: 978-1-4673-2581-3

[5] Campion, G., W. Chung 2008. Wheeled Robots. In *B. Siciliano and O. Khatib, editors, Springer Handbook of Robotics, chapter 17, pp. 391-410, Springer, Berlin*. ISBN: 978-3-540-23957-4, DOI: 10.1007/978-3-540-30301-5

[6] Chan, R.P.M., K.A. Stol, C.R. Halkyard 2013. Review of modelling and control of two-wheeled robots. *Annual Reviews in Control, vol. 37, 89-103*. DOI: 10.1016/j.arcontrol.2013.03.004

[7] Double Robotics 2014. Sunnyvale, CA. http://www.doublerobotics.com

[8] El-Sheimy, N., H. Hou, X. Niu 2008. Analysis and modeling of inertial sensors using Allan variance. *IEEE Transactions on Instrumentation and Measurement, vol. 57, 140-149*. DOI: 10.1109/TIM.2007.908635

[9] Gahinet, P., A. Nemirovski, A.J. Laub, M. Chilali 1995. *LMI Control Toolbox*, The MathWorks Inc., Natick, MA.

[10] Goodwin, G.C., S.F. Graebe, M.E. Salgado 2001. *Control System Design*, Prentice Hall, Upper Saddle River, NJ. ISBN: 9780139586538

[11] Gu, D.-W., P.Hr. Petkov, M.M. Konstantinov 2013. *Robust Control Design with MATLAB®, 2nd ed.*. Springer-Verlag, London. ISBN: 978-1-4471-4681-0

[12] Gugercin, S., A. C. Antoulas, H. P. Zhang 2003. An approach to identification for robust control. *IEEE Transactions on Automatic Control, vol. 48, 1109-1115*. DOI: 10.1109/TAC.2003.812821

[13] Howlader, A.M., N. Urasaki, A. Yona, T. Senjyu, A.Y. Saber 2013. Design and implement a digital H_∞ robust controller for a MW-class PMSG-based grid-interactive wind energy conversion system. *Energies, vol. 6, 2084-2109*. DOI:10.3390/en6042084

[14] Hristu-Varsakelis, D., W.S. Levine (Eds.) 2005. *Handbook of Networked and Embedded Control Systems*. Birkhäuser, Boston. ISBN: 978-0-8176-3239-7

[15] Isermann, R., M. Münchhof 2011. *Identification of Dynamic Systems. An Introduction with Applications*. Springer-Verlag, Berlin. ISBN: 978-3-540-78878-2, DOI: 10.1007/978-3-540-78879-9

[16] Jahaya, J., S.W. Nawai, Z. Ibrahim 2011. Multi input single output closed loop identification of two wheel inverted pendulum mobile robot. *2011 IEEE Student Conference on Research and Development (SCOReD), 19-20 Dec. 2011, Cyberjaya, Malaysia. IEEE, 138-143*. DOI: 10.1109/SCOReD.2011.6148723

[17] Landau, I.D., G. Zito 2006. *Digital Control Systems. Design, Identification and Implementation*. Springer-Verlag, London. ISBN 978-1-84628-055-9,

[18] Ljung, L.1999. *System Identification: Theory for the User, 2nd ed.*. Prentice-Hall, Inc. Englewood Cliffs, NJ. ISBN: 978-0136566953

[19] Ljung, L. 2013 *System Identification Toolbox®User's Guide*. The MathWorks Inc., Natick, MA. ISBN 0-13-881640-9

[20] Lunze, J., F. Lamnabhi-Lagarrigue (Eds.) 2009. *Handbook of Hybrid Systems Control. Theory, Tools, Applications*. Cambridge University Press, Cambridge. ISBN: 978-0-521-76505-3

[21] Lupián L.F., R. Avila 2009. Stabilization of a wheeled inverted pendulum by a continuous-time infinite-horizon LQG optimal controller. *IEEE Latin American Robotic Symposium LARS'2008, Natal, Rio Grande do Norte, Brazil, 29-30 Oct. 2008. IEEE, 65-70*. DOI: 10.1109/LARS.2008.33

[22] Nawawi, Ahmad, Osman 2010. Real-time control system for a two-wheeled inverted pendulum mobile robot. Chapter 6 in: I. Fürstner (Ed.), *Advanced Knowledge Application in Practice, Sciyo, Rijeka, Croatia, 299-312*. ISBN 978-953-307-141-1

[23] Muhammad, M., S. Buyamin, M.N. Ahmad, S.W. Nawawi 2011. Dynamic modeling and analysis of a two-wheeled inverted pendulum robot. *Third International Conference on Computational Intelligence, Modelling and Simulation CIMSiM 2011, 20-22 Sept. 2011, Langkawi, Malaysia. IEEE, 159-164*. DOI: 10.1109/CIMSim.2011.36

[24] Quinchia, A.G., C. Ferrer, G. Falco, E. Falletti, F. Dovis 2012. Analysis and modelling of MEMS Inertial Measurement Unit. *2012 International Conference on Localization and GNSS (ICL-GNSS), 25-27 June 2012, Starnberg, Germany. IEEE, 1-7*. DOI: 10.1109/ICL-GNSS.2012.6253129

[25] The MathWorks, Inc. 2013. *Embedded Coder*. Natick, MA. *Available from:* `http://www.mathworks.com/products/datasheets/pdf/embedded-coder.pdf`

[26] Spectrum Digital, Inc. 2007. *eZdspTMF28335 Technical Reference*. *Available from:* `http://c2000.spectrumdigital.com/ezf28335/docs/ezdspf28335c_techref.pdf`

[27] The MathWorks, Inc. 2013. *Simulink Coder*. Natick, MA. *Available from:* `http://www.mathworks.com/products/datasheets/pdf/simulink-coder.pdf`

[28] IEEEStd 952-1997, 1998. IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. *IEEE Aerospace and Electronic Systems Society*.

[29] Raafat, S.M., Akmeliawati, I. Abdulljabar 2012. Robust H_∞ controller for high precision positioning system, design, analysis, and implementation. *Intelligent Control and Automation, vol. 3, 262-273*. DOI:10.4236/ica.2012.33030

[30] Sánchez-Peña, R.S., M. Sznaier 1998. *Robust Systems. Theory and Applications*, John Wiley & Sons, Inc., New York. ISBN: 978-0-471-17627-5

[31] Segway Personal Transporters 2012. Bedford, NH. http://www.segway.com

[32] Skogestad, S., I. Postlethwaite 2005. *Multivariable Feedback Control, 2nd ed.*. John Wiley and Sons Ltd, Chichester, UK. ISBN: 978-0-470-01167-6

[33] Van den Hof, P. 2001. Identification of experimental models for control design. *Proceedings of the 18th Instrumentation and Measurement Technology Conference, (IMTC 2001), 21-23 May 2001, Budapest, Hungary. IEEE, vol. 2, 1155 - 1162*. DOI: 10.1109/IMTC.2001.928260

[34] Venkatesh, S. 2003. Identification of uncertain systems described by Linear Fractional Transformations. *Proceedings of the 42nd IEEE Conference an Decision and Control Maui, Hawaii USA, December 2003. IEEE, 5532-5537*. DOI: 10.1109/CDC.2003.1272518

[35] Yamamoto, Y. 2013. NXTway-GS (Self-Balanced Two-Wheeled Robot Controller Design). `http://www.mathworks.com/matlabcentral/fileexchange/`

[36] Zhou, K., J.C. Doyle, K. Glover 1996. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ. ISBN: 0-13-456567-3