

# Robust Real- and Discrete-Time Control of a Steer-by-Wire System in Cars

Eduard Reithmeier

*Institut für Meß- und Regelungstechnik  
Universität Hannover, 30167 Hannover, Germany  
Fax: +49-511-762-3234, Tel: +49-511-762-3331  
E-mail: reithmeier@imr.uni-hannover.de*

There is a series of objectives to remove the mechanical coupling between the steering wheel and the wheel suspension system in cars and establish a so-called steer-by-wire system. One main reason is related to driving comfort, that is, torque feedback characteristics may be individually adjusted to the driver. Other reasons are focused on providing a higher security level during driving. For instance, the car's stabilizing control system could be enhanced by gaining more influence, particularly in critical driving situations. Other advantages are related to manufacturing aspects, like assembly improvements or an entire new layout of the engine compartment.

The idea of a steer-by-wire system is to bring a torque-actuator interface – one for the steering wheel and one for the turning wheel system – into place where the steer-shaft is cut in half. Both actuator systems will be managed separately. However, an information exchange system is established between the two systems. This work is concerned with the steering wheel part. We use steer-angle, steer-angle-velocity and the actual steer-shaft torque as measured variables in order to control the torque motor connected to the shaft according to a given or desirable torque vs. steer-angle characteristic respectively. The system contains a series of uncertainties which need to be taken into account. Hence, the controller design is primarily based on robust control techniques. The main problems are caused around a zero steer-angle due to changes in torque direction. The steering-behavior around the zero steer-angle becomes sloppy and undefined. This led us to an appropriate additional mechanical design to implement a discrete time control scheme for the desired torque characteristic. The author will report on the current state of our experimental and theoretical investigations.

## 1 Introduction

X-by-wire systems (such as brake-by-wire, shift-by-wire or even steer-by-wire) are becoming more and more important to car manufacturers. The main reasons are the advantages and particular properties (see below) which car makers have to take into consideration for prospective intelligent transportation systems in the future. Also, continuously falling costs for general electronic equipment as well as for computational hardware and software products are pushing this general tendency.

Figure 1 shows the wheel suspension and steering system of an average car. In a steer-by-wire system the steering shaft is cut into two independent parts. After separation, each interface will be equipped with an electric or electro hydraulic actuator. This results in a *Turning Wheel Actuator System* generating the torque  $m_T$  and a *Steering Wheel Actuator System* (torque  $m_S$ ). Both systems communicate via the so-called CAN-Bus. Due to the loss of the mechanical coupling there is also a loss of basically two items of information related to two physical constraints, namely on the one hand

- the kinematic condition that there is exactly one turning wheel angle  $\psi$  for each steering wheel angle  $\varphi$  and, on the other hand,
- the kinetic condition that the two interface torques are the same in magnitude.

The actuator torque  $m_T$  necessary to turn the wheels depends on the dynamics of the whole suspension system and on the contact torque  $m_C$ . The torque  $m_C$  again depends on parameters and variables like the turning angle  $\psi$ , speed  $v$  of the car or the rolling resistance  $\mu_R$  (cf. Figure 1). The driver employs an appropriate torque  $m_D$  which finally constitutes the dynamical equilibrium of the whole system drawn in Figure 1. The main objective in designing the control of the steering wheel simulator is to establish a desired feedback characteristic to the driver. This feedback depends of course on the torque coming from the turning wheel actuator system. However, since the two systems are independent there might be more and different information processed therein.

A steer-by-wire system has some significant advantages in terms of security, comfort and manufacturing, and it is a necessary device for future transportation aspects. A traffic guided system, for instance, needs autonomous steering in order to apply appropriate control without being disturbed by the driver. It may be combined with automatic stability control (ASC) in order to support or to assist the driver in certain critical situations. And it can be easily used as a car locking device. Being able to adjust arbitrary torque feedback characteristics is considered to be a significant comfort aspect. The manufacturer is interested in more cost-effective assembly and exchange of spare parts. And, a separate steering system offers the possibility of redesigning the engine compartment in a completely new way.

## 2 Modeling of the Plant

In order to design a control scheme we need, of course, a model of the considered plant which, in our case, is the steering wheel simulator system. Our experimental setup consists basically of the hardware parts shown in Figure 2.

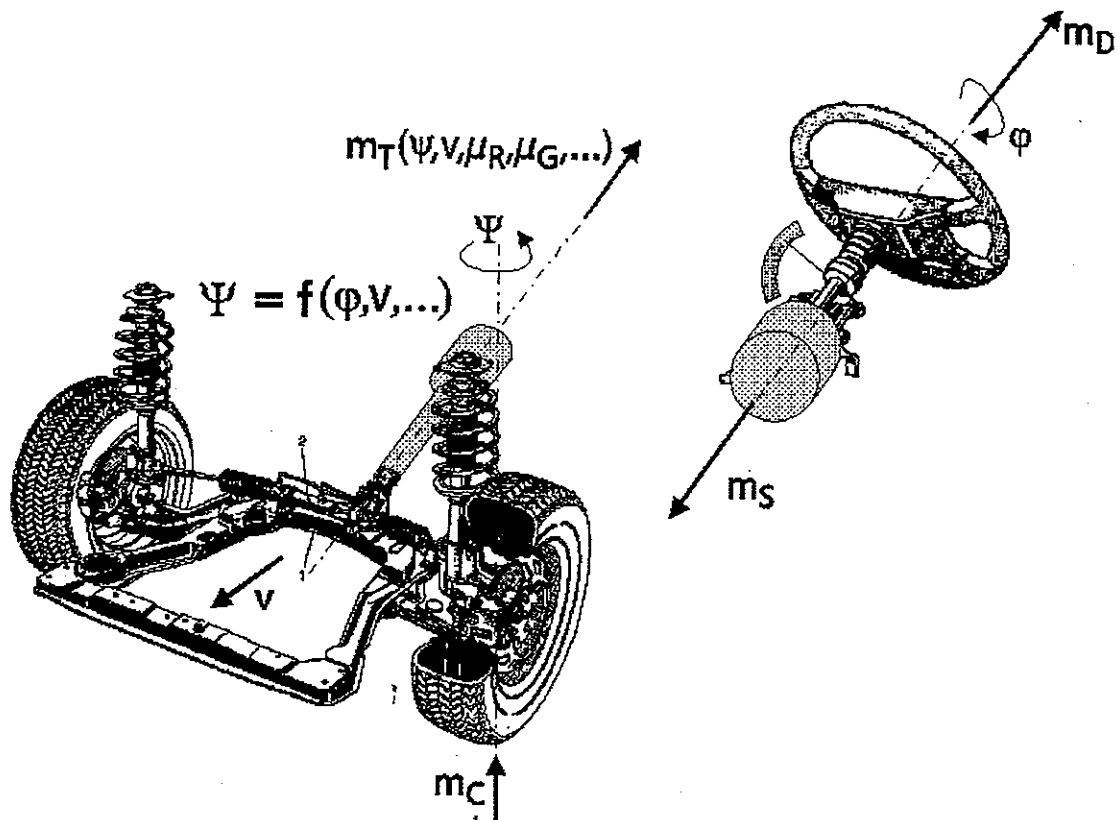


Figure 1 Wheel suspension and steering system of an average car.

First of all, there is a current-controlled electronic power supply with plant input voltage  $u_A$  for controlling a so-called torque motor. This motor, based on a three-phase-current system, supplies the power via the shaft torque and not the rotational velocity of the shaft. At the most we have only around  $\pm 2$  turns of the steering wheel but it is necessary to have torques up to 60 [Nm] available. The rotor of the torque motor is directly coupled to the wheel shaft, and a torque sensor is mounted in between. An additional feature is a spring in parallel connection to the rotor shaft in order to support the torque feedback to the driver.  $J$  is the moment of inertia of all turning parts, that is, rotor, shaft, torque sensor and steering wheel. They behave as one rigid body. The dynamics therefore are described by the steering angle  $\varphi$  and its time derivative  $\dot{\varphi}$ . These two state variables will be a direct output of the equation of motion which is represented by a flow chart according to Figure 2. The torque employed by the driver is considered as an external but measured disturbance. Bearing friction and similar unknown effects are considered to be uncertainties. Power supply and torque motor act according to a first order element followed by some time delay  $T_t$  (cf. Figure 3). The time constant  $T_T$  of the first order element depends on the temperature of the motor and has therefore to be considered as an uncertain parameter.

The last step in modeling the plant is concerned with the human-machine interface, that is, with the driver. Instead of treating the driver as a structural uncertainty, it turns out to be more appropriate to transform him/her into a simple model involving parameter uncertainties. For that purpose Figure 4 shows a qualitative closed loop model of the driver.

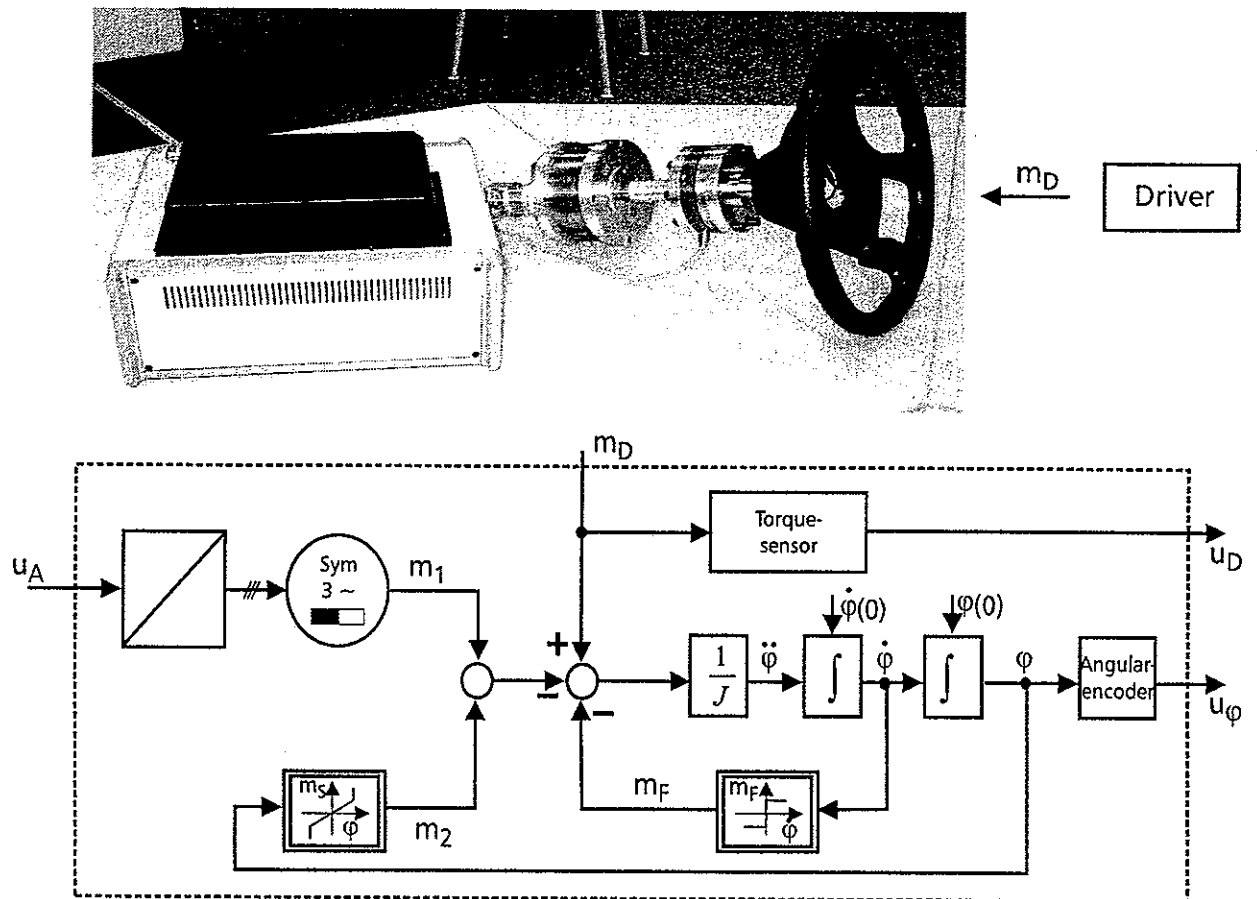


Figure 2 Model of the steering wheel simulator system.

Starting from some current road situation and some current location the driver will hopefully sense everything and will perform an analysis. The result will be some desired position of the car with respect to the road. After this, the driver will continue by giving a certain command to the actuators in order to steer the car into the desired position. In other words, the steering angle plays the role of the controlled variable, and the steering torque acts as input variable. Now, if the driver employs, say, a fictitious unit torque command to the steering wheel (cf. Figure 5), the system will respond similarly to a second order system (due to the damping  $d_0$  and elasticity  $\nu_0$  of the body parts). The strength of the muscles plays the role of the unknown amplification factor  $K_D$ . The magnitude and variation of the uncertain parameters involved can be estimated via experimental studies.

Employing the driver model in the plant leads to the flow chart given in Figure 6. It also shows the transfer function between the power supply input voltage  $u_A$  and the torque gauge output voltage  $u_D$ . Of course, this transfer function contains all model uncertainties. The objective is to follow a desired reference torque  $u_{D,Des}(\varphi)$  by properly commanding  $u_A$ .

In practice the control algorithm will be carried out by some micro controller, which requires digitized measurement values and which supplies digital input values. Within one sampling period there needs to be carried out a series of redundancy checks, AD and DA conversions as well as control algorithms with variable structure. This process takes a couple of milliseconds. That again makes it necessary to

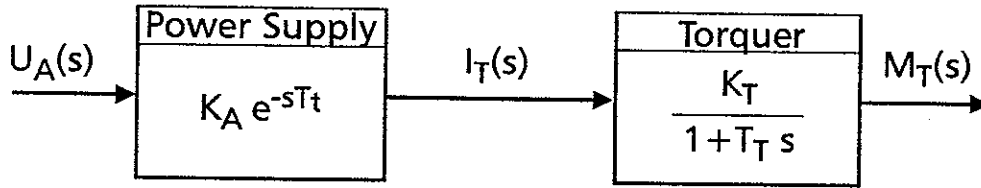


Figure 3 Transfer function of the torque motor.

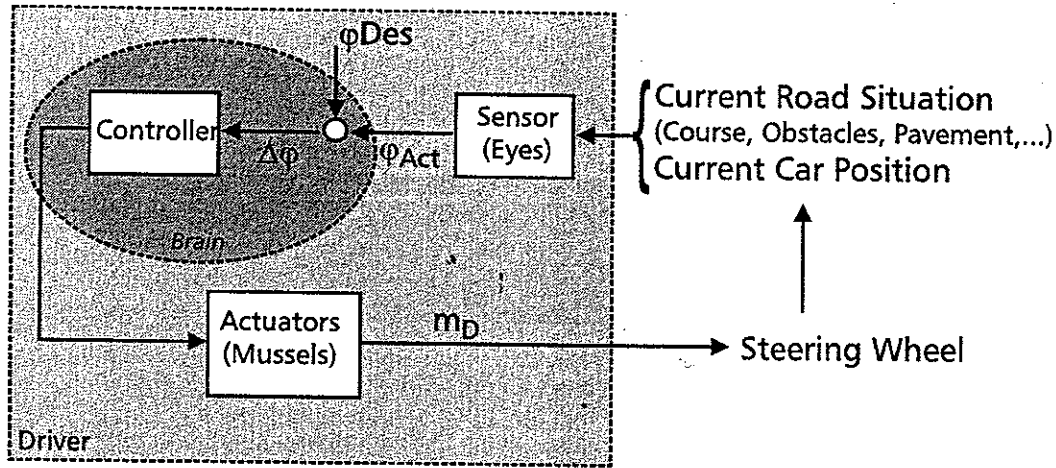


Figure 4 Simplified model of the driver.

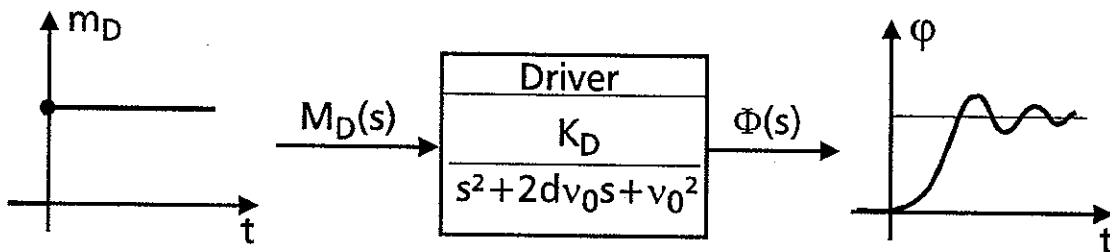


Figure 5 Transfer function of the simplified driver model.

consider a discrete time plant. If we describe the DA converter by the standard  $H_0$  modulator and the plant  $G_P$  by

$$G_P(s) = \left[ \frac{a}{s - s_1} + \frac{b}{s - s_2} + \frac{c}{s + \frac{1}{T_T}} \right] \cdot \exp(-sT_t) \quad (1)$$

with some properly chosen constants  $a$ ,  $b$  and  $c$ , then the corresponding rational Z-transfer function  $G$  is determined by

$$G(z) = \left[ \frac{a}{z - \exp(s_1 T)} + \frac{b}{z - \exp(s_2 T)} + \frac{c}{z - \exp(-\frac{1}{T_T} T)} \right] \cdot z. \quad (2)$$

In this case we used the fact that the time delay  $T_t$  of the power supply is much smaller than the sampling time  $T$ .

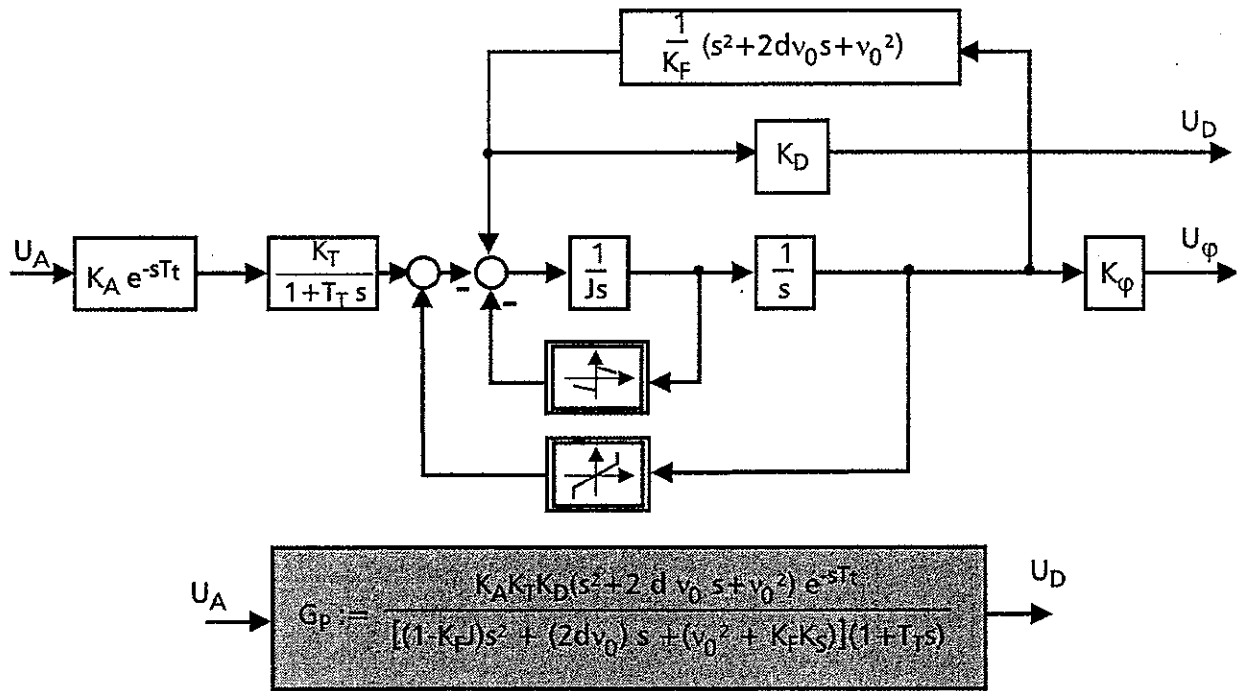


Figure 6 Transfer function of the simplified driver model.

### 3 Controller Design

Finally, the plant may be properly implemented into the closed loop system for the steering wheel simulator system according to Figure 7. The internal loop feeds back the torque directly. Micro controller board 2 contains the control algorithm and different administrative tasks such as communication with the power supply. The desired torque comes via the CAN bus from micro controller board 1 which is related to the turning wheel actuator system. In case there is a given arbitrary reference torque, we need also the actual angular position and its derivative which, in that case, will be also fed back via the CAN bus.

In this presentation we will concentrate on the design of a feedback controller which assures certain torque characteristics. The basic problem is that due to the structure of the given dynamical system it is not possible to follow any changes in the desired torque fast enough. A unit step response, for instance, looks qualitatively like the one in Figure 8.

We call  $t_{off}$  the practical settling time once the unit step response enters a certain tolerance interval and stays therein. The closed loop control objective is to reduce the settling time and lead the system to a dead beat behavior according to Figure 9.

Hence, our arrangement for reaching this goal is to design a reference transfer function  $G_R$  which behaves like a second order discrete time system with an additional damping term of the order integer  $q$ :

$$G_R(z) = (1 - z^{-1}) \cdot \mathcal{Z} \left\{ \frac{1}{s} \cdot \frac{K_0 \omega_0^2}{s^2 + 2D\omega_0 s + \omega_0^2} \cdot \left( \frac{\delta}{s + \delta} \right)^q \right\}. \quad (3)$$

The dead beat behavior is forced by the parameter  $D > 1$ . And the damping coefficient  $\delta$  is chosen in such a way that the poles of the second order system stay

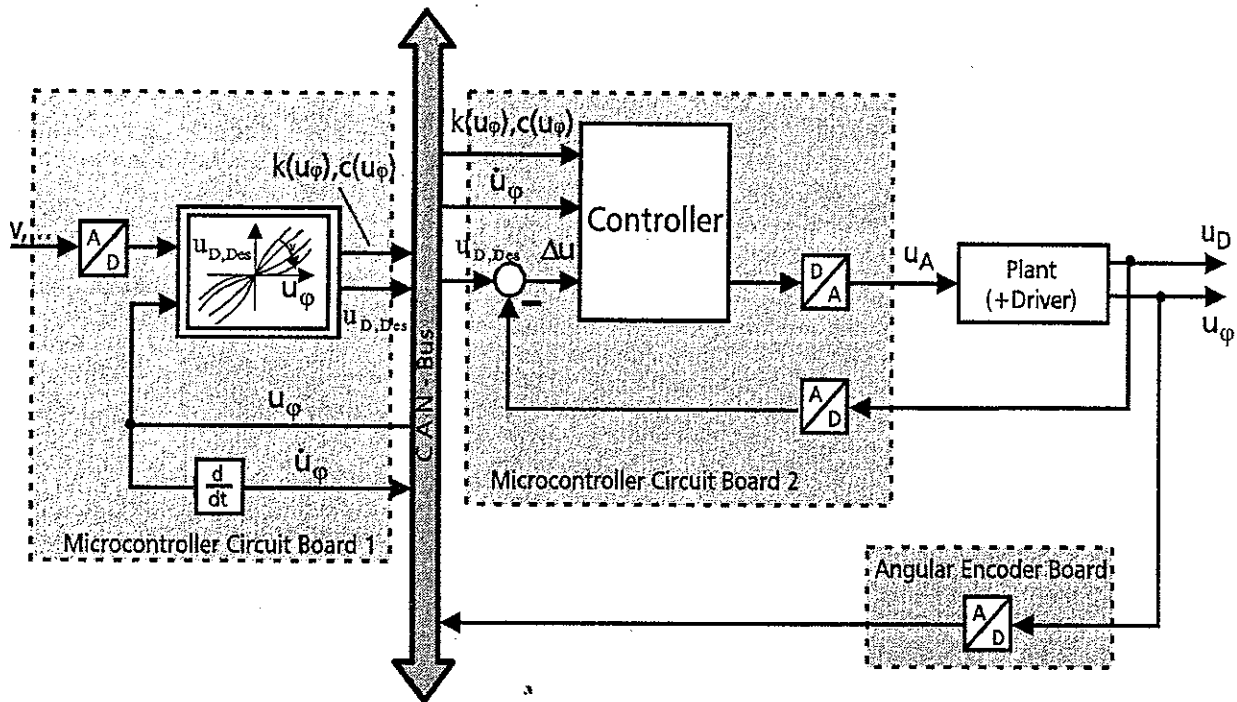


Figure 7 Discrete time closed loop system.

dominant. In order to get a feasible reference transfer function  $G_R$  we need to choose the pole surplus  $q$  in this case greater than or equal to 1. Once  $G_R(z)$  is designed, the controller  $G_C(z)$  based on plant  $G(z)$  may be directly determined from the standard closed loop according to Figure 10. For  $q = 1$ , for instance, this yields

$$G_C = \frac{G_R}{G[1 - G_R]} \quad (4)$$

or, respectively

$$G_C(z) = \frac{z^{-1}[\beta_1 + \beta_2 z^{-1} + \beta_3 z^{-2}]}{\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \alpha_3 z^{-3} + \alpha_4 z^{-4}} \quad (5)$$

with coefficients  $\alpha_i$  and  $\beta_i$ . These coefficients depend, in general, on a whole ensemble of parameters of the system. That is, uncertain parameters of the model and measured parameters as well as design parameters of the reference transfer function.

Of course, if the uncertain parameters vary in time or vary due to different drivers, the dominant poles will change their location and therefore the performance of the controller. To counter this situation we define a reference set of uncertain parameters and use, in addition, robustification procedures in order to improve the designed control scheme (see Section 4). Using the controller based on the reference set yields to a torque tracking behavior according to Figure 11 with respect to a feedback characteristic composed of two straight lines.

Figure 12a shows the actually measured torque feedback behavior of a real car. In that case, the driver turns the steering wheel at a speed of 50 km/h around 90 degrees and takes his/her hands off immediately after. At 90 degrees, the magnitude of the required torque is approximately 4 [Nm]. Figure 12b shows the same situation but produced by the steering wheel simulator.

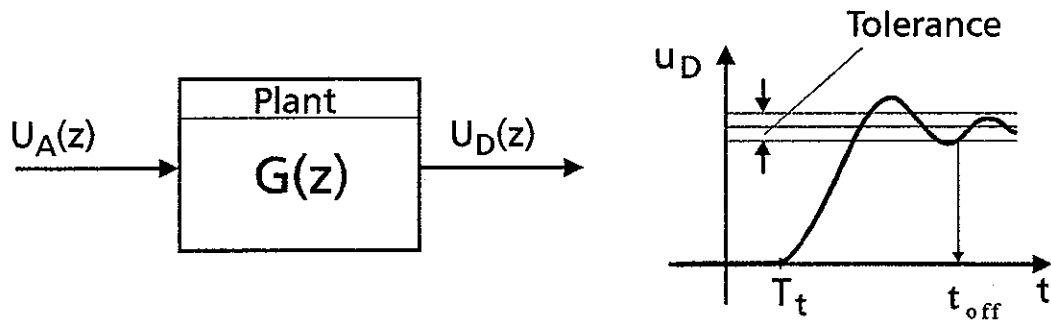


Figure 8 Unit step response of the plant (qualitatively).

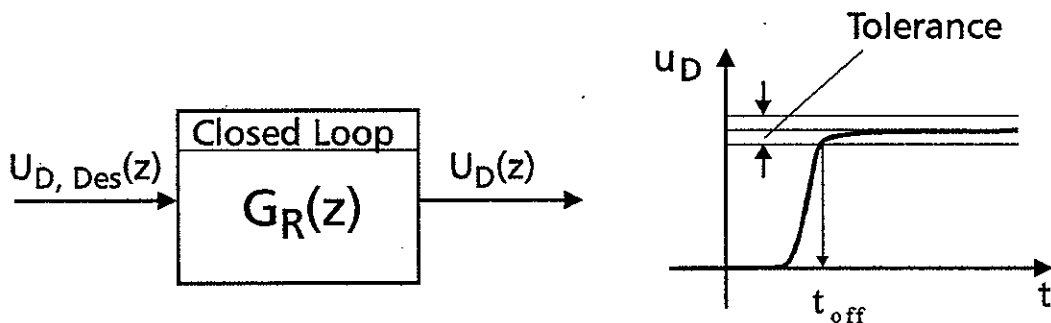


Figure 9 Unit step response of the closed loop (qualitatively).

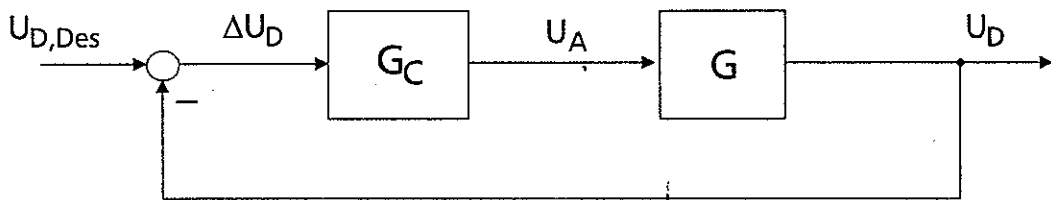


Figure 10 Standard closed loop.

## 4 Robustification

Despite the satisfying results stated in Figure 12, it turns out, in general, that it is better not to rely only on a nominal set of the uncertain parameters. To encounter this fact, it is necessary to improve the robustness of the closed loop with respect to these uncertainties. Since the controller was derived from the continuous time plant and a continuous time arrangement of the reference transfer function, the discrete time controller will also relate to some corresponding continuous time transfer element. That again makes it possible to analyze the system dynamics with respect to the corresponding continuous time elements of the discrete time open loop system.

A variation of the uncertain plant parameters results in a modified frequency response of the plant (see also Figure 13). The controller stays unchanged since it is based on the nominal set of plant parameters. The phase response of the



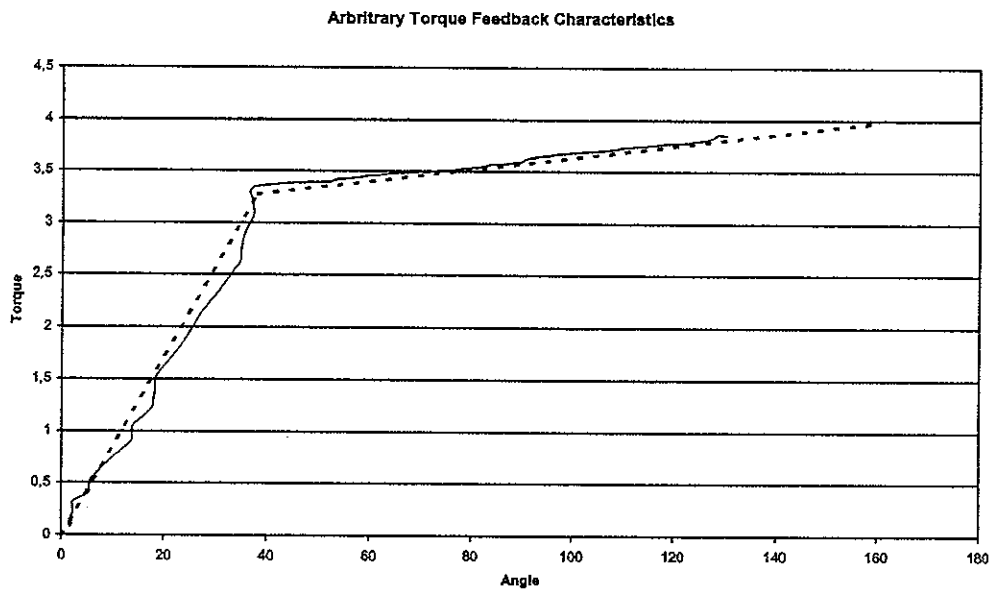
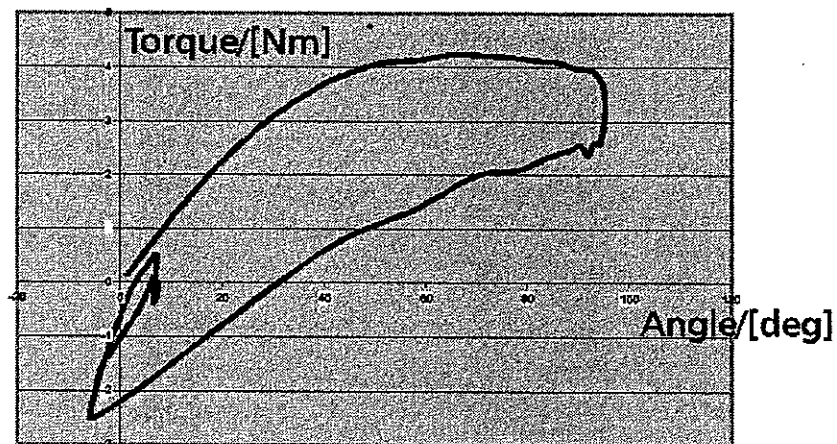


Figure 11 Actual torque (solid line), reference torque (dashed line).

a)



b)

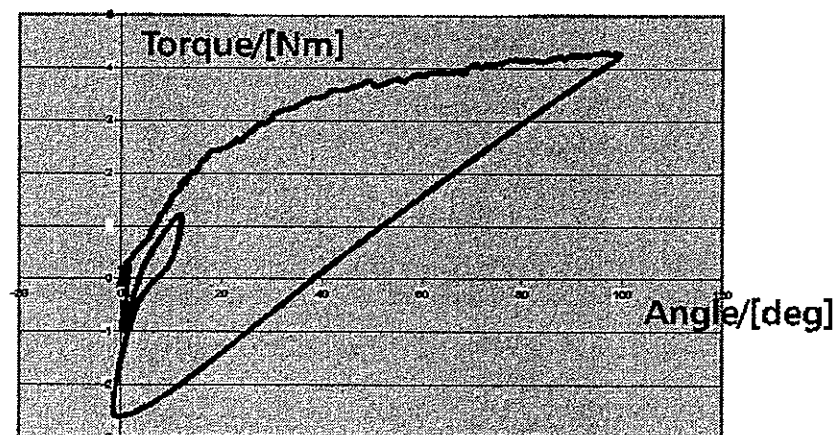


Figure 12 (a) Actual feedback characteristic. (b) Simulated feedback characteristic.

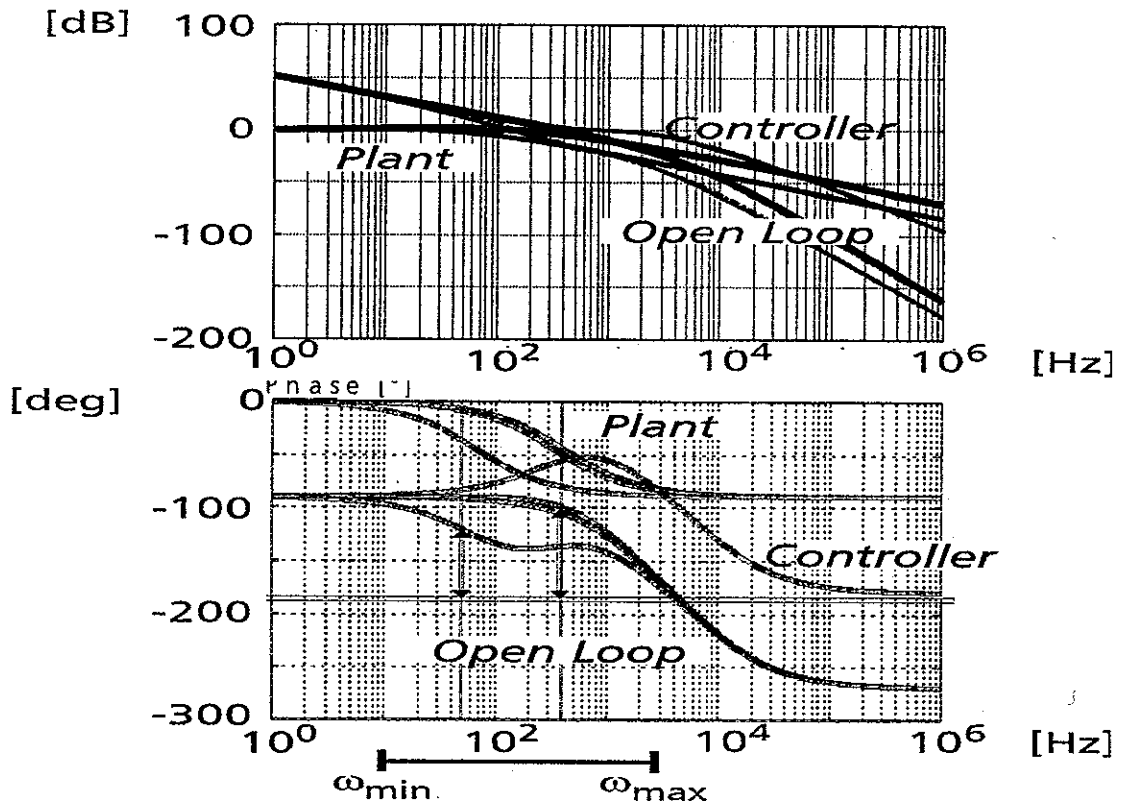


Figure 13 Frequency response of the plant, the nominal controller and the open loop.

plant is influenced only within a certain frequency range from  $\omega_{min}$  to  $\omega_{max}$  if the parameters are varied. Hence, it makes sense to lift the phase only inside that range in order to obtain a more robust phase margin. Using a transfer function

$$G_n(s) := k_0 \cdot \left[ \frac{1 + \frac{s}{\omega_L}}{1 + \frac{s}{\omega_H}} \right]^n \quad (6)$$

composed of  $0 < n < 1$  lead elements with cutoff frequencies  $\omega_L < 0.1 \omega_{min}$  and  $\omega_H < 10 \omega_{max}$  as well as some appropriate amplification  $k_0$  in serial connection to the controller will lift the phase of the open loop by  $n \cdot \pi/2$  (see Figure 14a). Since  $n$  is usually smaller than 1 we are forced to apply an approximation procedure by employing simple P, I or D elements. Using an approximation which consists of  $N$  of these basic elements with cutoff frequencies  $\omega_1, \dots, \omega_N$  and  $\omega'_1, \dots, \omega'_N$  yields a transfer function  $G_N$  given by

$$G_N(s) := k_0 \cdot \prod_{j=1}^N \left[ \frac{1 + \frac{s}{\omega'_j}}{1 + \frac{s}{\omega_j}} \right] \quad (7)$$

The corresponding frequency response is shown in Figure 15. This, again, results in a robust  $PID_N T_{q+1}$  controller

$$G_C^*(s) := G_C(s) \cdot G_N(s) \quad (8)$$

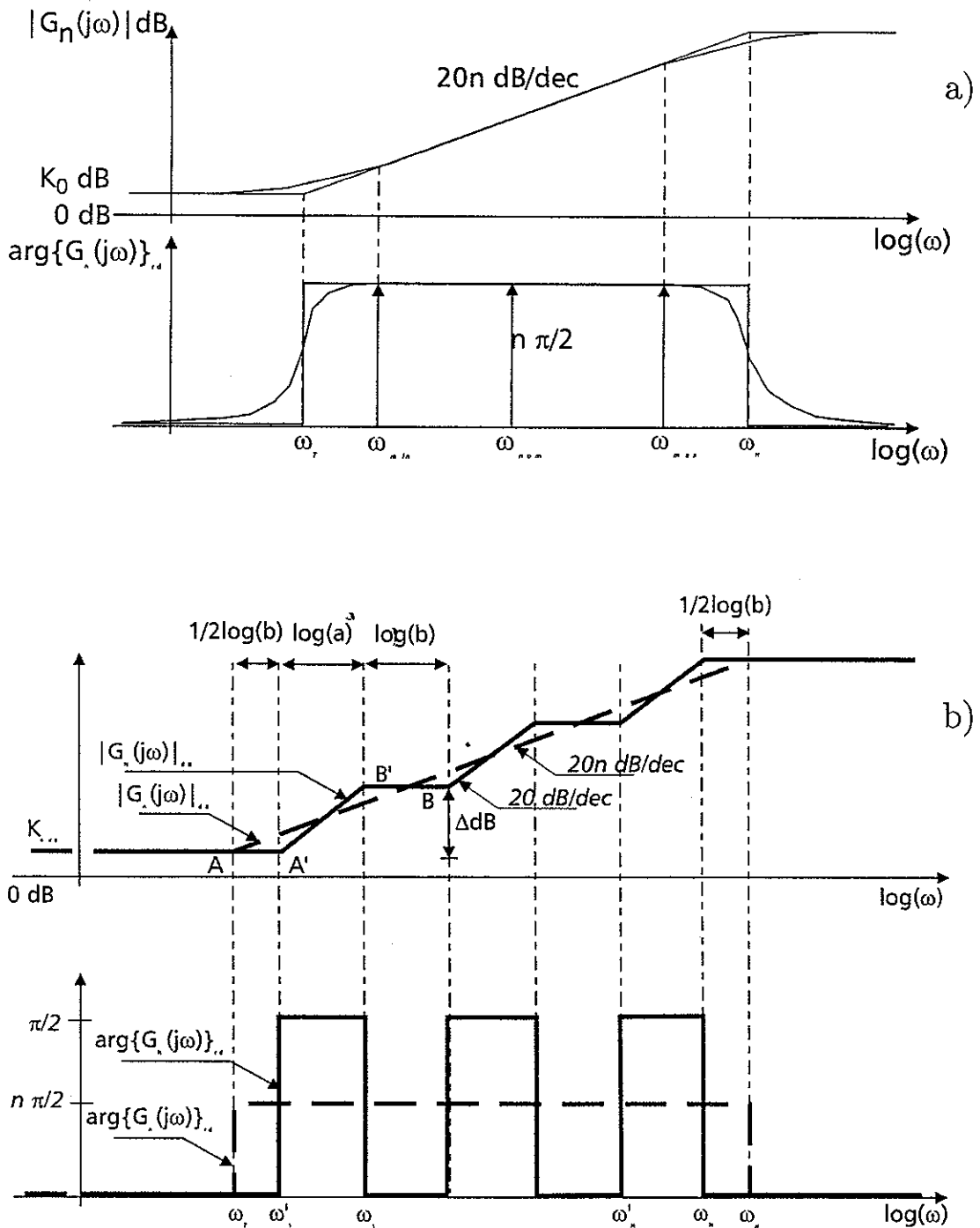


Figure 14 (a) Lift of the open loop phase margin. (b) Approximation  $G_N(i\omega)$  of  $G_n(i\omega)$ .

where  $q$ , in our case, is chosen to be 1. The integer number  $N$  relates to the approximation mentioned above (see also [1] for more details). The modified controller leads to the frequency response shown in Figure 15. The controller obviously lifts the phase inside the concerned frequency range and therefore leads to the desired result. Of course, the robust controller does not account for a better performance. On the contrary, the performance usually will deteriorate. This, however, is a general phenomenon which is encountered in the design of any robust concept. The right-hand side of Figure 16 shows the unit step response for different parameter settings. The robust controller leads to an extended settling time. On the other hand, overshooting will be significantly suppressed.

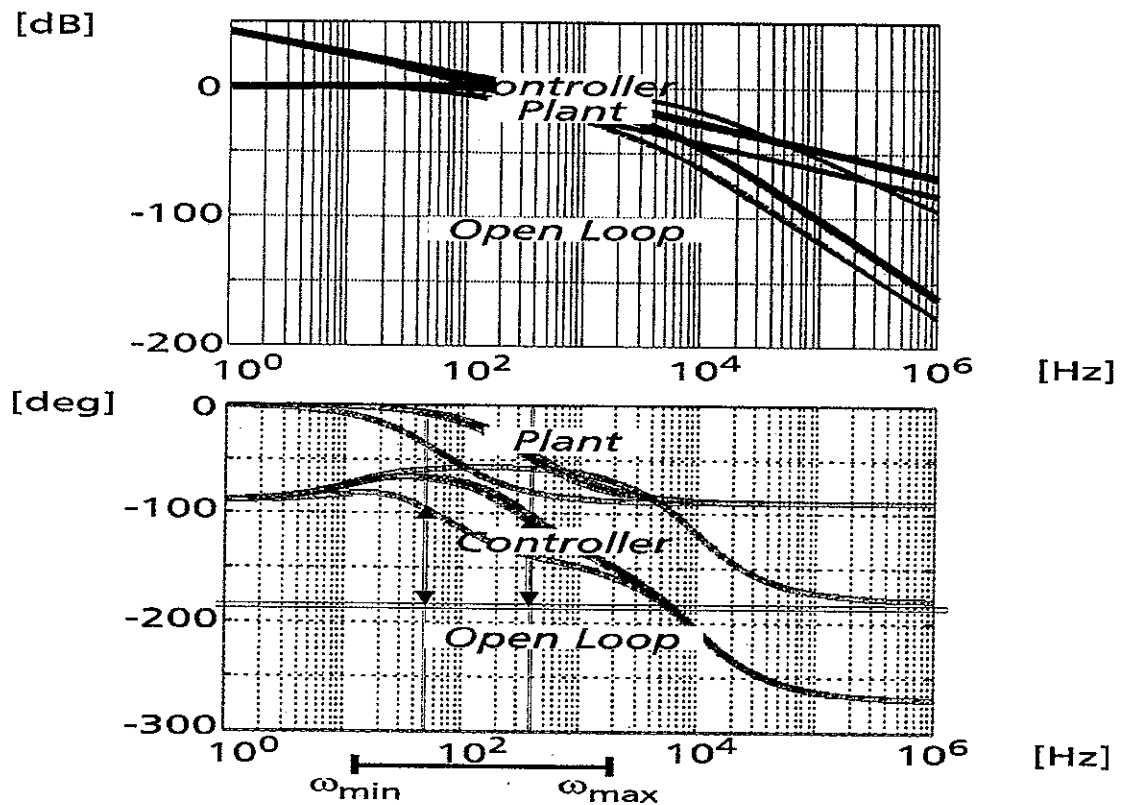


Figure 15 Frequency response of the plant, the robust controller and the open loop.

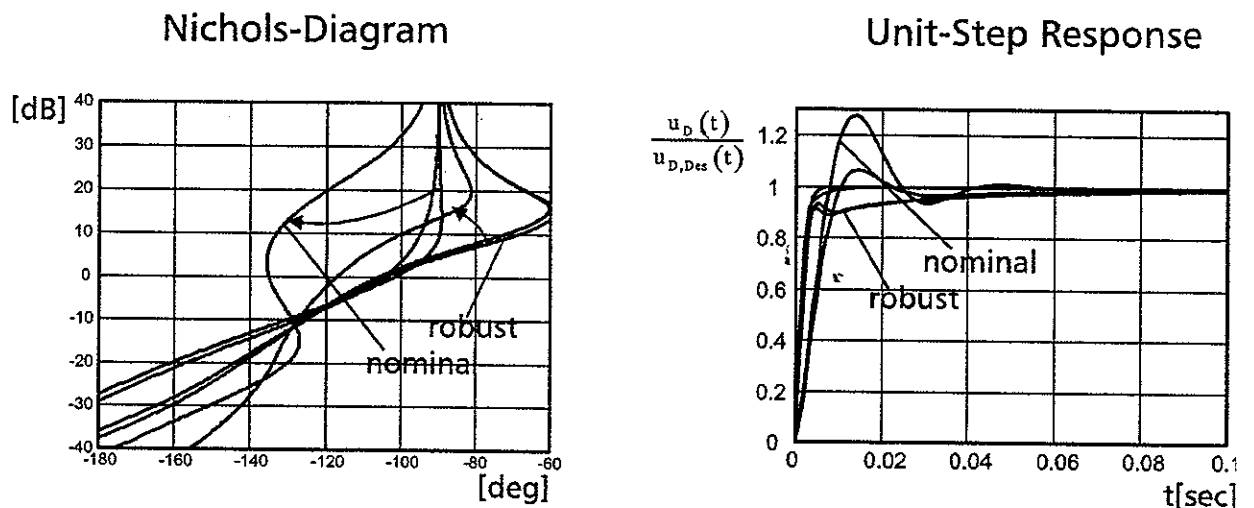


Figure 16 Closed loop behavior (nominal and robust control).

The left-hand side of Figure 16 shows the Nichols Diagram of the frequency response. Again, the results are shown for different parameter settings. The robust scheme leads obviously to a significant right shift of the characteristic curve and therefore to a more robust closed loop behavior.

In order to implement the control algorithm on some micro controller board, a time domain representation of the discrete time controller  $G_C$  is needed. Fortunately the nominal controller  $G_C$  as well as the robust controller  $G_C^*$  are given by

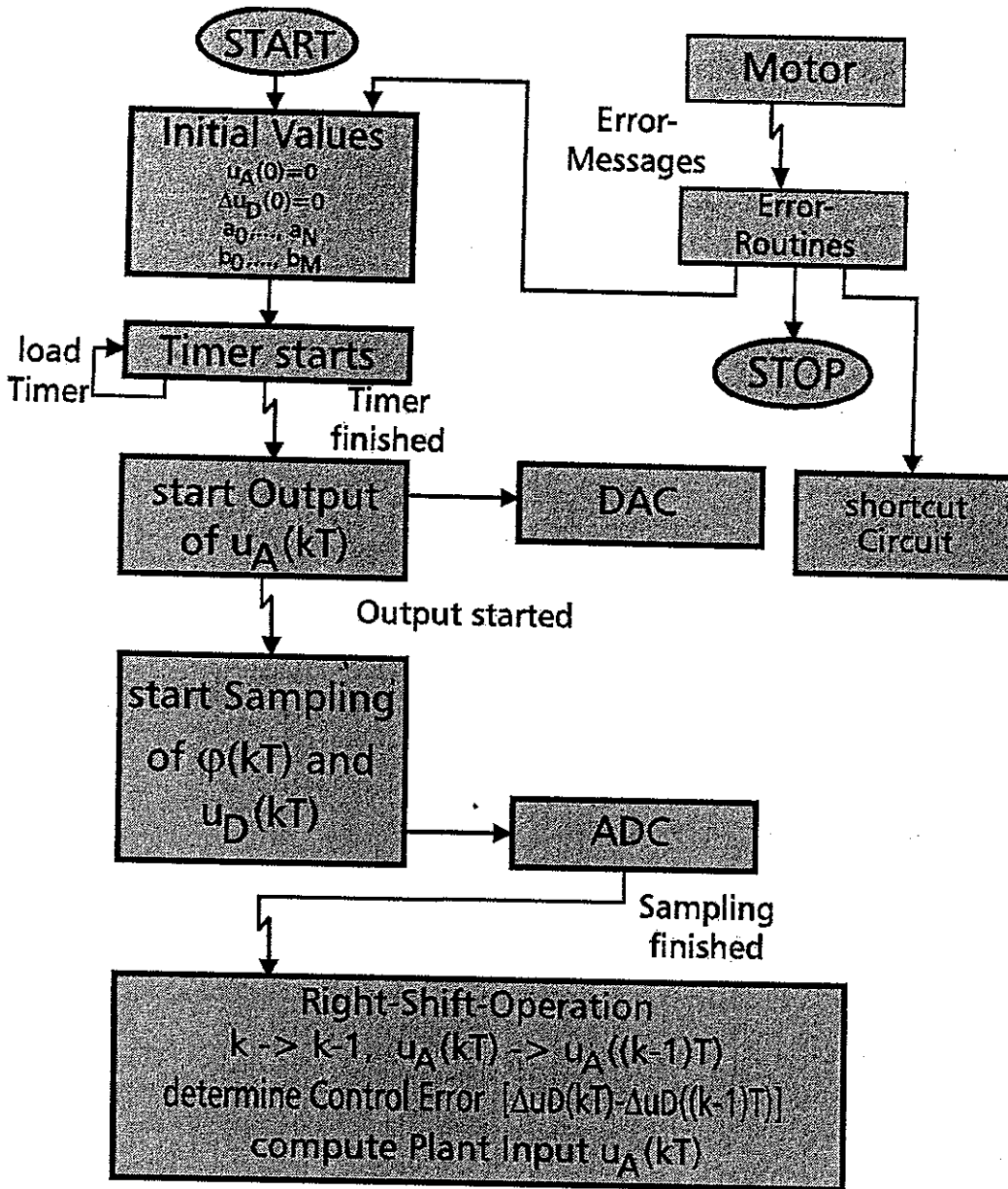


Figure 17 Implemented real time structure.

polynomial fractions in  $z^{-1}$ .

$$G_C(z) = \frac{U_A(z^{-1})}{\Delta U_D(z^{-1})} = \frac{\sum_{m=0}^M b_m z^{-m}}{\sum_{n=0}^N a_n z^{-n}}, \quad N \geq M. \quad (9)$$

This representation allows an easy application of the inverse Z-transformation. The result is an algorithmic mapping which gives the  $k$ -th control input  $u_A$  via the knowledge of the last  $M$  measured torque values and the last  $N$  computed control input values:

$$u_A(kT) = \frac{1}{a_0} \left[ \sum_{m=0}^M b_m \Delta u_D((k-m)T) - \sum_{n=0}^N a_n u_A((k-n)T) \right]. \quad (10)$$

See [2] for an optimization and real-time implementation of this algorithm. Compared to real-time programming, conventional programming does not allow for a fast response to external events, like a sensor or actuator failure. But this is exactly what is needed in security-related systems such as a steer-by-wire simulator. In addition, an interrupt-guided program eases the synchronization with respect to the sampling time  $T$ .

As shown in Figure 17, the implemented real-time structure is split into a series of independent parts. These parts are triggered via interrupts by external events. After setting the initial values a cycle timer will be loaded in time intervals  $T$ , where  $T$  denotes the sampling time. After each cycle an output routine is started by the first interrupt. It shifts the actual control input to the DAC register. The corresponding voltage will be kept at the torque motor until the next interrupt starts. Immediately after, a second interrupt starts the acquisition of the measurement values. The digitized values will be transferred to the micro controller where the control algorithm resides. The new control input value stays in the register until the timer starts the next cycle. In case an external failure is going to happen, the appropriate failure routine will be able to interfere at certain interrupt locations, depending on their interrupt priority. Finally, the triggered failure routine will carry out appropriate emergency procedures.

## References

1. Thielking, O. (1999) Entwurf und Realisierung einer diskreten Drehmomentregelung für das Lenkrad eines Steer-by-Wire Fahrzeuges. Diplomarbeit am Institut für Meß- und Regelungstechnik, Universität Hannover.
2. Budde, T. (2000) Komplettierung der Software und der mechanischen Komponenten eines mikrokontrollergeregelten Steer-by-Wire Systems. Diplomarbeit am Institut für Meß- und Regelungstechnik, Universität Hannover.
3. Krautstrunk, A., Uhler, R., Zimmer, M. and Mutschler, P. (2000) Elektrisch lenken: Handkraftaktor für Steer-by-Wire. Thema FORSCHUNG 1/2000, Seite 104–113.