Robust Control of a Radio Controlled Car Driving on Two Wheels

Daniel Robert Meier
danielme@student.ethz.ch
D-MAVT, ETH Zurich

***Abstract*— This paper presents the modelling and control of an RC-car driving on two wheels. In order to deal with a variety of uncertainties, a robust control approach is chosen. Therefore, perturbations are modelled in order to deal with actuator uncertainties as well as different geometries. The controller design with D-K iteration is compared to the design with a H_∞ controller.**

I. Introduction

This project presents a robust control approach in order to stabilize a radio controlled car on two wheels. Figure 1[1] shows this driving mode with a real car. A car on two wheels is an unstable and nonlinear system and therefore needs to be actively controlled. Due to several sources of uncertainty (such as geometry or an imprecise steering) a robust controller is desirable.

Fig. 1. Two-wheeled driving in a real car

The R/C car which will be modelled is a HPI Sprint, as depicted in Figure 2[2]. Its length is about 430mm with a weight of about 1.2kg. It can be controlled with the front wheels, which have an angle α, resulting in a turning rate Ψ. The car is actuated with a DC motor on the rear wheels and drives with the velocity v, which is assumed to be constant.

The controller has in- & outputs as depicted in Table I.

II. Prior Work

Arndt [1] presented an approach to control a car on two wheels, however not using an optimal controller. Moreover, their car has significantly bigger tires, more inertia and a higher center of gravity which facilitates control. Liu [2] developed a PID approach to control a car driving on two wheels.

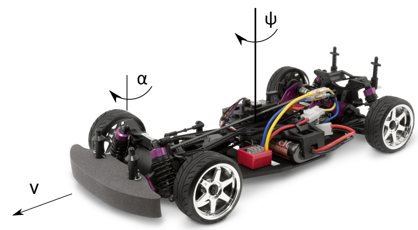

Fig. 2. HPI Sprint

	Input	Output
SISO	Roll angle ρ	Steering angle α

TABLE I

In- and outputs of the control system

III. Dynamic System Model

The system model consists of the steering mechanism, the roll angle dependant mapping of the steering angle and the vehicle body dynamics. It is set up as depicted in Figure 3.

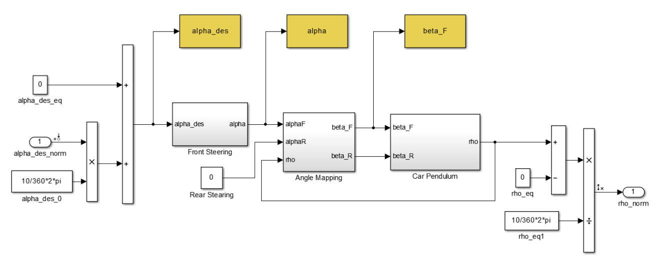

Fig. 3. Normalized plant

The angle ρ depicts the cars attitude as introduced in Figure 4[3]. Accordingly, there is one equilibrium point ($\dot{\Psi} = \dot{\rho} = \dot{\alpha} = 0$) which is dependant on the weight distribution of the car and its center of gravity (COG). This equilibrium point is also chosen at the operating point. As normalization factors, a steering angle $\alpha_0 = 10$ and a roll angle $\rho_0 = 10$ is chosen.

[1]Picture source: http://list25.com/
[2]Picture source: http://www.rcnitrotalk.com
[3]Source of car picture: http://www.4vector.com

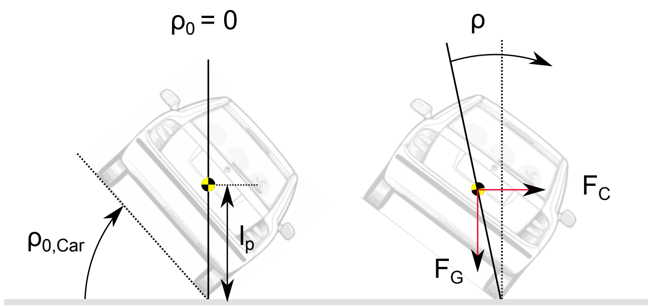

Fig. 4. Convention of roll angle: The state ρ depicts the angle around the equilibrium point (where the car's GOG is right above the line of the two wheels). $\rho_{0,car}$ is a geometrical constant which describes the cars attitude at the equilibrium.

A. Modelling Assumptions

- Constant velocity v. The dynamics of motor are slower than the dynamics of the steering and the balancing.
- Coriolis force is neglected, since it is small for small steering angles.
- Contact model: Tires have infinite grip and follow exactly the steering direction.

B. Steering

The front wheel steering is modelled as a second order system as described by equation 1. $\alpha_{desired}$ is the systems control input, ω and ζ are chosen to match the dynamics as realistic as possible.

$$\frac{d^2}{dt^2}\alpha = -2\zeta\omega\dot{\alpha} - \omega^2\left(\alpha - \alpha_{desired}\right) \tag{1}$$

To get the effective steering angle β, a nonlinear mapping is required. It is depicted by Figure 6 and implemented in the block *Angle Mapping*. It can be seen that steering inputs lead to high effective angles when operating at a high roll angle. Moreover the nonlinearity is more significant at higher roll angles ρ.

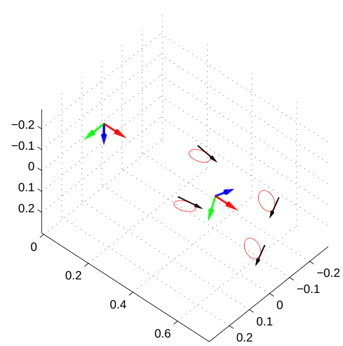

Fig. 5. Car with a non-zero roll angle. The effective steering angle β is different to the steering angle α

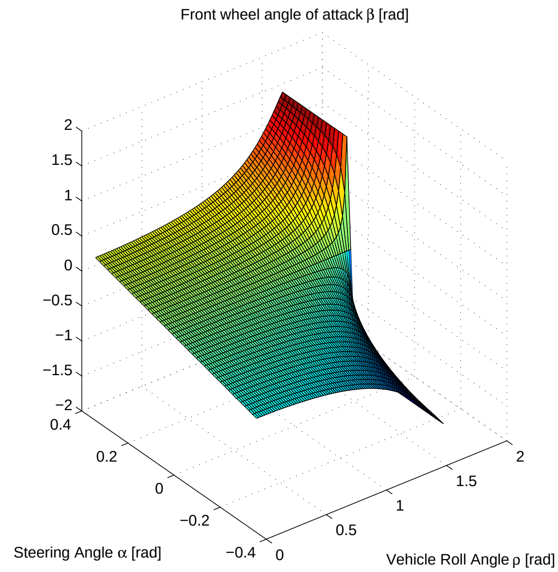

Fig. 6. Angle mapping: Steering angle α vs. effective steering angle β

C. Body dynamics

The vehicle is modelled as depicted in Figure 7. The gravitational force F_G and the centripetal force F_C are defined as:

$$F_G = m \cdot g \tag{2}$$

$$F_C = \cos\left(\gamma_c\right) \cdot m \cdot \dot{\psi}^2 \cdot l_{ZM} \tag{3}$$

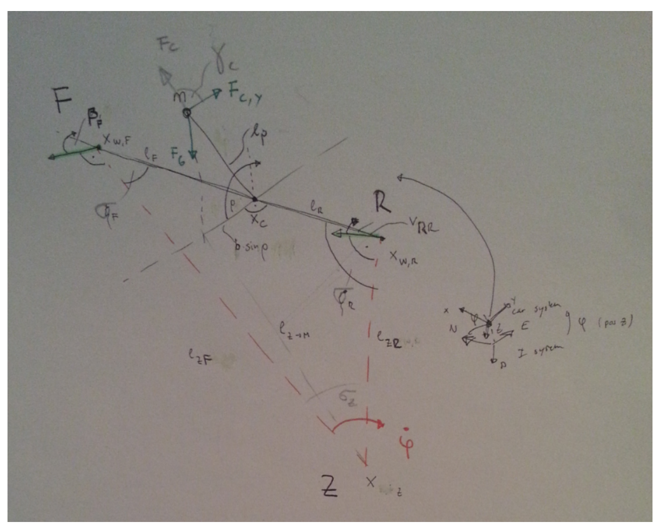

Fig. 7. HPI Sprint

m is the vehicles mass, l_{ZM} the distance between the center of rotation and the GOG and $\dot{\psi}$ the turning rate of the car. The factor $\cos\left(\gamma_c\right)$ takes into account that the centripetal force is not necessarily perpendicular to the roll axis of rotation of the car. This factor however can be neglected since it has only very small deviations from 1. With $\dot{\psi} = \frac{v}{l_{ZR}}$ and $l_{ZM} \approx l_{ZR}$, the centripetal force can therefore be expressed as:

$$F_C \approx m \cdot \frac{v^2}{l_w} \cdot \frac{\sin(\beta_F - \beta_R)}{\cos(\beta_F)} \quad (4)$$

v is the vehicles velocity, l_w the distance between the two wheels, β_F and β_R the front and rear effective steering angles resulting from the non-linear mapping. The resulting differential equation is given as:

$$\frac{d\rho}{dt} = \dot{\rho} \quad (5)$$

$$\frac{d\dot{\rho}}{dt} = \frac{1}{m \cdot l_p} (\sin\rho \cdot F_G - \cos\rho \cdot F_C) \quad (6)$$

D. States

As a result, the plant consists of the states shown in Table II.

Variable	Description	Unit	Operating point
α	Steering angle	rad	0
$\dot{\alpha}$	Steering rate	rad/s	0
ρ	Roll angle	rad	0
$\dot{\rho}$	Roll rate	rad/s	0

TABLE II
STATES OF THE NOMINAL PLANT

IV. Uncertainty Modelling

In order to deal with uncertainties, perturbations were introduced to the model. The uncertainty of the angle is modelled as a constant with value 0.1, which is equivalent to 1 degree. Also, the actuator has a uncertainty of 0.05 (0.5 degree).

$$W_{rho} = 0.1 \quad (7)$$

$$W_{alpha} = 0.05 \quad (8)$$

Performance weights are chosen as follows:

$$W_{rho,performace} = \frac{10}{10 \cdot s + 1} \quad (9)$$

$$W_{alpha,performance} = \frac{0.04 \cdot (1 + 0.4 \cdot s)}{100 + 0.1 \cdot s} \quad (10)$$

Noise is assumed to be constant over the whole spectrum. Disturbances are rejected in low frequencies.

$$W_{rho,noise} = 0.025 \quad (11)$$

$$W_{rho,disturbance} = \frac{2.5}{5 \cdot s + 1} \quad (12)$$

Figure 8 shows these weights graphically.

As a result, the weighted plant looks as depicted in 9. The inputs are described in Table III and the outputs in Table IV.

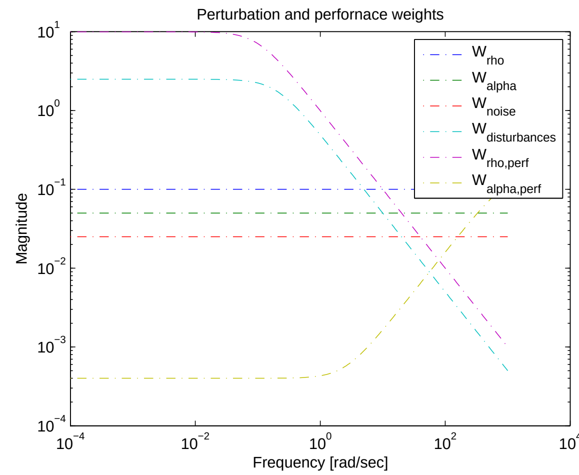

Fig. 8. Weights of the weighted plant

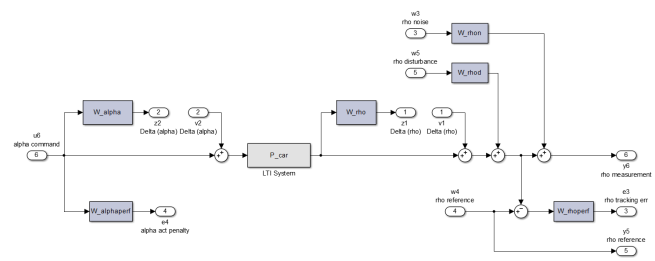

Fig. 9. Weighted plant

V. Robust Control Design

The robust control design consists of two parts: First, a H_∞ controller is developed using the weighted but unperturbed plant. Second, a D-K iteration is performed in order to achieve robust performance.

A. H_∞ Controller Design

For the controller design, the perturbations were removed. As a next step, a H_∞ controller was designed with MATLAB's function `hinfsyn()`. For simulation purposes, the unweighted plant as shown in Figure 10 is used. A robustness analysis for the resulting H_∞ controller is shown in figure 11.

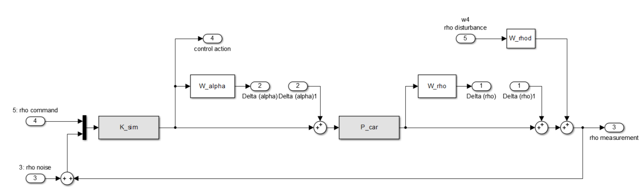

Fig. 10. Unweighted plant

B. D-K iteration

As a next step, a D-K iteration was performed. This had the result of an increased robustness, as Figure 12 shows.

Vector	Description
$v1$	Delta (ρ)
$v2$	Delta (α)
$w3$	ρ noise
$w4$	ρ reference
$w5$	ρ disturbance
$u6$	α command

TABLE III

INPUTS OF THE WEIGHTED PLANT

Vector	Description
$z1$	Delta (ρ)
$z2$	Delta (α)
$e3$	ρ tracking error
$e4$	α actuation penalty
$y5$	ρ reference
$y6$	ρ measurement

TABLE IV

OUTPUTS OF THE WEIGHTED PLANT

In order to compare the performance, a step response is shown in Figure 13. It can be seen that the system acts slower but with fewer oscillations.

VI. CONCLUSIONS

blablabla

REFERENCES

[1] ARNDT D. ET AL.: *Two-Wheel Self-Balancing of a Four-Wheeled Vehicle*. Article in IEEE Control Systems, 2011

[2] LIU K. ET AL.: *Two-wheel self-balanced car based on Kalman filtering and PID algorithm*. Conference Paper, IEEE IE&EM 2011

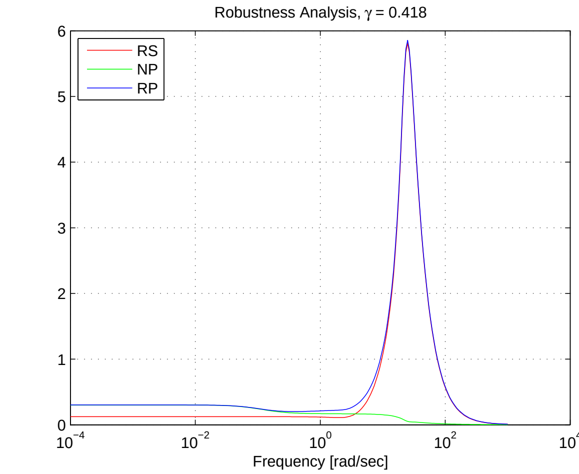

Fig. 11. Robustness analysis of a H_∞ controller

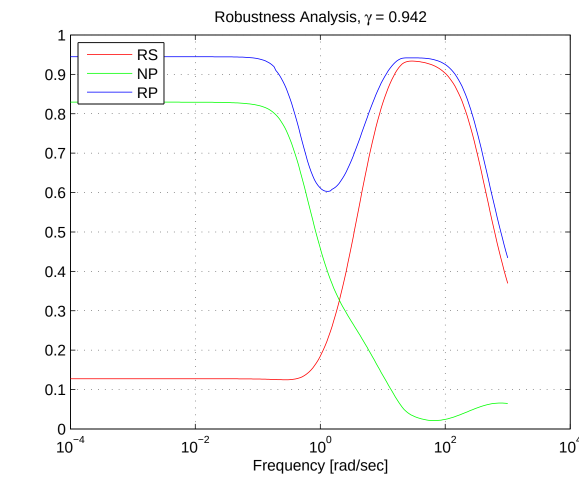

Fig. 12. Robustness analysis after D-K iteration

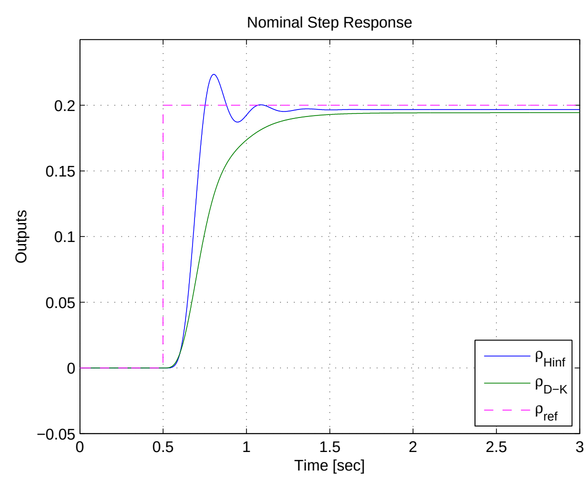

Fig. 13. Step response for different controllers