# Approach on a model based current regulator design for an electric drive unit using a holistic system design with driver and driving cycle

Harry Ott, M.Sc.
*(Cologne University of Applied Sciences, Germany*
*Uppsala University, Sweden);*

René Degen, M.Sc.
*(Cologne University of Applied Sciences, Germany*
*Uppsala University, Sweden);*

Prof. Dr. Eng. Mats Leijon
*(Uppsala University, Sweden);*

Prof. Dr. rer. nat. Margot Ruschitzka
*(Cologne University of Applied Sciences, Germany);*

## Abstract

Model based engineering is especially for the development of high performing control systems essentially. By means of suitable simplifications, they help to present technical relationships and express them mathematically. Thereby, active controllers to influence the system behavior could be developed in an efficient and reliable way.

This paper deals with the design of a holistic simulation environment for an e-bike with a wheel hub motor in the rear wheel and a torque and speed sensor in the bottom bracket. A model-based approach to development using rapid control prototyping is chosen. The model design is chosen similar to the system design of the control system. The interfaces between the main models are also the interfaces of the later controller, which makes it easier to implement the system afterwards. The engine dynamics has been tested and adjusted on the model using a driving cycle. A special focus is on the interpretation of the drivers inputs by the bottom bracket sensor. At the interface between the sensor and the subordinate engine control system, any desired driving condition can be set for different types of drivers and driving situations by means of different characteristic curves.

The scenarios investigated are derived from typical simulations needed during the development of e-bike drivetrains. They focus on the interaction of the hybrid system consisting of human driver and engine torque. Especially the synchronization of torques and the reaction to fast increasing stimuli were investigated.

The results show a valid performance of the developed algorithm. The e-motor torque oscillates quickly and the synchronization works fine. Additionally, the algorithm to smooth the pedaling fluctuations and thereby the torque fluctuations work quite well, whereby a smooth torque is implemented. Next steps are the integration of supporting modes and the demand-orientated control.

**Keywords**

e-bike control, model-based control, permanent magnet-excited synchronous machine, powertrain simulation, rapid control prototyping

## [redacted] Introduction

The e-bike industry has been booming for many years now and the market continues to grow [1]. The number of e-bikes sold is constantly increasing. Interest is no longer limited to private consumers who have rediscovered the fun of cycling with less effort. Companies are also looking for vehicles which, especially in large cities, can be moved locally with zero emissions and virtually independent of traffic. On the one hand, this offers great potential for expanding the e-bike market, for example through the area of cargobikes, and on the other hand, new demands are placed on existing drive solutions.

At present, the drive train topology for e-bikes that dominates the market is the mid-motor. It is distributed by a few large manufacturers, even though it has considerable disadvantages, for example the conventional bicycle frame can no longer be used and has to be extensively redesigned in the area of the bottom bracket.

High potential is offered by wheel hub motors in the rear wheel, which can be retrofitted into all conventional bicycles, as well as having range and safety advantages in the future. Previous solutions are either very expensive or of inferior quality, often both points are even true. For example, a simple step detection system is usually used which does not exploit the full potential of the drive. In addition, an adaptation of existing controllers to other motors or applications is only available to a limited extent.

## [redacted] Problem formulation

As a basis for a well-designed and individually adaptable engine control system, a simulation environment is to be created in which both engine control and the interpretation of driver inputs can be simulated. Thereupon a design of the control system should be possible already in the simulation for a later realization using rapid control prototyping (RCP) [2].

The entire powertrain is a hybrid system of driver and permanent magnet-excited synchronous machine (pmsm). Both drive systems have a significant influence on the propulsion of the wheel and influence each other. The system should be designed in such a way that the driver has the most natural driving experience possible. The only information available to the software for engine control comes from the torque sensor, more precisely, the torque generated by the left pedal and the cadence. The current gear ratio of the drive crank to the rear wheel is not known, like in real applications.

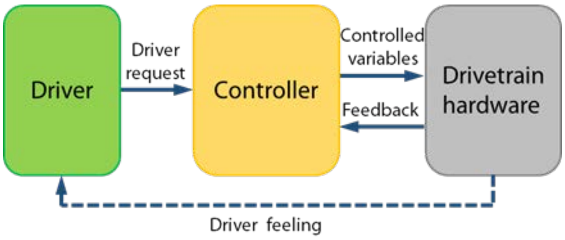

*Figure 1: Overall system with driver, controller and drivetrain hardware*

Apart from the natural riding feel, an adjustability for different riding modes is desirable. Since there are very different types of riders and situations in which the bike is to be used, it must be possible to adapt it almost completely to different riding profiles.

## 2.1 System definition

The system consists of an e-bike with a pmsm in the rear wheel, a cadence and torque sensor in the bottom bracket and a development board with microcontroller and power electronics. Three Hall sensors with an offset of 120° can determine the motor position. The power electronics consists of six MOSFETS, which can be switched selectively. The torque sensor in the bottom bracket measures the torsion at the middle of the shaft and can therefore only detect the torque generated by the left pedal. At the same time, two Hall sensors and eight pairs of poles can be used to determine the cadence and pedaling direction of the rider. The bike also has a derailleur system with 3x9 gears.

## [redacted]Model based approach

For the model-based approach the control design of the system is used as shown in figure 1. It consists of a model of the driver, the software on the controller and the hardware of the bicycle. The interfaces form the signal variables, which also come out of and into the controller during the later implementation of the system. A special feature is the driver feeling, which is shown as a dashed line in the figure 1. This is only used for simulation purposes and is not available in practice. However, it must be considered urgently for the control design and simulation, since the driver changes his own introduced force depending on the engine support.

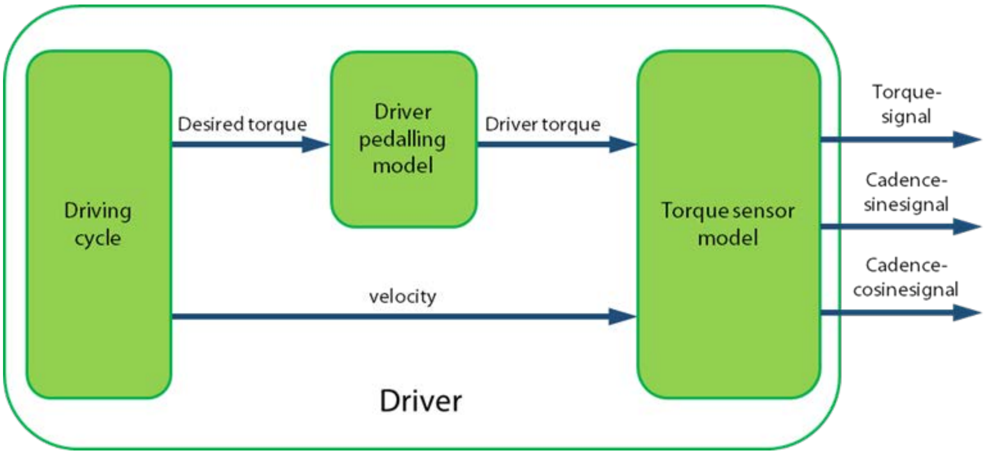

*Figure 2: Driver model*

Overall, this results in a cascade control with the engine control as inner loop and the driver as outer loop. The motor controller is designed in the rotor fixed coordinate system in d and q direction as described in [3].

## 3.1 Driver model

The driver model (figure 2) represents an important part of the overall system. Here, a possible driver or rather his driving behavior, is modeled. The basis for this is a driving cycle, from which a demand moment and the corresponding driving speed are determined. The actual momentum, which the driver introduces into the drive system at the pedals, results from the demanded torque. Based on these variables, the torque sensor can be modelled and the signals can be determined, which the sensor transmits to the controller during the subsequent ride.

### 3.1.1 Driving cycle

In order to generate plausible values as torque signals, it requires a drive cycle model which calculates the required torque and speed for a given target curve. The required torque is determined from the driving resistances. The required driving force $F_{Drive}$ is the sum of the driving resistances as described in [4] or [5].

$$F_{Drive} = F_{Acc} + F_{Air} + F_{Fric} + F_{Grad} \tag{3.1}$$

According to Newton's second axiom, the acceleration resistance $F_{Acc}$ corresponds to the product of mass $m$ and acceleration $a$ of the system.

$$F_{Acc} = ma \tag{3.2}$$

The air resistance $F_{Air}$ results from the air density $\rho_{Air}$, the drag coefficient $c_w$, the projected cross-sectional area $A$ and the square of the velocity $v$ according to the following formula:

$$F_{Air} = \frac{1}{2}\rho_{Air}c_w A v^2 \tag{3.3}$$

The friction force $F_{Fric}$ results from the tire friction and is dependent on the tire friction factor $f_R$ and the corresponding contact force of the wheel, which is the weight force of the system perpendicular to the road:

$$F_{Fric} = f_R mg\cos(\alpha) \tag{3.4}$$

The other component of the weight force is the gradient resistance, which acts parallel to the road surface.

$$F_{Grad} = mg\sin(\alpha) \tag{3.5}$$

The required drive torque$T_{Drive,res}$ can now be determined from the required drive force described above and the corresponding tire radius $r_{Tire}$.

$$T_{Drive,res} = F_{Drive}r_{Tire} \tag{3.6}$$

This must be made up of the applied torque of the $T_{Motor}$ and the proportion of $T_{Driver}$, which the driver must apply himself.

$$T_{Drive,res} = T_{Motor} + T_{Driver} \tag{3.7}$$

As a simplified assumption for the driver model, a proportional behaviour from engine torque to driver torque determined by the factor $k$ is specified.

$$T_{Motor} = k\, T_{Driver} \tag{3.8}$$

The actual moment that the driver has to apply in order to complete the driving cycle is therefore related as follows:

$$T_{Driver} = (1-k)\, T_{Drive} \tag{3.9}$$

### 3.1.2 Driving pedal model

The challenge now is to convert the driver's torque calculated from the driving cycle into the torque actually acting on the crank. Since the driver can almost only step on the pedals from above, a torque is generated over the entire crank revolution, which is similar to a sine. Since the torque sensor can only measure the torque in the left crank arm anyway, only this side is considered. The course of the rider's torque over the entire crank revolution is shown in idealized form in the left graph of figure 3. The first half of the crank revolution reflects the front half of the crank revolution, in which the driver can apply a force. On the second half of the crank revolution, the rider no longer applies active force to the pedal. However, since he leaves his foot on the pedal, the weight of the foot and leg acts on the pedal. This effect was also investigated by [6].

The idealized course is thus determined by the following formula:

$$T_{Pedal}(\varphi_{Pedal}) = \begin{cases} \sin(\varphi_{Pedal})\,, & 0 \leq \varphi_{Pedal} \leq \pi \\ \mathrm{k} \cdot \sin(\varphi_{Pedal}), & \pi < \varphi_{Pedal} < 2\pi \end{cases} \tag{3.10}$$

The factor $k$ compresses the sine wave to simulate the foot resting on the pedals. Since the torque signal is dynamic in this form and the motor controller receives a constant value as input in the best case, the signal must be smoothed. Assuming that the torque signal calculated from the driving cycle is constant over an entire pedal revolution, the following conclusion can be drawn from the integrals of the curves:

Since the torque signal is dynamic in this form and the motor controller receives a constant value as input in the best case, the signal must be smoothed. Assuming that the torque signal calculated from the driving cycle is constant over an entire pedal revolution, the following conclusion can be drawn from the integrals of the curves:

$$r \cdot \left[ \int_0^{\pi} \sin(\varphi_{Pedal})\, d\varphi_{Pedal} + \int_{\pi}^{2\pi} k \cdot \sin(\varphi_{Pedal})\, d\varphi_{Pedal} \right] \tag{3.11}$$

$$= \int_0^{2\pi} d\varphi_{Pedal}$$

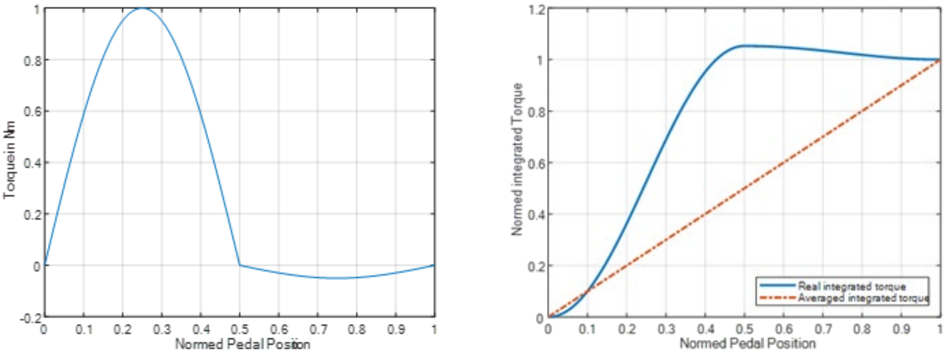

*Figure 3: Normed driver torque over pedal position (left) and integrated driver torque for one crank revolution (right)*

The first part of the equation is the integral of the sine waveform shown above. The second part is the integral of the assumed constant torque. The factor $r$ is a scaling factor, which scales the sine signal according to the choice of $k$ in such a way that the equation is fulfilled. $r$ must therefore be chosen in such a way that the total torque applied over one crank revolution corresponds to the constant course. The equation is therefore solved after $r$ to obtain the relationship between $k$ and $r$:

$$r = \frac{\pi}{1-k} \tag{3.12}$$

The right graph in figure 3 shows the integrated torque curves for $k = 0.05$ and corresponding $r$. The integrated torque of the sine wave rises faster initially and exceeds the final value. However, due to the negative component in the second half of the crank revolution, the final value is the same at the end. The scaled sine signal can now be multiplied by the demanded torque of the driver $T_{Driver}$. Since again only the torque at the left pedal is considered, a factor of 0.5 is added to the formula:

$$T_{Driver,real}(\varphi_{Pedal}, t) = \tfrac{1}{2} T_{Pedal}(\varphi_{Pedal}) T_{Drive}(t) \tag{3.13}$$

Figure 4 (left) shows as an example a driving cycle and the required torque of the driver, which he actually applies. A constant transmission ratio is assumed here. The lower graph shows that the cadence increases with increasing speed. The torque peaks also increase in the acceleration phase, while they are consistently low at constant speed.

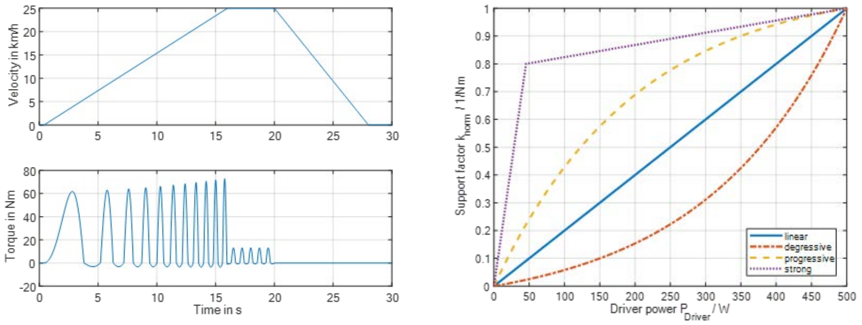

*Figure 4: Velocity and driver torque in an example driving cycle (left) and characteristic curves for the support factor k (right)*

### 3.1.3 Torque sensor model

The torque sensor model should simulate the measurements of torque and pedal speed and, depending on the input values, output the actual sensor signals, which can then be read from the evaluation controller. The inputs come from the rider model described above. The analog output signal results from the read-in sinusoidal torque of the rider and a conversion according to the characteristic curve of the sensor.

From the input signal for the pedal rotations the corresponding sine and cosine signal must be generated with special consideration of the direction of rotation. At this point it is assumed that the bicycle has a fixed gear ratio $i$. The pedalling speed $\Omega_{Pedal}$ then results from the speed from the cycle and the wheel circumference. If it is integrated, this results in the current pedal angle $\varphi_{Pedal}$. The cosine or sine of this angle is used to generate the corresponding signals of the torque sensor. In order to map a reverse pedaling, the two signals must be swapped with a negative sign of $\Omega_{Pedal}$.

### 3.2 Controller function modelling

The software of the controller is structured as shown in figure 5. The inputs of the model are on the one hand the signals of the torque sensor, on the other hand the sensor values from the motor required for control. The torque signals are evaluated and provide the setpoints for the motor current controller. This calculates the setpoints of the motor voltage, which are then transferred to the power electronics as duty cycles via the space vector modulation. The actual current values required for this are recorded by means of current sensors, the motor position and speed by means of Hall sensor interpolation. The interfaces of the error handler function are not explicitly drawn in the figure 5. This is a monitoring of the entire system, which, for example, switches off the system in the event of excessive temperatures or currents to prevent destruction.

### 3.2.1 Torque sensor evaluation

In torque sensor evaluation, the models for driver and sensor are inverted. The cosine and sine signals are sampled and the sectors are determined based on the model of an encoder. The speed and direction of rotation can then be calculated from the sector sequence as described in [7]. Ideally, the torque should also correspond to a constant value at constant speed. Since the torque is only picked up on one side, an averaged torque must first be determined from the contained sequence. According to the principle explained in "Driver pedal model", the torque can be reintegrated at this point for each revolution of the crank. At the same time the average value can be determined by measuring the time for the last pedal revolution. Via a holding element, which is updated after each pedal revolution, periodically constant torque recordings are created. After normalization, they then form the current setpoints for the motor controller.

At this point, the desired riding behaviour can be impressed on the e-bike. For example, multi-dimensional characteristic maps can be used to convert the now completely processed rider inputs into the target values for the motor. The input values of the maps consist of

- the torque of the driver $T_{Driver}$
- the pedalling speed of the driver $\Omega_{Pedal}$
- the speed of the bicycle $v$ and
- the level of support chosen, if applicable.

Additional complexity is brought into the control system by the gearshift system installed on the bicycle. If only one gear is available, there is always a fixed relationship between $\mathrm{T}_{Pedal}$, $\Omega_{Pedal}$ and $v$. Accordingly, in this case a fixed assignment between the inputs and the target currents can be made. If different levels of support are implemented, this relationship can be changed proportionally. If a gear shift is installed on the bicycle, which is assumed in this case, this assignment is no longer possible.

The remedy can be taken via a power control. From the rider torque and the pedalling speed the power $P_{Driver}$ is then calculated.

$$P_{Driver} = T_{Driver} \cdot \Omega_{Pedal} \tag{3.14}$$

Now the target torque of the motor can be determined from the following relationship.

$$P_{Motor,ref} = k\ P_{Driver} \tag{3.15}$$

Since the speed of the motor $\Omega_{rot}$ is also determined and known by the current driving status, the formula can be extended to the following form:

$$T_{Motor,ref} \cdot \Omega_{rot} = k\ P_{Driver} \tag{3.16}$$

Inserting the equations into each other and resolving them according to $T_{Motor,ref}$ results in:

$$T_{Motor,ref} = k\ \frac{T_{Driver} \cdot \Omega_{Pedal}}{\Omega_{rot}} \tag{3.17}$$

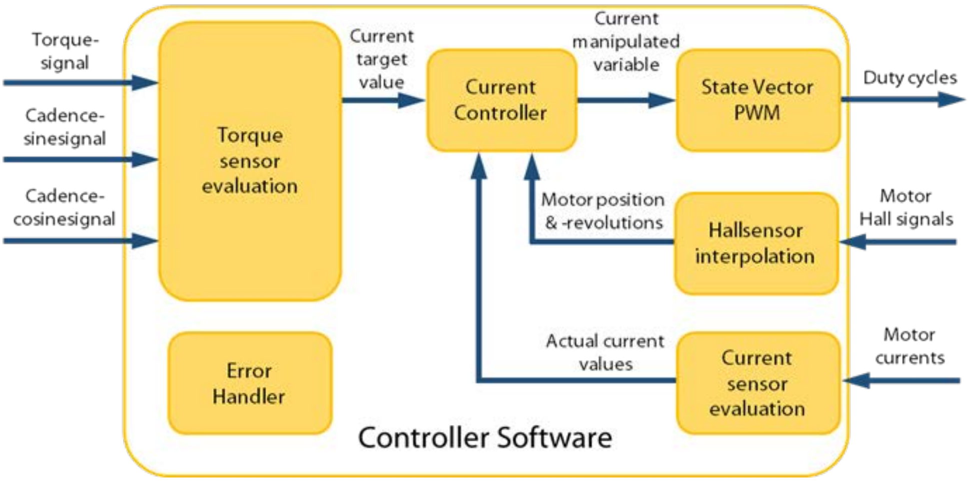

*Figure 5: Controller software models*

Since the current of the motor according to [3] is proportional to the torque, the torque-forming reference current $i_{q,ref}$ results:

$$T_{Motor,ref} = k\ \frac{T_{Driver} \cdot \Omega_{Pedal}}{\Omega_{rot}} \tag{3.18}$$

If $T_{Motor,ref}$ is again inserted into this equation, it follows as:

$$i_{q,ref} = k\frac{2}{3Z_p\Psi_{pmax}}\frac{\Omega_{Pedal}}{\Omega_{rot}}T_{Driver} \tag{3.19}$$

The term $\frac{2}{3Z_p\Psi_{pmax}}$is constant and $\frac{\Omega_{Pedal}}{\Omega_{rot}}$ represents the gear ratio of the circuit. The factor k can determine the degree of support of the motor via a characteristic diagram. The map can be adapted for different driving situations or driver types. Due to the dependency of $T_{Driver}$ and $\Omega_{Pedal}$, the current setting should work for both high and low cadence as well as for high and low torque.

For maximum motor efficiency, the current $i_{d,ref}$ is ideally set to zero. Figure 7 shows possible curves for a normalized support factor k$_{norm}$ as a function of the driver's performance. A maximum k$_{norm}$ corresponds to the maximum available motor torque. The linear characteristic curve depicts a very general case. The progressive characteristic curve corresponds to a more sporty design. The degressive characteristic curve only produces the higher engine power when the driver applies a large amount of power. With the very strong support, the engine torque is already very high at very low driver inputs.

Overall, the response of the engine can therefore be controlled via the characteristic curve. In order to create clean transitions between the current and torque setpoints and to avoid excessive jerking of the engine, a filter can be connected downstream to smooth the setpoint curves.

### 3.2.2 Current control

The direct motor controller is designed as current control with reference variable feedforward. This is the standard action with transforming the three motor phases in the rotor-fixed coordinate system by using Park-Clarke-transformation. The transformations and the current controller design is developed as in [8] or [9]. The setpoint variables are converted into the duty cycles for the inverter half-bridges using the space vector modulation method as described in [10]. The current rotor position and the rotor speed are determined from a Hall sensor interpolation, as is common with encoders. The motor currents are calculated by means of two measured phases and transformed into the rotor fixed coordinate system required for control. An single shunt measurement as described in [11] is also possible.

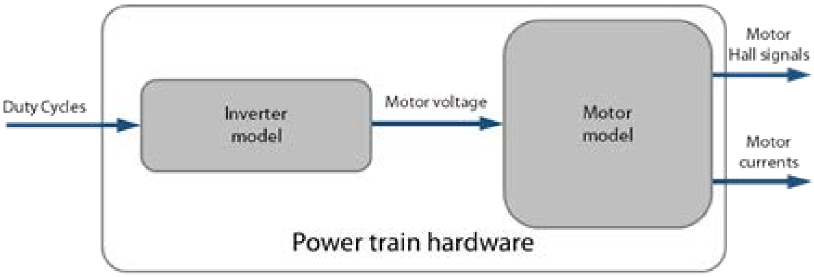

*Figure 6: Powertrain hardware model*

## 3.3 Power train models

This chapter deals with the drive hardware. The aim is to model the real components as simply as possible, but also as accurately as necessary, in order to test the controller in the simulation. For this reason, they are a very important part of the overall system when the controller functions are developed using RCP.

In other words, the drive hardware maps the actuators, the controlled system and parts of the sensor technology. One component is the power electronics, which is reflected in the inverter model, the other is the motor, which is located in the associated motor model. These two components together are the actuator of the system. The controlled mechanical system and sensors are also modelled in the motor model, because both are linked together very closely.

Figure 6 shows the structure and the interfaces of the drive hardware. The input, coming from the controller, transfers the duty cycles for the inverter half bridges. In the inverter model these are converted into the actual phase voltages and passed on to the motor model. From these, the Hall signals and the motor currents can be determined, which form the outputs to be read in the controller.

### 3.3.1 Inverter model

The inverter model is derived from [13] and can be looked up there. The resulting equations are the following:

$$u_a = u_{ZK}\left(\frac{2}{3}s_a - \frac{1}{3}s_b - \frac{1}{3}s_c\right) \quad (3.20)$$

$$u_b = u_{ZK}\left(\frac{2}{3}s_b - \frac{1}{3}s_a - \frac{1}{3}s_c\right) \quad (3.21)$$

$$u_c = u_{ZK}\left(\frac{2}{3}s_c - \frac{1}{3}s_a - \frac{1}{3}s_b\right) \tag{3.22}$$

### 3.3.2 PMSM model

The motor equations can be set up using these equivalent circuit diagrams with the help of Kirchhoff's mesh rule. $R_d$ and $R_q$ are the string resistances, which are normally equal in size and are therefore combined to $R_{Str}$. $L_d$ and $L_q$ are the string inductances which vary greatly depending on the design of the motor and therefore cannot be combined. The equations contain non-linear terms which are due to the flux linkage $\Psi_{pmax}$ of the motor and additionally depend on the rotor $\Omega_{el}$ [13] [14].

$$u_d = R_{Str}i_d + L_d\frac{di_d}{dt} - \Omega_{el}L_qi_q \tag{3.23}$$

$$u_q = R_{Str}i_q + L_q\frac{di_q}{dt} + \Omega_{el}\left(L_di_d + \Psi_{pmax}\right) \tag{3.24}$$

The torque is calculated by the following equation:

$$T_{el} = \frac{3}{2}Z_p\left[\Psi_{pmax}i_q + \left(L_d - L_q\right)i_di_q\right] \tag{3.25}$$

The electrical torque calculated here and the driver torque derived from the driver model together form the total drive torque. The current speed of the system is then calculated from this using the formulas for the driving resistances (see chapter "driving cycle").

## [redacted] **Simulation and results**

The specified system is implemented in Matlab Simulink as described. For the simulation, a ramp is run as a driving cycle, which is shown in figure 7. It is first accelerated to 50 km/h, this speed is held briefly and then decelerated to zero again.

The resulting power and torque are shown in the figure 7 as well. The parameters are adjusted in such a way that the driver and engine deliver almost the same power. Since the driver and engine intervene in the system at different points in the powertrain, the gear ratio of the gearshift has been deducted for comparability of the graphs.

As can be seen in the graphs, the driver must first overcome a short starting torque until the engine starts. Due to a proportional characteristic diagram in the torque sensor evaluation, the engine torque is proportional to the driver torque after a short settling period. During the transition to constant speed, the controller has to brake briefly for the driver to absorb the change in speed and counteract the changing inertia of the driver. It is to be expected that this effect will be less in real life, because a real driver would normally not try to maintain the given speed so abruptly. At the transitions between the discontinuities in the excitation, it can be seen, that the torque changes in a proportional behavior with the first order deceleration. This is a desirable effect, since too large torque jumps in the engine would lead to a jerk in the wheel. This effect is automatically achieved by the dynamic behavior of the current control loop. During the braking phase, the motor torque is controlled to zero and the driver has to brake the wheel with the mechanical brakes.

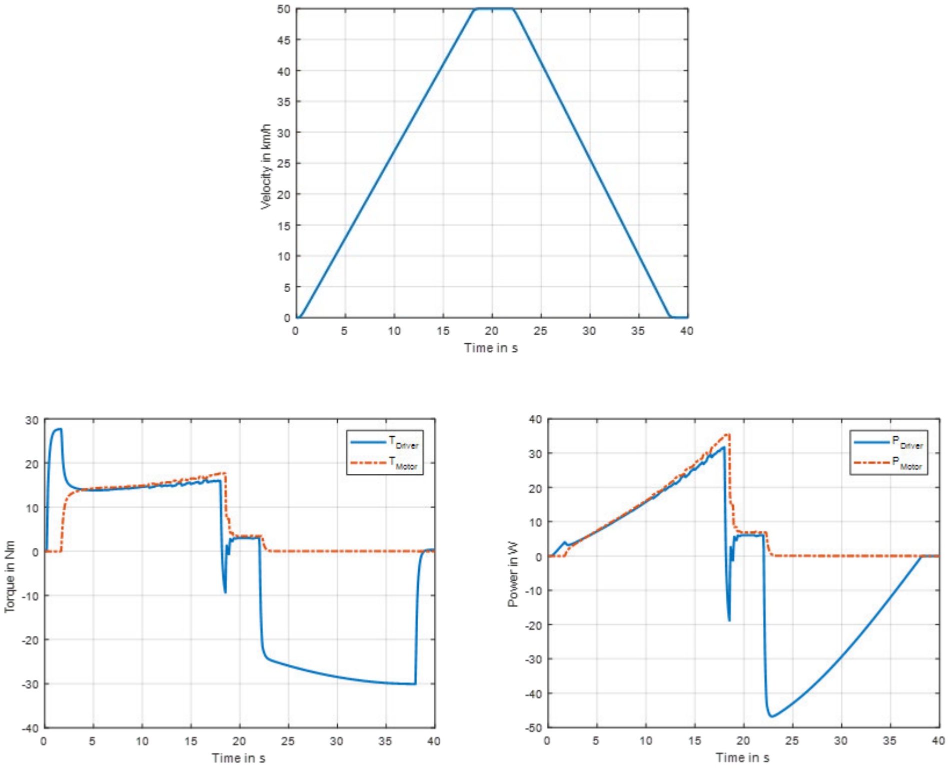

*Figure 7: Input driving cycle for overall system simulation (top center), driver and motor power in overall system simulation (bottom left) and driver and motor torque in overall system simulation (bottom right)*

## [redacted] Outlook and review

The simulation of the entire powertrain makes it easy to interpret and evaluate driver behavior. The model-based structure allows the individual components of the controller software to be tested and optimized. The direct response behavior of the engine and the setting of the required control variables can be mapped in the simulation. During the real implementation, only little fine-tuning is to be expected due to additional implementation effects.

The interpretation of the rider's inputs is the decisive component for the control of an e-bike motor, which takes a long time when designing the drive train. Due to the different rider types and environmental conditions that exist in reality, it is difficult to find even basic configurations of the controller. The model shown in this article allows not only to evaluate the rider's inputs, but also to adjust the appropriate engine behavior to meet the rider's specific needs.

Since the models and their interfaces were designed directly for the target hardware, they can be further tested in the next step with automatic code generation in software-in-the-loop and hardware-in-the-loop to enable a fast and error-free realization.

## [redacted] References

[1] F-wheel, „fwheel.cc," 5 March 2019. [Online]. Available: https://www.fwheel.cc/e-bike-market-research-report-from-dyu-fwheel/. [Accessed 23 July 2021].

[2] V. Hölttä, L. Palmroth und L. Eriksson, „Rapid control prototyping tutorial with application examples," in *ResearchGate*, Helsinki, 2004.

[3] U. Nuß, Hochdynamische Regelung elektrischer Antriebe, Berlin: VDE VERLAG GMBH, 2017.

[4] J. Kühlwein, „Driving resistances of lightduty vehicles in europe; present situation, trends and scenarios for 2025," International Council on Clean Transportation Europe, Berlin, December 2016.

[5] S. Rohde und F. Gauterin, Online Estimation of Vehicle Driving Resistance Parameters withRecursive Least Squares and Recursive Total Least Squares, Gold Coast, Australia: 2013 IEEE Intelligent Vehicles Symposium (IV), 2013.

[6] R. Wolff und J. Strunz, Biomechanische Leistungsdiagnostik und Feedbacktraining zur Optimierung der Trettechnik im Elitebahnradsport, Berlin, 2007.

[7] D. Collins, „Motion Control Tips,“ 30 July 2017. [Online]. Available: https://www.motioncontroltips.com/how-are-encoders-used-for-speed-measurement/. [Accessed 23 July 2021]

[8] Microsemi, Field Oriented Control of Permanent Magnet Synchronous Motors User’s Guide, Aliso Viejo CA, USA: Microsemi Corporate Headquarters, 2012.

[9] R. Ashok Kumar und K. Dr. Balaji, „Sensor less control of PMSM fed from three phase four switch,“ *International Journal of Engineering & Technology,* 2018.

[10] Y. Solbakken, „Switchcraft,“ 1 May 2017. [Online]. Available: https://www.switchcraft.org/learning/2017/3/15/space-vector-pwm-intro. [Accessed 23 July 2021]

[11] NXP Semiconductor, 3-phase Sensorless Single-Shunt Current-Sensing PMSM Motor Control Kit with MagniV MC9S12ZVM, Freescale Semiconductor, Inc., 2016.

[12] X. Liu, Improvement in inverter modeling and control, Kentucky: UKnowledge, 2017.

[13] J. Böcker, Gereglete Drehstromantriebe, Paderborn: Universität Paderborn, 2016.

[14] K. Belda, „Mathematical Modelling and Predictive Control,“ in *Transactions on Electrical Engineering*, Prague, Czech Republic, 2013.