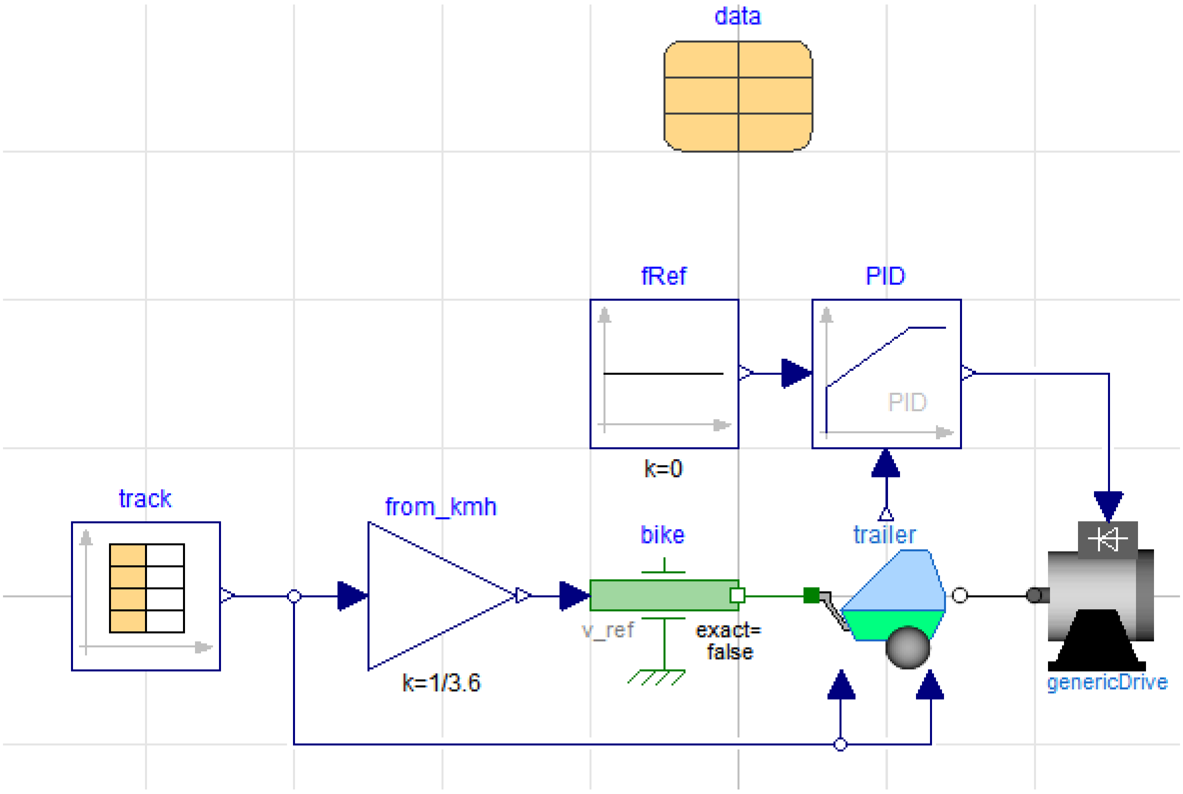
A single-axis bike trailer with two wheel hub motors shall be investigated.
In the drawbar, a traction force sensor is integrated.

For the beginning, only straight-line driving should be considered. For turns, the torque must be distributed to the two wheel hub motors in such a way that neither oversteer nor understeer occurs. Additional sensors are required to achieve this goal.



The bicycle is modeled in a simplified way using a specified riding speed. Speed, gradient and wind speed are specified by a time-dependent table:
`Modelica.Mechanics.Translational.Sources.Speed`
`Modelica.Blocks.Sources.CombiTimeTable`

The trailer is implemented as a separate model:

- Modelica.Mechanics.Translational.Components.Vehicle describing:
  - Mass of trailer
  - Wheels
  - Moment of inertia of wheels
  - Resistances (can be specified constant or by an input):
    - Air drag resistance
      $$F_{Drag} = c_W \cdot A \cdot \rho \cdot \frac{(v - v_{Wind})^2}{2}$$
    - Rolling resistance
      $$F_{Roll} = c_R \cdot m \cdot g \cdot sin(\alpha)$$
    - Inclination resistance
      $$F_{Grav} = m \cdot g \cdot cos(\alpha)$$
      Note: Inclination is the tangent of α.
- Translational multi-sensor, providing traction force, speed and power

The drives are summarized in one external generic drive model:

- Limiter for maximum and minimum achievable torque
  `Modelica.Blocks.Nonlinear.Limiter`
- Second-Order (Reference transfer function for an optimal designed current controller)
  `Modelica.Blocks.Continuous.TransferFunction`
- Gesteuertes Drehmoment
  `Modelica.Mechanics.Rotational.Sources.Torque`

Control oft the two wheel hub drives to achieve zero traction force at the drawbar.

Proportional gain can be calculated from the relationship between force and torque:

$$\tau = F \cdot \frac{D_{wheel}}{2}$$

The reference transfer function for an optimal designed current controller can be written as:

$$\frac{\tau}{\tau_{Ref}} = \frac{I}{I_{Ref}} = \frac{1}{1 + sT_{sub} + s^2 \frac{T_{sub}^2}{2}}$$

$T_{sub} = 2T_\sigma$ is the substitute time constant. T_σ is the sum of small time constants on the current control loop (dead time respectively delay of the power converter as well as current measurement).

A PID-controller with ideal differential part cannot be achieved. Therefore a PIDT$_1$-controller is choosen:

$$1 + \frac{1}{sT_i} + \frac{sT_d}{1 + skT_d} = \frac{1 + s(T_i + kT_d) + s^2 T_i T_d (1 + k)}{sT_i(1 + skT_d)}$$

$k < 1$ determines how near the behaviour of the DT$_1$-term approximates the ideal differentiator,

The transfer function of the open loop can be written as:

$$\frac{1 + s(T_i + kT_d) + s^2 T_i T_d (1 + k)}{sT_i(1 + skT_d)} \cdot \frac{1}{1 + sT_{sub} + s^2 \frac{T_{sub}^2}{2}}$$

Compensation of the denominator of the plant using the numerator of the controller delivers:

$$T_i + kT_d = T_{sub}$$

$$T_i T_d (1 + k) = \frac{T_{sub}^2}{2}$$

Therefor we can derive the parameterization oft he controller:

$$\frac{T_i}{T_{sub}} = \frac{1}{2} \pm \sqrt{\frac{1}{4} - \frac{k}{2(1 + k)}} = \frac{1}{2}\left[1 \pm \sqrt{\frac{1 - k}{1 + k}}\right]$$

$$\frac{T_d}{T_{sub}} = \frac{T_{sub}}{T_i} \frac{1}{2(1 + k)} = \frac{1}{1 + k \pm \sqrt{1 - k^2}}$$

To get only real solutions, k is limited:

$$k \leq 1$$

Thus we get the transfer function of the open loop:

$$\frac{1}{sT_i(1 + skT_d)}$$

The reference transfer function shows second order behaviour:

$$\frac{\frac{1}{sT_i(1+skT_d)}}{1+\frac{1}{sT_i(1+skT_d)}} = \frac{1}{1+sT_i+s^2T_ikT_d} = \frac{1}{1+2\vartheta Ts+T^2s^2}$$

Compariosn of coefficients reveals:

$$T_i = 2\vartheta T$$

$$T_ikT_d = T^2$$

$$T = \sqrt{kT_iT_d} \rightarrow \frac{T}{T_{sub}} = \sqrt{\frac{k}{2(1+k)}}$$

$$\vartheta = \frac{T_i}{2T} = \frac{\sqrt{2}}{4} \cdot \left[\sqrt{\frac{1}{k}+1} \pm \sqrt{\frac{1}{k}-1}\right] \geq 1$$

The aperiodic borderline case is achieved for damping $\vartheta = 1$:

$$k \leq \frac{8}{17}$$

For all solutions of quadratic equations the positive sign of the root is taken.

Comparison of PID – controller Modelica – Simulink (ideal)

$$k_p \cdot \left(1 + \frac{{}^1/_{T_i}}{s} + \frac{T_d \cdot s}{1+s\cdot k\cdot T_d}\right) = P \cdot \left(1 + \frac{I}{s} + \frac{D\cdot s}{1+s\cdot {}^1/_N}\right)$$

Conversion of the parameters:

$$P = k_p$$

$$I = {}^1/_{T_i}$$

$$D = T_d$$

$$N = \frac{1}{k\cdot T_d}$$