Stabilizing a Bicycle into its upright position and rejecting disturbances using a Model Predictive Control approach

Jean Kuppers (4593928), Pierre-Antoine Denarié (4451805)

***Abstract*—This paper proposes a model predictive control (MPC) strategy for controlling an unmanned bicycle with linear model dynamics. The linear model used is the validated benchmark model by Schwab et all. The state-space matrices are parameterized using the forward velocity. The cyclist is assumed to be rigidly attached to the bicycle frame and the leaning torque which is normally exerted by the cyclist is assumed to be directly applicable. Next to the leaning torque the system can be also controlled with a steering torque. A state-based regulation MPC and an output-feedback disturbance rejection MPC are designed for a constant velocity. Furthermore, an output-feedback adaptive MPC is considered which adapts to a changing forward velocity. The proposed closed-loop MPC designs are simulated with Matlab using the CVX package. The final results show that for adequately chosen weighting matrices and prediction horizons, the closed-loop systems are able to regulate the system and reject simulated wind and road disturbances.**

I. Introduction

The dutch association for Bicycles and Automobiles (Rijwiel en Automobiel Industrie vereniging) estimated that in 2019 around 22.9 million bicycles were present in the Netherlands [1]. 5.6 million more than the estimated amount of inhabitants. This large difference in numbers illustrates the importance of the bicycle as a travel medium for the dutch. Surprisingly, the dynamics of such a commonly used and seemingly simple object are far from being straightforward. The handling and stabilizing of a bicycle are complex tasks which practiced humans can perform effortlessly. However, for a control system these are demanding and difficult tasks especially at velocities for which the bicycle is not self-stabilising anymore.

The first attempts of modeling the dynamics of a bicycle date back to 1869 with a publication by William Rankine [2] on the steering and leaning observations of a bicycle rider. In the following years many expanded on his work until meteorologist Francis Whipple presented in 1899 [3] the first fully fledged non-linear equations of motion that govern the dynamics of an upright uncontrolled bicycle. Apart from a few typographical errors, these highly non-linear equations were confirmed to be correct in the years after in [4], [5], [6]. Whipple and many others [4] [6] [7] have linearized the non-linear model around the upright position using d'Alemberts principle. As computer technology advanced, researchers started deriving and testing the bicycle model using computer software such as Autosim [4] to the point where an extensive model and stability analysis where done by Schwab et all [8] in Autosim and the numerical multibody dynamics program SPACAR. Schwab found that the speed-dependent linearized equations together with his benchmark model are suitable for dynamical modelling and analysis of a bicycle.

The validated linear benchmark model led to the proposition of different control strategies for controlling an unmanned bicycle such as output-zeroing control [9], Fuzzy control [10] and Model Predictive Control (MPC) [11][12]. The latter however, has not been explored fully in depth yet as the proposed MPC controllers lack insight into the design process and stability analysis.

In this paper the problem of stabilizing an unmanned bicycle into its upward position is addressed by proposing a model predictive control strategy based on the validated linear benchmark model of a bicycle. This includes a stability analysis, a method of estimating states and simulating disturbances and a proposal for a controller that adapts to changing dynamics.

It is structured as follows. The linear bicycle model is presented and its approximated nonlinear dynamics are briefly explained. Section I is then concluded with the discretization of the model. The next section will elaborate on the MPC formulations and will be followed by the stability analysis of section III which includes a method for finding the terminal set $\mathbb{X}_\mathrm{f}$. The analysis consists of a systematic approach to proving stability by showing that the closed-loop system satisfies certain assumptions of which one is shown to hold a posteriori. In section IV the controller weight design and performance of the closed-loop system will be assessed through simulation in MATLAB with the CVX package. The paper will be concluded in section V with a summary of the main findings related to the MPC strategy implementation.

A. Bicycle model

The Whipple model [3] represents a mechanical bicycle model build up of four rigid bodies that connected to one another either rigidly or with revolute hinges. These bodies comprise of the rear frame (including the rigidly attached rider), the two wheels and the front frame consisting of the front fork and handle bar. All of which the specifications are shown in appendix A and are taken from [8].

The linearized equations of motion shown in equation 2 are derived in [7] by approximating small changes in angles so

that they all can be expressed in terms of the variables ϕ, the rear frame roll angle, δ, the steering angle and $\dot{\theta}_r$, the rear wheel rotational velocity respectively. The latter is substituted by the forward velocity of the bicycle v given in equation 1:

$$v = -\dot{\theta}_r R_{rw} \tag{1}$$

with R_{rw} the radius of the rear wheel.

The linear differential system equation from [7] is shown below:

$$\mathbf{M}\ddot{\mathbf{q}} + [\mathbf{C_1} \cdot v]\dot{\mathbf{q}} + \left[\mathbf{K_0} + \mathbf{K_2} \cdot v^2\right]\mathbf{q} = \mathbf{f}, \quad \mathbf{q} = \begin{bmatrix}\phi \\ \delta\end{bmatrix} \tag{2}$$

Here matrices $\mathbf{M}$, $\mathbf{C_1}$, $\mathbf{K_0}$ and $\mathbf{K_2}$ are constant matrices of which the numerical values are computed using the equations from [8] and can be found in 3, 4, 5 and 6. These matrices represent the bicycles mass matrix, "damping" matrix proportional to the forward speed v, the constant part of the stiffness matrix and the part of the stiffness matrix that is proportional to the square of the forward speed v respectively.

$$\mathbf{M} = \begin{bmatrix} 80.81210000000002 & 2.32343142623549 \\ 2.32343142623549 & 0.30126570934256 \end{bmatrix} \tag{3}$$

$$\mathbf{K_0} = \begin{bmatrix} -794.119500000000 & -25.739089291258 \\ -25.739089291258 & -8.139414705882 \end{bmatrix} \tag{4}$$

$$\mathbf{K_2} = \begin{bmatrix} 0 & 76.40620875965657 \\ 0 & 2.67560553633218 \end{bmatrix} \tag{5}$$

$$\mathbf{C_1} = \begin{bmatrix} 0 & 33.77386947593010 \\ -0.84823447825693 & 1.70696539792387 \end{bmatrix} \tag{6}$$

The parameter v is thus taken as a state-space defining parameter which can be used to approximate the bicycle dynamics at different forward velocities. The input vector $\mathbf{f}$ consists of the roll or lean torque T_ϕ and the steering torque T_δ:

$$\mathbf{f} = \begin{bmatrix}T_\phi \\ T_\delta\end{bmatrix} = u \tag{7}$$

The steering torque T_δ is the torque that a cyclist can apply to the front wheel by applying a force to each handlebar in counter direction to one another. The lean or roll torque T_ϕ is the torque that the cyclist can apply to the bicycle by leaning to a side and shifting his center of gravity.

Rewriting these differential equations into the standard state-space form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{8}$$

gives the following state-space system:

$$\begin{bmatrix}\dot{\phi} \\ \dot{\delta} \\ \ddot{\phi} \\ \ddot{\delta}\end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix}\phi \\ \delta \\ \dot{\phi} \\ \dot{\delta}\end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M^{-1}} \end{bmatrix} u \tag{9}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

$$A_{21} = -\mathbf{M^{-1}}(\mathbf{K_0} + \mathbf{K_2}v^2) \quad A_{22} = -\mathbf{M^{-1}}\mathbf{C_1}v \tag{11}$$

with $A \in \mathbb{R}^{4x4}$, $B \in \mathbb{R}^{4x2}$, $C \in \mathbb{R}^{2x4}$, $x \in \mathbb{R}^4$ and $u \in \mathbb{R}^2$. The outputs of the system are chosen to be the two angles ϕ and δ. All angles, angular velocities and angular accelerations are defined in rad, rad/s and rad/s^2 respectively.

The presented equations of motion can be misleading so it must be noted that the linearization is not a linearization around the forward velocity v but in fact a linearization of small angles within the inner bicycles' dynamics which is extensively described in [7]. Furthermore, no relations between the forward velocity v and the angles ϕ and δ are stated which makes it impossible to include v as a state to turn the model into an actual non-linear model.

B. Stability of the Bicycle Model

The stability of the linear model in 2 has already been thoroughly researched by Schwab in [8]. Schwab found that by analysing the model using a bifurcation diagram with varying forward velocities v, there is a range of velocities for which the A-matrix only has negative eigenvalues. This means that the system is asymptotically stable in this region and thus self-stabilizing, up to certain angles ϕ and δ. The velocity range is shown in 12.

$$\begin{gathered} v_w < v < v_c \\ v_w = 4.301611 \text{ m/s}, \quad v_c = 6.057011 \text{ m/s} \end{gathered} \tag{12}$$

C. Discretization of the model

The state-space model in 9-11 can be rewritten into its discrete form counterpart:

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= C x_k \end{aligned} \tag{13}$$

using the zero-order-hold sampling theorem:

$$\Phi = e^{Ah}, \quad \text{and} \quad \Gamma = \int_0^h e^{As} ds B \tag{14}$$

Here, h denotes the sampling time which is chosen as 0.1. This sampling time was chosen by trial and error with the goals of being fast enough to capture the systems dynamics as good as possible but also being slow enough such that the computational effort would not be too taxing. The discretization was done in Matlab using the `c2d` command.

The pair (Φ, Γ) is controllable as the controllability matrix $\mathcal{W}_c = \begin{bmatrix} \Gamma & \Phi\Gamma & \Phi^2\Gamma & .. & \Phi^3\Gamma \end{bmatrix}$ is full rank and the pair (Φ, C) is observable as the observability matrix $\mathcal{O} = \begin{bmatrix} C & C\Phi & C\Phi^2 & .. & C\Phi^3 \end{bmatrix}^T$ is also of full rank.

II. Model predictive control design

The MPC strategy proposed in this paper consists of 3 different MPC controller designs that consecutively increase in complexity.

1) A state-feedback regulation MPC controller
2) An output-feedback Offset-free MPC controller with disturbance tracking and rejection.
3) An adaptive MPC controller that can control the system at changing forward velocity v.

A. Constrained control problem

The main advantage of using Model Predictive Control is that both state and input constraints can be taken into account when computing the optimal control input. This is done by minimizing a control cost.

1) Cost function: The MPC problem will be solved at each time step by minimizing the following cost function:

$$\begin{aligned} J\left(x_0, u\right) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N)) \\ \text{s.t. } u \in \mathbb{U}, x \in \mathbb{X} \end{aligned} \tag{15}$$

consisting of a stage cost $\ell(x(k), u(k))$ and a terminal cost $V_f(x(N))$ both shown in 16:

$$\begin{aligned} \ell(x(k), u(k)) &= \frac{1}{2} x(k)^T Q x(k) + \frac{1}{2} u(k)^T R u(k) \\ V_f(x(N)) &= \frac{1}{2} x(N)^T \, \mathrm{P} \, x(N) \end{aligned} \tag{16}$$

The Q and R matrices are diagonal weighting matrices that can be adjusted to specify the importance of certain states or inputs in the optimization. Large values for the diagonal elements in Q for example reflect the intent to drive the state faster to 0 or to the reference state at the expense of large control inputs. The weight matrix P is the solution to the Discrete Algebraic Riccati Equation (DARE) used to solve the unconstrained infinite horizon-control problem which is found using the `idare` command in Matlab. All three weighting matrices have the properties of being symmetric and positive definite.

2) State Constraints: The state constraints of the optimization problem are found by interpreting physical limitations of the angles and angle velocities. In [13] it was found that a regular person (non-cyclist) can control a bicycle with a maximal roll angle of 30 deg and roll rate of 25 deg/s which translate roughly into $\pi/6$ rad and $0.42\pi/3$ rad/s. The same paper describes how the steering rate would not exceed 55 deg/s equal to $0.84\pi/3$ rad/s. The maximal steering angle was chosen to be 30 $degrees$. These state constraints are visible in 17 in the form $Fx(k) \leq e$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \phi \\ \delta \\ \dot{\phi} \\ \dot{\delta} \end{bmatrix} \leq \pi/3 \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.42 \\ 0.42 \\ 0.84 \\ 0.84 \end{bmatrix} \tag{17}$$

The goal of the optimization is to find the optimal control input at the current time-step. As such the state-constraints $x \in \mathbb{X}$ are rewritten into a set of input constraints using the prediction relationship $x_N = Px_0 + Zu_N$ to decrease the amount of optimization parameters and thus speed up the optimization process. This yields new constraints on the optimal input sequence of the form:

$$I_N \otimes F(Gx_0 + Zu_N) \leq e_N \tag{18}$$

$$Z = \begin{bmatrix} \Gamma & 0 & 0 & \dots & 0 \\ \Phi\Gamma & \Gamma & 0 & \dots & 0 \\ \Phi^2\Gamma & \Phi\Gamma & \Gamma & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ \Phi^{N-1}\Gamma & \Phi^{N-2}\Gamma & \dots & \Phi\Gamma & \Gamma \end{bmatrix} \tag{19}$$

$$u_N = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad G = \begin{bmatrix} \Phi \\ \Phi^2 \\ \Phi^3 \\ \vdots \\ \Phi^N \end{bmatrix}, \quad e_N = \begin{bmatrix} e \\ e \\ e \\ \vdots \\ e \end{bmatrix} \tag{20}$$

3) Input Constraints: It was found in [11] that for the steering torque a constraint of $5Nm$ would be reasonably realistic. For the roll torque it was explained in section I that it is applied by the cyclist by moving his center of mass. However, the linear model assumes that the cyclist is rigidly attached to the rear frame so this control action can only be simulated. In [14] the rolling motion of the bicycle is stabilized with the use of a flywheel. The model in that paper is based on the same set of linearized equations in 2. This proposed flywheel can deliver a maximum rolling torque of 40 Nm and the bicycle weighs $29,2$ kg which gives a torque to weight ratio of 1.37. Assuming that the model 9 in this paper uses a perfect flywheel large enough to counter the full body weight of the cyclist and that can deliver immediate torque, the torque was taken as the entire model weight multiplied by the found torque to weight ratio: $T_{phi,max} = 94 \cdot 1.37 = 128.8Nm$.

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} T_\phi \\ T_\delta \end{bmatrix} \leq \begin{bmatrix} 128.8 \\ 128.8 \\ 5 \\ 5 \end{bmatrix} \tag{21}$$

4) Terminal Constraint: Next to the input and state constraints a terminal constraint set was also introduced which was derived from the estimation of the terminal set $\mathbb{X}_f$. The terminal constraint set represents the constraints that the MPC optimization has to satisfy such that the system reaches the maximum control invariant set $\mathbb{X}_f$ in the given amount of steps. Within this set $\mathbb{X}_f$ the optimal MPC control law is equal to the control law of the unconstrained infinite-horizon LQR problem.

The set $\mathbb{X}_f$ was estimated using the extended algorithm presented in [15] which is based on the algorithm from [16]. This algorithm finds the maximum input and state admissible invariant set by checking if the computed optimal MPC control actions are equal to those of the unconstrained infinite-horizon LQR problem. This is done for each state and input constraint separately after which the found final set is projected onto $\mathbb{R}^4$. This gives 36 new linear state constraints of the form $Hx \leq h$. By rewriting these in a similar fashion as was done for the state constraints in 18-20 the constraints can be implemented as input constraints into the MPC optimization problem. Due to the relatively small size of the terminal constraint set it was opted not to omit them as this mattered little in computation time for the regulator and output-feedback strategy.

B. State-feedback Regulation MPC

For the state-feedback MPC the goal is to stabilize the states and bring the system to the equilibrium point which is the 0-state. This is when the bicycle is standing in an upright position. To reach the equilibrium an MPC controller was designed that optimized the cost function 15.
For the state-feedback regulation the velocity was taken as constant and in the unstable range mentioned in 12. In [8] it was found that system is more unstable and harder to control at low velocities then at higher velocities as for higher velocities the increased amount of energy present in the system makes it easier to balance the bicycle upwards. Therefore the velocity was chosen as $v = 2m/s$.

An initial state x_0 was introduced for the regulation task in which the roll angle and roll angle velocity were set to $10\ rad$ and $10\ rad/s$:

$$x_0 = \begin{bmatrix} 10 \\ 0 \\ 10 \\ 0 \end{bmatrix} \tag{22}$$

With this initial state the tuning procedure for obtaining the best performance was conducted. This tuning procedure is explained in section IV-A. The weighting matrices were chosen as $Q = I$ and $R = I$ and P as the solution to the DARE. The prediction horizon was set at $N = 8$.

C. Output-feedback offset-free MPC

Because of the limited access to all the states, the second MPC design is based on an output-feedback MPC strategy. The outputs of the system are the roll- and steering angle ϕ and δ as these states can be measured accurately by sensors as stated in [17]. Again the velocity of the bicycle was taken constant at $v = 2m/s$ for the same reasons mentioned in II-B. This system was subjected to a wind disturbance that acts on the roll angular acceleration. This new angular accelarion component $\ddot{\phi}_{wind}$ is introduced in the continuous state-space and then discretized as was shown in section I-C. This disturbance acting on the system was computed as:

$$F_{wind} = \frac{1}{2}\rho A v_{wind}^2 \tag{23}$$

with ρ the air density and A the surface the wind acts on. This surface was roughly estimated to be $0.5\ m^2$. Equation 23 was then used to calculate $\ddot{\phi}_{wind}$:

$$\ddot{\phi}_{wind} = \frac{F_{wind}}{m_{rider} z_{rf}} \tag{24}$$

with m_{rider} the mass of the rider and bicycle combined and z_{rf} the vertical distance from center of mass to the ground obtained from [8]. This wind disturbance will be simulated as a squared sine wave with a max amplitude of $\ddot{\phi}_{wind}$.

Next to the wind disturbance a second disturbance d_{steer} acting on the steering angle δ is added in the model to simulate the effects of a "rough" road. This disturbance was simulated in the form of a white noise signal with a maximum amplitude of 0.1% of the steering angle constraint $0.84 \cdot \frac{\pi}{3}$. This however, is purely done as an experiment as none of the MPC controllers have been specifically tuned to address this disturbance. This leads to the following continuous and discrete time state-space representations:

$$\begin{aligned} \dot{x} &= Ax + Bu + B_d d \\ y &= Cx \end{aligned} \tag{25a}$$

$$B_d = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ \frac{1}{m z_{rf}} & 0 \\ 0 & 0 \end{bmatrix} \tag{25b}$$

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k + \Gamma_d d \\ y_k &= C x_k + C_d d \end{aligned} \tag{26a}$$

$$C_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{27a}$$

$$d = \begin{bmatrix} F_{wind} \\ d_{road} \end{bmatrix} \tag{27b}$$

In order to estimate the disturbances an augmented state-space system is introduced as shown in 28:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u_k \\ y_k &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} \end{aligned} \tag{28}$$

As stated in section I-C the pair (Φ, Γ) is observable which is required in order to estimate the states. For the augmented state-space this property holds as well. In addition the following equation holds:

$$rank \begin{bmatrix} I - \Phi & -\Gamma_d \\ C & C_d \end{bmatrix} = n + n_d = 5 \tag{29a}$$

$$n_d \leq dim(y) \tag{29b}$$

which results in the fact that a Luenburger observer can be used to estimate the augmented state-space. This Luenberger observer is of the following form:

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u_k + L \left(y - \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} \right) \tag{30}$$

with L as the Kalman observer gain of which the weighting matrices were chosen as $Q = diag(1, 1, 1, 1, 10^4, 1)$ and $R = I$. This will be showcased in the next section.

For the reference tracking and disturbance rejection an optimal target selection problem was solved in order to obtain the reference states x_{ref} and reference inputs u_{ref}. These parameters were obtained online by solving the following optimization problem

$$(x_{\text{ref}}, u_{\text{ref}}) \left(\hat{d}, y_{\text{ref}} \right) \in \begin{cases} \underset{x_r, u_r}{\operatorname{argmin}} J(x_r, u_r) \\ s.t. \\ \begin{bmatrix} I - \Phi & -\Gamma \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} \Gamma_{\text{d}} \hat{d} \\ y_{\text{ref}} - C_{\text{d}} \hat{d} \end{bmatrix} \\ (x_{\text{r}}, u_{\text{r}}) \in \mathbb{Z} \\ C x_{\text{r}} + \hat{d} \in \mathbb{Y} \end{cases} \tag{31}$$

where $J(x_r, u_r)$ is of the form

$$J(x_r, u_r) = \begin{bmatrix} x_r & u_r \end{bmatrix}^T H \begin{bmatrix} x_r & u_r \end{bmatrix} \tag{32}$$

with H the identity matrix of dimension 6. y_{ref} was set equal to zero as the desired output is the upright position.

With the OTS defined and the sampling time kept equal to $0.1s$, the control horizon and weighting matrices were again tuned. N was set to 16, $R = I_2$ and Q as:

$$Q = \begin{bmatrix} 10000 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

which is elaborated more in section IV-B. With these parameters defined and the online OTS solving x_{ref} and u_{ref} at each time iteration the MPC problem was solved with the same constraints mentioned in II-B

D. Adaptive MPC

For the case that the forward velocity v is not constant but time-varying, the state-space model described in 9-11 will have different A and B matrices for continuous time and different Φ and Γ matrices for discrete time at time-steps where the velocity has changed. This changes nothing for the solving of the MPC problem. The thing that does have to be taken into account is the fact that at every time-step the discretization and nearly all previously defined constraints have to be recomputed which was found to increase the computation time drastically.

For this MPC controller the same output-feedback disturbance tracking set up as described in the previous section was used. A similar wind disturbance was added twice on the system during a run of 1 min which is equal to 600 steps. During this run the systems forward velocity increased linearly from 2 m/s to 8 m/s. The wind signal was introduced at $v = 2.5$ m/s and at $v = 5.5$ m/s. This means that for the first gust of wind the controller would have to stabilize the system starting in an unstable region $v < v_w$ while entering the stable region $v_w < v < v_c$ and for the second region the other way around starting in $v_w < v < v_c$ while entering $v > v_c$. Apart from the linearly increasing velocity and the slightly different way of disturbing the MPC problem was solved similar to the one described in II-C. The sampling time, control horizon and weighting matrices were kept the same.

III. Stability analysis

When implementing a model predictive control strategy it is imperative that the closed-loop system is stable or otherwise such an exercise would be futile. This section illustrates a systematic approach to prove that the proposed MPC strategy turns the closed-loop system asymptotically stable. This will be done by proving that the regulation strategy satisfies a set of assumptions from [18] which in turn when satisfied prove the stability of the system using a theorem from the same book.

***Assumption 2.2**:* (Continuity of system and cost) In the proposed MPC strategy the linear system function f and the two quadratic cost functions ℓ and V_f are all continuous and satisfy $f(0,0) = 0$, $\ell(0,0) = 0$ and $V_f(0,0) = 0$. This means the condition is satisfied.

***Assumption 2.3**:* (Properties of constraint sets). Both constraint sets $\mathbb{X}$ and $\mathbb{U}$ are closed, which in turn means that set $\mathbb{Z}$ is closed. The input set $\mathbb{U}$ and previously found control invariant terminal set $\mathbb{X}_f$ which is a subset of $\mathbb{X}$, are bounded so hence also compact. Lastly, all sets contain the origin which means that for the regulation problem the assumption is satisfied.

***Assumption 2.14**:* (Basic stability assumption). The functions $\ell = \frac{1}{2}\left(x^\top Q x + u^\top R u\right)$ and V_f satisfy the following:

$$\begin{aligned} \ell(x,u) &\geq \frac{1}{2} x^\top Q x \geq \frac{1}{2} \lambda_{\min}(Q) |x|^2 = \alpha_1(|x|) \\ V_{\text{f}}(x) &= \frac{1}{2} x^\top P x \leq \frac{1}{2} \lambda_{\max}(P) \, |.x|^2 = \alpha_f(|x|) \end{aligned} \tag{34}$$

with $\alpha_1(|x|)$ and $\alpha_f(|x|)$ being $\mathcal{K}_\infty$ functions.

Furthermore, $\mathbb{X}_f$ was already proven to be control invariant. The last part of this assumption is the Lyapunov decrease:

$$V_f(x+u) - V_f(x) \leq -\ell(x,u) \tag{35}$$

This assumption was checked a posteriori. In figure 1 the Lyapunov decrease is clearly visible which also satisfies

the above condition 35. This means that the assumption is satisfied.

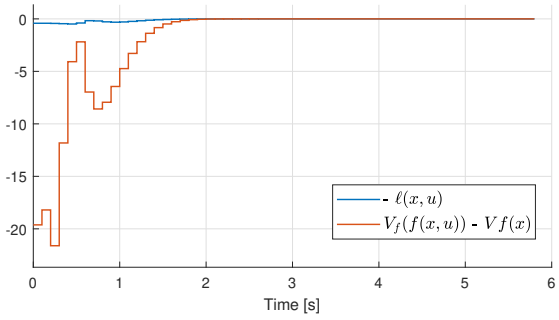

Fig. 1. Lyapunov drecrease of the state-feedback regulator MPC

Assumption 2.17: (Weak controllability). As the system was already proven to be controllable this assumption is automatically satisfied.

All assumptions have been validated, so by theorem 2.19 and 2.23 from [18] the closed-loop system has been proven to be stable with the origin begin exponentially stable.

A. Controllable set $\mathcal{X}_N$

The controllable is set $\mathcal{X}_N$ is the set of initial states that can be steered into the terminal set $\mathbb{X}_f$ within the total amount of steps in the prediction horizon N. In this paper the estimation of the controllable set is not covered. However, several initial states have been checked for if they lie within this set to give a general impression of what this set might look like. These are shown below:

$$S_{1,x_0} = \{ \begin{bmatrix} 0 \\ 10 \\ 0 \\ 10 \end{bmatrix}, \begin{bmatrix} 10 \\ 0 \\ 10 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 25 \\ 0 \\ 10 \end{bmatrix}, \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -10 \\ 10 \\ 0 \\ 0 \end{bmatrix} \} \in \mathcal{X}_N$$

$$\mathrm{S}_{2,x_0} = \{ \begin{bmatrix} 25 \\ 0 \\ 10 \\ 0 \end{bmatrix}, \begin{bmatrix} 10 \\ -10 \\ 10 \\ 10 \end{bmatrix} \begin{bmatrix} -10 \\ 10 \\ -10 \\ 10 \end{bmatrix}, \begin{bmatrix} 25 \\ 25 \\ -25 \\ -25 \end{bmatrix} \} \notin \mathcal{X}_N(36)$$

IV. Numerical simulations

In this section all choices that were made during the tuning processes are elaborated and plots are shown to support these choices. The final closed-loop system performances are evaluated too.

A. State-feedback Regulation MPC

To find the optimal parameters for the state-feedback MPC, first the control horizon was tuned. For control horizons lower than $N = 8$ the optimization problem was infeasible as the inputs would immediately go out of bounds and give a `NaN` as result. $N = 8$ therefore is the minimal control horizon for which the problem could be solved.

It would seem logical that the states would converge faster towards 0 if the prediction horizon N is increased, as the MPC controller would be able to predict further in the future. From Figure 2 one can see that for this system this is not the case. For control horizons chosen larger than $N = 8$, the systems regulation response would be slower as can be seen for the $N = 12, 16$ and 16 trajectories.

The reason behind this effect can be explained using figure 3. In this figure the response is shown for a prediction horizon of $N = 16$ but with variable Q weights. For $Q = I$ the trajectory is the exact same as the trajectory for $N = 16$ in figure 2. As the weight on the states is increased for the optimizing of the cost function, the trajectory increasingly resembles the trajectory for $N = 8$ from figure 2. This means that the slower responses for higher control horizons from figure 2 are a result of the MPC design finding a lower control cost for these responses as it can predict further into the future. Setting a higher penalty on the states, or in other words a lower penalty on the control inputs results in the MPC design finding a less costly trajectory that resembles the trajectory of $N = 8$. From this it was concluded that the prediction horizon of $N = 8$ was the optimal horizon for the regulator as it delivered the best performance for the least amount of computational effort.

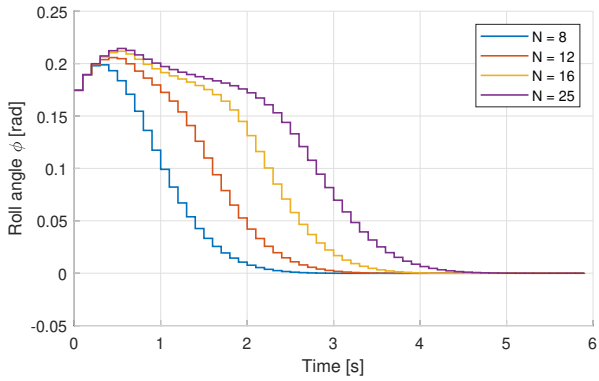

Fig. 2. State-feedback MPC roll angle ϕ with $v = 2m/s, Q = I, R = I$

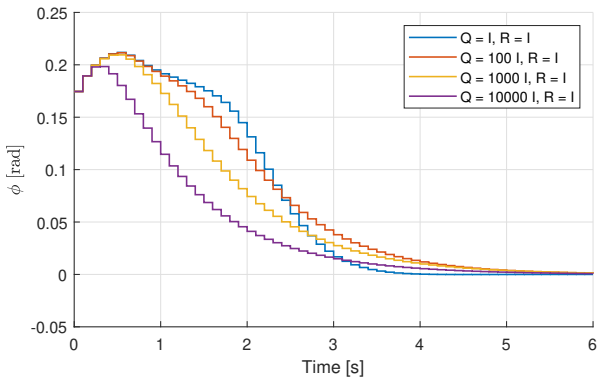

Fig. 3. State-feedback MPC ϕ roll angle with variable Q and $v = 2m/s, R = I$.

B. Output-feedback offset-free MPC

For the output-feedback MPC design a control horizon of $N = 16$ was chosen. This differs from the chosen horizon of the state-feedback regulator MPC. In figure 4 the wind disturbance rejection performance of the output-feedback MPC controller is shown for different control horizons. It was found that for horizons smaller dan $N = 12$ the disturbance caused a fluctuating behaviour that would remain in the rest of the simulation as can be clearly seen in 4 for $N = 8$. For horizons larger than $N = 18$ the system would reach the 0 state the fastest but at the expense of larger fluctuations in the roll angle ϕ which the trajectory of $N = 25$ shows. After iteratively testing different control horizons in between $N = 12$ and $N = 25$ the preferred horizon was chosen at $N = 16$ as here the intermediate fluctuations were of minimal amplitude which means that the bicycle has a smoother rolling motion.

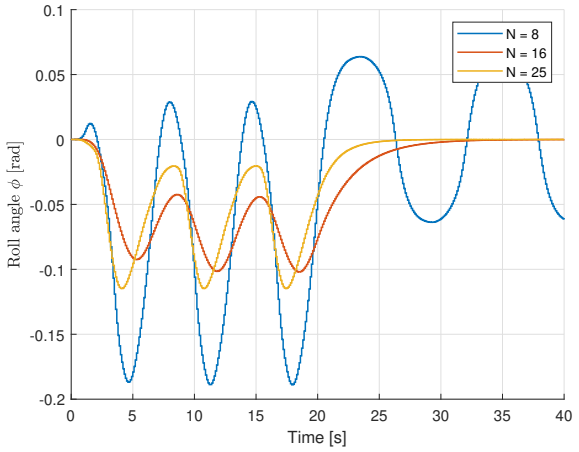

Fig. 4. Output-feedback output roll angle for different N with $v = 2m/s, Q = diag\{10e4, 10e4, 1, 1\}, R = I$

Figure 5 shows the disturbance tracking responses of the system for several different Q weights when excited with 2 step inputs. The behaviour for $Q = I$ shows that a lot of unwanted fluctuations are taking place. Increasing the weight on the first disturbance channel results in a better tracking behaviour by the observer of the wind disturbance. The final value for the fifth diagonal entry of Q was chosen to be 10^4.

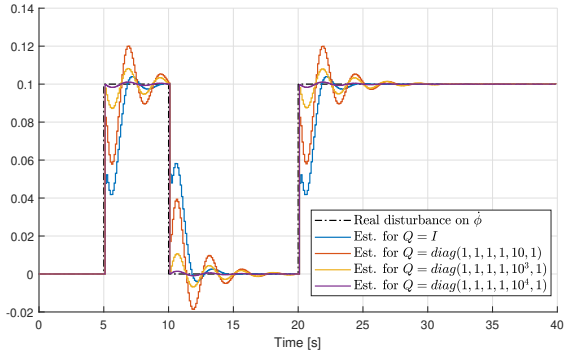

Fig. 5. Different tunings for the weighting matrix Q for the Kalman observer gain L

For dampening the response of the disturbance in the roll state ϕ, the relevant weights in the Q matrix were tested for different values. These turned out to be the two angle states ϕ and δ. Figure 6 shows the effect of these weights on the system response. It is clear that setting a higher weight on the angle states, or in other words punishing these states more when they deviate from the desired 0 value, results in better dampened peaks of the roll angle ϕ without slowing the total response too much. As a result, the final weight was chosen as $Q = diag(10^4, 10^4, 1, 1$. The tuning of the R weight is not discussed here as it is about adjusting the ratio between the Q and R weights which is identical to only tuning Q. R was kept as I.

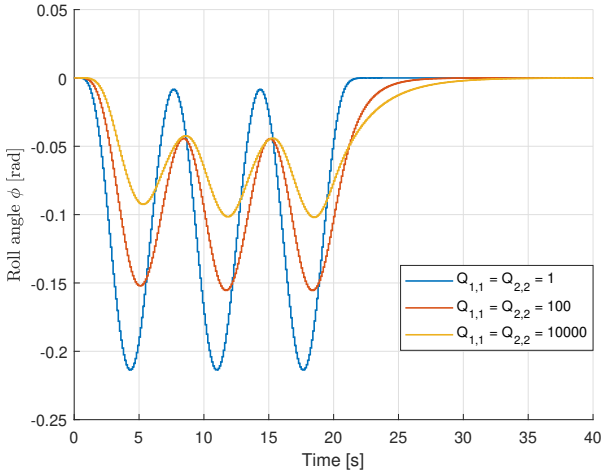

Fig. 6. Output-feedback MPC and output roll angle for different Q and with $v = 2m/s, R = I$

The final output-feedback MPC controller was tested once with only the wind disturbance shown in figure 7 and once with the wind disturbance and the rough road disturbance that acts on the steering angle shown in figure 8.

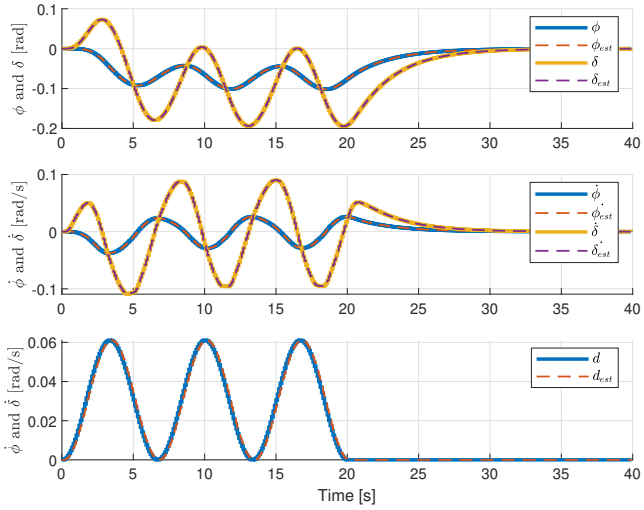

Fig. 7. Output-feedback MPC output all states and disturbance with estimates

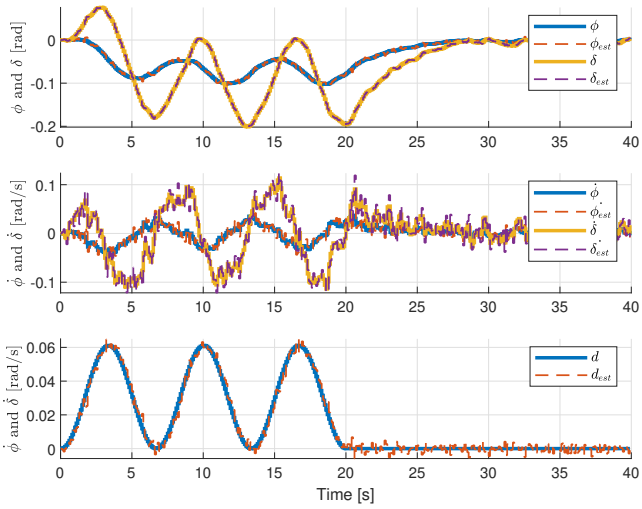

Fig. 8. Output-feedback MPC output all states and disturbance (including white noise) with estimates

The figures show the angle states with their estimates in the upper plot, the angular velocity states with their estimates in the mid plot and the disturbances with their estimates in the third plot. It clearly shows that the observer functions properly as all states are being tracked nicely. Besides that, the system manages to address the disturbances adequately.

In figure 8 it is visible that the noisy steering disturbance has its biggest impact on the angular velocity states. This is due to the fact that small high frequency changes in the steering wheel angle cause the angular velocity of the steering wheel to fluctuate even more. The effect is less notable on the roll angle velocity which is desired as you do not want the rider and bicycle to swing at a high rate from left to right due to the bumpy "rough" road.

C. Adaptive MPC

For the adaptive MPC only one simulation was conducted. This was due to the very large computational time. Figure 9 shows the true and estimated angular states and the increasing velocity over time. When looking at ϕ it becomes apparent that at higher speeds the noisy road disturbance has a greater impact on this state. The observer starts to have difficult time to correctly estimate the true state. A possible solution to this could be variable observer weights that change with the velocity.

The second subplot shows that the estimation of the steering angle remains solid over all velocities with just a slight vibration in the high velocities. As the velocity increases the system has to rotate less in both angles to stabilize itself which is best seen in the steering angle.

V. Conclusion

In conclusion, the bicycle model proved to be an interesting system to control using the model predictive control strategy. However, this is not so straightforward as assumptions had to be made to simplify the model.

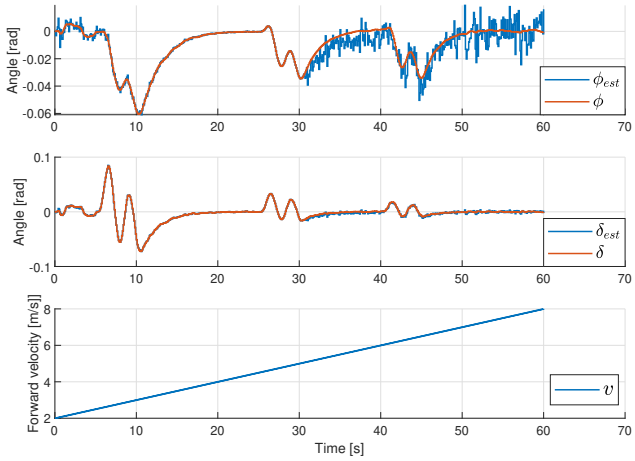

Fig. 9. Angular states ϕ and δ of the adaptive output feedback MPC with their respective estimates with a varying velocity from $2m/s$ to $8m/s$

It was found that it definitely is possible to control the bicycle in its upright position using the proposed MPC designs. The output-feedback design represents a more realistic approach to this problem than the state-feedback design. The designs were only tuned to a constant forward velocity of $2m/s$ as the model was the hardest to control at this velocity. A concept for an adaptive MPC controller that adapts to the changing forward velocity was briefly introduced. However, a static tuning was applied which did not work equally well for all velocities. As such it is recommended for future studies to design several tunings at several velocities and to make the controller switch tunings in between.

References

[1] M. Es and M. Slütter. Fietsen in cijfers. [Online]. Available: https://www.fietsersbond.nl/ons-werk/mobiliteit/fietsen-cijfers

[2] A. Schwab. Bicycle dynamics. [Online]. Available: http://bicycle.tudelft.nl/schwab/Bicycle/BicycleHistoryReview/index.html

[3] F. J. Whipple, ""The Stability of the Motion of a Bicycle," *The Quarterly Journal of Pure and Applied Mathematics*, vol. 30, pp. 312–348, 1899.

[4] R. S. Sharp, "On the Stability and Control of the Bicycle," *Applied Mechanics Reviews*, vol. 61, no. 6, 2008.

[5] E. Dohring, "Stability of single-track vehicles," *Institut fur Fahrzeugtechnik*, vol. 21, no. 2, pp. 50–62, 1955.

[6] J. I. Neimark and N. A. Fufaev, "Dynamics of non-holonomic systems," *A.M.S Providence RI*, 1972.

[7] J. M. Papadopoulos, "Bicycle Steering Dynamics and Self-Stability: a Summary Report on Work in Progress," *Technical report, Cornell Bicycle Research Project*, pp. 1–23, 1987.

[8] A. L. Schwab, J. P. Meijaard, and J. M. Papadopoulos, "Benchmark Results on the Linearized Equations of Motion of an Uncontrolled Bicycle," *Journal of Mechanical Science and Technology*, vol. 19, no. S1, pp. 292–304, 2005.

[9] L. Keo and M. Yamakita, "Control of an autonomous electric bicycle with both steering and balancer controls," *IFAC Proceedings Volumes*, vol. 25, no. 1-2, pp. 1–22, 2011.

[10] C. K. Chen and T. Dao, "Fuzzy control for equilibrium and roll-angle tracking of an unmanned bicycle," *Multibody System Dynamics*, vol. 51, no. 4, pp. 321–346, 2006.

[11] T. D. Chu and C. K. Chen, "Modeling and Model Predictive Control for a Bicycle-Rider System," *Vehicle System Dynamics*, vol. 56, no. 1, pp. 1–24, 2017.

[12] A. Saccon, J. Hauser, and R. Frezza, "Control of a bicycle using model predictive control strategy," *IFAC Proceedings Volumes*, vol. 37, no. 13, pp. 633–638, 2004.

[13] S. M. Cain, J. A. Ashton-Miller, N. C. Perkins, and J. M. Haddad, "On the skill of balancing while riding a bicycle," *PLOS ONE*, vol. 11, no. 2, pp. 1–18, 2016.
[14] M. R. Garica, D. A. Mantaras, J. C. Alvarez, and D. B. F., "Stabilizing an urban semi-autonomous bicycle," *IEEE Access*, vol. 6, pp. 5236–5246, 2018.
[15] A. E. Berndt and J. M. Bekendam. , "quadrotor-balancing-pendulum-mpc. [Online]. Available: https://github.com/alexberndt/Quadrotor-Balancing-Pendulum-MPC
[16] E. Gilbert and K. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
[17] J. Moore. Davis instrumented bicycle — human control of a bicycle: Jason k. moore. [Online]. Available: https://moorepants.github.io/dissertation/davisbicycle.html
[18] J. Rawlings and M. D. Q., *Model Predictive Control Theory and Design*. Nob Hill Pub, Llc, 2009.

APPENDIX

A. Bicycle model parameters of the Schwab Benchmark model

Parameter	Symbol	Value
Wheel base	w	1.02 m
Trail	t	0.08 m
Head angle	α	$\arctan(3)$
Gravity	g	9.81 N/kg
Forward speed	v	*variable* m/s
Rear wheel		
Radius	R_{rw}	0.3 m
Mass	m_{rw}	2 kg
Mass moments of inertia	(A_{xx}, A_{yy}, A_{zz})	$(0.06, 0.12, 0.06)$ kgm^2
Rear frame		
Position centre of mass	(x_{rf}, y_{rf}, z_{rf})	$(0.3, 0, -0.9)$ m
Mass	m_{rf}	85 kg
Mass moments of inertia	$\begin{bmatrix} B_{xx} & 0 & B_{xz} \\ & B_{yy} & 0 \\ sym. & & B_{zz} \end{bmatrix}$	$\begin{bmatrix} 9.2 & 0 & 2.4 \\ & 11 & 0 \\ & & 2.8 \end{bmatrix}$ kgm^2
Front frame		
Position centre of mass	(x_{ff}, y_{ff}, z_{ff})	$(0.9, 0, -0.7)$ m
Mass	m_{ff}	4 kg
Mass moments of inertia	$\begin{bmatrix} C_{xx} & 0 & C_{xz} \\ & C_{yy} & 0 \\ sym. & & C_{zz} \end{bmatrix}$	$\begin{bmatrix} 0.0546 & 0 & -0.0162 \\ & 0.06 & 0 \\ & & 0.0114 \end{bmatrix}$ kgm^2
Front wheel		
Radius	R_{fw}	0.35 m
Mass	m_{fw}	3 kg
Mass moments of inertia	(D_{xx}, D_{yy}, D_{zz})	$(0.14, 0.28, 0.14)$ kgm^2