Model Predictive Control for a ball balancing plate

Kenrick Trip (4661826), Laurens Hoogenboom (4609638),

***Abstract*—This paper aims to control the balancing of a ball on a controllable surface through MPC. Target regulation, output-feedback and state-feedback MPC are implemented and their properties are discussed. Stability analysis is be performed on the system to show the effectiveness of MPC. Target regulation is be compared to LQR control, after which state and output MPC are compared on disturbance rejection, reference tracking and trajectory tracking performance.**

I. Introduction

This paper concerns the stabilization of a ball balancing plate using model predictive control (MPC). The MPC is first implemented for target regulation, and then as state feedback and output feedback. Reference tracking is performed online, enabling trajectory tracking with live updating of the trajectory. The derivation of the system Dynamics was done by ourselves, of which the notes can be found in [1]. The system was linearized, resulting in a linear MIMO system in state space representation. The target regulation is compared to an unconstrained LQR controller and the tuning of the cost function will be elaborated. The full code repository can be found at [2], here it is clear that YALMIP was used as a solver.

II. Model Dynamics

The dynamics of the system can be describe in terms of three coordinate frames. Namely, the world frame ($\mathcal{F}_w$) as seen in figure 1, the plate frame ($\mathcal{F}_p$) and the ball frame ($\mathcal{F}_b$). Both seen in figure 2.

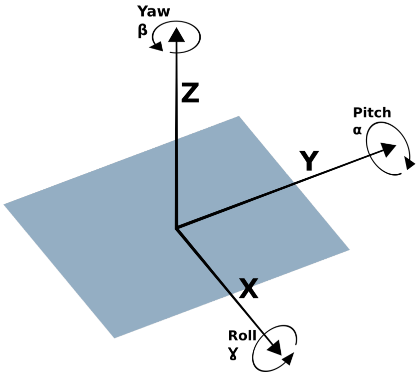

Fig. 1. World fixed frame $\mathcal{F}_w$

A. Frames

The origin $\mathcal{F}_w$ will be taken at the center (of rotation) of the plate in horizontal pose. $\mathcal{F}_p$ never translates in $\mathcal{F}_w$, but does rotate with just pitch and roll. Yaw will not be considered. $\mathcal{F}_b$ never rotates in $\mathcal{F}_p$, but does translate in the X_pY_p-plane in $\mathcal{F}_p$.

This means that the frame transformation from $\mathcal{F}_w$ to $\mathcal{F}_p$ will be as follows:

$$
{}^{w}\mathcal{F}_p = \begin{bmatrix} cos(\gamma) & -sin(\gamma)cos(\alpha) & sin(\gamma)sin(\alpha) \\ sin(\gamma) & cos(\gamma)cos(\alpha) & -cos(\gamma)sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \quad (1)
$$

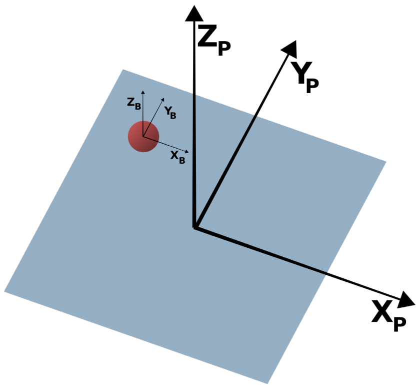

Fig. 2. Body fixed frames $\mathcal{F}_p$ & $\mathcal{F}_b$

B. System Dynamics

The system is actuated by 4 electric motors, connected halfway on the edges of the square plate. Each motor exerts a torque on a linkage that converts it to a horizontal and vertical force acting in the fixed world plane $\mathcal{F}_w$. These forces can be viewed a the input of the system. The inputs are therefore constrained by the maximum torque that the motor could apply at a given rotational speed.

It is assumed that the motor linkages have zero mass and that the ball never slips and that pure rolling can be assumed. Furthermore, since the system is symmetric and due to the linkage system connecting the motors to the plate, the plate is assumed to rotate purely around its center of mass (COM) without translating.

With these assumptions a state space description can be constructed using the equations of motion of the plate and the ball in the LTI form. Since the system is strongly non-linear, it is linearized around the origin which is an equilibrium point of the system.

$$
\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (2)
$$

Where u is the input force vector and x is that state vector of the system, denoted as follows.

Kenrick Trip and Laurens Hoogenboom are master students at TU Delft, The Netherlands. E-mail addresses: {k.trip, l.p.hoogenboom}@student.tudelft.nl.

$$\begin{aligned} x &= [x_b, \dot{x}_b, y_b, \dot{y}_b, \alpha, \dot{\alpha}, \gamma, \dot{\gamma}]^T \\ u &= [F_{y1}, F_{y2}, F_{y3}, F_{y4}]^T \end{aligned} \tag{3}$$

The system has 8 different states, namely the (x, y) position of the ball x_b and y_b, the (x, y) velocities of the ball $\dot{x}_b$ and $\dot{y}_b$, the pitch (α) and roll (γ) angles of the plate and the rotational velocities of the plate $(\dot{\alpha})$ and $(\dot{\gamma})$. Due to the linearization, the horizontal forces acting on the plate can be left out of the input vector u. This is a logical result as these forces do not do any work, but are constraint forces that prevent the COM of the plate from translating. For that reason, the system has 4 inputs, which are the 4 vertical forces applied by the motors on the plate $F_{yn}, n = 1, 2, 3, 4$. Forces are chosen as the system inputs so that different motor configurations could be implemented with different linkages or motor torque curves.

The system is discretised for a time step of 0.1 seconds with a prediction horizon N of 15, the choices of which will become clear in section III.

III. MPC CONTROLLER

An MPC controller is implemented with the aim of stabilising and obtaining the desired system behaviour in different scenarios.

- **Scenario 1:** Target regulation
- **Scenario 2:** Disturbance rejection
- **Scenario 3:** Reference trajectory tracking

The MPC controller exploits a specified plant model, which is unique to each scenario. The controller consists of a prediction model in a combination with an optimisation algorithm. A reference signal $r(k)$ and output feedback $y(k)$ or state feedback $x(k)$ are converted by the controller to a control input $u(k)$.

The MPC controller aims to achieve optimal control, by minimising the cost function $V_N(x_0, u_N)$ that consists of a stage cost $l(x(k), u(k))$ and a terminal cost $V_f(x(N))$ for a certain prediction horizon N. This is an optimisation problem that needs to be solved for each new current state x_0, as described in equation 4. Coupled with the system dynamics, the system constraints and the terminal constraint, the Optimal Control Problem (OCP) can be defined as in equation 5 [3].

$$V_N(x_0, u_N) = \ell(x(k), u(k)) + V_f(x(N)) \tag{4}$$

$$P_N(x_0, t) : \begin{cases} \min\limits_{u_N} \sum_{k=0}^{N-1} V_N(x_0, u_N) \\ s.t. \quad system\ dynamics \\ \qquad\ \ system\ constraints \\ \qquad\ \ x(N) \in X_f \end{cases} \tag{5}$$

A. Target regulation MPC

In this scenario, full information about the system is assumed. This means that there are no disturbances or noise signals that interfere with the system described in equation 2. The MPC aims to control the system to a target equilibrium point $x_e = 0$ to obtain stability.

The control problem can be formulated as a Linear Quadratic (LQR) problem with an infinite prediction horizon N. The stage costs can be formulated with a quadratic function depending on a matrix Q and R and the terminal costs can be formulated using a matrix P, which is obtained by solving the Discrete Algebraic Riccati Equation (DARE). The formulation of the stage cost, terminal costs and the DARE are given in equations 6, 7 and 8 respectively [3].

$$\ell(x(k), u(k)) = x(k)^T Q x(k) + u(k)^T R u(k) \tag{6}$$

$$V_f(x(N)) = x(N)^T P x(N) \tag{7}$$

$$P = A^T P A - (A^T P B)(R + B^T P B)^- 1(B^T P A) + Q \tag{8}$$

In order to fully define the optimal control problem, the constraints of the system must be imposed. The inputs of the system are constrained by the maximum torque T_{max} that the motors can exert on the linkages and the length of the maximum moment arm r_{max}, so that $F_y n \leq \frac{T_{max}}{r}$. Hence, u is constrained as described in equation 9.

$$\mid u_i \mid \leq \frac{T_{max}}{r_{max}} = 0.5N, \qquad i = 1, 2, 3, 4 \tag{9}$$

There are also state constrains present, introduced by the size of the plate and the maximum rotation angle of the plate. Since the plate has a square surface of $(300, 300)mm$ and the frame $\mathcal{F}_p$ has its origin at the center. Furthermore, the system configuration allows the maximum angles α en γ that the plate can reach to be bounded to $|45\deg|$. There is not a significant constraint of the velocities of the ball $\dot{x_b}$ and $\dot{y_b}$ or the rotational velocity of the plate $\dot{\alpha}$ and $\dot{\gamma}$, however high velocities would increase the wear of the system over time. For that reason, some constraints can be set. This leads to the state constraint formulation stated in equation 10.

$$\mid x \mid \leq \begin{pmatrix} 0.15\ m \\ 2.0 \quad ms^{-1} \\ 0.15\ m \\ 2.0 \quad ms^{-1} \\ \frac{\pi}{4} \quad rad \\ 3.0 \quad rads^{-1} \\ \frac{\pi}{4} \quad rad \\ 3.0 \quad rads^{-1} \end{pmatrix} \tag{10}$$

These constraints form a subspace $\mathbb{Z}$ based on the domain of input constraints $\mathbb{U}$ and state constraints $\mathbb{X}$, $\mathbb{X} \times \mathbb{U} := \mathbb{Z}$. This results in an admissible control sequence $\mathbb{U}_N$ for some prediction horizon N, as states in equation 11 [3]. Here $\mathbb{X}_f$ is the terminal constraint that should be reachable in N steps from controllable states $\mathbb{X}_N = \{x_0 \in \mathbb{R}^n \mid \mathbb{U}_N(x_0) \neq \emptyset$, which is further discussed in section IV.

$$\begin{aligned} \mathbb{U}_N(x_0) = \{u_N \mid x(k) = Ax(k-1) + Bu(k-1), \\ (x(k), u(k)) \in \mathbb{Z}, x(N) \in \mathbb{X}_f\} \subseteq \mathbb{U}_N \quad \forall k \in \mathbb{N} \end{aligned} \tag{11}$$

B. Disturbance rejection MPC - output feedback

In order to control the system to a certain point, a reference output of the system y_{ref} can be set. An offset-free MPC can be achieved using measured output feedback. For this model, the outputs are the x_b and y_b positions of the ball at each instant k. These positions can be measured by a fixed camera system, that measures these positions in the $\mathcal{F}_w$ frame, similar to what is done in [4]. The measurements should then be converted to the $\mathcal{F}_w$ frame, using the ${}^{w}\mathcal{F}_p$ matrix from equation 1. Any system is subject to disturbances d and any sensor is subject to noise v, w, which is often zero mean. This leads to an augmented system for the linearized model discussed in section II, described by equation 12 [3].

$$\begin{bmatrix} x^+ \\ d^+ \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} w \\ w_d \end{bmatrix}$$

$$y = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + v \tag{12}$$

The aim for the offset-free controller is for $y(k)$ to approach y_{ref} even if an unknown disturbance acts on the system. To achieve this, a reference input and state must be derived to track this reference using optimal target selection (OTS), as stated in equation 14[3]. Since the system is subject to disturbance, the Luenberg observer can be used to estimate this disturbance. The disturbance can be estimated using equation 13 [3] and is used for optimal target selection. Here $\tilde{A}$, $\tilde{C}$ and y are derived from the augmented state space 12 and $\tilde{L}$ is the Luenberg gain for which $\tilde{A} - \tilde{L}\tilde{C}$ is stable.

$$\begin{bmatrix} \hat{x}^+ \\ \hat{d}^+ \end{bmatrix} = \tilde{A} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \tilde{L}(y - \tilde{C} \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix}) \tag{13}$$

$$(x_{ref}, u_{ref}) : \begin{cases} \min\limits_{x_r, u_r} J(x_r, u_r) \\ s.t. \quad \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} B_d \hat{d} \\ y_{ref} - C_d \hat{d} \end{bmatrix} \\ \qquad (x_r, u_r) \in \mathbb{Z} \\ \qquad C x_r + \hat{d} \in \mathbb{Y} \end{cases} \tag{14}$$

The optimal target selection is computed online, after which the control problem is solved for $\tilde{x} = x - x_{ref}$ and $\tilde{u} = u - u_{ref}$ and the state x^+ is updated according to equation 12. The system can then be controlled offset free to a reference output y_{ref} with disturbance rejection.

C. Reference and trajectory tracking MPC - state feedback

For tracking a fixed reference, the optimal target selection can be computed offline by solving the OTS problem in equation 14 for the fixed y_{ref}, $B_d = C_d = \hat{d} = 0$ in the initialisation of the controller. This also means that no observer is required since we assume that no disturbances act on the system.

The ultimate aim is to control the system along a certain trajectory, a reference output of the system y_{ref} can be set that varies over time. In order to track the trajectory, r_{ref} needs to be updated at each time step, therefore the optimal target selection should be recomputed at each time step. Since the OTS problem is solved at each time step, which obtains state feedback MPC. For this, the current state x needs to be measured and (A, B) should be controllable. In section III-B, it is described how we can measure the output using a camera and feature recognition. This information can then be used to estimate $(\dot{x}_b, \dot{x}_y)$ by dividing the change in position over time. A IMU can be used to measure the orientation and rotational velocities of the plate $(\alpha, \dot{\alpha}, \gamma, \dot{\gamma})$. It is clear that for this system measuring the state requires significantly more hardware and other overhead.

For trajectory tracking, the reference positions of the ball can be calculated in the $\mathcal{F}_b$ frame for a specific rotational velocity ω and radius r of the reference trajectory around the center of the plate. Changing the radius with time, a spiralling trajectory can be created. The reference positions $y_{ref}(k)$ are updated at every time step k, according to equations 15 and 16, where t is the total simulation time.

$$x_{b_{ref}}(k) = \frac{k}{t} r cos(\omega k) \tag{15}$$

$$y_{b_{ref}}(k) = \frac{k}{t} r sin(\omega k) \tag{16}$$

$$y_{ref}(k) = \begin{bmatrix} x_{b_{ref}}(k) \\ y_{b_{ref}}(k) \end{bmatrix}$$

Random disturbance d and zero mean noise v can be added to the system in order to simulate more realistic operating conditions. This results in the augmented state space model described by equation 17. This is similar to the augmented state space obtained in section III-B, again we use the Luenberg observer to estimate the disturbance $\hat{d}^+$ described in equation 18 [3].

$$\begin{bmatrix} x^+ \\ d^+ \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + v \tag{17}$$

$$\hat{d}^+ = (I - L)\hat{d} + L(y - Cx) \tag{18}$$

The optimal target selection described in equation 14 can be repeated for $B_d = 0$ and $C_d = I$. Since disturbances are added to the system and the reference is updated at each tie step, this problem needs to be solved online. This leads to the following cost functions, described in equation 19 [3]. The cost function needs to be minimised for the admissible control sequence $\mathbb{U}_N(x_0, y_{ref})$ and the terminal constraint $\mathbb{X}_f(\hat{d}, y_{ref})$.

$$\begin{aligned} V_N(x_0, \hat{d}, y_{ref}, u_N) = \ell(x(k) - x_{ref}(\hat{d}), u(k) - u_{ref}(\hat{d}) \\ + V_f(x(N)) - x_{ref}(\hat{d}) \end{aligned} \tag{19}$$

IV. Stability evaluation

To achieve stability, the system must meet the following assumptions for $\forall x \in \mathbb{X}_f$ and $\exists u \in \mathbb{U}$ [3].

- f, linear ℓ, V_f, quadratic
- $\mathbb{X}$ closed, $\mathbb{U}(t), \mathbb{X}_f \subseteq \mathbb{X}(t)$ compact, for all t
- $f(x, u) \in \mathbb{X}_f$
- $V_f(x, u) \leq V_f(x) - \ell(x, u)$

In the following subsections, the design choices of the controller are motivated so that the controller achieve optimal control whilst fulfilling the aforementioned stability conditions.

A linearised system is controllable if the controlability matrix (A, B) is of full rank. In order to solve the MPC problem the Q and R matrices described in equation 4 should be positive definite for the quadratic cost function. Q and R are chosen as diagonal matrices with positive values, varying for each MPC controller considered in this paper. The P matrix in the terminal cost is chosen as the solution to the DARE. With these choices, it can be assured that f is linear and ℓ, V_f is quadratic.

Furthermore, since the selected boundary conditions for $\mathbb{X}$ and $\mathbb{U}$ apply on the absolute values, $\mathbb{X}$ and $\mathbb{U}$ are closed and compact. With these conditions we can construct the MPC controllers.

A. Terminal state estimation

An MPC problem is feasible if the system can be controlled to a terminal state $\mathbb{X}_f$ within k steps, where $k \in (0, N)$, while adhering to all the input and output constraints mentioned in section II. The system must therefore be able reach the terminal state in less or equal to than the prediction horizon N amount of steps. One method to ensure stability is to meet the condition stated in equation 20 to hold for some constant $c > 0$, where P is the stabilising solution to the DARE equation stated in equation 8 and K is the belonging state-feedback gain.

$$\begin{aligned} V_f(x) &= \frac{1}{2} x^T P x \\ \mathbb{X}_f &= \{x \mid v_f(x) \leq c\} \\ \mid Kx \mid &\leq \frac{1}{2}, \quad \text{for all } x \in \mathbb{X}_f \end{aligned} \tag{20}$$

Solving equation 20 for c, returns a value of $c = 0.56$. This means that all values v on the ellipse spanned by the eigenvectors of p hold that $v < 0.5$. This is a relatively large value, meaning that the ellipsoidal invariant set for the terminal constraint is relatively large.

B. Feasible set estimation

The feasible $\mathbb{X}_N$ set of the system describes a set of initial states x_0 for which the MPC control problem is feasible for the horizon N, if we assume that the system converges towards $\mathbb{X}_f$ [5]. Such a feasible set can be found by starting from many random initial conditions x_0 that lay within the state constraints stated in equation 10 and verifying whether they converge to $\mathbb{X}_f$ in N steps, as shown in figure 3. Due to the large domain of computations that would be needed to compute an entire set of converging initial states, a minor subset of initial states x_0 we obtained iteratively.

C. Lyapunov decrease

In order to achieve Lyapunov asymptotical stability, the assumption $V_f(x, u) \leq V_f(x) - \ell(x, u)$ must hold. This is validated in figure 4 for $Q = 0.5I$ and $R = 0.3I$.

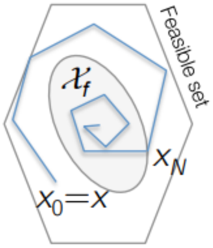

Fig. 3. Feasible set $\mathbb{X}_n$ [5]

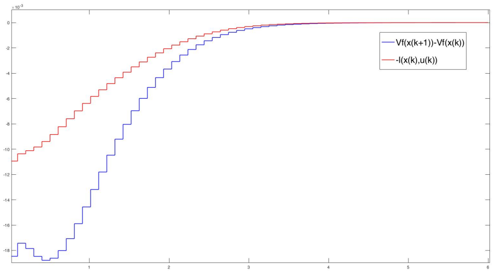

Fig. 4. Position of ball for different prediction horizons with identical tuning

If this condition is not met, the cost function will not converge and stability will not be achieved. From figure 4, it can be observed that the system adheres to this condition with the aforementioned quadratic cost function formulation. With all these conditions met, the MPC controllers can be constructed.

V. Numerical simulations

A. Prediction horizon

The prediction horizon determines (N) how far the prediction model tries to estimate forward in time. A small N implies less computationally expensive control, but also poses a risk of not predicting far ahead enough, preventing the system from converging to the terminal set.

In order to demonstrate this a series of simulations was performed with identical settings, apart from the prediction horizon. The initial state here was $x_0 = \begin{bmatrix} 0 & 0 & 0 & .01 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$. i.e. The ball is placed in the middle with a initial velocity of 10^{-2} m/s in one primary direction. Figure 5 shows the location of the ball in this primary direction for different horizons. Closer inspection shows that in this case $N = 12$ is not stable, while $N = 13$ does converge. However, $N = 15$ shows a significantly faster convergence. $N = 20$ shows a similar convergence, but is computationally significantly more expensive. Hence, the trade off will be made that $N = 15$ is sufficiently fast, yet cheap enough.

B. Comparison with LQR

In order to compare target regulation MPC and LQR the performance was plotted in identical scenarios. However, the MPC controller was constrained, while the LQR controller was

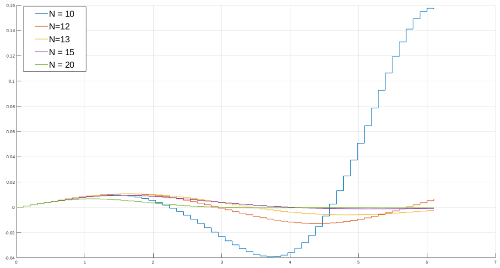

Fig. 5. Position of ball for different prediction horizons with identical tuning

not. We believe this plays a role in the LQR controller having better performance when solely the position and velocity of the ball are considered, as can been seen in figure 6. This counts for both an initial state x_0 that is inside and outside the terminal set $\mathbb{X}_f$.

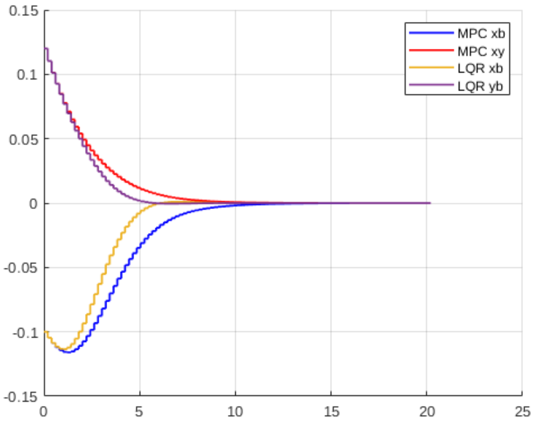

Fig. 6. Position of ball for MPC and LQR controllers with identical x_0 outside $\mathbb{X}_f$

Looking at figures 6 and 7 there is no clear difference in behaviour of the ball for the different controllers in either scenario. However, it is visible that the difference in performance is smaller when the x_0 is outside of $\mathbb{X}_f$. While the LQR controller uses more agressive control action, the MPC controller approaches similar performance with less agressive action. This can be explained as the model prediction taking its effect. The LQR controller still manages to have faster converge, but requires higher input gains and more extreme angles and angular velocities of the plate. If the LQR controller were constrained, this might not have been the case.

We did not put constraints on the LQR controller as the focus was shifted to improving the MPC. It is possible that doing so would result in the MPC controller performing better outside of $\mathbb{X}_f$. We can see that the input gains and behaviour of the plate is significantly larger in the LQR controller. Impeding the LQR controller in doing so will likely cause it to decrease performance quality.

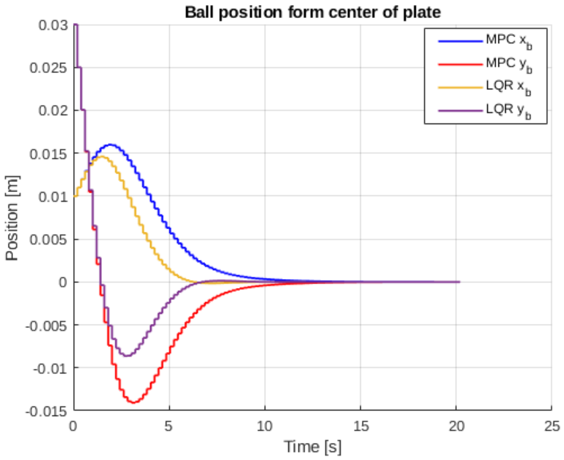

Fig. 7. Position of ball for MPC and LQR controllers with identical x_0 inside $\mathbb{X}_f$

C. Cost function tuning

Figures 8 and 9 show comparisons of one of the ball position states for different Q and R matrices respectively. In this example the initial states for either axis were identical so the exact axis this is on is trivial. In MPC the Q, R and P matrices are used in the cost function. However, as mentioned before the P matrix is found through solving the Riccati equations and is thus not tuned. The stage cost as in equation 6 can be tuned with Q and R, depending on the desired behaviour. The Q matrix represents the state cost and the R matrix represents the input cost. In a simple scenario both these matrices are diagonal, where each element corresponds with one state or input. Depending on which states and inputs should be penalised most one might individually tune the elements in the Q and R matrices. However, In this case all elements in the Q matrix are identical, as is the case with the R matrix. The sole reason for this is time constraints on the project.

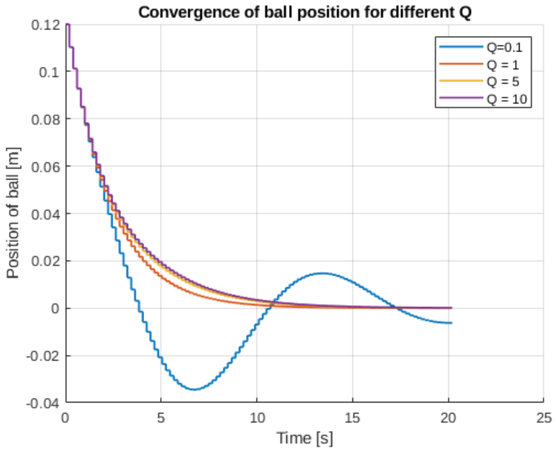

Fig. 8. Position of ball with regulation MPC controller for different R values.

Figure 8 shows that a higher matrix norm of Q Will cause the system to converge eventually without an overshoot, but will increase the time before it first reaches the target state.

The value of $Q = 0.1$ shows a longer transient behaviour. One could interpret this as having a low penalty on the states, meaning that be comparison the input cost is high. In the case of a ball balancing plate this means the angular acceleration of the plate will be low. The controller will allow the ball to have high velocity and reach the edges of a tilted plate, rather than having to use a lot of force to make the ball converge to the centre of the plate. The converse can be said for having a high state cost and a low input cost. Logically it follows form the kinematics that the former will allow for overshoot, while this is undesirable in the latter case.

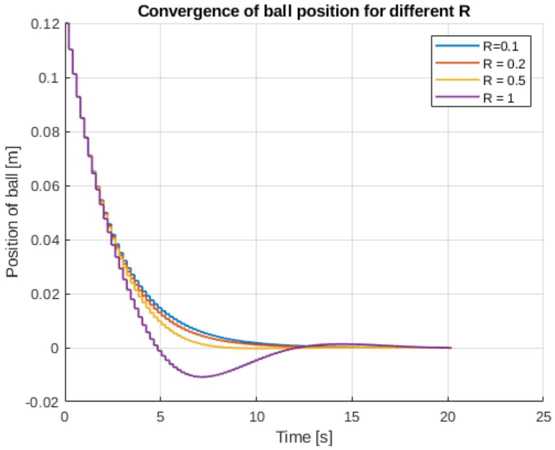

Fig. 9. Position of ball with regulation MPC controller for different R values.

This explains why increasing the norms of the Q and R matrices have an inverse effect on the stabilisation behaviour, as can be seen in figures 8 and 9. From a process of trial and error the Q and R matrices were tuned to be $Q = 0.5I$ & $R = 0.3I$. Further tuning of the individual elements could result in a better nuanced controller where, for example, the position of the ball has a higher state cost than its velocity. i.e. We care more about the ball's overshoot being reduced than the ball remaining slow.

D. Disturbance rejection

In order to reject disturbances, a Luenberg observer can be used to estimate the state and disturbance according to an augmented state space model as is described in section III-B. This allows the system to track a reference whilst rejecting disturbances on the system. Two different techniques are used to obtain offset free MPC, namely output feedback and state feedback. In output feedback the system outputs are measured and with state feedback the states of the system are measured. Figure 10 shows the results obtained using output feedback and figure 11 shows the results obtained using state feedback. Here, a step disturbance is directly placed on the output. In other words, the ball is physically disturbed in its position and the MPC aims to adjust to the offset and get the ball back towards the reference position. During this time, a step change in the reference is also introduced to further explore system behavior.

It can be observed that bot plots look very similar, the main difference between the state and output feedback methods is that in the output feedback method the state is estimated. The similarity of the graphs indicates that the observer gain L was well chosen. The orientations and rotational velocities of the plate can also be plotted as shown in figure 12, to indicate the effect on other states.

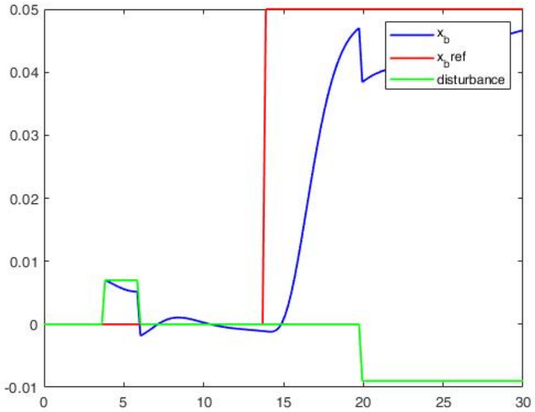

Fig. 10. Disturbance rejection with output feedback

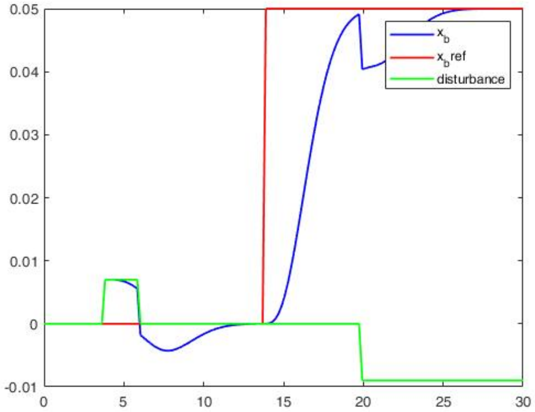

Fig. 11. Disturbance rejection with state feedback

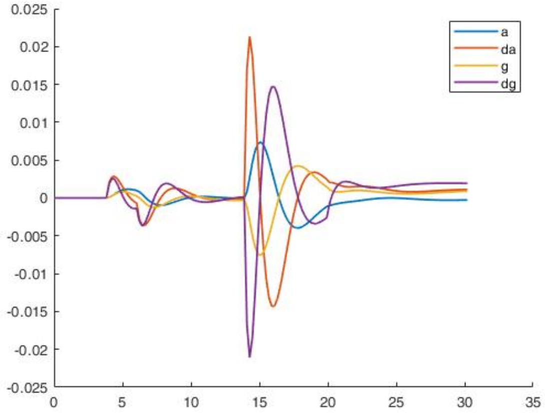

Fig. 12. States $(\alpha, \dot{\alpha}, \gamma, \dot{\gamma})$ during disturbance rejection

The system has a response time of around 5 seconds to move the ball to the desired location.

E. Trajectory tracking

In order to track a trajectory the OTS problem described in equation 14 can be solved online and a new y_{ref} can be calculated according to the desired trajectory, as described in section III-C. Figure 5 shows the reference tracking as described before, using the state feedback MPC.

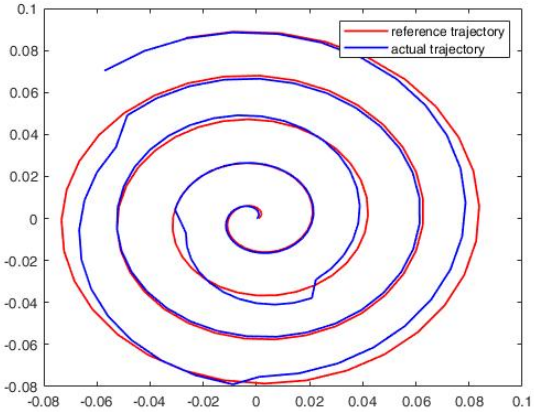

Fig. 13. Position of ball for different prediction horizons with identical tuning

Since the system response time was rather slow, the trajectory could not be accurately followed for large rotational velocities. It can also be observed that disturbances were added to the system, hereby proving the robustness of the MPC design as it is still able to track the reference trajectory relatively accurately. Since we constructed an output feedback MPC in section III-B and the results on disturbance rejection in section V-D were comparable, the output feedback MPC was also tested the follow this trajectory. As can be observed in figure 14, the results are very similar, as expected.

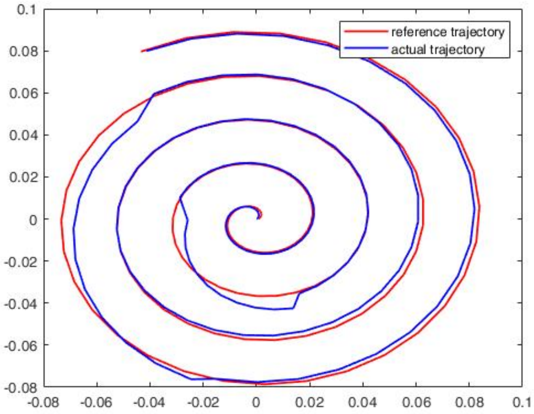

Fig. 14. State feedback MPC - Trajectory tracking with disturbance

VI. Conclusion

The implementation of MPC on a ball balancing plate proved feasible. The derivation of the dynamics was performed by ourselves, which proved more time consuming than initially anticipated. The dynamics were thereafter linearized, to simplify the implementation of the controllers. All in all MPC proved useful in stabilizing the system, which aligns with previous research conducted on MPC and LQR control for similar systems [6]. The MPC was expanded by the addition of trajectory tracking an disturbance rejection. This enabled the MPC controller to perform more complicated tasks regarding the motion of the ball and clearly has potential to be more than just a stabilising controller.

Unfortunately, the target regulation MPC did not surpass LQR in this project when regarding solely position convergence, but with more work we believe this is certainly possible.

In further work we would like to elaborate on both the MPC and LQR controllers to be able to make a better comparison and to get more insight in when to use the controllers. We would like to improve response time of the system by changing the physical properties of the system to improve reference tracking. For future research, adaptive MPC could also be applied to the system to further improve reference tracking.

References

[1] L. Hoogenboom and K. Trip, "System Dynamics ball balancing plate," http://studiocodeorange.nl/MPC/MPC_assignment.pdf, accessed: 2021-04-09.

[2] ——, "Github Repository ball balancing plate," https://github.com/LHoogenboom/BallBalancingPlate, accessed: 2021-04-09.

[3] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[4] C. J. Bay and B. P. Rasmussen, "Exploring controls education: A re-configurable ball and plate platform kit," *2016 American Control Conference (ACC)*, 2016.

[5] C. Jones and M. Zeilinger, "Constraint satisfaction and stability: Theory in model predictive control," 2010. [Online]. Available: https://www.uiam.sk/pc11/data/workshops/mpc/MPC_PC11_Lecture1.pdf

[6] J. F. Pasha and S. Mija, "Discrete laguerre based mpc for constrained asymptotic stabilization of 4 dof ball balancer systems," *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR)*, 2019.