A model predictive control approach for multi link flexible manipulators.

Pim de Bruin (4545702), Bram Benist (4281926)

***Abstract*—A model predictive control (MPC) approach for a multi link flexible manipulators is designed and simulated. Two different MPC are designed, one for where we assume the ability of perfect state measurement and one where we only measure the noisy output with a constant disturbance. Assumptions are verified for asymptotic stability and the controllers are simulated for a reference signal. A comparison to state feedback controller from a different paper is also made.**

I. Introduction

The topic for this control design is the control of multi link flexible manipulators. This system comprises of two flexible links connected by a joint. A payload mass is attached to the end of link 2. link 1 is connected to the fixed world by another joint. Figure 1 presents the system as drawn in [1].

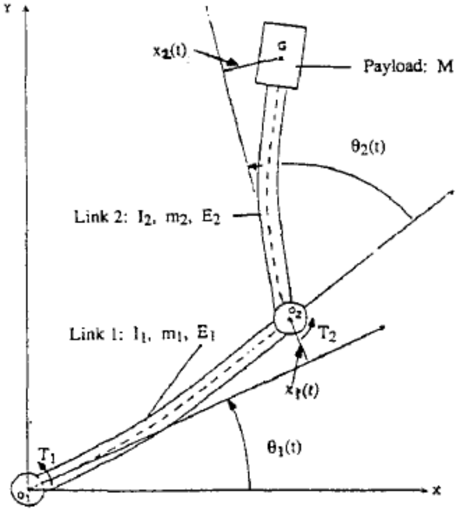
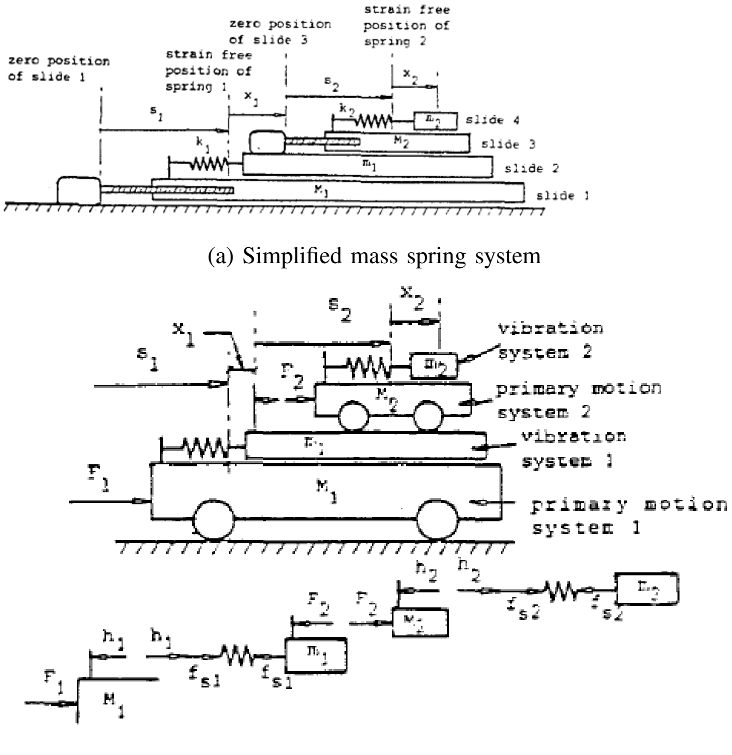


Fig. 1: multi link flexible system from [1].

The control of flexible manipulators can be applied to a wide range of industries. Flexible manipulation would allow for a less stiffness in the links, thus potentially reducing cost and weight while maintaining proper control of the system. This would result in lighter and cheaper manipulators for all sorts of applications. The traditional approach to multi-link manipulation does not take into account any form of deflection from the links and can thus easily be solved by using a form of feed-forward control, or a PID type feedback controller. When information about the flexibility of the joints is available, a more efficient and optimal control strategy can be achieved by an MPC controller as will be shown in this paper.

The main topic of interest in [1] is wether the multi link flexible manipulator can be controlled using a simplified model derived from the setup presented in figure 2a. This system consists of multiple links of masses and springs. Further abstraction of this system is done in figure 2b to obtain a simple mass-spring system which can be modelled in a linear fashion. The full derivation can be found in [1].



(a) Simplified mass spring system

(b) Further simplified system modelling

Fig. 2: overview of the simplified mass spring system.

For the modelling, the inputs are chosen as the forces on the two moving carts in figure 2b:

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{1}$$

Which correspond to the input torques T_1, T_2 in the original system in figure 1. The outputs of the system are chosen as the distance both primary motion systems have travelled from some reference point shown in figure 2a:

$$y = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{2}$$

Pim de Bruin and Bram Benist are master students at TU Delft, The Netherlands. E-mail addresses: {P.E.deBruin, B.P.Benist}@student.tudelft.nl.

Which corresponds to θ_1 and θ_2 in the original system. The states of the system are chosen as:

$$\mathbf{x} = \begin{bmatrix} x_1 & \dot{x}_1 & s_1 & \dot{s}_1 & x_2 & \dot{x}_2 & s_2 & \dot{s}_2 \end{bmatrix}^T \tag{3}$$

Which also incorporate the deflection of the two beams x_1 x_2 as states of the system.

Following the modelling procedure described in [1], the system is put into standard continuous time state space form:

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

With matrices:

$$A =$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_1\left(\frac{1}{M_1} + \frac{1}{m_1}\right) & -\frac{c_1}{m_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{k_1}{M_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_2\left(\frac{1}{M_2} + \frac{1}{m_2}\right) & -\frac{c_2}{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\mathrm{k}_1}{\mathrm{m}_1} & \frac{\mathrm{c}_1}{\mathrm{m}_1} & 0 & 0 & \frac{\mathrm{k}_2}{\mathrm{M}_2} & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} 0 & 0 \\ -\frac{1}{M_1} & -\frac{1}{m_1} \\ 0 & 0 \\ \frac{1}{M_1} & 0 \\ 0 & 0 \\ 0 & -\frac{1}{M_2} \\ 0 & 0 \\ 0 & \left(\frac{1}{m_1} + \frac{1}{M_2}\right) \end{bmatrix} \tag{5}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

The model thus falls into the linear Multi-input Multi-output category with 2 inputs, and 2 outputs.

To be able to implement this model in a MPC, it first needs to be discretised. This is done by applying the Zero Order Hold method using a sampling time $d_t = 0.1$s. This sampling time is found by finding a compromise between low sampling time and computational cost. Since the dynamics of the model are quite slow, a high sampling time is fine to still capture all relevant dynamics. The sampled system is represented by:

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx \end{aligned}$$

This system will be used in the design of the controllers.

II. Model predictive control design

To create a MPC we first set up the constrains of our system where the controller needs to operate in section II-A. After this we design an ideal state-measurement model predictive controller in section II-B. Since in reality it is not very common that all states are known we also design an output-measurement model predictive controller which will also incorporate disturbance estimation and rejection in section II-C.

A. Constraints

For this problem there are state and input constraints. The state constraints concern the angles of the joints as well as their angular velocity. Constraints are also applied to the deformation to make sure the beams are not deformed out of proportion. These linear constraints are presented in equation 7 and 8. These constraints represent feasibility of the solution. Equation 8 represents the workspace constraint where the entire manipulator is to remain within a certain workspace.

$$\begin{bmatrix} -\pi \\ -\infty \\ -\pi \\ -2\pi \\ -\pi \\ -\infty \\ -\pi \\ -2\pi \end{bmatrix} \le \begin{bmatrix} x_1(k) \\ \dot{x}_1(k) \\ s_1(k) \\ \dot{s}_1(k) \\ x_2(k) \\ \dot{x}_2(k) \\ s_2(k) \\ \dot{s}_2(k) \end{bmatrix} \le \begin{bmatrix} \pi \\ \infty \\ \pi \\ 2\pi \\ \pi \\ \infty \\ \pi \\ 2\pi \end{bmatrix}, \quad \forall k \tag{7}$$

$$-\frac{\pi}{2} \le x_1(k) + s_1(k) + x_2(k) + s_2(k) \le \frac{\pi}{2} \tag{8}$$

These constraints can be re-written in compact form as:

$$Fx(k) \le e$$

By splitting the inequality constraints up in a lower and upper bound, and multiplying the lower bound by -1. This is then added to the F and e matrix as 9 new constraints.

Since the input in reality can't be arbitrarily large, these have to be constrained as well. The two inputs of the system correspond to the two forces on the carts, or the two torques in the original model. The constraints on these are represented in equation 9. The values of the constraints are chosen iteratively to still allow for fast convergence, but not arbitrarily large input forces.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{u} \le \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}, \quad \forall k \tag{9}$$

To be able to implement the state constraints over the receding horizon of the MPC controller, the state evolution **x**(t) needs to be computed. This can be done by exploiting the linear structure of the model as:

$$\mathbf{x}_{N+1} = Tx_0 + S\mathbf{u}_N \tag{10}$$

$$= \begin{bmatrix} 1 \\ A \\ A^2 \\ \vdots \\ A^{N-1} \end{bmatrix} x_0 + \begin{bmatrix} 0 & \dots & & \dots & 0 \\ B & 0 & & & \vdots \\ AB & B & 0 & & \\ \vdots & & & \ddots & \vdots \\ A^{N-2}B & A^{N-3}B & \dots & B & 0 \end{bmatrix} \mathbf{u}_N \tag{11}$$

In the following controllers, a terminal constraint will also be implemented. This constraint states that the final state of the simulation horizon in the MPC has to be in the terminal set and can be written as:

$$x(N) \in \mathbb{X}_f \tag{12}$$

The terminal set $\mathbb{X}_f$ will be defined in the next section as a function of T which will be investigated in section III.

B. Ideal state-measurement controller

For the design of this model predictive controller it is assumed all state values for x_0 are measured and correct. Therefore we can design the stage and terminal cost function as in Equation 13.

$$\begin{aligned}\ell(x,u) &= x(k)^T Q x(k) + u(k)^T R u(k)\\ V_\mathrm{f}(x) &= \frac{1}{2} x(N)^T P x(N),\end{aligned} \tag{13}$$

Here Q and R are weighting matrices of size 8x8 and 2x2 respectively and P is the solution from the DARE equation. Since the objective state is a reference state, the states are translated to the origin by subtracting x_r and u_r. From this we obtain the optimisation problem as in Equation 14. Here Equation 11 is used to calculate $x(k)$ and $x(N)$

$$\begin{aligned}\min_{\boldsymbol{u}_N \in \mathbb{R}^{N\times 2}} & \sum_{k=0}^{N-1} \{\ell\left(x(k) - x_r, u_N(k)\right)\}\\ & + \frac{1}{2}(x(N) - x_r)^\top P (x(N) - x_r)\\ s.t. \quad & Fx(k) \le e\end{aligned} \tag{14}$$

To summarise the state-measurement controller, the entire problem is formulated in the standard optimisation scheme as in Equation 15. Here we use only x_r since we know all states and therefore assume we have the preferable state values. Where $V_N(x_0, \mathbf{u}_N, x_r) = \sum_{k=0}^{N-1} \ell(x - x_r, u) + V_\mathrm{f}(x - x_r)$

$$\begin{aligned}& \underset{\mathbf{u}_N}{min}\; V_N(x_0, \mathbf{u}_N, x_r)\\ & s.t.\; x^+ = Ax + Bu\\ & \quad F\mathbf{u}_N \le e - FTx_0\\ & \quad \begin{bmatrix} I \\ -I \end{bmatrix} \mathbf{u}_N \le \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}\\ & \quad x(N) \in \mathbb{X}_f\end{aligned} \tag{15}$$

C. Output-measurement controller

When looking at the state vector in equation 3, it is clear most states can not directly be measured like the deformation of the beams. Thus, an output measurement MPC is also designed.

Here x is not measurable. However the A and B matrix are controllable and A and C matrix are observable. Therefore we can create an observer to obtain an approximation for our states. The observer equation for our system is given as

$$\begin{aligned}\hat{x}^+ &= (A - KC)\hat{x}(k) + Bu(k) + Ky(k)\\ \hat{y}^+ &= C\hat{x}\end{aligned} \tag{16}$$

where the observer gain K is chosen such that the matrix $(A - KC)$ has much faster poles then the original system to make sure the observer converges quick enough.

This observer structure can easily be utilised to estimate constant disturbances for which can then be accounted for. This is relevant for this paper since multi link manipulators can have some sort of stick, slip and friction acting in the joints or have permanent deformation in the links. This can therefore be estimated by a constant disturbance.

This disturbance is modelled by adding an auxiliary state to the original dynamics. This auxiliary state d represents the constant disturbance. In addition, a measurement noise v is added to the systems output. A physical interpretation of this load disturbance could be be a friction phenomena, or an incorrect angle measurement.

$$\begin{bmatrix} x^+ \\ d^+ \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \tag{17}$$

$$y = \begin{bmatrix} C & C_d \end{bmatrix} + v \tag{18}$$

The original system is observable but to make sure the augmented system is also observable the order of the augmented system needs to be the order of the original system plus the order of the order of the constant disturbance:

$$rank \begin{bmatrix} I - A & -B \\ C & C_d \end{bmatrix} = n + n_d \tag{19}$$

Since this extra disturbance state is not known but observable if the previous equation holds. It can be added to the observer as given in equation 16 which gives the augmented observer equation as:

$$\begin{bmatrix} \hat{x}^+ \\ \hat{d}^+ \end{bmatrix} = \left(\begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} - \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix} \right) \begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} y \tag{20}$$

Using this, an Optimal target selection problem needs to be formulated and solved to be able to find the correct inputs that account for this estimated disturbance. This problem is compactly formulated as:

$$(x_r, u_r)(\hat{d}, y_r) \in \begin{cases} \underset{x_r, u_r}{argmin} & J(x_r, u_r)\\ s.t. & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ y_r - \hat{d} \end{bmatrix}\\ & (x_r, u_r) \in \mathbb{Z}\\ & Cx_r \in \mathbb{Y} \end{cases} \tag{21}$$

This problem needs to be solved online to find the optimal x_r and y_r which cancel out the disturbances $\hat{d}$ as much as possible. This all will create the following cost function for the optimisation problem:

$$\begin{aligned}V_N(x_0, \hat{d}, y_r, \mathbf{u}_N) = & \sum_{k=0}^{N-1} \left\{ \ell\left(x(k) - x_{\hat{d}}, u_N(k) - u_r(\hat{d}) \right) \right\}\\ & + V_\mathrm{f}(x(N) - x_r(\hat{d}))\end{aligned} \tag{22}$$

III. Asymptotic stability

In this section, the stability of the MPC controller designed in section II will be discussed. This will be done by discussing a set of assumptions that are necessary for determining stability. The ultimate result will be the proof of the exponential stability of the origin in $\mathbb{X}_N$ by the verification of the assumptions done in Theorem 2.19 and 2.21. Since the cost function defined in 13 is quadratic, and the state and input constraints are all linear and time invariant, the approach in section 2.5.4 of [2] will be followed.

Assumption 2.2

Verifying assumption 2.2 from [2], does not require extra computations. Recall that the model defined in the introduction is linear, and thus $f : \mathbb{Z} \to \mathbb{R}^n$ with $n = 8$ is continuous. As for the cost function, the stage cost function $l(x, u)$ as defined in section II is a quadratic function with weighting matrices $Q, R \succ 0$, which is also continuous as well as the terminal cost function with $P \succ 0$. Since the model itself is linear it has an equilibrium point at $x, u = 0$. Therefore Assumption 2.2 holds.

Assumption 2.3

The second assumption that needs to be verified is the assumption that $\mathbb{X}_f \subseteq \mathbb{X}$ and is compact.
The terminal set $\mathbb{X}_f$ is defined as: $\mathbb{X}_f = \{x \in \mathbb{R}^8 | V_f(x') \leq T\}$ where the terminal cost function $V_f(x)$ is smaller or equal to some positive parameter T. This T will later be chosen accordingly to be able to prove stability. The terminal cost function $V_f(x)$ is defined in section II as $V_f(x) = \frac{1}{2}x'^T P x'$ with $x' = x(N) - x_r$. P is the solution to the DARE. Since P is symmetric and positive definite, $x'^T P x'$ spans an 'ellipsoid' in $\mathbb{R}^8$. The eigenvectors of P define its principal axes. This is used to describe the terminal set $\mathbb{X}_f$ as the region where $V_f(x') \leq T$. To do this, P is first diagonalised as $P = SDS^T$ with S an orthonormal matrix containing the eigenvectors of P, and D a diagonal matrix with the eigenvalues of P on its diagonal. A coordinate change $x = Sy$ with y being the principal axes of P such that the terminal set can be described as $\frac{1}{2}y^T D y \leq T$ is defined. This equation describes the ellipsoid of the terminal set $\mathbb{X}_f$. The principal axes y_i intersect the boundary of this ellipsoid at the locations $\pm\sqrt{\frac{2T}{D_{i,i}}}$ where $D_{i,i}$ is the eigenvalue of P corresponding to principle axis y_i. To be able to prove this region is within the constrained set of stated $\mathbb{X}$, the ellipse can be bounded by a box tangent to the ellipse at the intersection of the principal axes and the boundary of the ellipse which is shown in figure 3 for the two dimensional case. Since the ellipsoid is bounded by the box and $\mathbb{X}$ is a convex set due to the linear constraints, checking the vertices of this box will be sufficient to check if $\mathbb{X}_f \subseteq \mathbb{X}$. For each of these vertices , it can be checked whether it lies within $\mathbb{X}$ by checking if it falls within the set constrains defined in section II.This can be formulated as:

$$[\text{lower bounds}] \leq \bar{y}_i \leq [\text{Upper bounds}] \tag{23}$$

Where $\bar{y}_i$ represents the i'th vertex of the bounding box. Since the full system contains 8 states, the vertices span an 8 dimensional hypercube with $2^8 = 256$ vertices. The computation of every vertex is done using matlab. Iteratively, a value of T = 0.1 is found for which the set of vertices are all contained within $\mathbb{X}$. Thus $\mathbb{X}_f$ is contained in the admissable states $\mathbb{X}$.

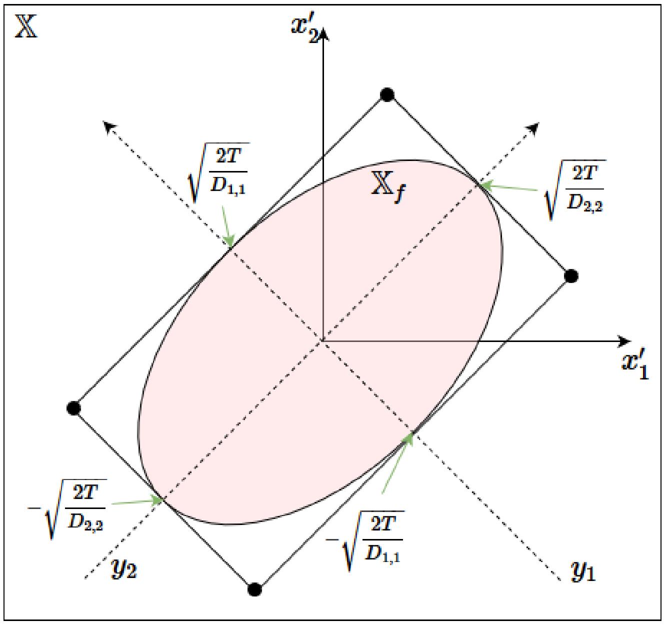

$\mathbb{X}$
x'_2
$\sqrt{\frac{2T}{D_{1,1}}}$
$\mathbb{X}_f$
$\sqrt{\frac{2T}{D_{2,2}}}$
x'_1
$-\sqrt{\frac{2T}{D_{2,2}}}$
$-\sqrt{\frac{2T}{D_{1,1}}}$
y_2
y_1

Fig. 3: Discription of bounding box for terminal set for the two dimensional case.

This result can be extended to also check whether $u \in \mathbb{U}$ by taking $u = Kx$ with K being the LQ optimal control gain. Again, we can empose the coordinate change $x = Sy$ with S being a vector containing the eigenvectors of P and $y^T D y \leq 0.2$. This can then be formulated as $K\mathbb{X}_f \in \mathbb{U}$. Which, can easily be checked by multiplying the vertices found in the previous proof by K, and checking whether this is still within the admissable control set:

$$\begin{bmatrix}-100\\-100\end{bmatrix} \leq K\bar{y}_i \leq \begin{bmatrix}100\\100\end{bmatrix} \tag{24}$$

With the upper and lower bounds defined in section II. Again, using matlab the vertices were computed and all lie within the admissable set. This thus concludes the proof of assumption 2.3.

Assumption 2.14

2.14(a) One important assumption is the lyapunov decrease of the cost function. Mathematically, this assumption is represented as:

$$V_f(f(x, u)) - V_f(x) \leq -l(x, u) \tag{25}$$

This assumption is verified after designing the state feedback MPC by plotting the left and right hand side of 25. This is shown in figure 4

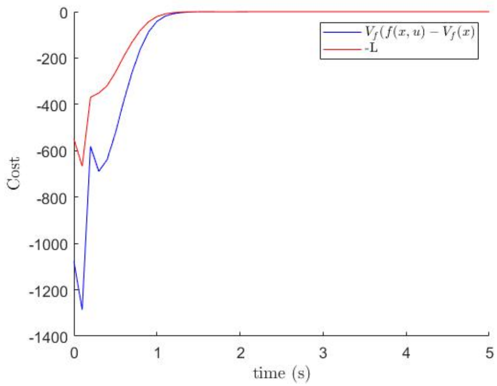


Fig. 4: Terminal cost difference and minus stage cost plotted over entire simulation

It is clearly shown that assumption 2.14(a) holds since minus the stage cost is always larger then the difference in terminal cost.

2.14(b) can be checked by looking at the defined cost function for the stage cost and the terminal cost defined in equation 13. It can be seen that the stage cost $\ell(x', u')$ is a quadratic function which can be lower bounded by a $\mathcal{K}_\infty$ function which in terms can be found exploiting the fact that Q, R are positive definite and the triangle inequality.

$$\begin{aligned} x'(k)^T Q x'(k) + u'(k)^T R u'(k) &\geq x'(k)^T Q x'(k) \\ &\geq \underbrace{\lambda_{min}(Q)||x'(k)||^2}_{\alpha_1(||x'||)} \end{aligned} \tag{26}$$

Where $\lambda_{min}(Q)$ is the smallest eigenvalue of Q. This can also be applied to the terminal cost as:

$$\frac{1}{2} x'(N)^T P x'(N) \leq \underbrace{\lambda_{max}(P)||x||^2}_{\alpha_f(||x'||)} \tag{27}$$

Where $\lambda_{max}(P)$ is the largest eigenvalue of P. Since both functions are quadratic, these results hold globally.

Conclusion

Since assumptions 2.2, 2.3 and 2.14 are shown to hold and $\mathbb{X}_f$ holds the origin in its interior, the conclusion of section 2.5.4 of [2] can be used. By theorem 2.19 and 2.21, the origin is exponentially stable for initial conditions within $\mathcal{X}_N$ for the MPC control design in this paper.

IV. Numerical simulations

In this section, we run several numerical simulations of the MPC designed in II as well as the original controller for the multi link flexible manipulators [1].

A. State feedback control

To set an initial benchmark, the step response for the original state feedback controller designed in [1] is shown in figure 5. Plotted are the two joint angles $s_1 s_2$ aswell as the link deformations $x_1 x_2$. The reference state is $\pi/3$ for the joint angles, and 0 for the velocities and deformations.

The controller reaches its settling time in 25 steps.(the y values stay within the 99% of the reference). This can also be seen in in figure 5 where there seems to be no off-set for x_1 and x_2. The corresponding feedback matrix K for the state feedback can be found in [1].

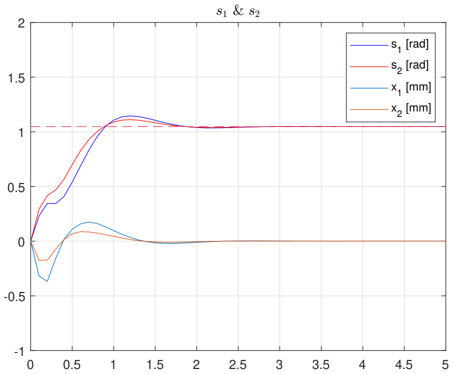


Fig. 5: Output of the dynamical system with a state feedback controller

B. Horizon

Our horizon is set to 11 as to meet the required terminal constrain derived in section III. A shorter horizon will cause the terminal constrain to make the optimisation problem invalid. A larger horizon is unnecessary since a difference in settling time and other values is not obtained for our current setup. Therefore, different horizons are also not plotted.

C. State Measurement

For the tuning of the weighting matrices we will use the state measurement MPC on an ideal dynamical system. The values of the state measurement MPC will be compared with the values of the state feedback controller. Only improved values are shown and x_1 and x_2 will be compared using figures and not exact values.

First the weighting matrices of the stage cost as given in Equation 13 are tuned with a horizon of 3. For the weighting matrices only the diagonal values are tuned and given a non-zero value. This to make tuning by hand possible and using the physical representation of the states as a guideline for the values. Our Q matrix is good with the following values on the diagonal [20 50 500 40 20 50 500 40]. R is left to be a 2x2 zero matrix since we have limited our maximum values for u in the constrains and penalising the control input more drastically lowers the performance on the controller.

Currently the dynamical system reaches has a settling time of 22 steps and has suppressed overshoot more compared to the state feedback controller, as can be seen in Figure 6. Though the settling time of x_1 and x_2 seem to have a small increase.

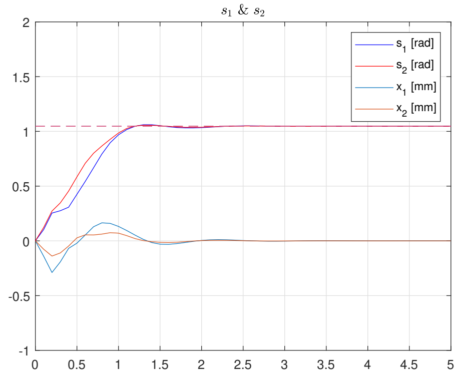


Fig. 6: Output of the dynamical system with a state measurement controller

D. output measurement MPC

The ouput MPC controller described in section II-C has also been implemented. This is done by augmenting the design with the described observer and adding an optimal target selection. The reference is kept the same as previous simulations. To be able to show the effect of measurement noise and the constant disturbance on controller performance first nominal performance is shown. This can be seen in figure 7. It is shown that the controller performs the same as the state measurement controller without any noise or load disturbance. Which is to be expected.

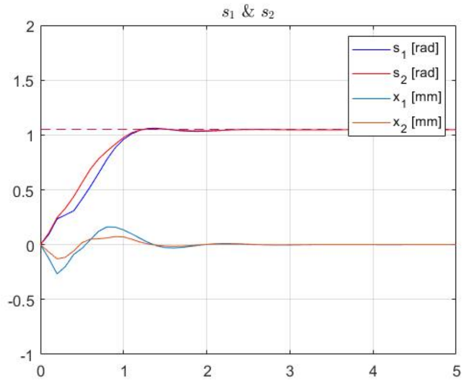
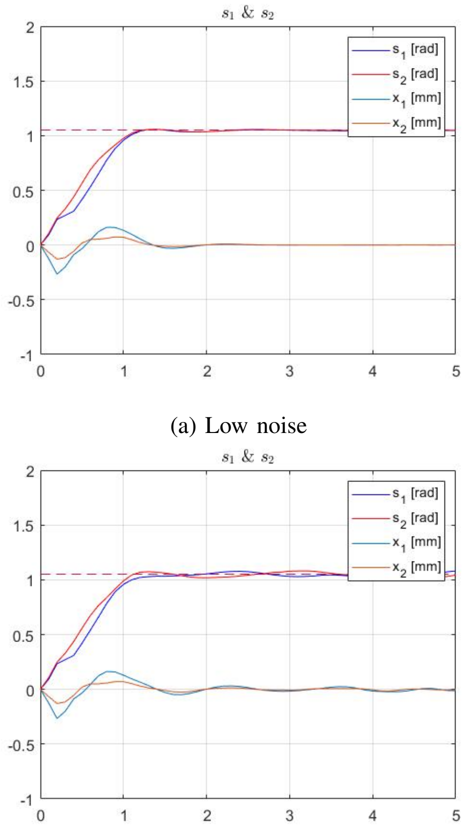


Fig. 7: Nominal performance for the output measurement controller

As main source of noise, output measurement noise is added first. For this, two levels are considered with different Signal to Noise ratio (SNR).

- Low, SNR: 80dB
- High, SNR: 60dB

First, These results are shown in figure 8 for both noise levels. It is clear that the low noise setting introduces some but very little ringing in the output. This ringing becomes more extensive in the high noise setting. The noise however barely effects the initial step response as the settling time is still the same. Since the signal to noise ratio of 60dB is quite high, the controller does a good job in attenuating this disturbance.



(a) Low noise

(b) High noise

Fig. 8: Output feedback controller under different levels of measurement noise.

Next, a constant load disturbance is added to the two angles x_1 and x_2 of :

$$\text{Dist} = \begin{bmatrix} 0.001 \\ 0.001 \end{bmatrix} [m] \tag{28}$$

To be able to correctly see the entire response, the simulation time is extended to 10 seconds. This response is shown in figure 9

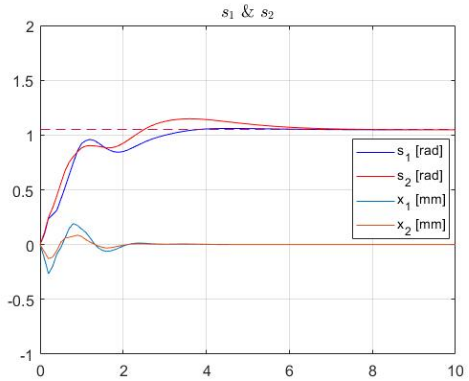


Fig. 9: Response for constant state disturbance.

It can clearly be seen that the controller takes much longer to converge then the state measurement controller or the state measurement MPC. This can be explained by looking at the error between the observed disturbance and the true disturbance states throughout the simulation which is also shown in figure 10.

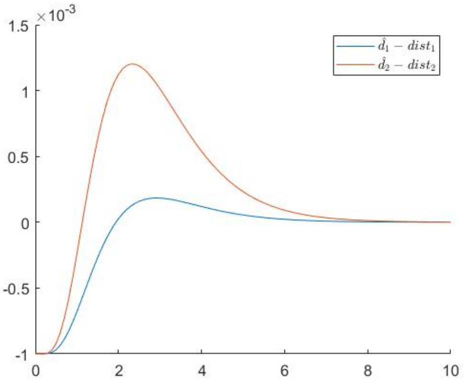


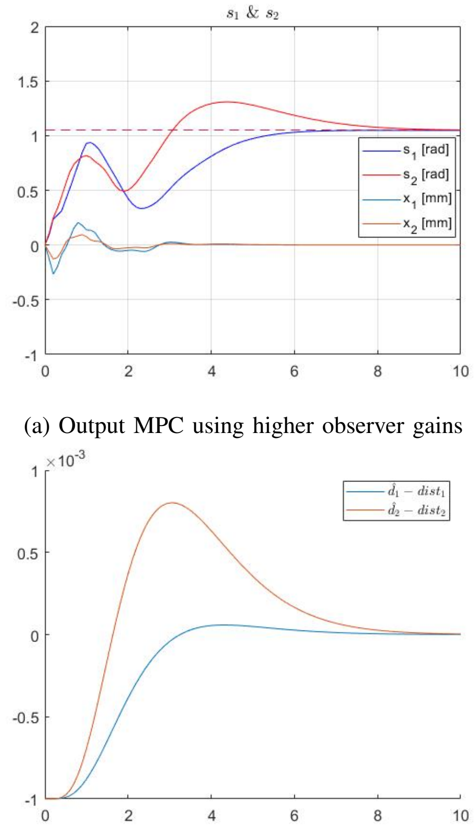
Fig. 10: Observer error for the disturbance states

It can be seen that this error only converges to 0 at around 6 seconds, which can clearly also be seen in figure 9. The long settling time is thus caused by the observer states taking a long time to converge to the true states. Other characteristics of the state error dynamics can also be recognised in the system response such as the large peak in the observer error at around 2.5 seconds. The observer dynamics thus influence the controller performance significantly. Increasing the speed of the observer poles is considered to speed up the disturbance estimation. The result is shown in figure 11.

It can be seen that the overshoot of the estimation of the disturbance is much lower. The time of convergence however does not change significantly, and the system response looks much worse. This is probably due to the higher feedback gains and thus higher susceptibility to measurement noise. Increasing the feedback gains thus does not improve performance. Further tuning is necessary however to find a trade-off between high observer gains and convergence times.

V. Conculsion

In this paper two model predictive controllers are designed for multi link flexible manipulators. A state measurement controller was able to have a better performance then the state feedback controller designed in [1]. It had a settling time of 22 steps which translate to 2.2 seconds. The output measurement model predictive controller made more realistic assumptions regarding possible measurements done on a dynamic system. Though the noise on the output measurement hardly affected the controller a disturbance on the two outputs did increase the settling time. The estimate error as seen in 10 was a big cause. Though making the observer poles faster did not increase the performance. Overall the model predictive controller outperformed the state feedback controller though a small disturbance did cause the model predictive controller to have an increased settling time.



(a) Output MPC using higher observer gains

(b) High noise

Fig. 11: Response for increasing the observer gains

References

[1] D. L. P. Bopearatchy and G. C. Hatanwala, “State space control of a multi link robot manipulator by a translational modelling technique,” *Proceedings. 5th IEEE International Symposium on Intelligent Control 1990*, vol. 1, no. -, pp. 285–290, 1990. [Online]. Available: https://doi.org/10.5772/61312

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