Disturbance Rejection MPC for Tracking of Wheeled Mobile Robot

Group 4 (Anthony Chen *(zc2812)* and Xinhe Yang *(xy2651)*)

(Based on Paper by Z. Sun, Y. Xia et al. , 2017)

April 29, 2025

View Paper on IEEE/ASME Transactions on Mechatronics

ELENE 6907 Model Predictive Control Spring 2025

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Overview of the System

Background Issue:

Disturbances exist in most of the mobile vehicle system, which generally subject to many constraints.

Even though we can track the location of the mobile vehicle system well using method like EKF, the disturbance prevents us from tracking desired trajectory.

Limitation of pure MPC:

Pure MPC considers the worst case of realizations of disturbances and lead to a poor performance.

Tube-based MPC:

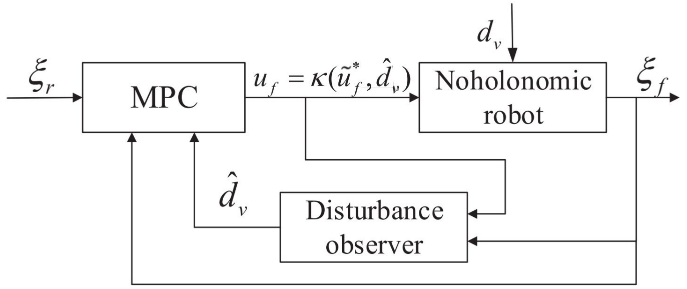
Optimization + Error-Based Feedback Control Law

Previous Limitations:

1.Ignorance of the input constraints

2.No guarantees of feasibility and stability

DRMPC (disturbance rejection MPC)



Key Ideas:

Addition to the MPC, use a observer, which can stabilize disturbance estimation error, to compensate the disturbance.

Overview of the System

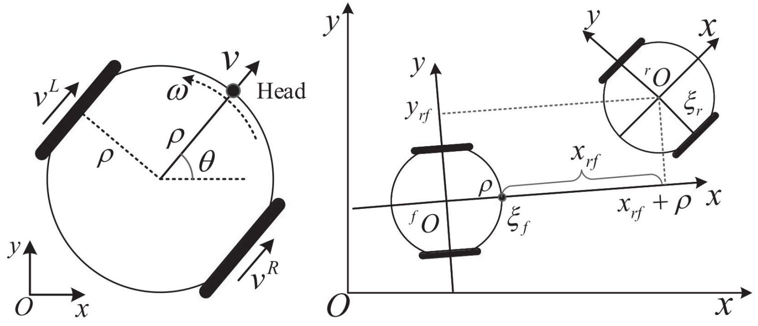
Reference trajectory generated by a typical nonholonomic mobile robot

$$\dot{\xi}_r = f(\xi_r, u_r) = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix}$$

Where $\xi_r = [p_r, \theta_r]^\top \in R^2 \times (-\pi, \pi]$

$p_r = [x_r, y_r]^\top$

Tracking



The follower robot head position kinematics

$$\dot{\xi}_f = f_h(\xi_f, u_f, d_v) = \underbrace{\begin{bmatrix} \cos\theta_r & -\rho\sin\theta_r \\ \sin\theta_r & \rho\cos\theta_r \\ 0 & 1 \end{bmatrix}}_{B_1(\theta_f))} \underbrace{\begin{bmatrix} v_f \\ w_f \end{bmatrix}}_{u_f} + \underbrace{\begin{bmatrix} \cos\theta_r \\ \sin\theta_r \\ 0 \end{bmatrix}}_{B_2(\theta_f))} d_v$$

Where

$\xi_f = [p_f, \theta_f]^\top \in R^2 \times (-\pi, \pi]$

$p_f = [x_f, y_f]^\top$

Tracking Error Dynamics

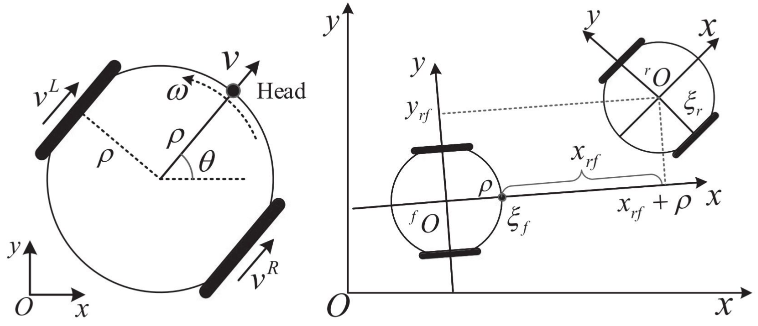
Assumed that the robot can get the relative position from the reference coordinate with respect to its own coordinate that is a moving frame fixed on the robot.

$$p_{rf} = R(-\theta_f)(p_r - p_f)$$

$$\theta_{rf} = \theta_r - \theta_f$$

Where $R(-\theta_f)=\begin{bmatrix} \cos(\theta_f) & \sin(\theta_f) \\ -\sin(\theta_f) & \cos(\theta_f) \end{bmatrix}$

Tracking



The position tracking error dynamics is then given by taking derivatives

$$\dot{\tilde{p}}_{rf} = \begin{bmatrix} \dot{\tilde{x}}_{rf} \\ \dot{\tilde{y}}_{rf} \end{bmatrix} = \begin{bmatrix} 0 & \omega_f \\ -\omega_f & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_{r_f} \\ \tilde{y}_{r_f} \end{bmatrix} + \tilde{u}_{r_f}$$

with the input error

$$\tilde{u}_{rf} = \begin{bmatrix} -v_f + v_r\cos\theta_{rf} \\ -\rho w_f + v_r\sin\theta_{rf} \end{bmatrix}$$

Case 1: DOB Design for Unknown Disturbance

No information (e.g. frequency, amplitude, and phase) of the disturbances is available.

$$\begin{cases} \dot{z}_1 = -\ell_1(\xi_f)\left[B_2(\theta_f)\left(q_1(\xi_f) + z_1\right) + B_1(\theta_f)\,u_f\right] \\ \hat{d}_v = z_1 + q_1(\xi_f) \end{cases}$$

$\hat{d}_v \in \mathbb{R}$ is the estimation of the disturbance

$z_1 \in \mathbb{R}$ is an intermediate variable

$$q_1(\xi_f) = L_1(x_f \cos\theta_f + y_f \sin\theta_f)$$

$\ell_1(\xi_f) = \dfrac{\partial q_1(\xi_f)}{\partial \xi_f^{\top}}$ is the observer gain

Let the estimation error be $e_{d_v} = d_v - \hat{d}_v$

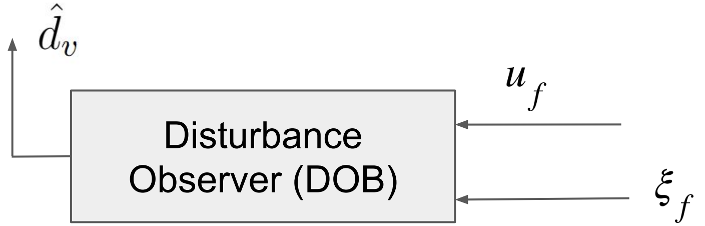
Then, taking the derivative of e_{d_v} yields the estimation error dynamics

$$\dot{e}_{d_v} = \dot{d}_v - \ell_1(\xi_f)B_2(\theta_f)e_{d_v}$$

$$\Rightarrow \quad \dot{e}_{d_v} = \dot{d}_v - L_1 e_{dv}$$

This system is input-to-state stable w.r.t. $\dot{d}_v$

(Proof on Appendix 1)



Case 2: DOB Design for Known Harmonic Frequencies

Disturbances with only known harmonic frequencies is developed.

If we know that the disturbances consist of m harmonics with frequencies c_i, the disturbances can be generated by a neutral stable system:

$$\begin{cases} \dot d(t) = W d(t) \\ d_v(t) = C d(t) \end{cases}$$

$W = diag\{W_1, W_2, ..., W_m\}$ with

$$W_i = \begin{bmatrix} 0 & c_i \\ -c_i & 0 \end{bmatrix} \quad where\ i = 1, 2, \ldots m$$

The DOB form is:

$$\begin{cases} \dot z_2 = (W - \ell_2(\xi_f) B_2(\theta_f) C) z_2 + W q_2(\xi_f) \\ \qquad - \ell_2(\xi_f)\left(B_2(\theta_f) C q_2(\xi_f) + B_1(\theta_f) u_f\right) \\ \hat d = z_2 + q_2(\xi_f) \\ \hat d_v = C d \end{cases}$$

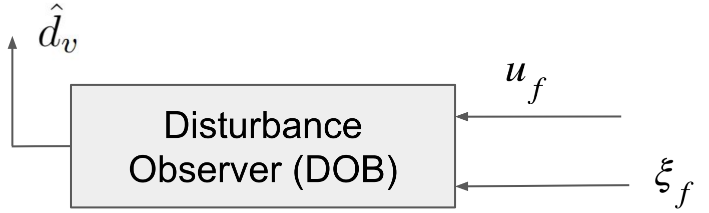
Hurwitz regardless of ξ_f

Also, $\dot e_d = (W - L_2 C) e_d$

$$e_{d_v} = C e_d$$

Then the system is asymptotically stable: $\hat d_v(t) \longrightarrow d_v(t)$

(Proof on Appendix 2)



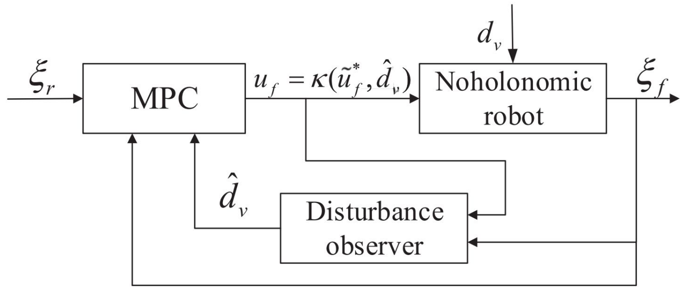
Role of MPC

DRMPC (disturbance rejection MPC)

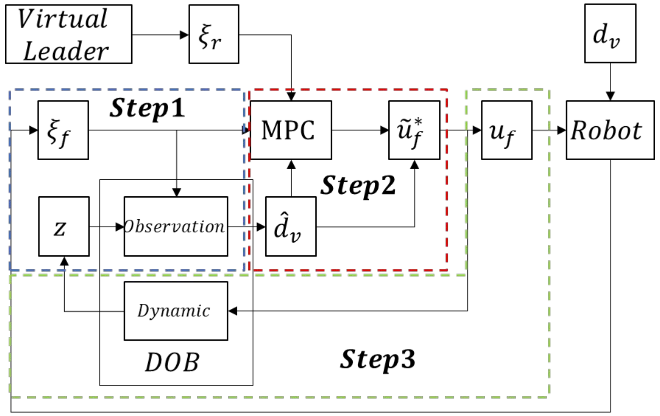
- An MPC with a quadratic tracking error cost is used to roughly trace the virtual leader
- The proposed DOBs provided disturbance compensations to improve the tracking performance

A single loop include 3 steps:

1. Estimate the disturbance through current state and the latent variables of DOBs
2. Use the MPC to solve an optimization problem that minimizes the tracking error cost
3. Execute the compound control input and update the latent variables of DOBs



Decompose



MPC Formulation

Decision Variable (Actually Discrete)

$$\min_{\tilde{\mathbf{U}}_{\mathbf{r}}} J, \tilde{U}_r = \{\tilde{u}_r(0), \tilde{u}_r(\delta), \tilde{u}_r(2\delta), \dots, \tilde{u}_r(T = N\delta)\}$$

Objective Function

$$J = \int_0^T \left(\tilde{p}_{rf}(t)^T P \tilde{p}_{rf}(t) + \tilde{u}_{rf}(t)^T Q \tilde{u}_{rf}(t)\right) dt + \frac{1}{2}\left\|\tilde{p}_{rf}(T)\right\|^2$$

Subject to

IC and Transfer Constraints

$$\tilde{\xi}_r(0) = \xi_{r0}, \tilde{\xi}_f(0) = \xi_{f0}$$

$$\dot{\tilde{\xi}}_r(t) = f(\tilde{\xi}_r(t), \tilde{u}_r(t)), \dot{\tilde{\xi}}_f(t) = f_h(\tilde{\xi}_f(t), \tilde{u}_f(t), 0)$$

$$\equiv \dot{\tilde{p}}_{rf}(t) = f_e(\tilde{p}_{rf}(t), \tilde{u}_{rf}(t))$$

Physical Constraints of Car

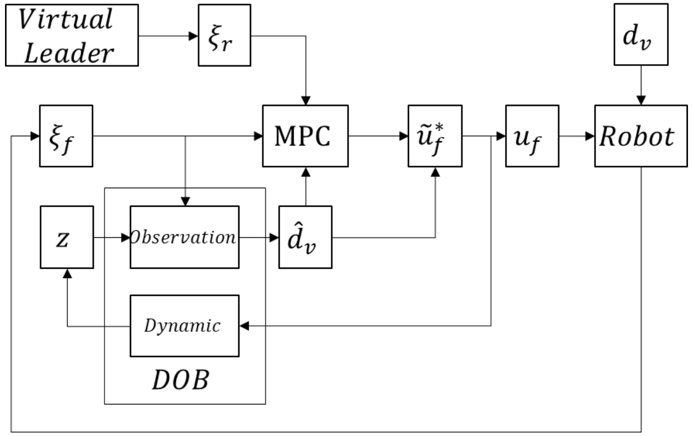
$$\tilde{u}_f(t) - \hat{d}_v(t) \in \mathbb{U}, \mathbb{U} = \left\{[u, \omega]^T, \frac{|u|}{a} + \frac{|\omega|}{b} \le 1\right\}$$

$$\|\tilde{p}_{rf}(t)\| \le \frac{rT}{t}$$

Feasibility Guarantee Constraints

$$\|\tilde{p}_{rf}(T)\| \le \epsilon$$

$$r = \frac{(a-\varrho)(1-\lambda_r)}{\sqrt{k_1^2+k_2^2}}, \lambda_r = \frac{\sqrt{2}}{a}\max|v_r|, \max|d_v| \le \varrho, \epsilon \le r$$



Compound Execution $u_f(0) = \tilde{u}_f^*(0) - [\hat{d}_v(0), 0]^T, t \in [0, \delta]$

Paper proved that the DRMPC is persistent feasible with feasibility guarantee constraints.

And if the estimation error of the disturbance is limited by ϱ, which is guarantee by the stabilities of proposed DOBs, even we actually execute an compound control input, that property still holds.

MPC Formulation

Discretization:
Everything (Robot models, DOBs, MPC) until now is continuous, how can we implement a digital controller？
Simplest Way: 1st order Euler Integration!

Robot models (only use in MPC):

$$\tilde{\xi}_{f/r}[k+1] = \tilde{\xi}_{f/r}[k] + \delta \cdot \dot{\tilde{\xi}}_{f/r}(k \cdot \delta) \Rightarrow \tilde{p}_{rf}[k+1] = \tilde{p}_{rf}[k] + \delta \cdot \dot{\tilde{p}}_{rf}(k \cdot \delta)$$

DOBs (be careful):

$$z[k+1] = z[k] + \delta \cdot \dot{z}(k \cdot \delta)$$

MPC:

$$J = \sum_{k=t}^{N} \left(\tilde{p}_{rf}[k]^T P \tilde{p}_{rf}[k] + \tilde{u}_{rf}[k]^T Q \tilde{u}_{rf}[k]\right) + \frac{1}{2}\left\|\tilde{p}_{rf}[N]\right\|^2$$

$$\tilde{\xi}_r[0] = \xi_{r0}, \tilde{\xi}_f[0] = \xi_{f0}$$

$$\tilde{p}_{rf}[k+1] = \tilde{p}_{rf}[k] + \delta \cdot \dot{\tilde{p}}_{rf}(k \cdot \delta)$$

$$\tilde{u}_f[k] - \hat{d}_v(k) \in \mathbb{U}$$

$$\|\tilde{p}_{rf}[k]\| \leq \frac{rN}{k}$$

$$\|\tilde{p}_{rf}[N]\| \leq \epsilon$$

Some Risks:

Will the DOB explode?
It might! So, we need to be careful about the gain L.

$$\dot{e}_{d_v} = \dot{d}_v - L_1 e_{d_v}, (DOB1)$$
$$\Rightarrow e_{d_v}[k+1] = (1 - \delta \cdot L_1)\, e_{d_v}[k] + \delta \cdot \dot{d}_v \Rightarrow 0 < L_1 < 2/\delta$$

$$\dot{e}_{d_v} = (W - L_2) e_{d_v}, (DOB2)$$
$$\Rightarrow e_{d_v}[k+1] = (I + (W - L_2))\, e_{d_v}[k]$$
$$|eig((I + (W - L_2))| < 1 \Rightarrow -2/\delta < eig(W - L_2) < 0$$

Does the persistent feasibility still holds?
Yes! The the paper actually has proved that.

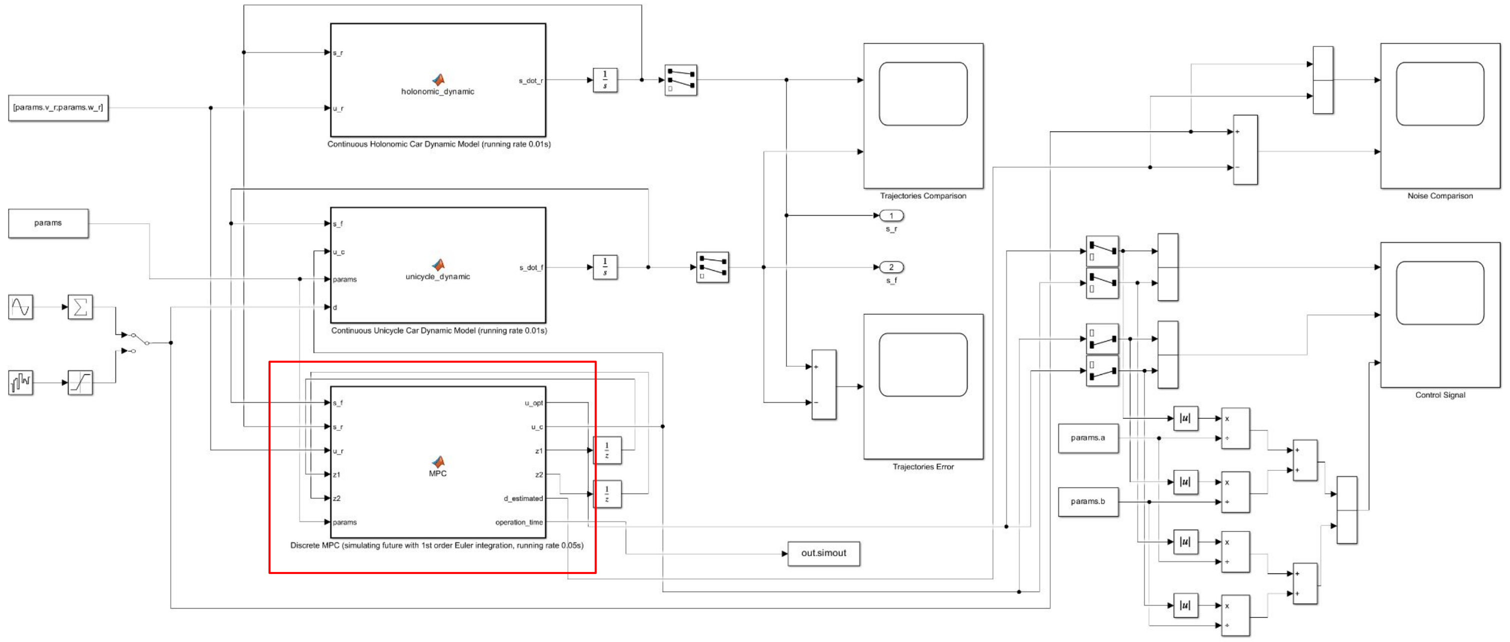
Can we apply this discrete MPC to the continuous model and still get a good performance?
Yes! We will see that later.

Matlab/Simulink Implementation

Continuous Robot Dynamic Model

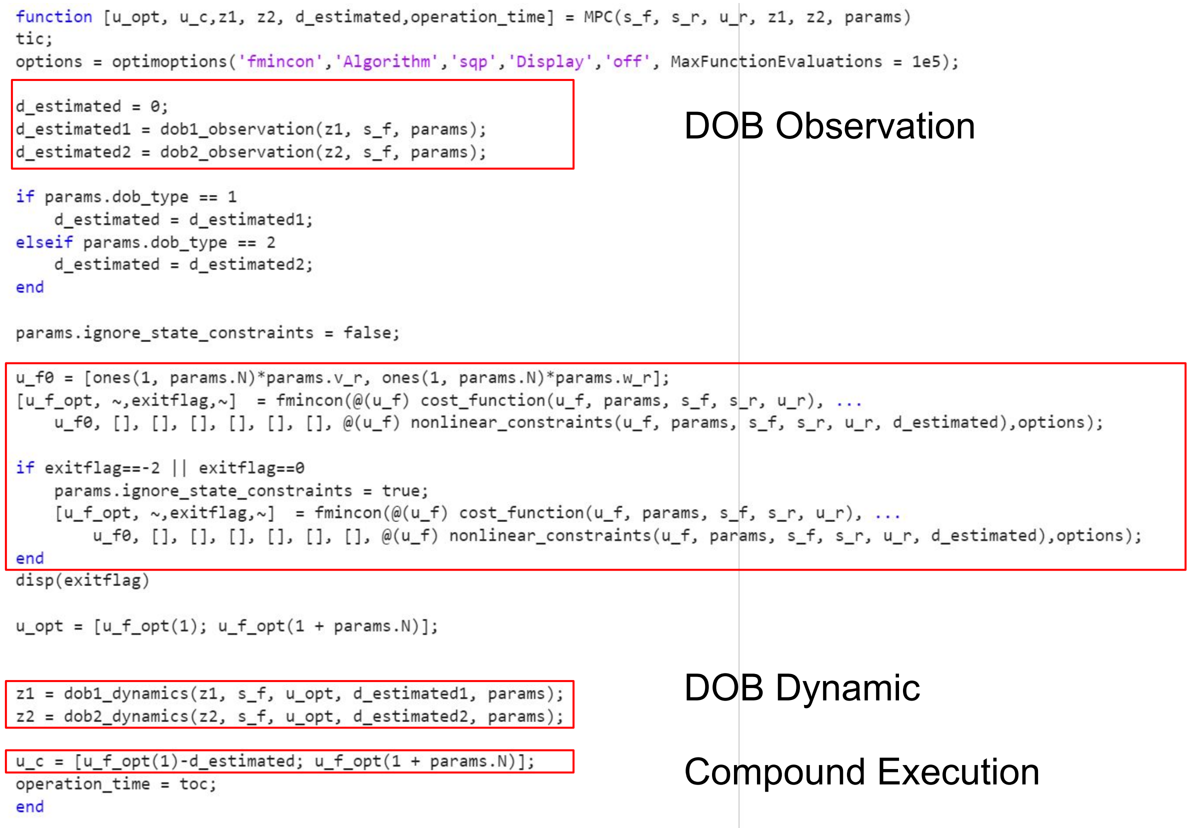
Simulink Step Size=δ=0.05s (synchronous simulation)

Simulink has better approximation of continuous system



Discrete DOB and MPC controller

Matlab/Simulink Implementation



```
function [u_opt, u_c,z1, z2, d_estimated,operation_time] = MPC(s_f, s_r, u_r, z1, z2, params)
tic;
options = optimoptions('fmincon','Algorithm','sqp','Display','off', MaxFunctionEvaluations = 1e5);

d_estimated = 0;
d_estimated1 = dob1_observation(z1, s_f, params);
d_estimated2 = dob2_observation(z2, s_f, params);

if params.dob_type == 1
    d_estimated = d_estimated1;
elseif params.dob_type == 2
    d_estimated = d_estimated2;
end

params.ignore_state_constraints = false;

u_f0 = [ones(1, params.N)*params.v_r, ones(1, params.N)*params.w_r];
[u_f_opt, ~,exitflag,~]  = fmincon(@(u_f) cost_function(u_f, params, s_f, s_r, u_r), ...
    u_f0, [], [], [], [], [], [], @(u_f) nonlinear_constraints(u_f, params, s_f, s_r, u_r, d_estimated),options);

if exitflag==-2 || exitflag==0
    params.ignore_state_constraints = true;
    [u_f_opt, ~,exitflag,~]  = fmincon(@(u_f) cost_function(u_f, params, s_f, s_r, u_r), ...
        u_f0, [], [], [], [], [], [], @(u_f) nonlinear_constraints(u_f, params, s_f, s_r, u_r, d_estimated),options);
end
disp(exitflag)

u_opt = [u_f_opt(1); u_f_opt(1 + params.N)];


z1 = dob1_dynamics(z1, s_f, u_opt, d_estimated1, params);
z2 = dob2_dynamics(z2, s_f, u_opt, d_estimated2, params);

u_c = [u_f_opt(1)-d_estimated; u_f_opt(1 + params.N)];
operation_time = toc;
end
```

DOB Observation

Solving Optimal Control Problem through MATLAB ***fmincon***

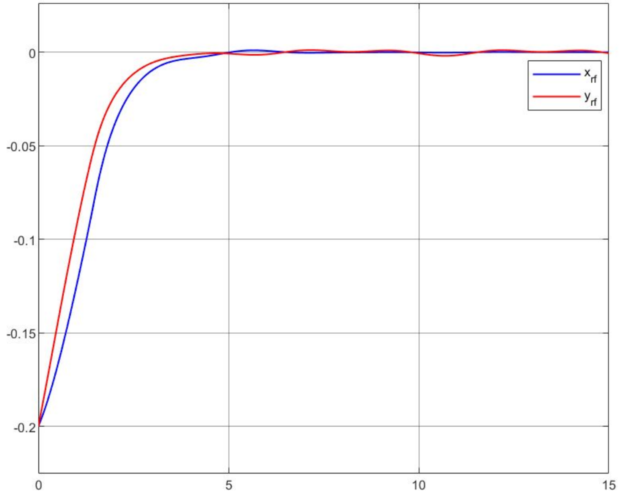
DOB Dynamic

Compound Execution

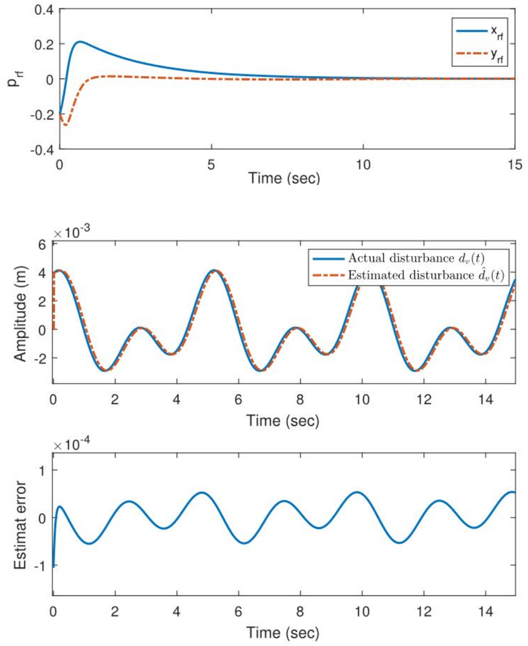
Result Comparison DOB1

Blank comparison:

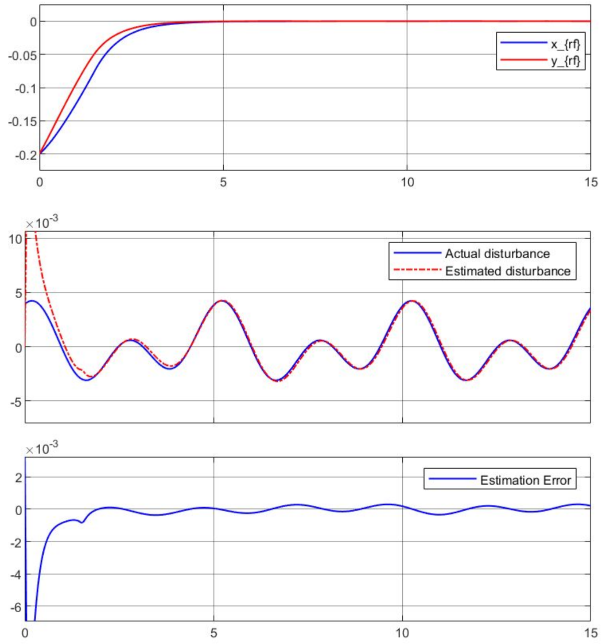
- **No DOB**
- **No Feasibility Guarantee Constraints**



Result Provided by Paper

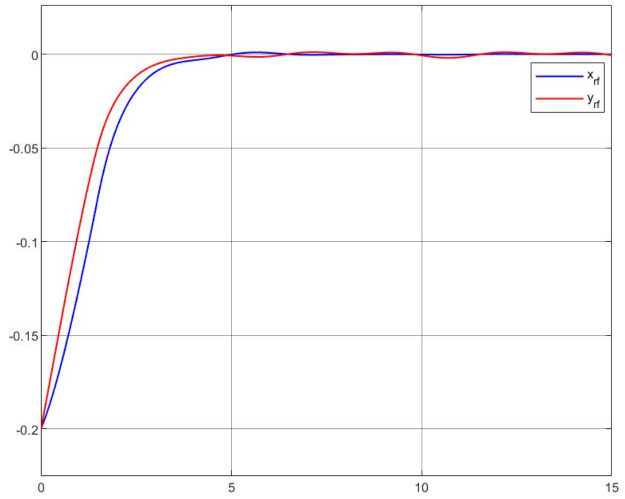


Our Simulation

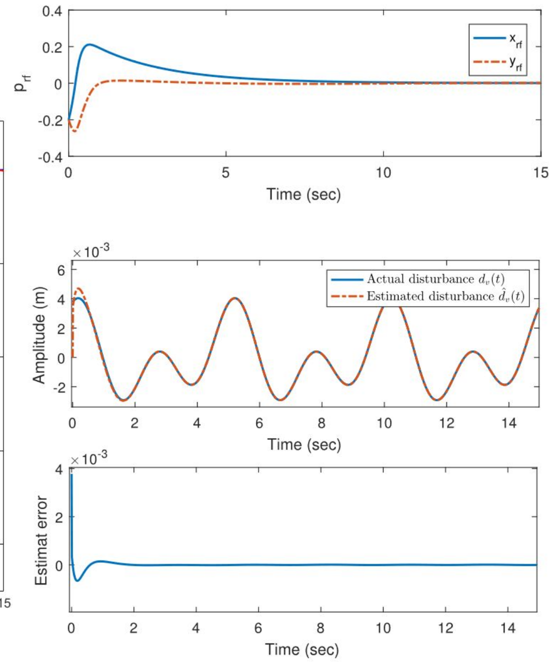


Result Comparison-DOB2

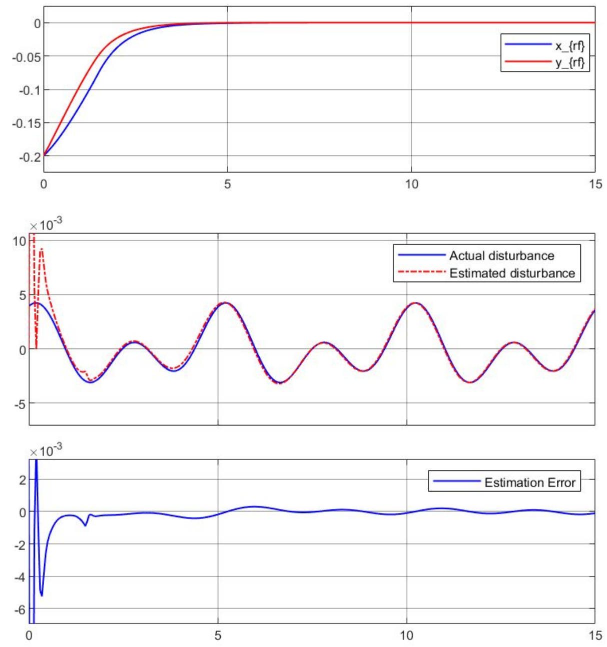
Blank comparison



Result Provided by Paper



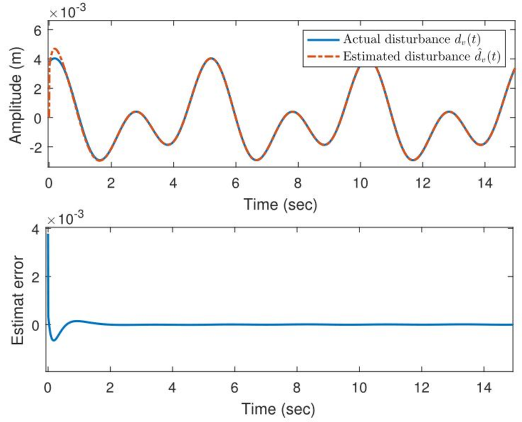
Our Simulation



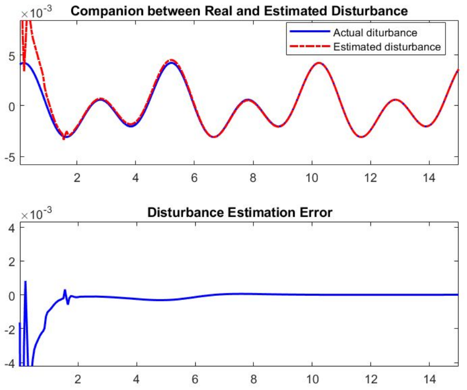
The performance of DOB2 is better than DOB1 but not as good as paper, the estimation error does not decrease to 0.

Result Comparison-Continued

Result Provided by Paper



Our Simulation*



Interesting, if we also let the robot dynamic model **using the discretization method** in MPC, then the performance of DOB2 is similar to the paper.

Why still other different?

- Simulation Configuration
- Discretization Method
- MPC solver

Extension1: Shortened Horizon for Real-time Implementation

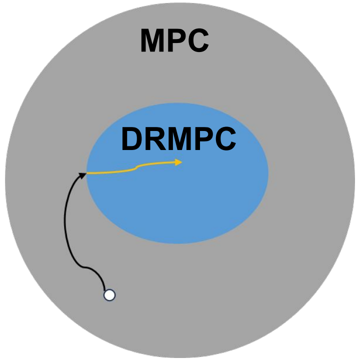
Real-time implementation Issue:
In the original paper $T = 4s, \delta = 0.05s \Rightarrow N = 80$
This resulting in an average MPC solving time $\bar{t}_o \approx 0.34s \gg \delta = 0.05s$, which is meaningless for real-time implementation.

How to fix?
Shortening the Horizon! E.g. $T' = 0.4s \Rightarrow N' = 8$
However, MPC becomes infeasible at the **beginning** since the robot are too far away from the virtual leader, and a short horizon cannot satisfies the Feasibility Guarantee Constraints (they are very strict state constraints, first requires the error decrease linearly, second requires a terminal region).

$$\|\tilde{p}_{rf}(t)\| \leq \frac{rT}{t} \qquad \|\tilde{p}_{rf}(T)\| \leq \epsilon$$

Lucky, since the pure MPC itself can already track the virtual target, we can **deactivate these constraints** until they can be satisfied (car is far away from the target), and then they will be satisfied from persistently.

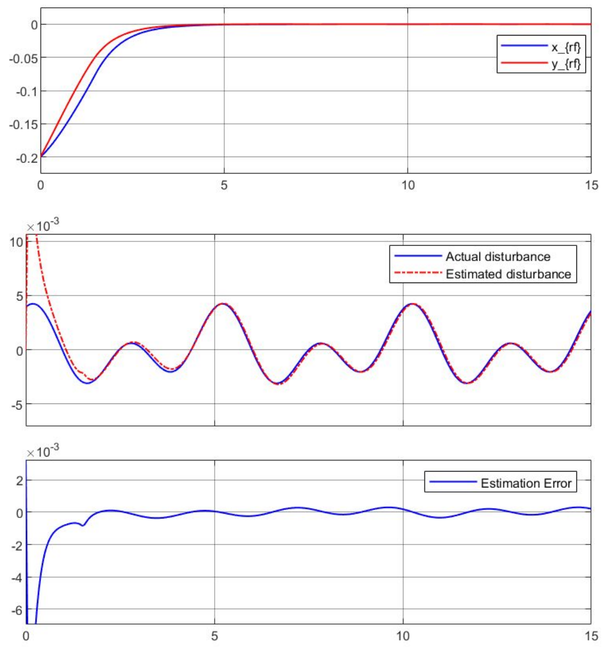
In Summary, we can use pure MPC (we still always estimate the disturbance) until system reach the **Positive Invariant Set of DRMPC** for given horizon.



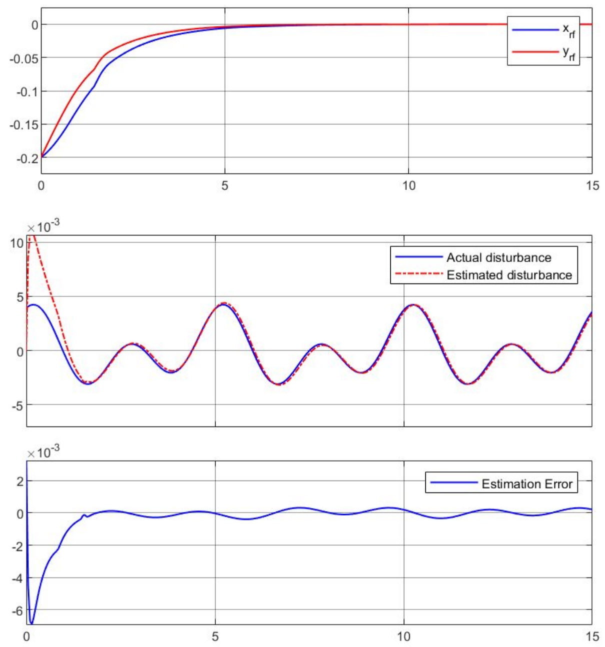
Extension1: Shortened Horizon for Real-time Implementation-DOB1

Works for both DOBs. Slower tracking but Same Steady-State Performance

DOB1 N=80



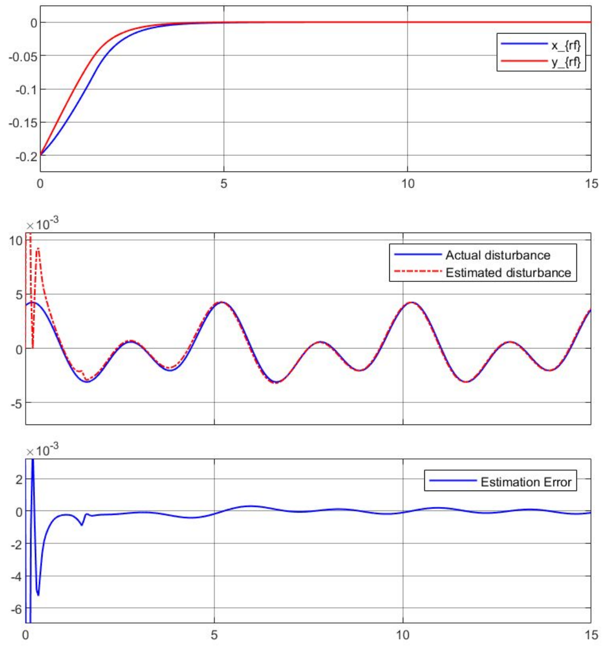
DOB1 N=8



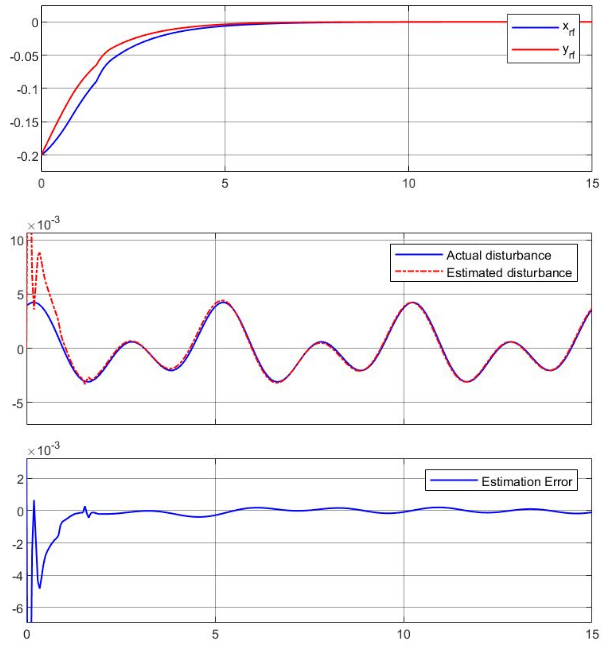
Extension1: Shortened Horizon for Real-time Implementation-DOB2

Significant drop in average MPC solving time $\bar{t}_o^{80} \approx 0.34s \gg \delta = 0.05s \rightarrow \bar{t}_o^{8} \approx 0.0043s \ll \delta = 0.05s$

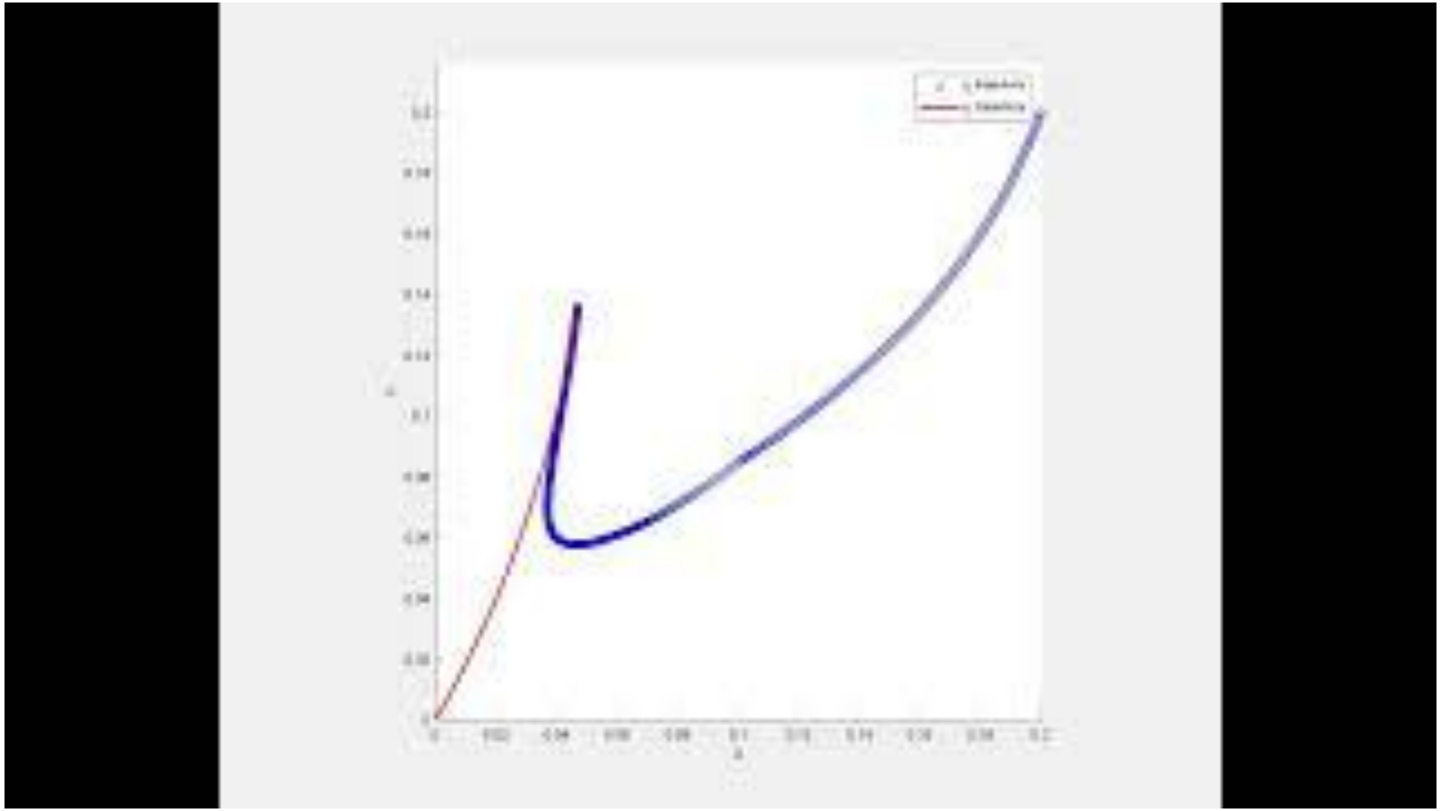
DOB2 N=80



DOB2 N=8



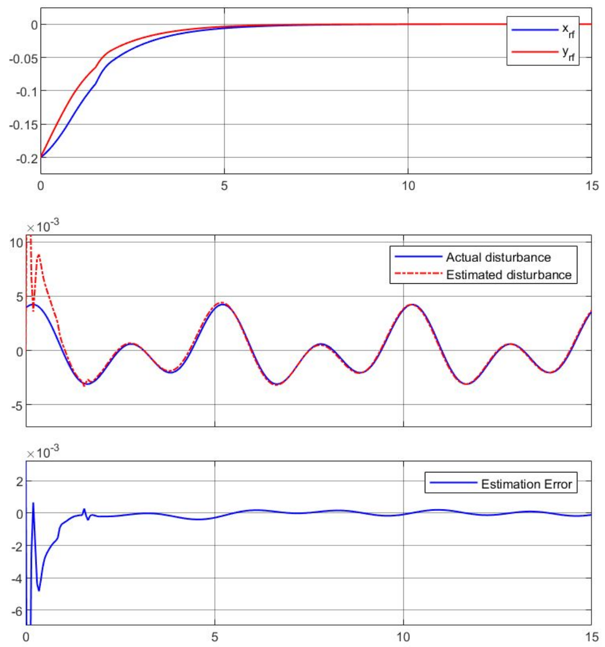
Extension1: Shortened Horizon for Real-time Implementation-Video



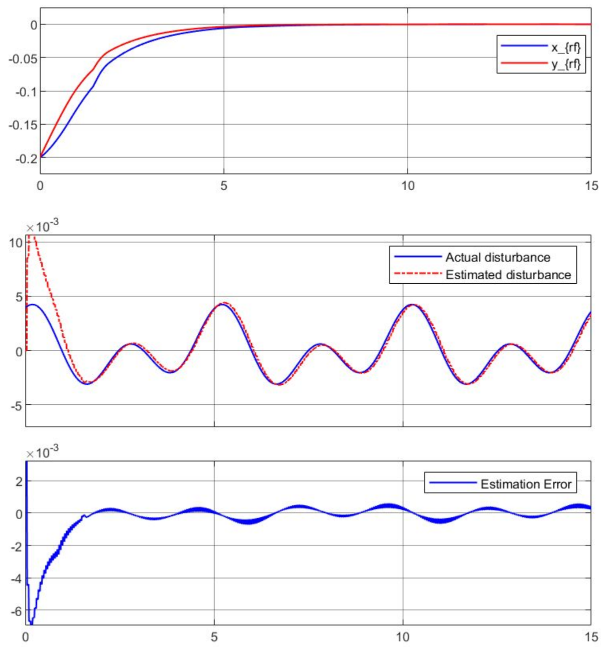
Extension2: Asynchronous Control-DOB1

Furthermore, what if we let the Robot Dynamic Models run at a more faster speed (simulate more close to a real continuous system).→Simulink Step Size=δ/5=0.01s (Asynchronous simulation)

DOB1 N=8 T_sim=0.05s



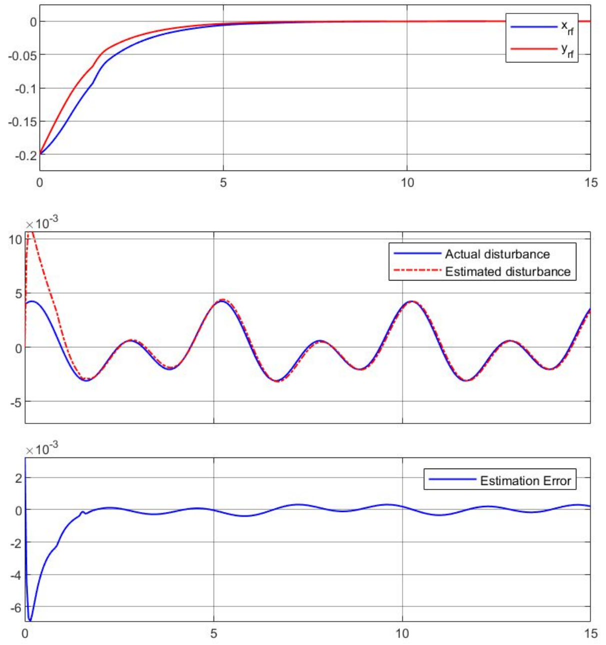
DOB1 N=8 T_sim=0.01s



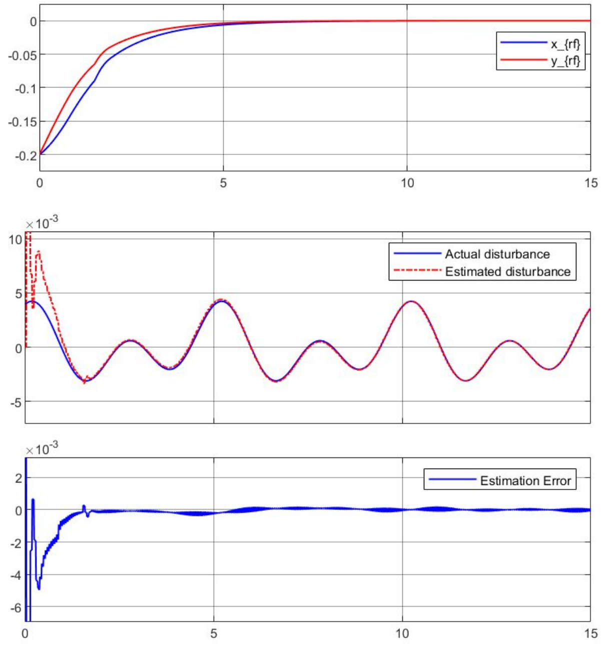
Extension2: Asynchronous Control-DOB2

Disturbance estimation error rises, but still retains good performance.

DOB2 N=8 T_sim=0.05s



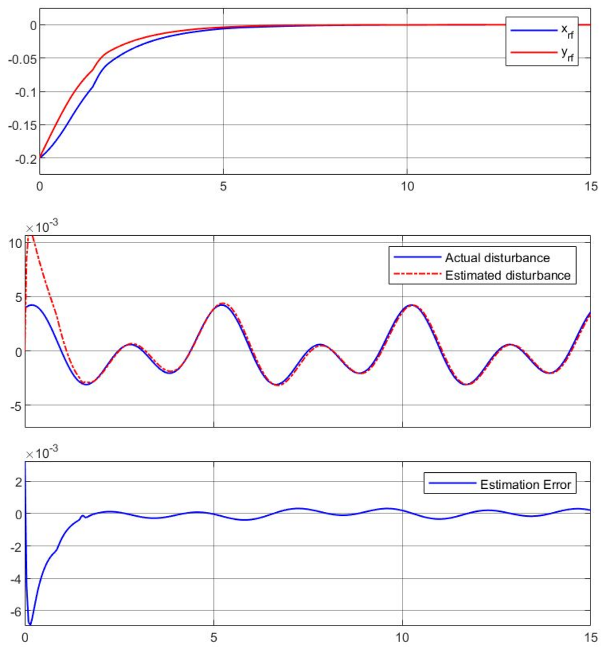
DOB2 N=8 T_sim=0.01s



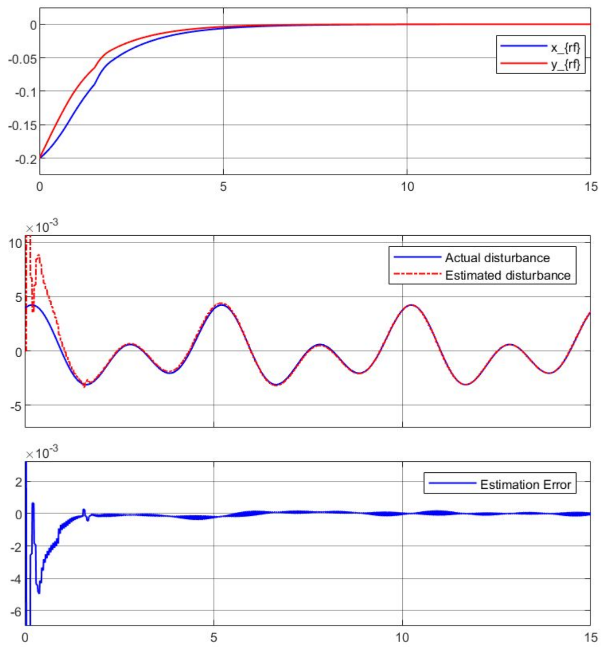
Extension2: Asynchronous Control-DOB2

Disturbance estimation error rises, but still retains good performance.

DOB2 N=8 T_sim=0.05s



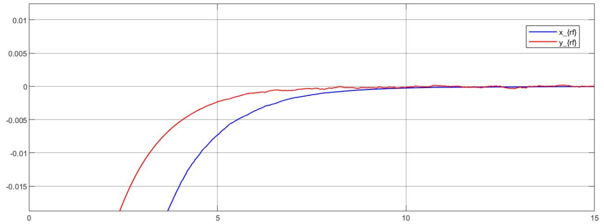
DOB2 N=8 T_sim=0.01s



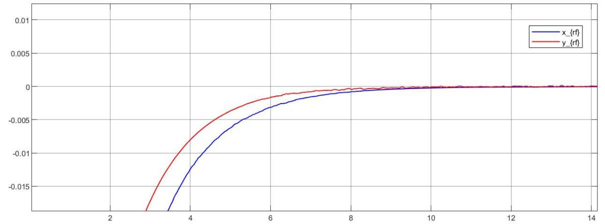
Extension 3: Performance Under Nonharmonic Noise

Finally, paper tested harmonic noise, what about Gaussian noise (limited amplitude)?

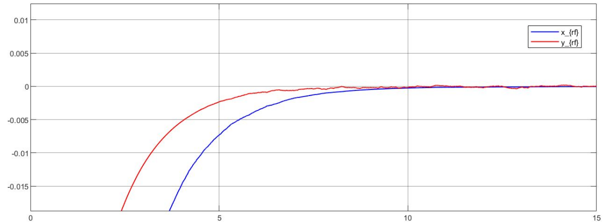
Blank N=8 T_sim=0.05s

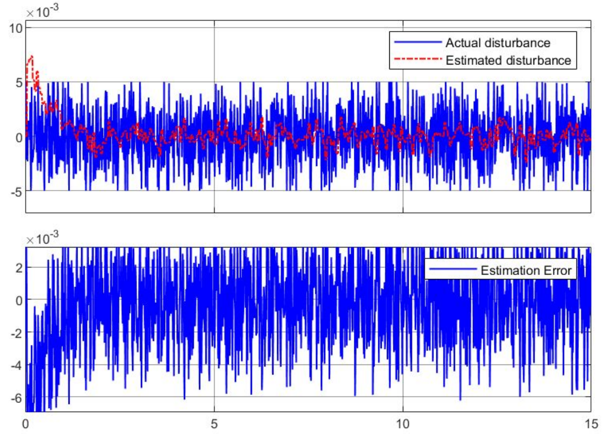


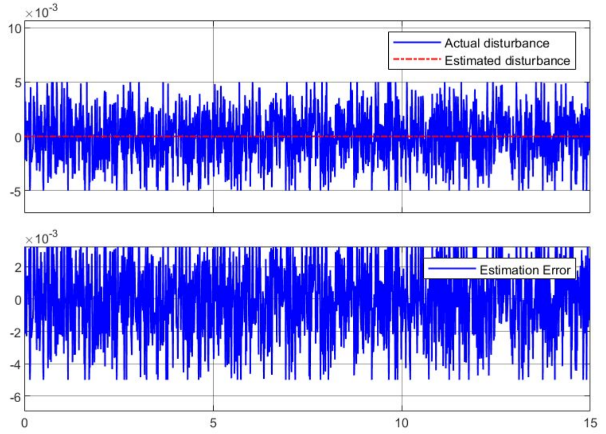
DOB1 N=8 T_sim=0.05s



DOB2 N=8 T_sim=0.05s







- DOB1 managed to do something, although not good
- DOB2 just crashed since the known frequency assumption no longer exists

Conclusion and Key Insights

Challenges:

- Understanding the whole control process of the DRMPC
- Implementing the discrete version of DRMPC
- Exploring the possibilities of DRMPC in real applications

Conclusion:

- The disturbance observer proposed in this article can significantly improve the trajectory tracking performance of mobile robots when the disturbance is harmonic signal
- DRMPC manages to give good performance under discrete controllers and asynchronous control situations

Lessons learned

- The accuracy of the discrete model of the MPC has a significant impact on its performance as a discrete controller when used in continuous environments

Appendix 1

Proposition 1: There always exists a nonlinear function q1(ξf) such that system is ISS with respect to d ̇v .

$$\dot{e}_{d_v} = \dot{d}_v - \dot{z}_1 - \frac{\partial q_1(\xi_f)}{\partial \xi_f^T}\dot{\xi}_f = \dot{d}_v + l_1(\xi_f)\big(B_2(\theta_f)(q_1(\xi_f) + z_1) + B_1(\theta_f)u_f\big) - l_1(\xi_f)\dot{\xi}_f$$

$$\dot{e}_{d_v} = \dot{d}_v + l_1(\xi_f)\left(B_2(\theta_f)(q_1(\xi_f) + z_1) + \left(B_1(\theta_f)u_f - \dot{\xi}_f\right)\right)$$

$$\dot{\xi}_f = B_1(\theta_f)u_f + B_2(\theta_f)d_v, q_1(\xi_f) + z_1 = \hat{d}_v$$

$$\Rightarrow \dot{e}_{d_v} = \dot{d}_v + l_1(\xi_f)\big(B_2(\theta_f)\hat{d}_v - B_2(\theta_f)d_v\big) = \dot{d}_v - l_1(\xi_f)B_2(\theta_f)e_{d_v}$$

$$-l_1(\xi_f)B_2(\theta_f) = -L_1\begin{bmatrix}\cos\theta_f & \sin\theta_f & y_f\cos\theta_f - x_f\sin\theta_f\end{bmatrix}\begin{bmatrix}\cos\theta_f \\ \sin\theta_f \\ 0\end{bmatrix} = -L_1 < 0$$

Appendix 2

Proposition 1: There always exists a nonlinear function q2(ξf) such that system is ISS with respect to d ̇v .

Define $q_2(\xi_f)$ as $\begin{bmatrix} l_1(\ x_f cos\theta_f + y_f sin\theta_f) \\ l_2(\ x_f cos\theta_f + y_f sin\theta_f) \\ \dots \\ l_m(\ x_f cos\theta_f + y_f sin\theta_f) \end{bmatrix}$

Then $\ell_2(\xi_f) = \begin{bmatrix} l_1 cos\theta_f & l_1 sin\theta_f & l_1(y_f cos\theta_f - x_f sin\theta_f \\ l_2 cos\theta_f & l_2 sin\theta_f & l_2(y_f cos\theta_f - x_f sin\theta_f \\ \dots & \dots & \dots \\ l_m cos\theta_f & l_m sin\theta_f & l_m(y_f cos\theta_f - x_f sin\theta_f \end{bmatrix}$

Then $-\ell_2(\xi_f)B_2(\theta_f) = -L_2 = -[l_1\ l_2 \dots l_m]^\top$

Assume the output matrix is:

$C = [c_{1,1}, c_{1,2}, c_{2,1}, c_{2,2} \dots, c_{m,1}, c_{m,2}],$

Where $[c_{i,1}, c_{i,2}]$ corresponds to the states of the ith subsystem. For each subsystem (W_i, C_i), the observability matrix is:

$$O_i = \begin{bmatrix} C_i \\ C_i W_i \end{bmatrix} = \begin{bmatrix} c_{i,1} & c_{i,2} \\ -c_i c_{i,2} & c_i c_{i,1} \end{bmatrix}$$

Since $c_i \neq 0$, and $[c_{i,1}, c_{i,2}] \neq [0,0]$

$$\det(O_i) = c_i\,(c_{i,1}^2 + c_{i,2}^2) \neq 0$$

So, the subsystem is observable and detectable. For all subsystems are observable, the entire system is observable and detectable.

Therefore, we can design L_2 such that $A - L_2 C$ is Hurwitz and using pole placement to stabilize the system.