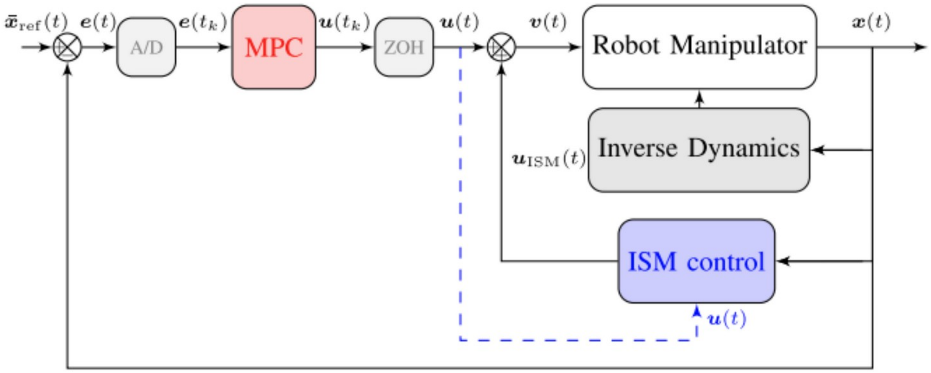
MPC for Robot Manipulators With Integral Sliding Modes Generation

Incremona, Ferrara, Magni; IEEE/ASME Transactions on Mechatronics

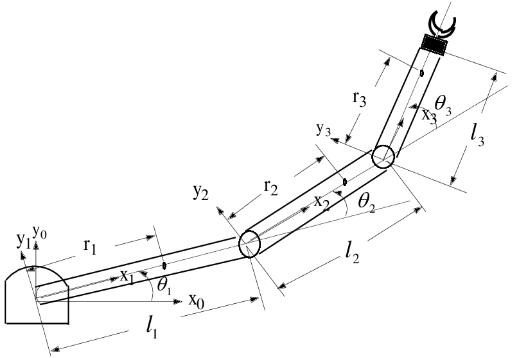
Swapneel Bhatt & Onur Calisir
Team 1

Problem Statement

- MPC struggles with robot systems due to:
  - Nonlinear MIMO dynamics
  - Modelling uncertainties
  - High computational cost
- Proposed Solution:
  - Hierarchical control: Inverse Dynamics + ISM + MPC
  - ISM handles uncertainties first
  - Leads to a simplified MPC optimization
- Key Novelty:
  - ISM enforces sliding modes from initial time, creating uncertainty-free dynamics
  - Multi-rate control (fast ISM, slower MPC)
  - Significantly reduced computational complexity



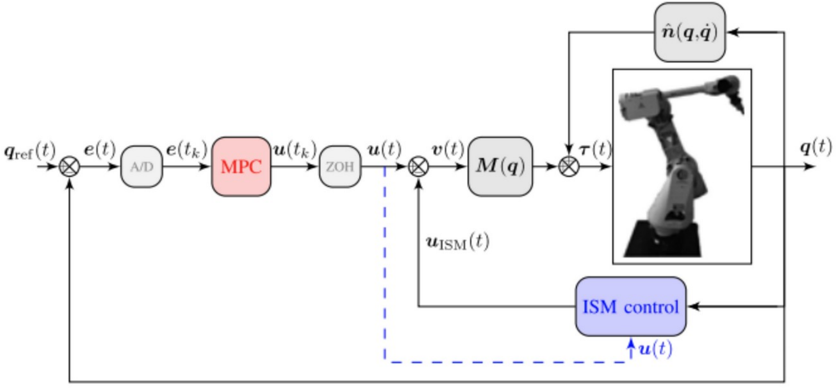
$$M(q)\ddot{q} + h(\dot{q}, q) + g(q) = F(t)$$



Controller Overview

Three Layer Control Architecture

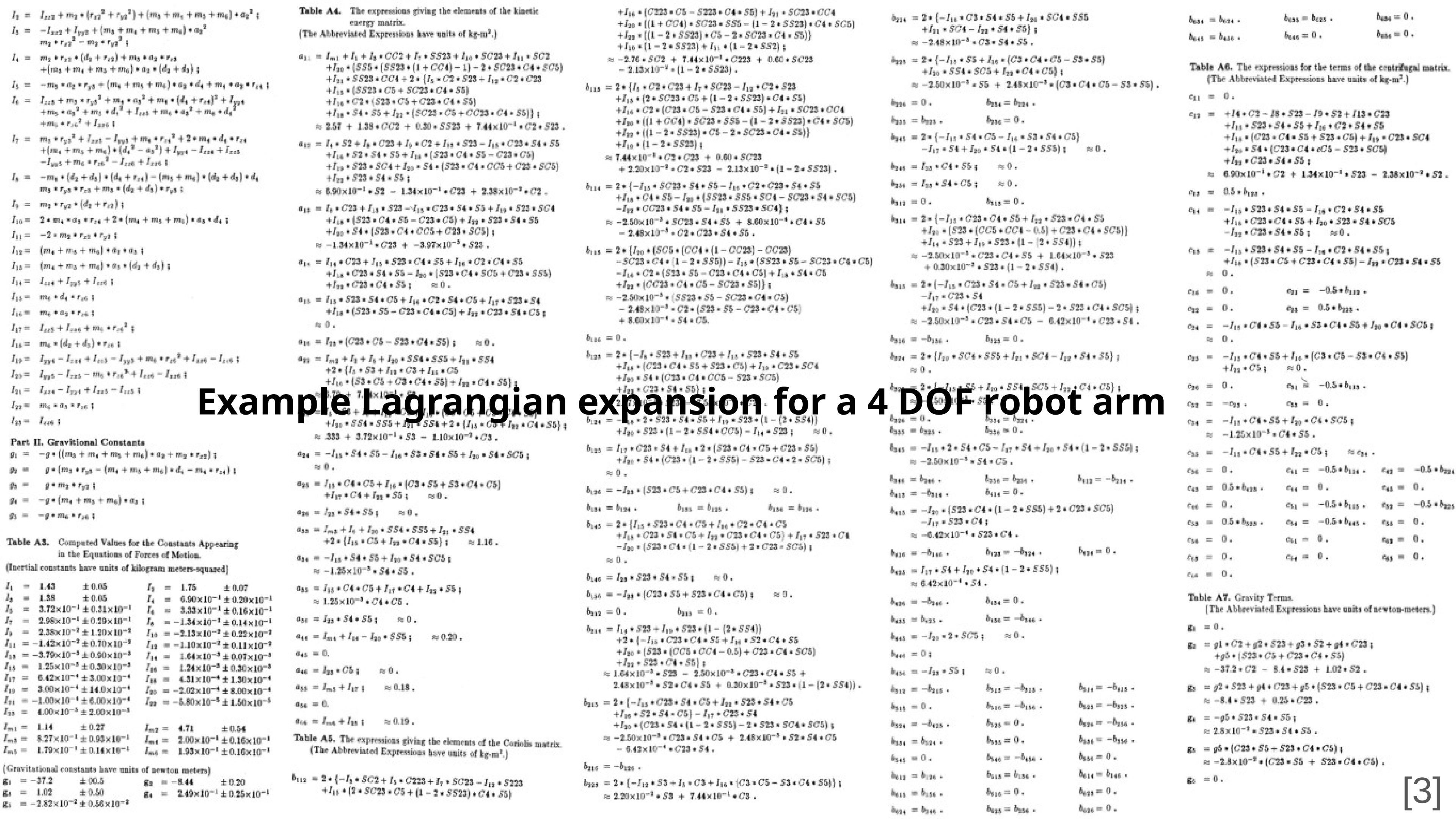
1) Inner Loop: Inverse Dynamics Controller
   a) Linearizes nonlinear MIMO system
   b) Transforms to n decoupled double integrators
2) Middle Loop: Integral Sliding Mode (ISM)
   a) Rejects matched uncertainties
   b) Fast sampling rate (high frequency)
3) Outer Loop: MPC
   a) Optimal trajectory tracking
   b) Handles constraints
4) Combine Control Law:
   a) v = u_MPC + u_ISM



The full coupled robot model:

$$\underbrace{M(q)}_{\text{inertia matrix}} \ddot{q} + \underbrace{n(q,\dot{q})}_{\substack{\text{Coriolis/centripetal} \\ \text{gravity + friction}}} = \tau$$

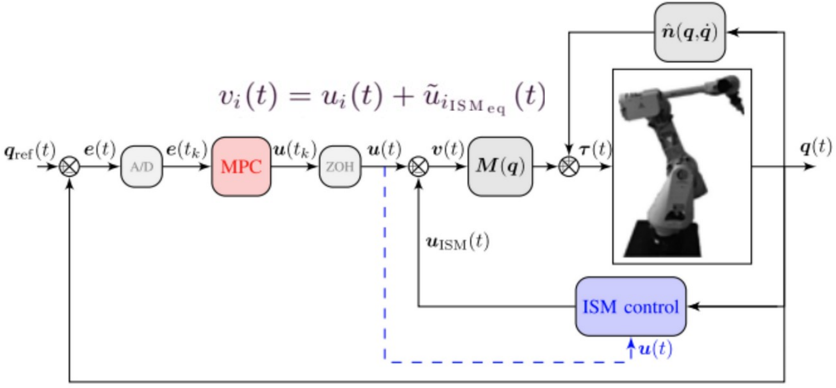
$$e = q_{\text{ref}} - q$$



Example Lagrangian expansion for a 4 DOF robot arm

[3]

Controller Formulation - ISM



<u>Three Layer Control Architecture</u>

1) Inner Loop: Inverse Dynamics Controller

$$\ddot{q} = v - \eta(q, \dot{q}) \qquad \eta(q, \dot{q}) = -M^{-1}(q)(\hat{n}(q, \dot{q}) - n(q, \dot{q}))\ .$$

N-decoupled

Double Integrators

$$\begin{cases} \dot{x}_{1_i}(t) = x_{2_i}(t) \\ \dot{x}_{2_i}(t) = v_i(t) - \eta_i(t) \end{cases}$$

Sliding Mode Control

$$\sigma_i = S_i \left[x_i(t) - x_i(t_0) - \int_{t_0}^{t} \begin{bmatrix} x_{2i}(\zeta) \\ u_i(\zeta) \end{bmatrix} d\zeta \right], \qquad S_i = [c_i \ \ 1]$$

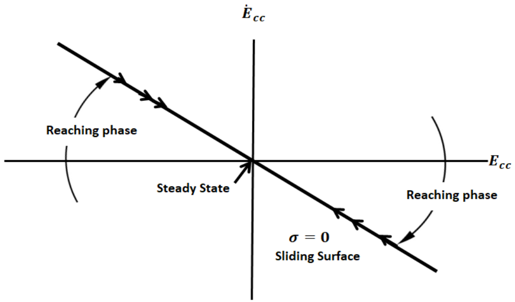
$$u_i^{\mathrm{ISM}} = -U_i^{\max} \operatorname{sgn}(\sigma_i)$$

2) Middle Loop: Integral Sliding Mode (ISM)

$$\tilde{u}_{i_{\mathrm{ISM}_{\mathrm{eq}}}}(t) = \frac{1}{\mu_i} \int_{t_0}^{t} \mathrm{e}^{-\frac{1}{\mu_i}(t-\zeta)} u_{i_{\mathrm{ISM}}}(\zeta) d\zeta \qquad \begin{cases} \dot{x}_{1_i}(t) = x_{2_i}(t) \\ \dot{x}_{2_i}(t) = u_i(t) \end{cases}$$

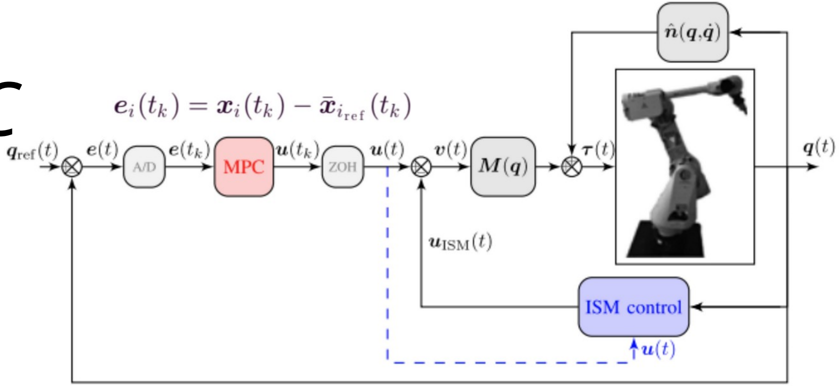
$$\sigma_i(x_i(t)) = $$

$$S_i \left(x_i(t) - x_i(t_0) - \int_{t_0}^{t} [x_{2_i}(\zeta),\ v_i(\zeta) - u_{i_{\mathrm{ISM}}}(\zeta)]^T d\zeta \right) \qquad \dot{x}_i(t) = A_i x_i(t) + B_i u_i(t)$$



Sliding surface and sliding variable. [2]

Controller Formulation - MPC



Outer Loop: MPC

$$\ddot{q} = v + M^{-1}(q)(\hat{n}(q,\dot{q}) - n(q,\dot{q})) = v - \eta(q,\dot{q}))$$

Recap:

$$v_i(t) = u_i(t) + \tilde{u}_{i_{ISM_{eq}}}(t)$$

$$x_{k+1} = \tilde{A}\,x_k + \tilde{B}\,u_k, \qquad \tilde{A} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \ \tilde{B} = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$$

Cost Function:

$$J(e_i(t_k), \mathbf{u}_{i_{[t_k, t_{k+N-1}|t_k]}}, N) =$$

$$\sum_{j=0}^{N-1} \|e_i(t_{k+j})\|^2_{Q_i} + \|u_i(t_{k+j})\|^2_{R_i} + \|e_i(t_{k+N})\|^2_{\Pi_i}$$

Subject to:

$$x_i(t_{k+j}) \in \mathcal{X}$$

$$x_i(t_{k+N}) \in \mathcal{X}_f$$

$$\|u_i(t_{k+j})\| \leq v_{i_{max}} - U_{i_{max}}$$

Terminal Set and Terminal Control:

$$\mathcal{X}_f := \{x_i \mid \|x - \bar{x}_{i_{ref}}\|^2_{\Pi} \leq \rho\}, \quad \mathcal{X}_f \subseteq \mathcal{X}$$

K_{LQ} is the control gain of an infinite horizon Linear Quadratic (LQ) controller with the same cost function

$$\kappa_{i_f}(e_i(t_k)) = K_{LQ}\, e_i(t_k)$$

$$x_i(t_k) \in \mathcal{X}_f$$

$$\|\kappa_{i_f}(e_i(t_k))\| \leq v_{i_{max}} - U_{i_{max}}$$

Solution to the Riccati Equation

$$(\widetilde{A}_i - \widetilde{B}_i K_{LQ})^T \Pi_i (\widetilde{A}_i - \widetilde{B}_i K_{LQ}) - \Pi_i = -Q_i - K_{LQ}^T R_i K_{LQ}$$

Paper Results

- 3-joint COMAU Smart3-S2 industrial robot manipulator
- Realistic uncertainty injection (identified from real robot data)
- Constraints: See Table I (position, velocity, acceleration limits)
- Target position: q_ref = [π/4, π/3, 2π/4]ᵀ, from q_0 = [0, 0, 0]
- Controller parameters:
    - Ci = [10, 10, 10]
    - ISM gains: [20, 35, 85]
    - MPC: Q = diag(100,100), R = 0.1,
    - Terminal Weight
    - Horizon N=10
    - Sampling: MPC = 20ms, ISM = 1ms

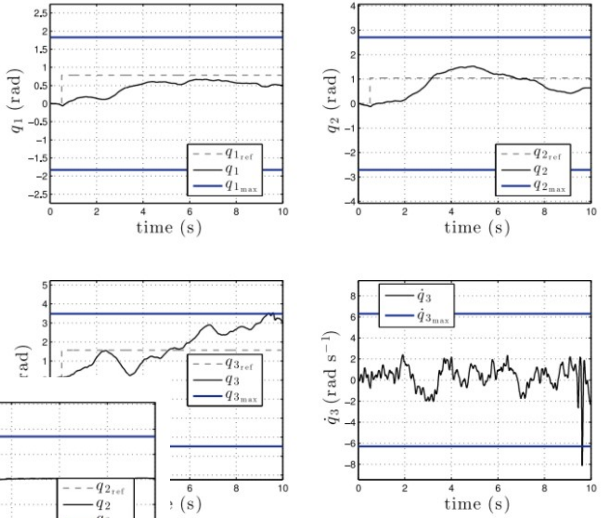
$$\Pi_i = \begin{bmatrix} 5213.4 & 165.8 \\ 165.8 & 221.3 \end{bmatrix}$$

Key Results:

- MPC/ISM reduces RMS error dramatically:
    - Joint 1: 0.2105 → 0.0070 rad (30× decrease)
    - Joint 3: 1.0458 → 0.0152 rad (69× decrease)
- Similar control effort but better performance
- Computational efficiency:
    - MPC: 18ms average execution time
    - ISM: only 29μs average execution time
    - Suitable for real-time implementation
- MPC alone violates constraints,
- MPC/ISM ensures constraint satisfaction

TABLE I
STATE AND INPUT CONSTRAINTS FOR EACH JOINT

Joint i	$q_{i\,max}$ (rad)	$\dot{q}_{i\,max}$ (rad s^{-1})	$v_{i\,max}$ (rad/s^2)
1	1.83	2	145
2	2.71	3.5	250
3	3.49	6.3	350



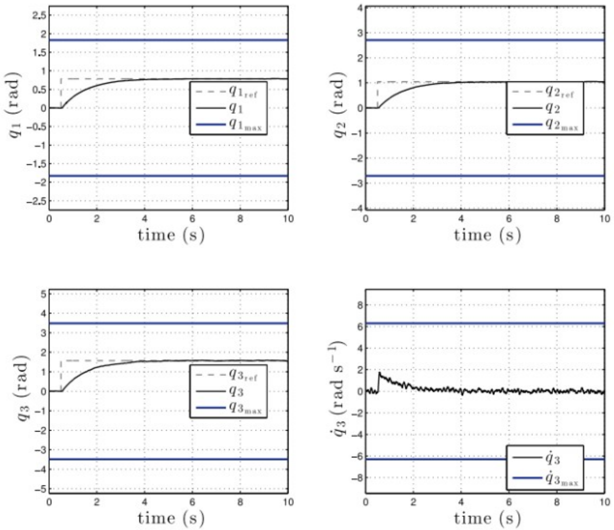


TABLE II
PERFORMANCE INDEXES FOR EACH JOINT

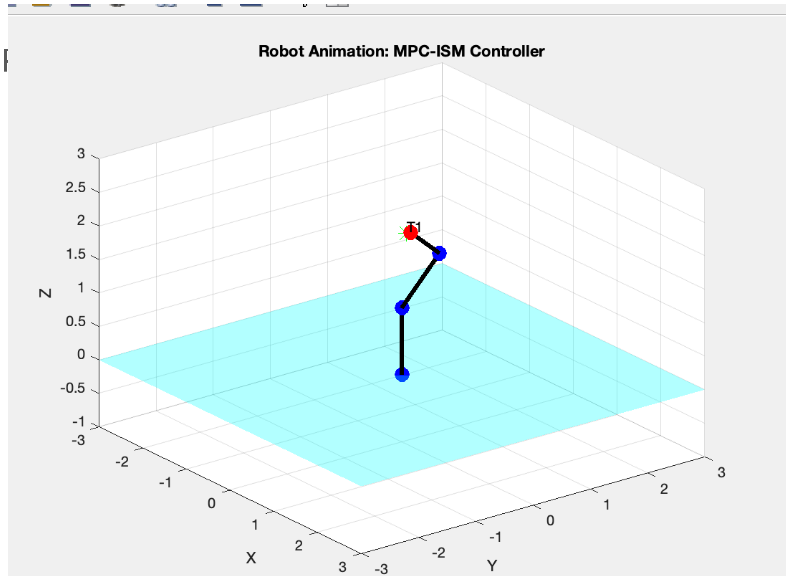
Strategy	Joint i	e_{RMS_i} (rad)	E_{c_i} (rad/s^2)
MPC	1	0.2105	8.0303
	2	0.3749	13.4839
	3	1.0458	41.2172
MPC/ISM	1	0.0070	8.0600
	2	0.0102	13.4217
	3	0.0152	34.8530

TABLE III
TIME CONSUMPTION OF THE PROPOSED CONTROL STRATEGY IN SECONDS

Algo.	Mean	Min.	Max.	Std. dev.
MPC	0.018	0.017	0.71	0.011
ISM	2.9×10^{-5}	2.7×10^{-5}	0.0086	1.4×10^{-4}

Implementation on Matlab

1) Robot Joint Angle target : pi/2, pi/3, pi/4
2) Joint trajectory from MPC and MPC-ISM controllers compared to reference cubic polynomial trajectory
3) Sinusoidal noise injected to see the benefits of the MPC-ISM Combined Controller Algorithm



Uncertainty Formulation and Injection

1. Uncertainty formulation:

```
% Deterministic component based on position
pos_comp = 0.5 * max_eta * sin(q(i));

% Deterministic component based on velocity
vel_comp = 0.2 * max_eta * sign(q_dot(i)) * min(abs(q_dot(i)), 1);

% Random component (bounded)
rand_comp = 0.3 * max_eta * (2*rand() - 1);

% Combined uncertainty
eta(i) = pos_comp + vel_comp + rand_comp;
```

2. Uncertainty injection:

   Where M is the inertia matrix, n contains the nonlinear terms (friction, coriolis effect), and η is the uncertainty

$$\ddot{q} = M^{-1}(u - n + \eta)$$

```
% Compute acceleration: q_ddot = M⁻¹(τ - n + η)
q_ddot = M \ (u - n + eta);
```

3. This is done for all 3 joints.

Results

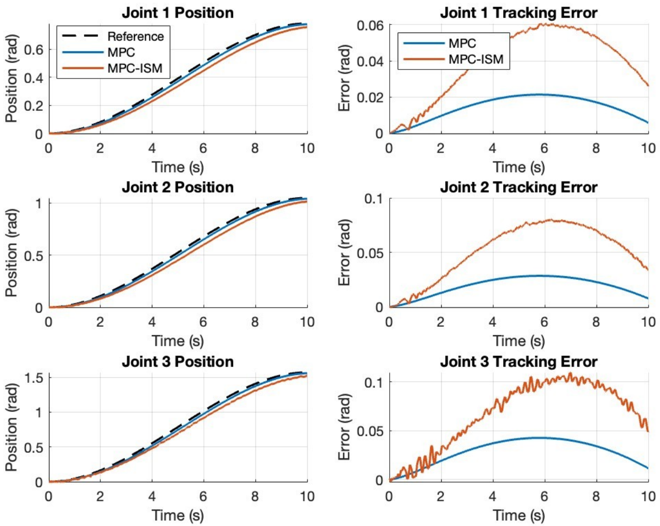
1) The results were not what was expected based on the paper.
2) MPC Controller was very well tuned to account for the noise and gave highly accurate results.
3) MPC-ISM performed slightly worse than MPC.
4) However, the difference seems negligible - both controllers have an error within hundredths of a radian.

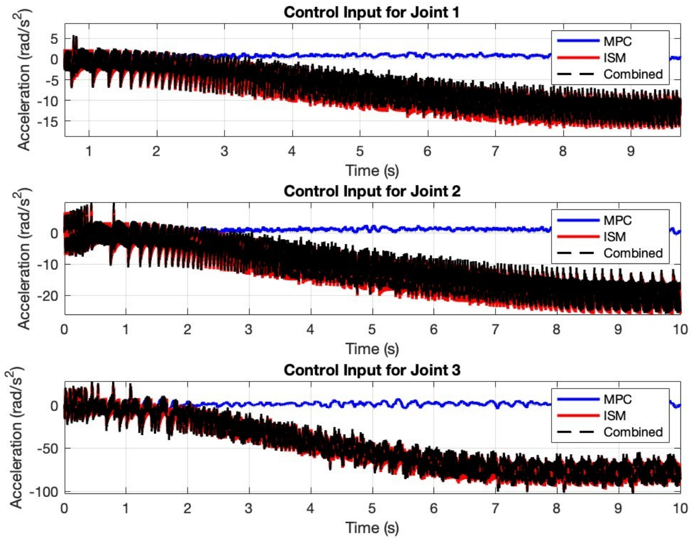
```
Controller: MPC
RMS Tracking Error (rad):
  Joint 1: 0.0156
  Joint 2: 0.0207
  Joint 3: 0.0311
Control Effort (rad/s^2):
  Joint 1: 0.0259
  Joint 2: 0.0345
  Joint 3: 0.0518
Steady-state Error (rad):
  Joint 1: 0.005942
  Joint 2: 0.007923
  Joint 3: 0.011884
```

```
Controller: MPC-ISM
RMS Tracking Error (rad):
  Joint 1: 0.0434
  Joint 2: 0.0571
  Joint 3: 0.0748
Control Effort (rad/s^2):
  Joint 1: 8.4462
  Joint 2: 13.5679
  Joint 3: 57.2287
Steady-state Error (rad):
  Joint 1: 0.026736
  Joint 2: 0.034350
  Joint 3: 0.050732
```

Results

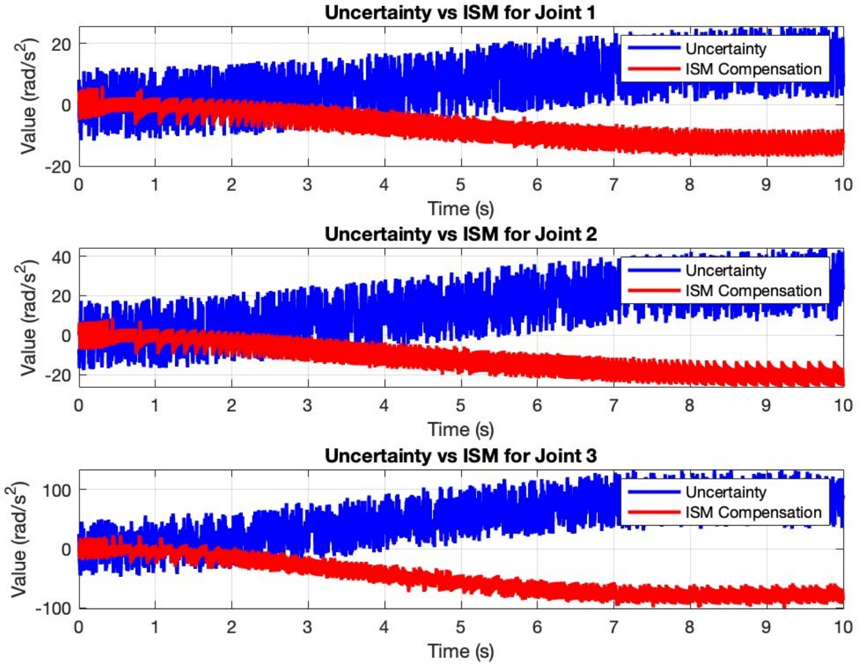
1) We can see however, that the ISM Controller adjusts the control input for the joints to account for the uncertainty





Results

1) As the uncertainty goes higher in magnitude, the ISM controllers counter action increases as well



Possible considerations

1) Uncertainty modeling - The simulated uncertainty might have not been aggressive enough to impact MPC performance as the uncertainty injected by the authors
2) Hardware Differences: The COMAU Smart3-S2 robot in the paper may have different characteristics than our simulated model
    a) The paper notes that their model of the COMAU Smart3-S2 on Matlab Simulink was based on data from real experiments.
    b) However, we are simulating a perfect model on which the MPC acts upon. Due to this, the performance of MPC itself is highly accurate.
    c) The performance difference between MPC and ISM is marginally different, and seems to be negligible due to this .

References

- [1] Incremona, Gian Paolo, et al. “MPC for robot manipulators with integral sliding modes generation.” IEEE/ASME Transactions on Mechatronics, vol. 22, no. 3, June 2017, pp. 1299–1307, https://doi.org/10.1109/tmech.2017.2674701.
- [2] Adaptive chaos control of a humanoid robot arm: a fault-tolerant scheme - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Sliding-surface-and-sliding-variable_fig1_370320084
- [3] *ECE 5463 taught at Ohio State University by Prof. Wei Zhang https://storage1.ucsd.edu/slides/CSE291Robo/L9_lagrangian.html#/title*