From Data to Control: A Two-Stage Simulation-Based Approach




Federico Dettù **Braghadeesh Lakshminarayanan** Simone Formentin Cristian R. Rojas

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden

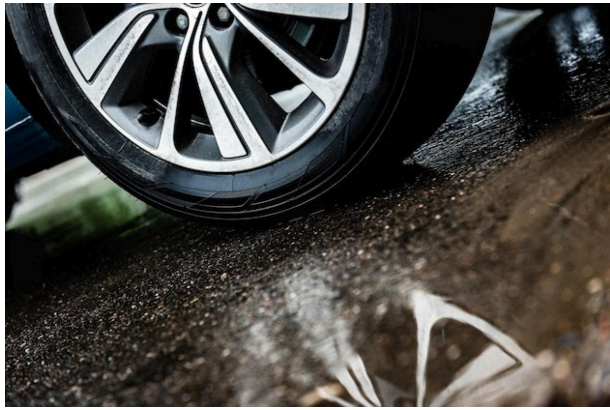
ECC 2024, Sweden
June 28, 2024

Motivation

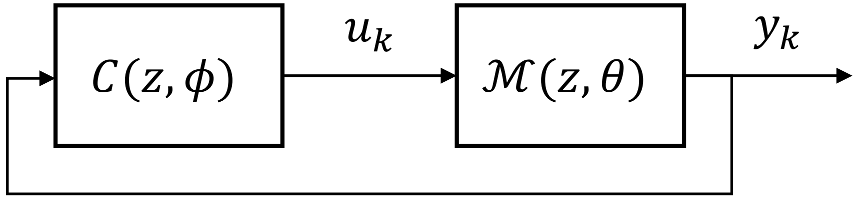
- **Control systems** often required to **operate in uncertain/varying** conditions;

- **Some plant knowledge** is usually **available**. But **key parameters might change** over time;

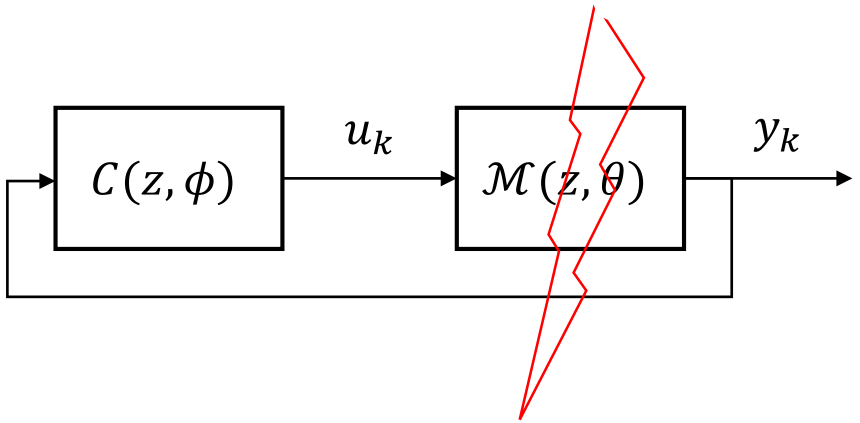
- This **undermines to some extent traditional model-based approaches**, requiring re-tuning of the controller.


Motivation



Motivation

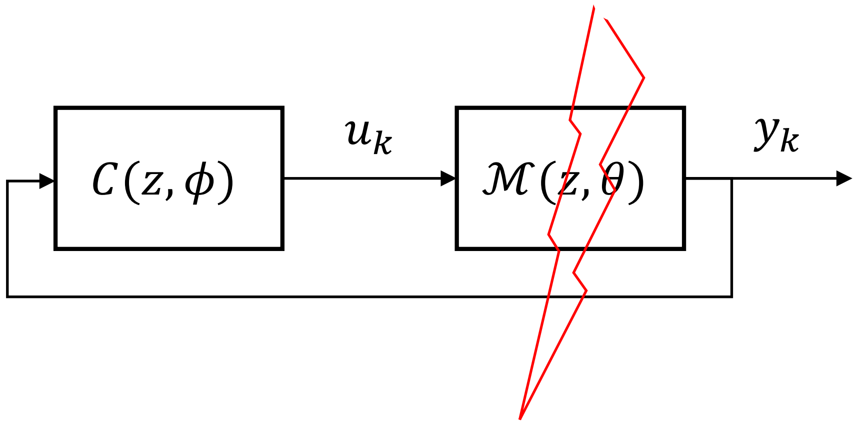


$$\theta_\delta \in \Theta = \{\theta : ||\theta - \theta_0||_2 \leq \rho\}$$

perturbed parameters

maximum deviation (ρ)

Motivation



$$\theta_\delta \in \Theta = \{\theta : ||\theta - \theta_0||_2 \leq \rho\}$$

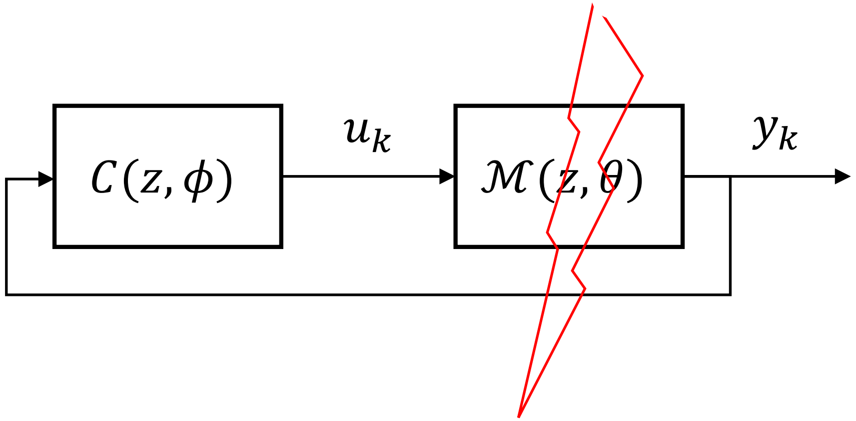
perturbed parameters

maximum deviation (ρ) known

POLITECNICO MILANO 1863

KTH VETENSKAP OCH KONST

Motivation



$$\theta_\delta \in \Theta = \{\theta : ||\theta - \theta_0||_2 \leq \rho\}$$

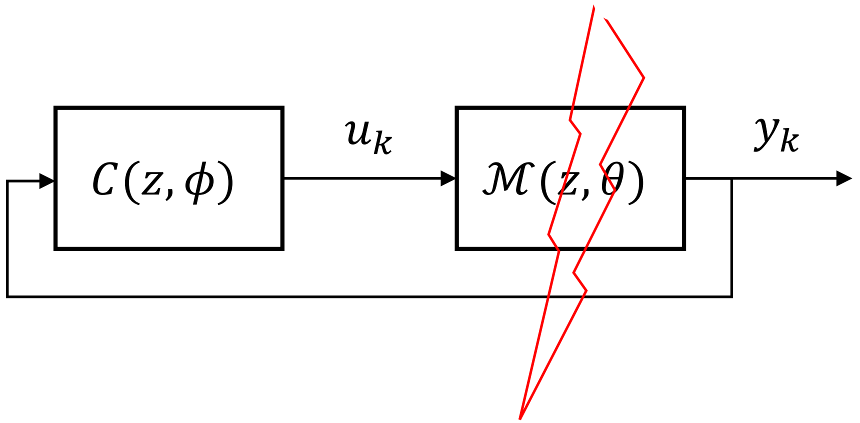
perturbed parameters

maximum deviation (ρ) known

$\mathcal{M}(z,\theta)$
(simulator)

available

Motivation



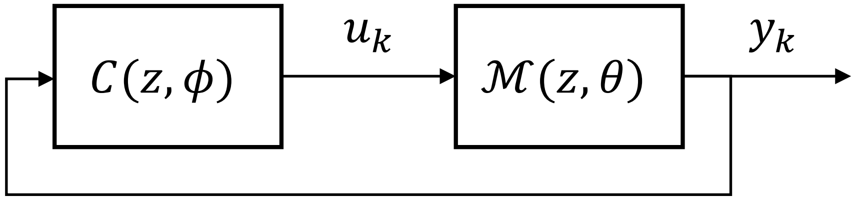
$$\theta_\delta \in \Theta = \{\theta : ||\theta - \theta_0||_2 \leq \rho\}$$

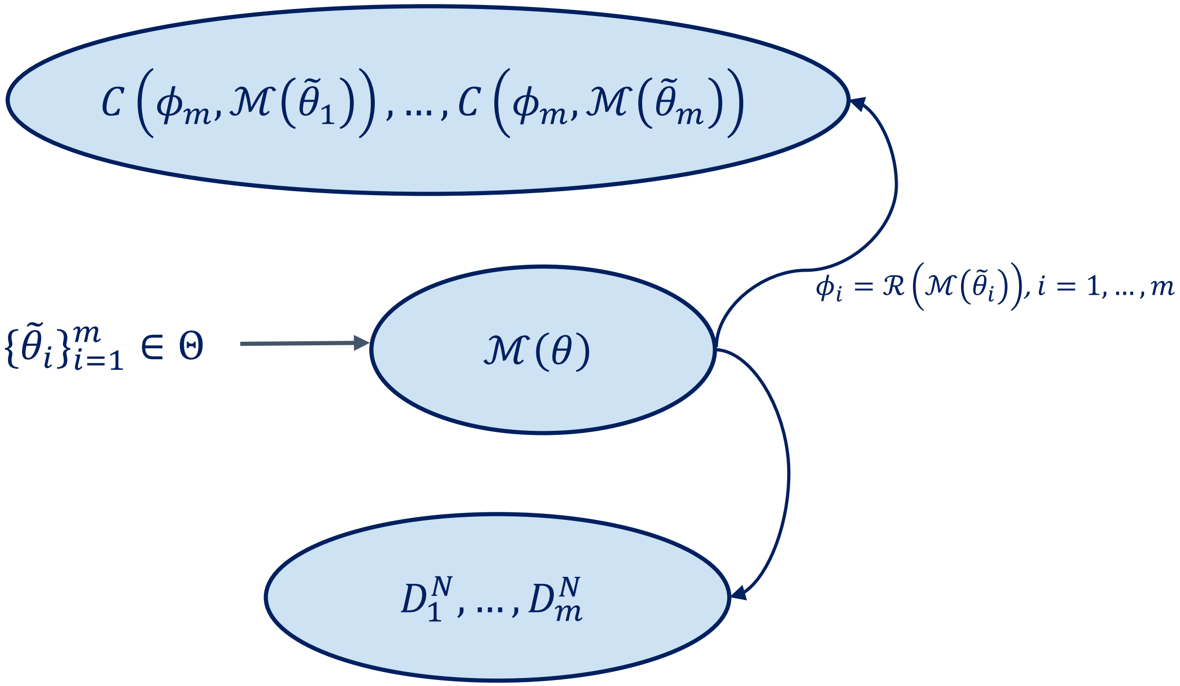
perturbed parameters

Can we leverage this information to fully automate controller tuning ?

- maximum deviation (ρ) — known
- $\mathcal{M}(z,\theta)$ (simulator) — available

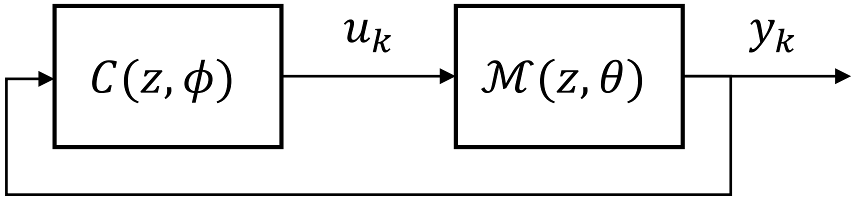
Setup



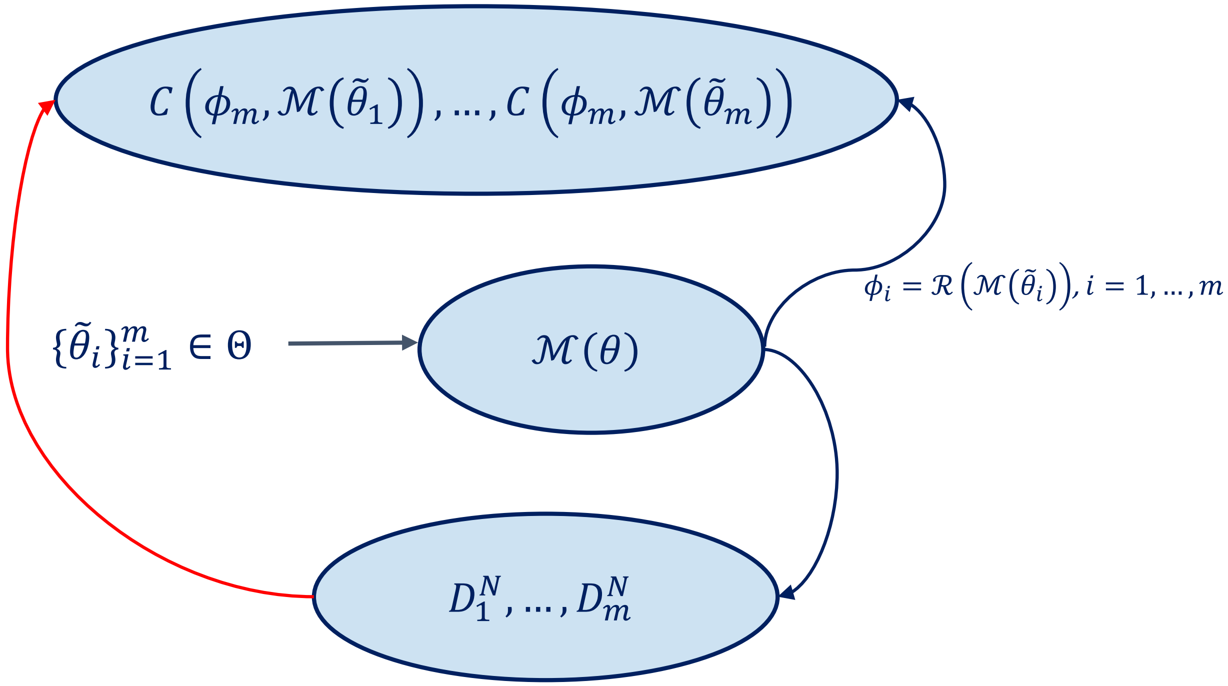


$\tilde{\theta}_1$	$D_1^N = \{(u_1^{(1)}, y_1^{(1)}), \dots, (u_N^{(1)}, y_N^{(1)})\}$	$C\left(\phi_1, \mathcal{M}(\tilde{\theta}_1)\right)$
$\tilde{\theta}_2$	$D_2^N = \{(u_1^{(2)}, y_1^{(2)}), \dots, (u_N^{(2)}, y_N^{(2)})\}$	$C\left(\phi_2, \mathcal{M}(\tilde{\theta}_2)\right)$
$\vdots$	$\vdots$	$\vdots$
$\tilde{\theta}_m$	$D_m^N = \{(u_1^{(m)}, y_1^{(m)}), \dots, (u_N^{(m)}, y_N^{(m)})\}$	$C\left(\phi_m, \mathcal{M}(\tilde{\theta}_m)\right)$

Setup

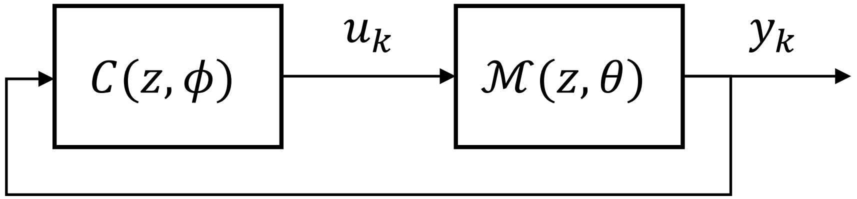


$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2$$



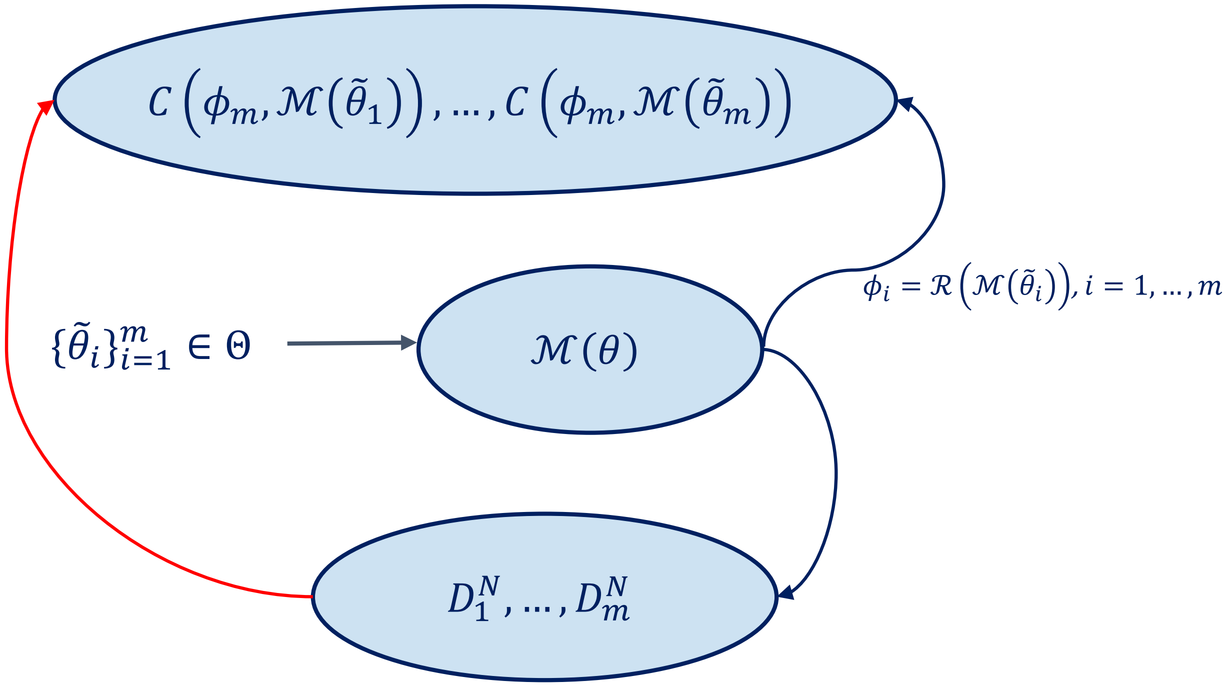
$\tilde{\theta}_1$	$D_1^N = \{(u_1^{(1)}, y_1^{(1)}), \dots, (u_N^{(1)}, y_N^{(1)})\}$	$C\left(\phi_1, \mathcal{M}(\tilde{\theta}_1)\right)$
$\tilde{\theta}_2$	$D_2^N = \{(u_1^{(2)}, y_1^{(2)}), \dots, (u_N^{(2)}, y_N^{(2)})\}$	$C\left(\phi_2, \mathcal{M}(\tilde{\theta}_2)\right)$
$\vdots$	$\vdots$	$\vdots$
$\tilde{\theta}_m$	$D_m^N = \{(u_1^{(m)}, y_1^{(m)}), \dots, (u_N^{(m)}, y_N^{(m)})\}$	$C\left(\phi_m, \mathcal{M}(\tilde{\theta}_m)\right)$

Setup



$$\boldsymbol{f}^* = \underset{\boldsymbol{f}}{\operatorname{argmin}} \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \left\| \boldsymbol{\phi}_{\boldsymbol{i}} - \boldsymbol{f}\left(\boldsymbol{y}_{\mathbf{1}}^{(\boldsymbol{i})}, \boldsymbol{u}_{\mathbf{1}}^{(\boldsymbol{i})}, \dots, \boldsymbol{y}_{\boldsymbol{N}}^{(\boldsymbol{i})}, \boldsymbol{u}_{\boldsymbol{N}}^{(\boldsymbol{i})}\right) \right\|_{\mathbf{2}}^{\mathbf{2}}$$

Use of Two-Stage parameter estimation paradigm to meta-learn $\boldsymbol{f}^*$



$\tilde{\theta}_1$	$D_1^N = \{(u_1^{(1)}, y_1^{(1)}), \dots, (u_N^{(1)}, y_N^{(1)})\}$	$C\left(\phi_1, \mathcal{M}(\tilde{\theta}_1)\right)$
$\tilde{\theta}_2$	$D_2^N = \{(u_1^{(2)}, y_1^{(2)}), \dots, (u_N^{(2)}, y_N^{(2)})\}$	$C\left(\phi_2, \mathcal{M}(\tilde{\theta}_2)\right)$
$\vdots$	$\vdots$	$\vdots$
$\tilde{\theta}_m$	$D_m^N = \{(u_1^{(m)}, y_1^{(m)}), \dots, (u_N^{(m)}, y_N^{(m)})\}$	$C\left(\phi_m, \mathcal{M}(\tilde{\theta}_m)\right)$

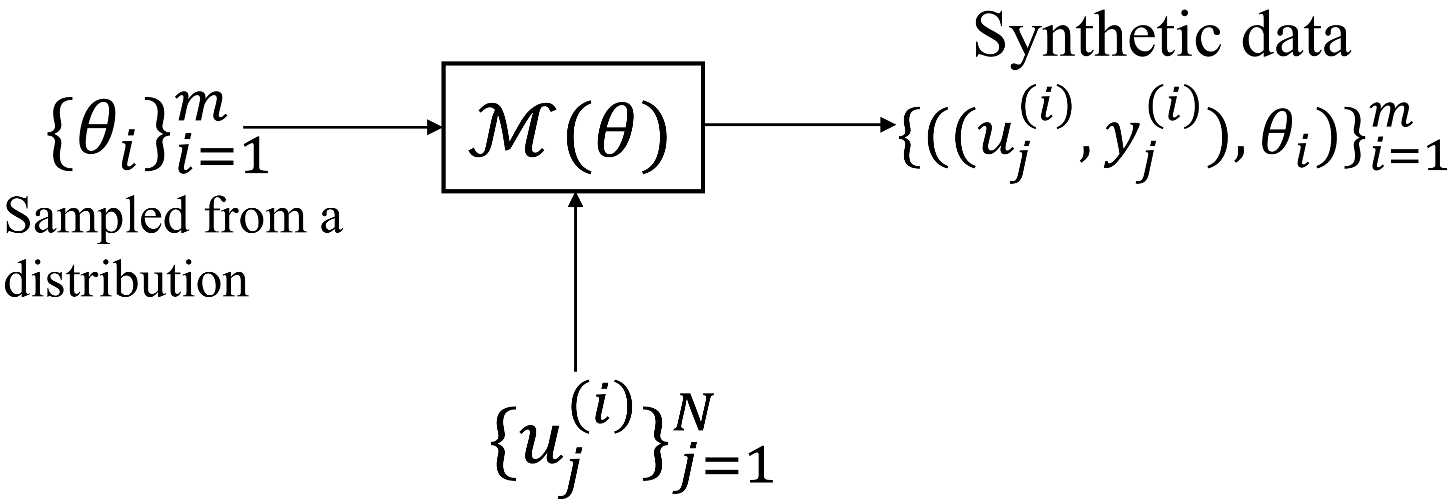
Outline

- ❑Two-stage (TS) estimation paradigm
- ❑TS for controller tuning
- ❑Numerical study
- ❑Conclusion

Outline

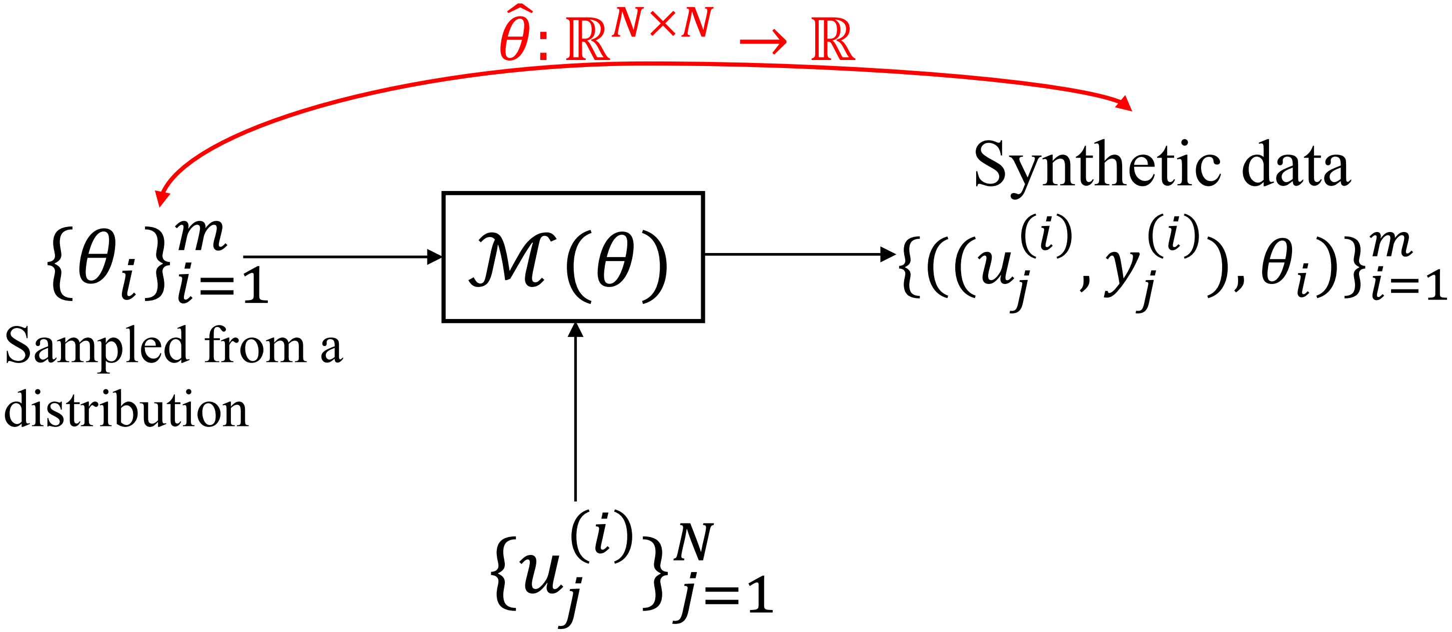
- ❑Two-stage (TS) estimation paradigm
- ❑TS for controller tuning
- ❑Numerical study
- ❑Conclusion

TS estimation paradigm



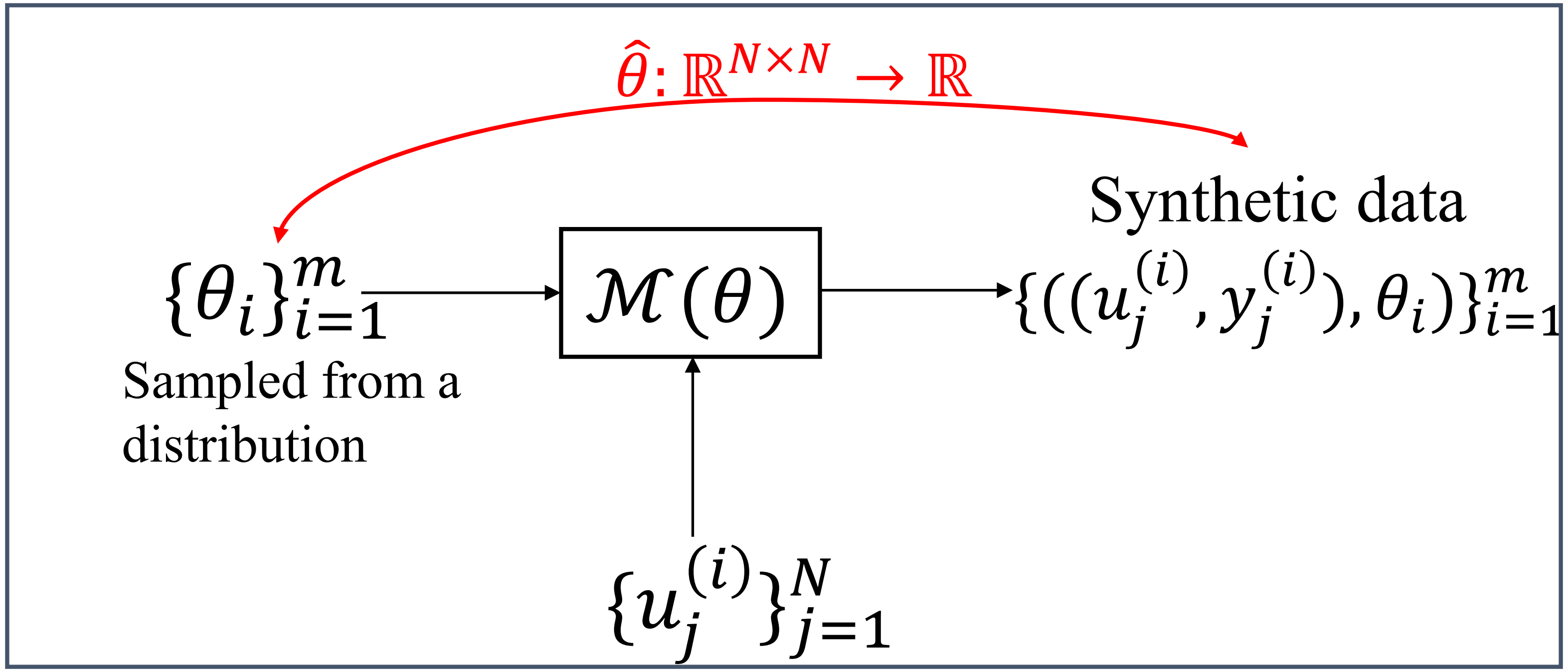
S. Garatti, S. Bittanti. "A new paradigm for parameter estimation in system modeling". Int. J. Adapt. Control Sig. Proc., 2013

TS estimation paradigm



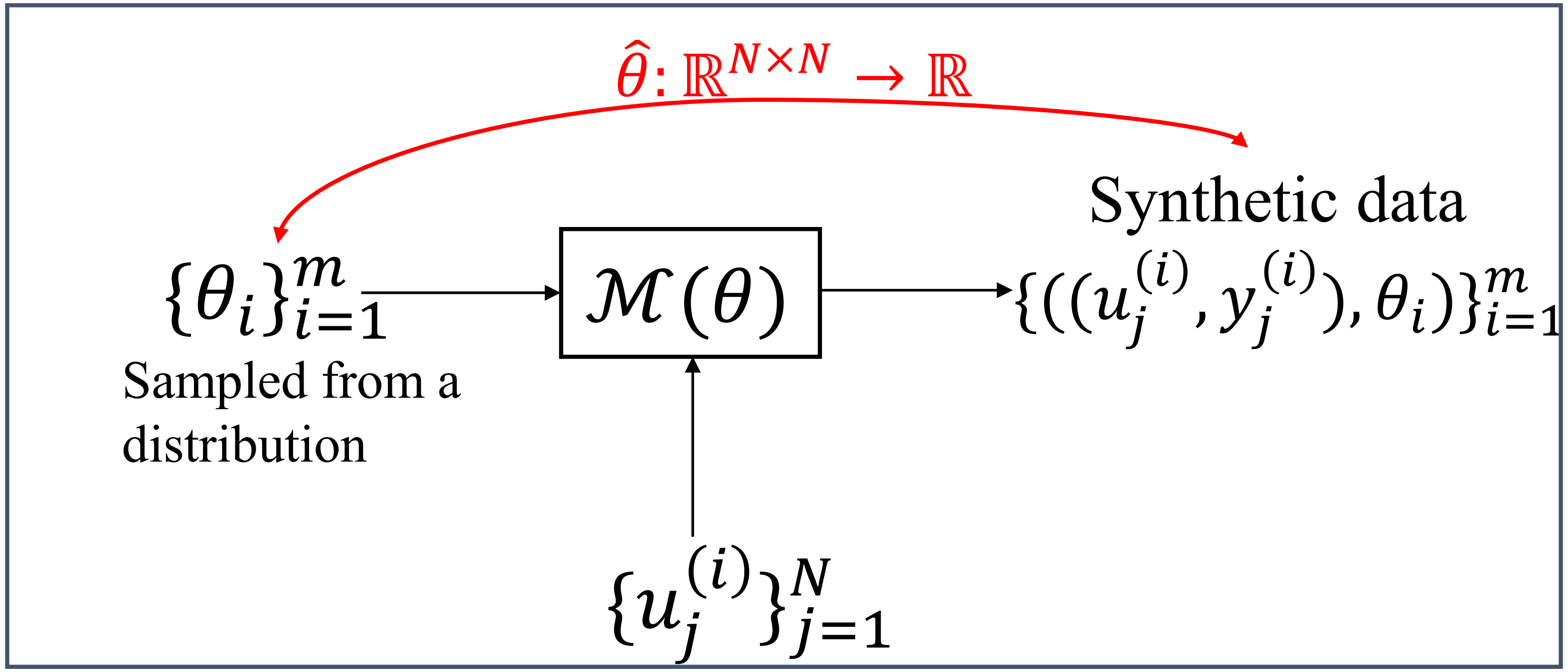
S. Garatti, S. Bittanti. "A new paradigm for parameter estimation in system modeling". Int. J. Adapt. Control Sig. Proc., 2013

TS estimation paradigm



Inverse (meta) learning

TS estimation paradigm



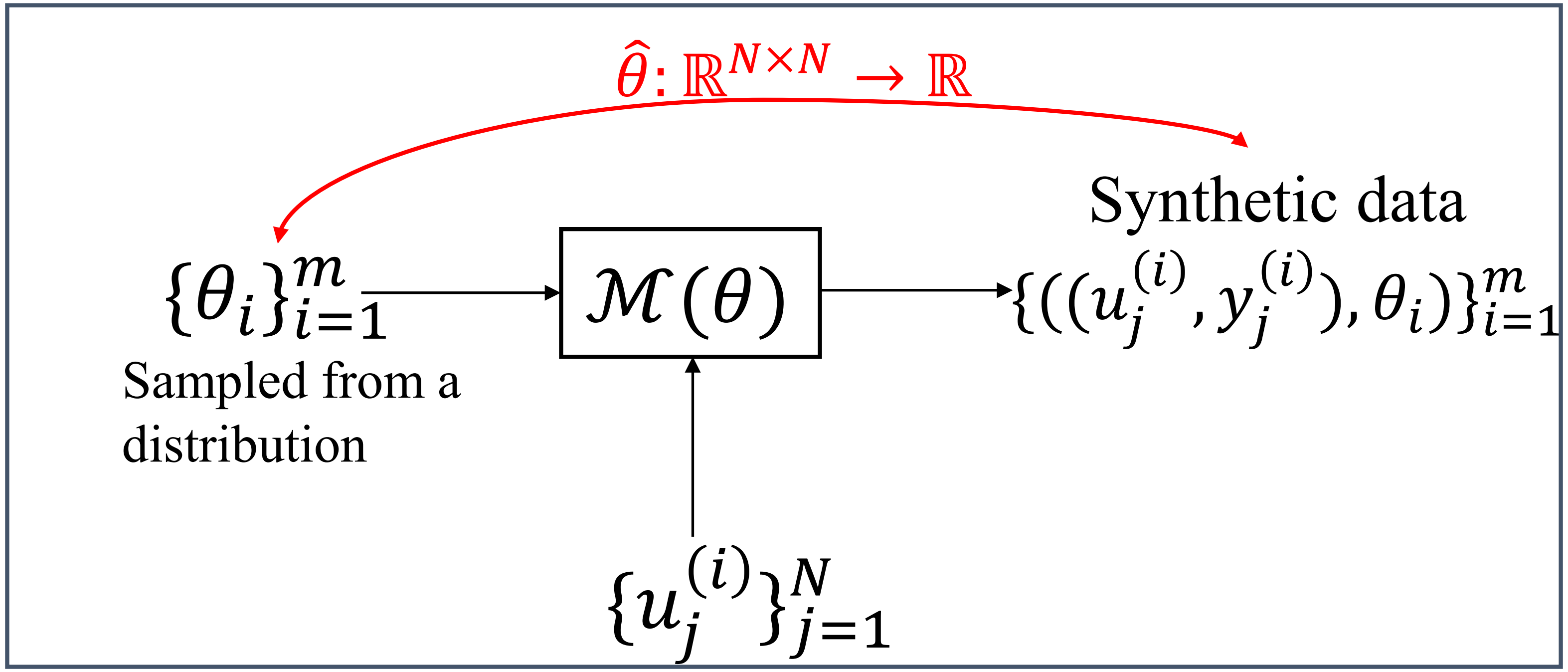
Inverse (meta) learning

$$g^* = \arg\min_{g \in \mathcal{G}} \frac{1}{M} \sum_{i=1}^{m} \left\| \theta_i - g\left(h(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}) \right) \right\|_2^2$$

$$\hat{\theta}_{TS} = g^* \circ h$$

S. Garatti, S. Bittanti. "A new paradigm for parameter estimation in system modeling". Int. J. Adapt. Control Sig. Proc., 2013

TS estimation paradigm



Inverse (meta) learning

$$g^* = \arg\min_{g\in\mathcal{G}} \frac{1}{M}\sum_{i=1}^{m}\left\|\theta_i - g\left(h(y_1^{(i)}, u_1^{(i)}, \ldots, y_N^{(i)}, u_N^{(i)})\right)\right\|_2^2$$

$$\hat{\theta}_{TS} = g^* \circ h$$

$h: \mathbb{R}^{N\times N} \rightarrow \mathbb{R}^n$ Fixed
Data compressor

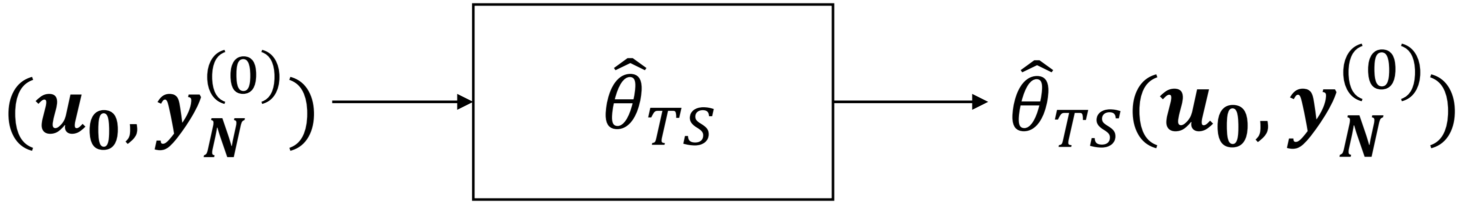
$g: \mathbb{R}^n \rightarrow \mathbb{R}^{d_\theta}$ Optimized
Function approximator

S. Garatti, S. Bittanti. "A new paradigm for parameter estimation in system modeling". Int. J. Adapt. Control Sig. Proc., 2013

How and why?

$\hat{\theta}_{TS}$ is an estimator functional

- Need to be tested on a new observation



Why promising?

- Compression step – helps in dimensionality reduction
- Second stage – A continuous map of compressed samples

Outline

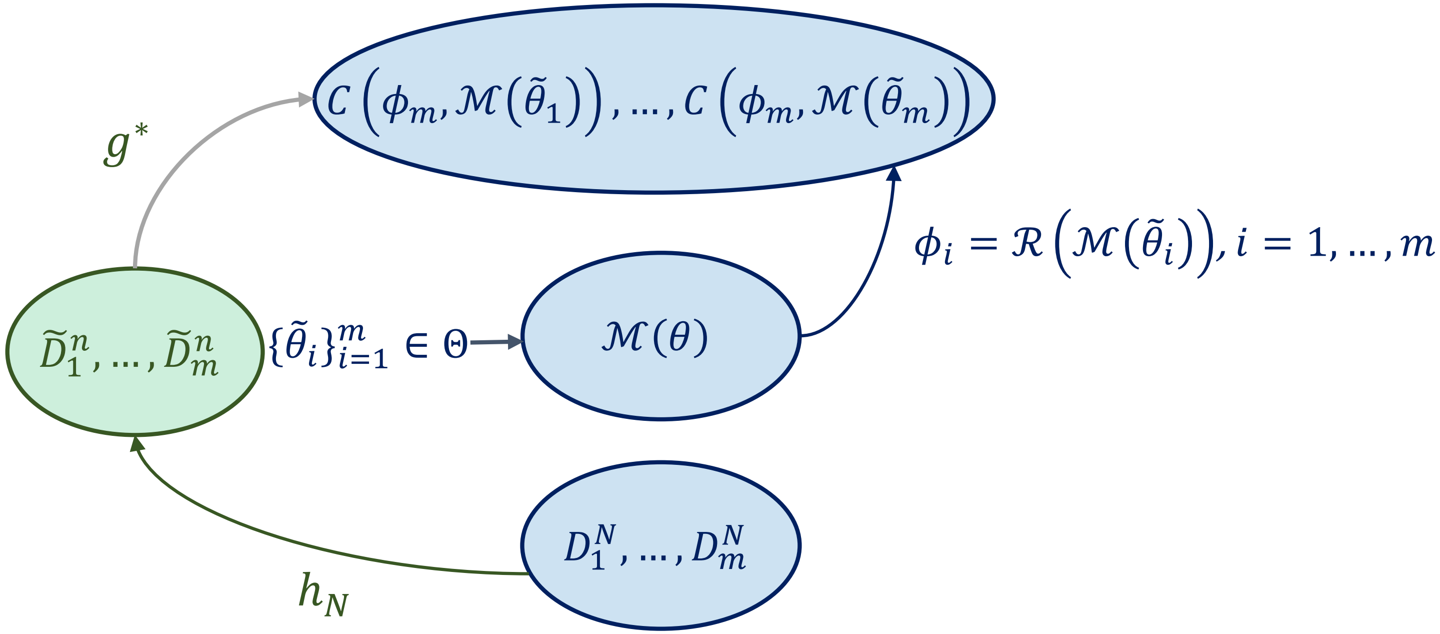
- ❑Two-stage (TS) estimation paradigm
- ❑TS for controller tuning
- ❑Numerical study
- ❑Conclusion

TS for controller tuning

Objective: Learn function $\boldsymbol{f}^*$ such that:

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2 = g^* \circ h_N$$

$$g^* = \underset{g}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - g\left(h_N\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right)\right) \right\|_2^2$$

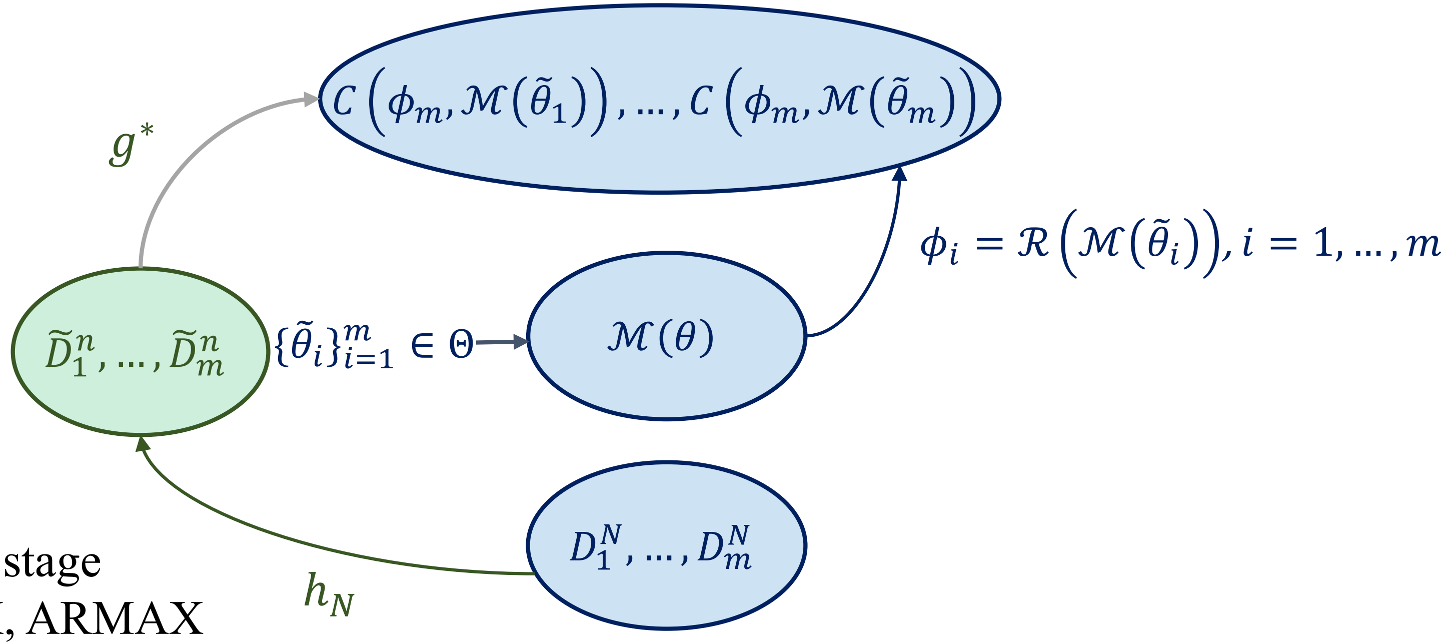


TS for controller tuning

Objective: Learn function $\boldsymbol{f}^*$ such that:

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2 = g^* \circ h_N$$

$$g^* = \underset{g}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - g\left(h_N\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right)\right) \right\|_2^2$$



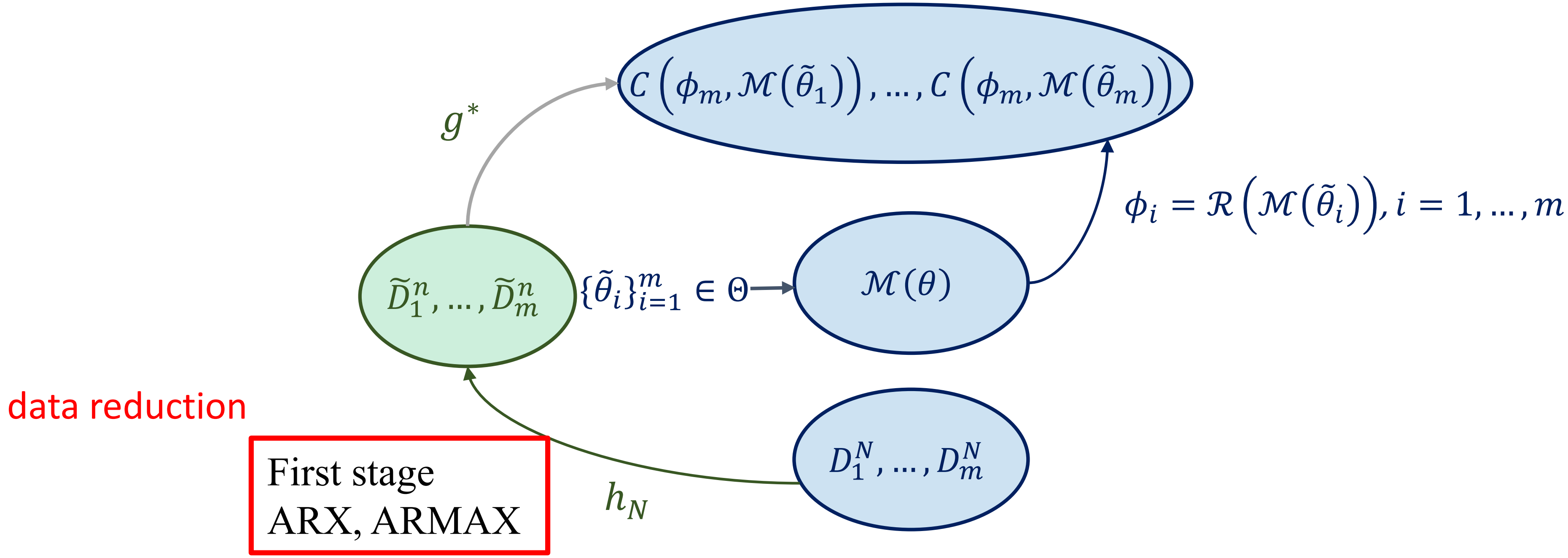
First stage
ARX, ARMAX

TS for controller tuning

Objective: Learn function $\boldsymbol{f}^*$ such that:

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2 = g^* \circ h_N$$

$$g^* = \underset{g}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - g\left(h_N\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right)\right) \right\|_2^2$$

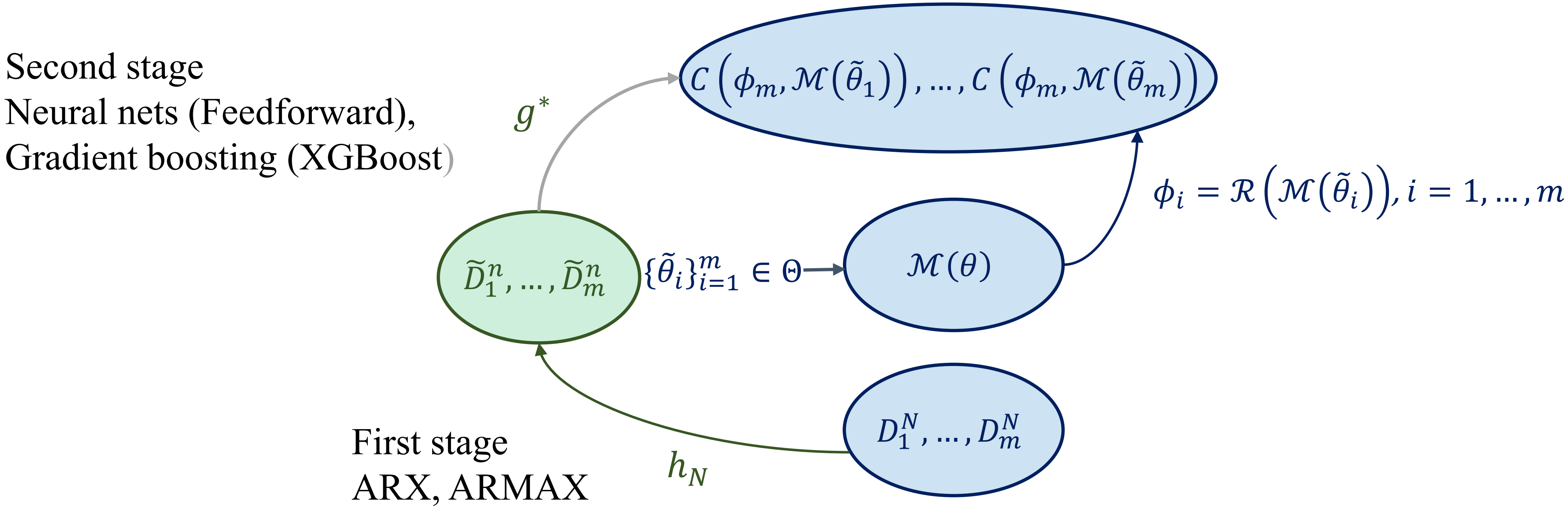


TS for controller tuning

Objective: Learn function $\boldsymbol{f}^*$ such that:

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2 = g^* \circ h_N$$

$$g^* = \underset{g}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - g\left(h_N\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right)\right) \right\|_2^2$$



TS for controller tuning

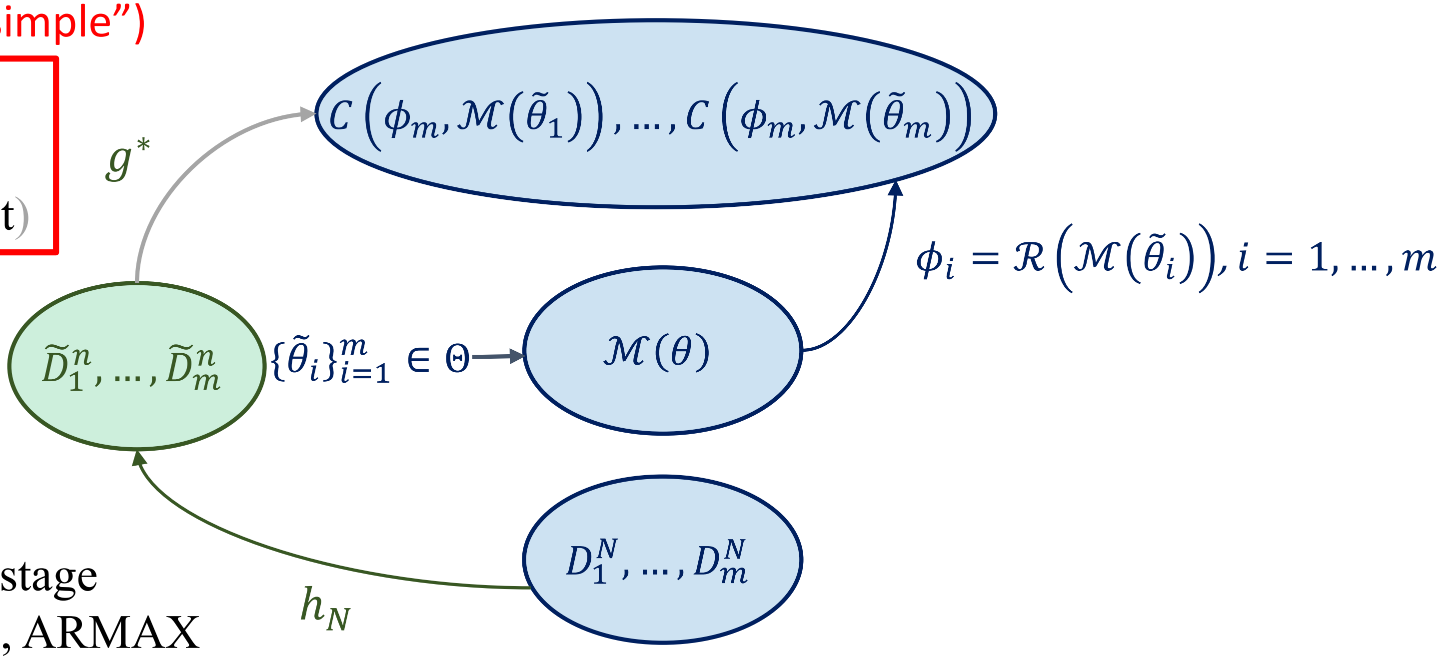
Objective: Learn function $\boldsymbol{f}^*$ such that:

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - f\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right) \right\|_2^2 = g^* \circ h_N$$

$$g^* = \underset{g}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \left\| \phi_i - g\left(h_N\left(y_1^{(i)}, u_1^{(i)}, \dots, y_N^{(i)}, u_N^{(i)}\right)\right) \right\|_2^2$$

Function approximation ("simple")

Second stage
Neural nets (Feedforward),
Gradient boosting (XGBoost)



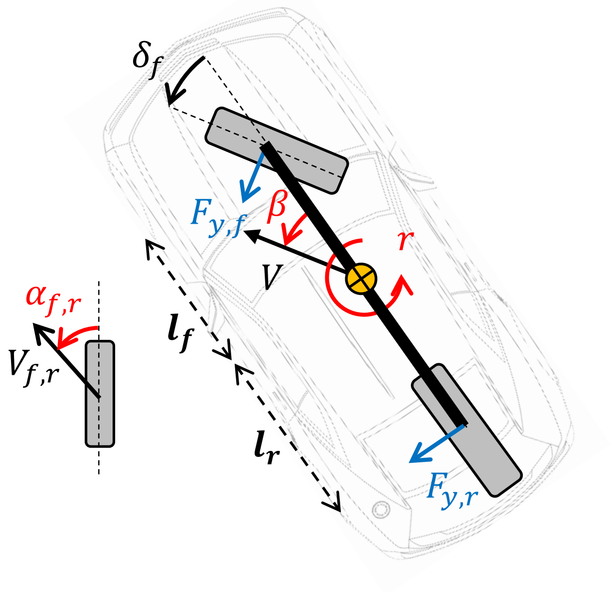
First stage
ARX, ARMAX

Outline

- ❑Two-stage (TS) estimation paradigm
- ❑TS for controller tuning
- ❑Numerical study
- ❑Conclusion

Numerical study

Objective: Achieve desired yaw-rate r



Vehicle dynamics:

state: $\boldsymbol{x} = (\beta, r, \alpha_f, \alpha_r, z, \delta_f)^T$
input $u = \delta_f^{cmd}$
output $y = r$

$$\dot{\beta} = -r - \frac{C_f \alpha_f}{M_{veh}\, v_x} - \frac{C_r \alpha_r}{M_{veh}\, v_x}$$

$$\dot{r} = -\frac{l_f C_f \alpha_f}{J_z} + \frac{l_r C_r \alpha_r}{J_z}$$

$$\dot{\alpha}_f = -\frac{v_x}{l_{rel,f}}(\alpha_f - \alpha_f^{kin}), \qquad \alpha_f^{kin} = -\delta_f + \beta + \frac{L_f}{v_x} r$$

$$\dot{\alpha}_r = -\frac{v_x}{l_{rel,r}}(\alpha_r - \alpha_r^{kin}), \qquad \alpha_r^{kin} = \beta - \frac{L_r}{v_x} r$$

$$\dot{z} = -\omega_n^2\, \delta_f + \omega_n^2 \delta_f^{cmd}$$

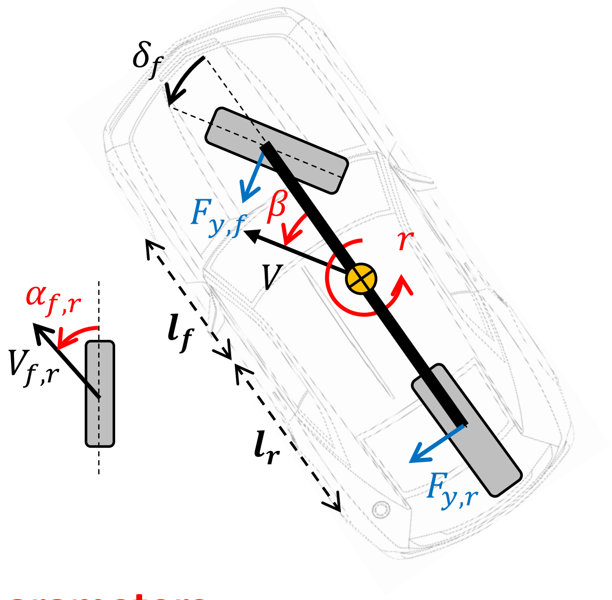
$$\dot{\delta}_f = z - 2\,\xi \omega_n \delta_f$$

M_{veh} [kg]	J_z [kgm^2]	l_f [m]	l_r [m]	C_f [N	C_r [N/rad]	T_s [s]	ω_n [rad/s]	ξ
1895	2400	1.18	1.53	$1.24 \cdot 10^5$	$1.66 \cdot 10^5$	0.01	2π . 5	0.9

1. T. Hiraoka et al, "Model-following sliding mode control for active four-wheel steering vehicle," Review of Automotive Engineering, 2004.

Numerical study

Objective: Achieve desired yaw-rate r



Vehicle dynamics:

state: $\boldsymbol{x} = (\beta, r, \alpha_f, \alpha_r, z, \delta_f)^T$
input $u = \delta_f^{cmd}$
output $y = r$

$$\dot{\beta} = -r - \frac{C_f \alpha_f}{M_{veh}\, v_x} - \frac{C_r \alpha_r}{M_{veh}\, v_x}$$

$$\dot{r} = -\frac{l_f C_f \alpha_f}{J_z} + \frac{l_r C_r \alpha_r}{J_z}$$

$$\dot{\alpha}_f = -\frac{v_x}{l_{rel,f}}(\alpha_f - \alpha_f^{kin}), \qquad \alpha_f^{kin} = -\delta_f + \beta + \frac{L_f}{v_x} r$$

$$\dot{\alpha}_r = -\frac{v_x}{l_{rel,r}}(\alpha_r - \alpha_r^{kin}), \qquad \alpha_r^{kin} = \beta - \frac{L_r}{v_x} r$$

$$\dot{z} = -\omega_n^2\, \delta_f + \omega_n^2 \delta_f^{cmd}$$

$$\dot{\delta}_f = z - 2\,\xi \omega_n \delta_f$$

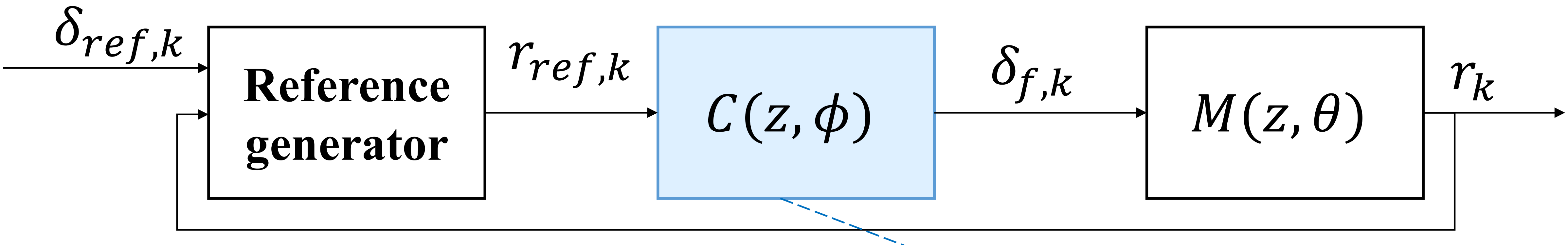
Uncertain parameters

M_{veh} [kg]	J_z [kgm^2]	l_f [m]	l_r [m]	C_f [N	C_r [N/rad]	T_s [s]	ω_n [rad/s]	ξ
1895	2400	1.18	1.53	$1.24 \cdot 10^5$	$1.66 \cdot 10^5$	0.01	2π . 5	0.9

$$M_{veh} = M_0 + M_\delta$$
$$C_{f,r} = C_{f,r}^{nom}\, \mu_s$$
$$M_\delta \in [0,300]$$
$$\mu_s \in [0.3,1.1]$$

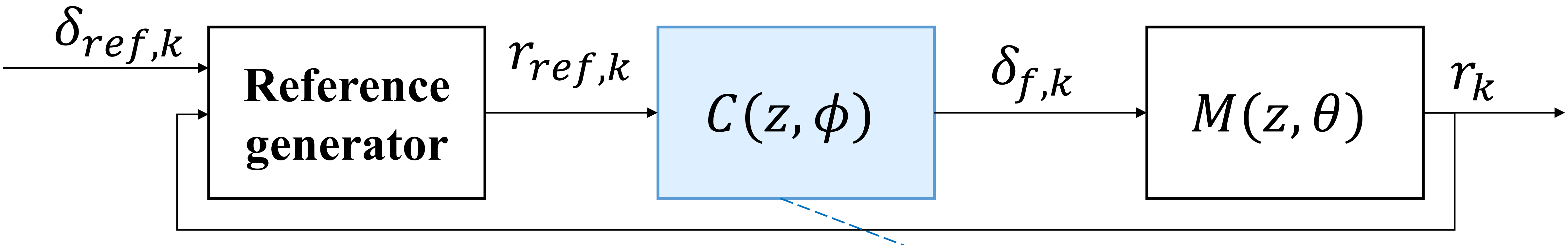
1. T. Hiraoka et al, "Model-following sliding mode control for active four-wheel steering vehicle," Review of Automotive Engineering, 2004.

Control design



A **Proportional-Integral controller** suffices for our purposes. We design it via loop shaping, such as to guarantee phase margin $\phi_m \geq 60^{\circ}$ and cutting frequency $\omega_c \geq 1.5\ Hz$.

Control design



Train and test data:

$m = 1500, N = 10000, T_s = 0.01\ s$

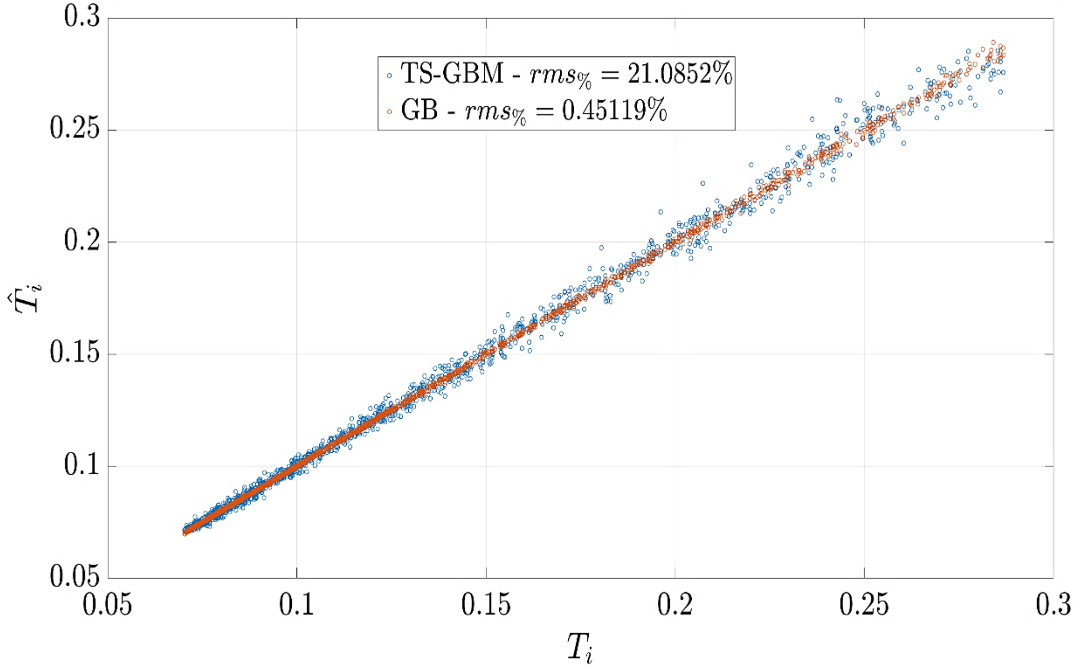
$\delta_{f,k}$ - PRBS
r_k - Perturbed with Gaussian white noise

A **Proportional-Integral controller** suffices for our purposes. We design it via loop shaping, such as to guarantee phase margin $\phi_m \geq 60^{\circ}$ and cutting frequency $\omega_c \geq 1.5\ Hz$.

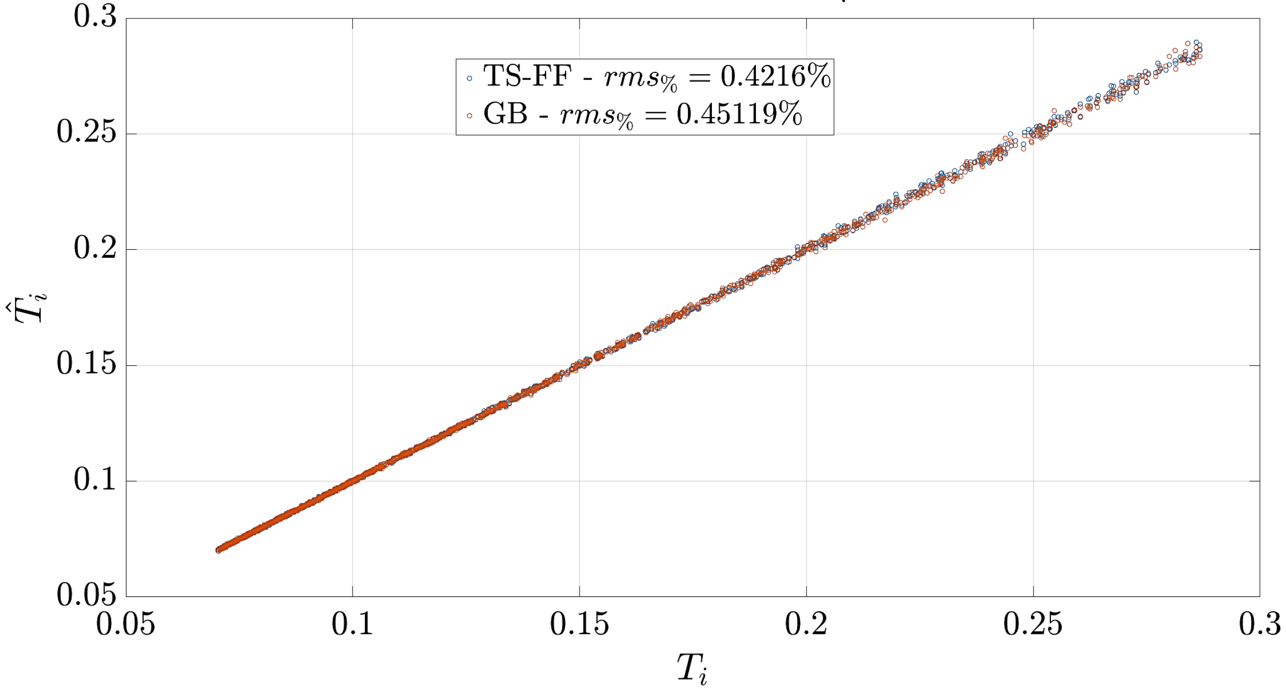
Results

T_i regression performance

$$rms\,\%\,(\nu,\hat{\nu}) = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\left(100\frac{\nu_i-\hat{\nu}_i}{\nu_i}\right)^2}$$



XGBoost

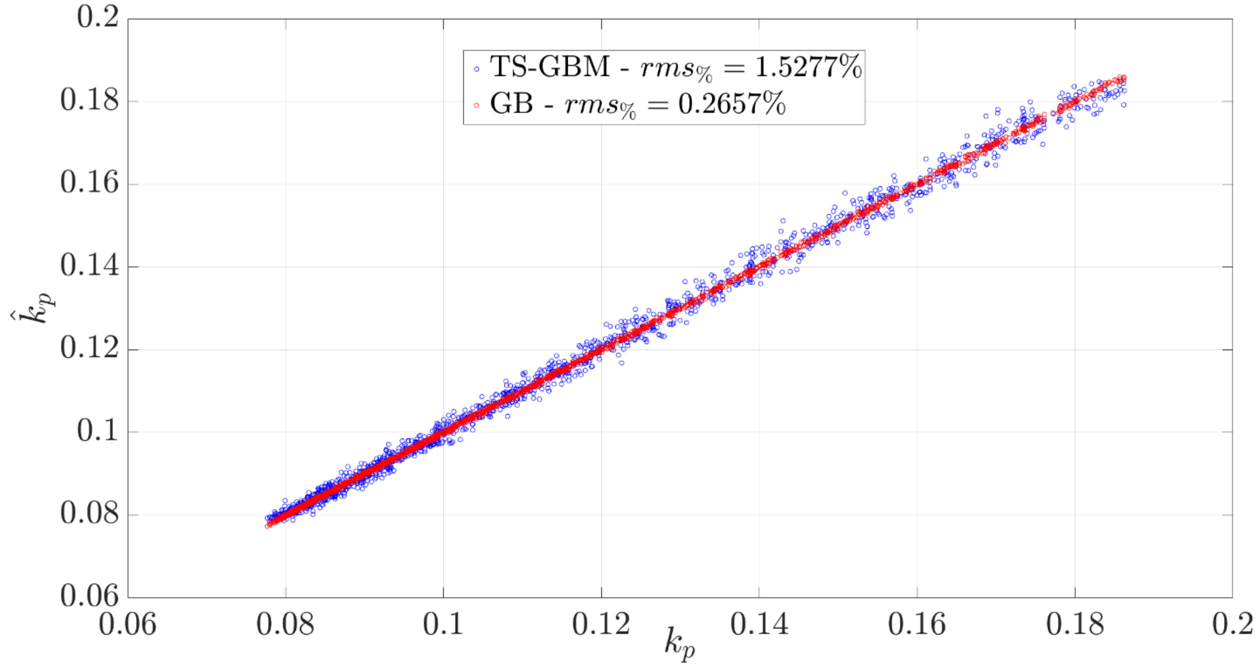


FF-NN

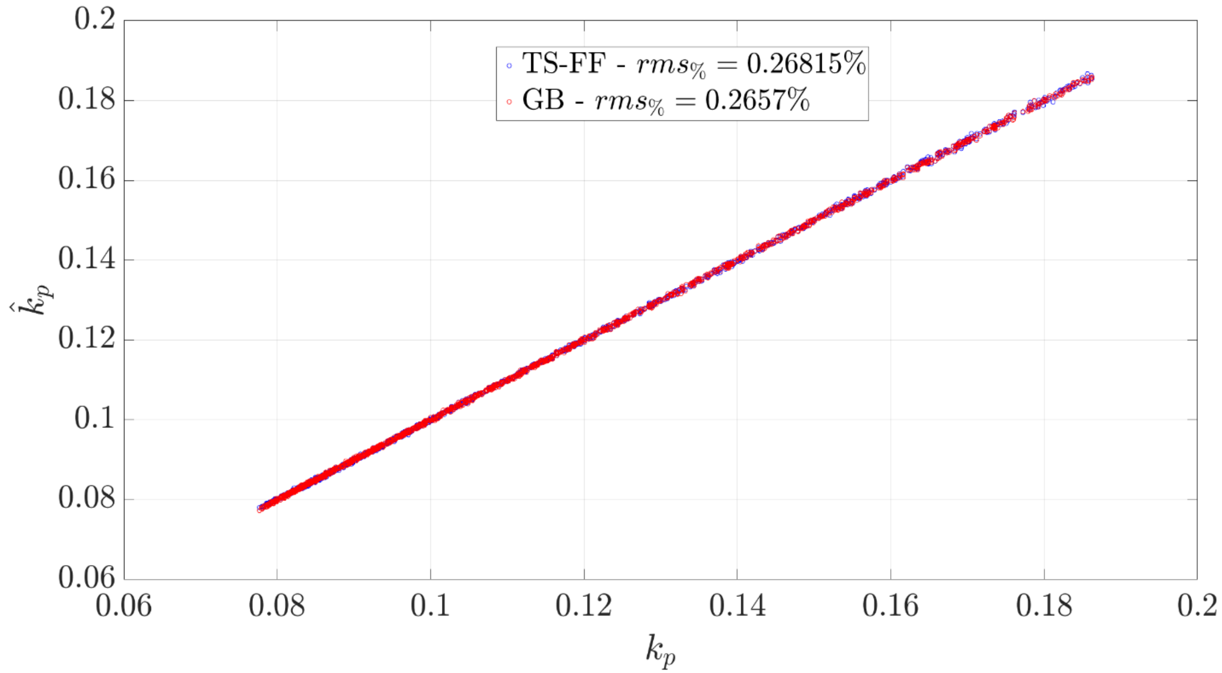
Results

k_p regression performance

$$rms\,\%\,(\nu,\hat{\nu}) = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\left(100\frac{\nu_i-\hat{\nu}_i}{\nu_i}\right)^2}$$



XGBoost



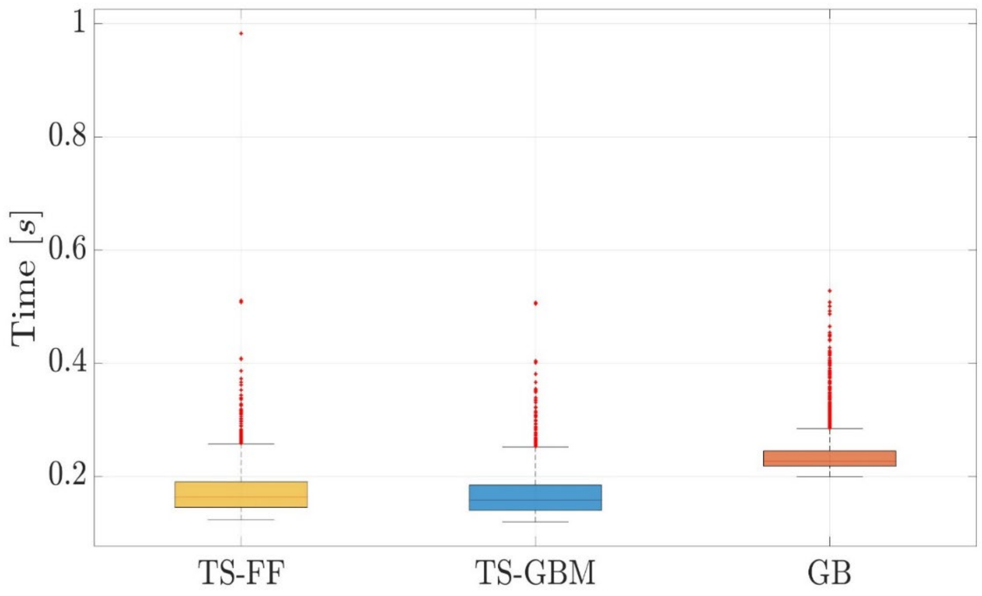
FF-NN

Results

Closed loop performance

Method	$\varphi_m[deg]$		ω_c	
	Mean	Std.	Mean	Std.
GB	59.98	0.2385	1.5	0.0037
TS-FF	59.99	0.1714	1.5	0.0032
TS-GBM	60.01	0.69	1.5	0.0143

Computation time analysis



Outline

- ❑Two-stage (TS) estimation paradigm
- ❑TS for controller tuning
- ❑Numerical study
- ❑Conclusion

Conclusion

- Designed a meta-learning based controller tuning using TS

Conclusion

- Designed a meta-learning based controller tuning using TS
- First stage - ARMAX, second stage – Feed-forward neural network and XGBoost

Conclusion

- Designed a meta-learning based controller tuning using TS

- First stage - ARMAX, second stage – Feed-forward neural network and XGBoost

  - First stage helps in model reduction, while second stage acts as function approximator

Conclusion

- Designed a meta-learning based controller tuning using TS
- First stage - ARMAX, second stage – Feed-forward neural network and XGBoost
  - First stage helps in model reduction, while second stage acts as function approximator
- Improved computation time (at the inference step) with same closed loop guarantees as Grey-Box procedure

Conclusion

- Designed a meta-learning based controller tuning using TS
- First stage - ARMAX, second stage – Feed-forward neural network and XGBoost
  - First stage helps in model reduction, while second stage acts as function approximator
- Improved computation time (at the inference step) with same closed loop guarantees as Grey-Box procedure
  - GBM – Training and testing are fast compared to FF

Conclusion

- Designed a meta-learning based controller tuning using TS
- First stage - ARMAX, second stage – Feed-forward neural network and XGBoost
  - First stage helps in model reduction, while second stage acts as function approximator
- Improved computation time (at the inference step) with same closed loop guarantees as Grey-Box procedure
  - GBM – Training and testing are fast compared to FF
  - FF- Better accuracy in terms of controller parameters regression performance

POLITECNICO
MILANO 1863

KTH
VETENSKAP
OCH KONST

THANK YOU FOR YOUR ATTENTION ☺