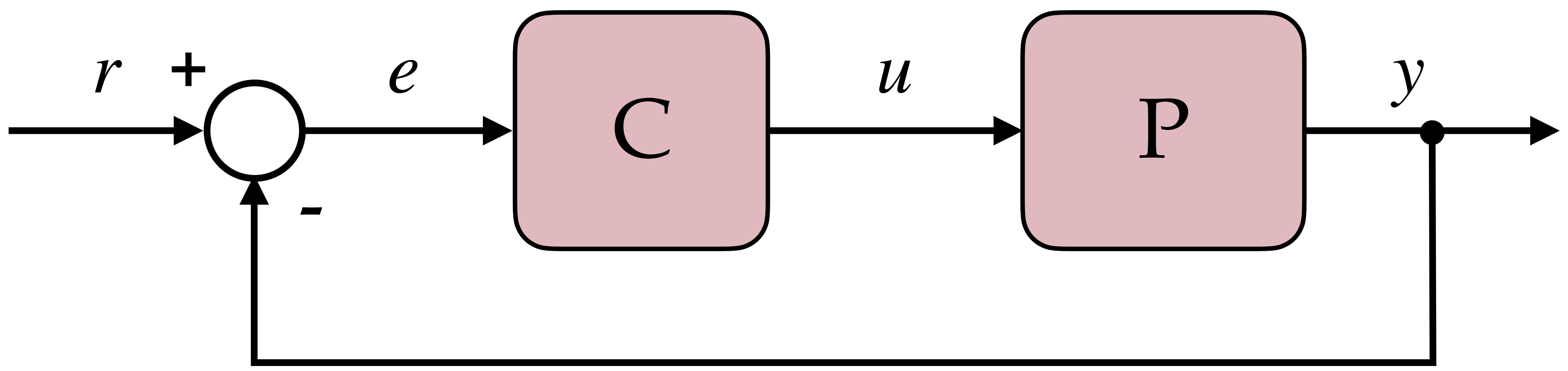
Lecture 13 Online Adaptation

Recursive Least Square

Online Adaptation

- Online adaptation can be applied to:
  - Offline obtained control policies (either designed or learned) — direct adaptive control
  - Offline obtained dynamic models (either designed or learned) — indirect adaptive control
    - Once the model is adapted, the model-based controller will also be changed.



Online Adaptation vs Online Filtering

- Mathematically, adaptation and filtering solve the same problem.
- Practically, people use the term “filtering” for system states and the term “adaptation” for system parameters.
- We will see that the widely used recursive least square algorithm for online adaptation is just another form of the Kalman filter.

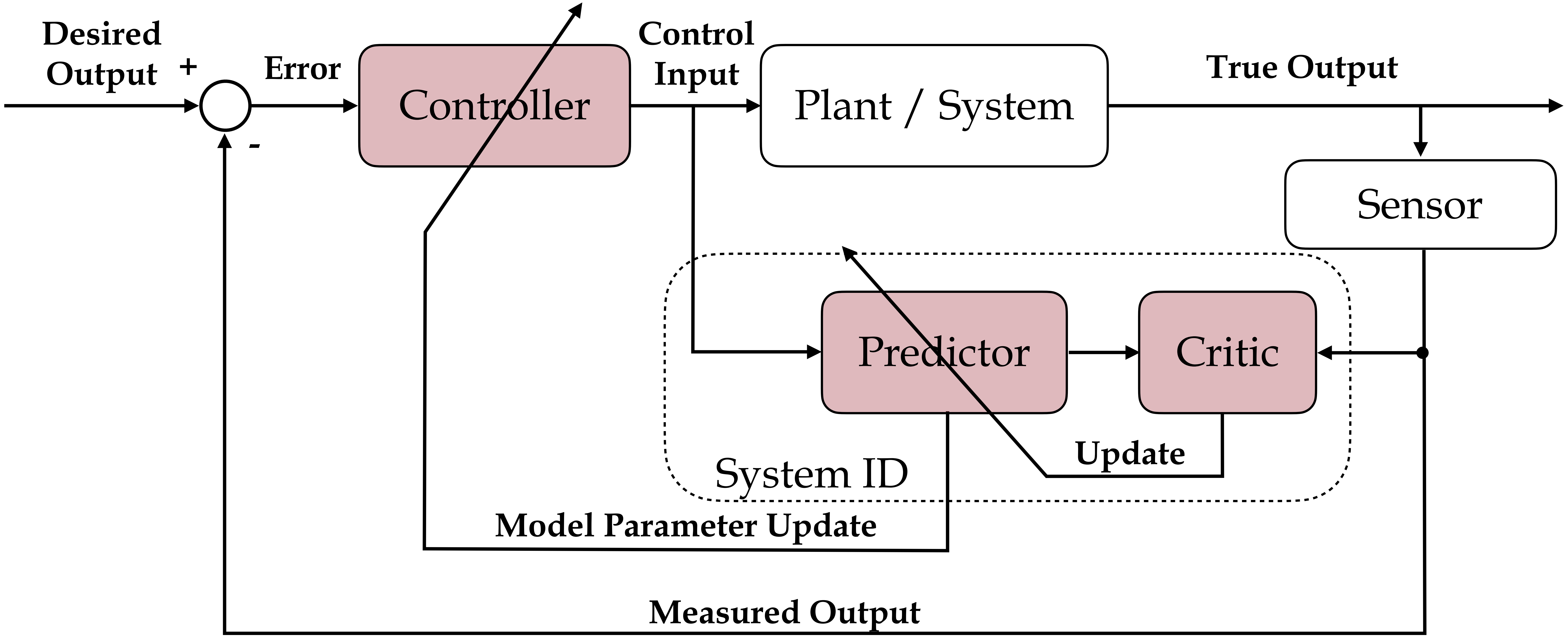
Big Picture

- In the following weeks, we will discuss:
  - Model adaptation
  - Indirect adaptive control (model adaptation + optimal control)
  - Direct adaptive control - Reinforcement learning
- Today, we will discuss:
  - Recursive least square for model adaptation

Online Adaptation

- Problem Formulation
- Recursive Least Square
- Similarity with Kalman Filters

Indirect Adaptive Control



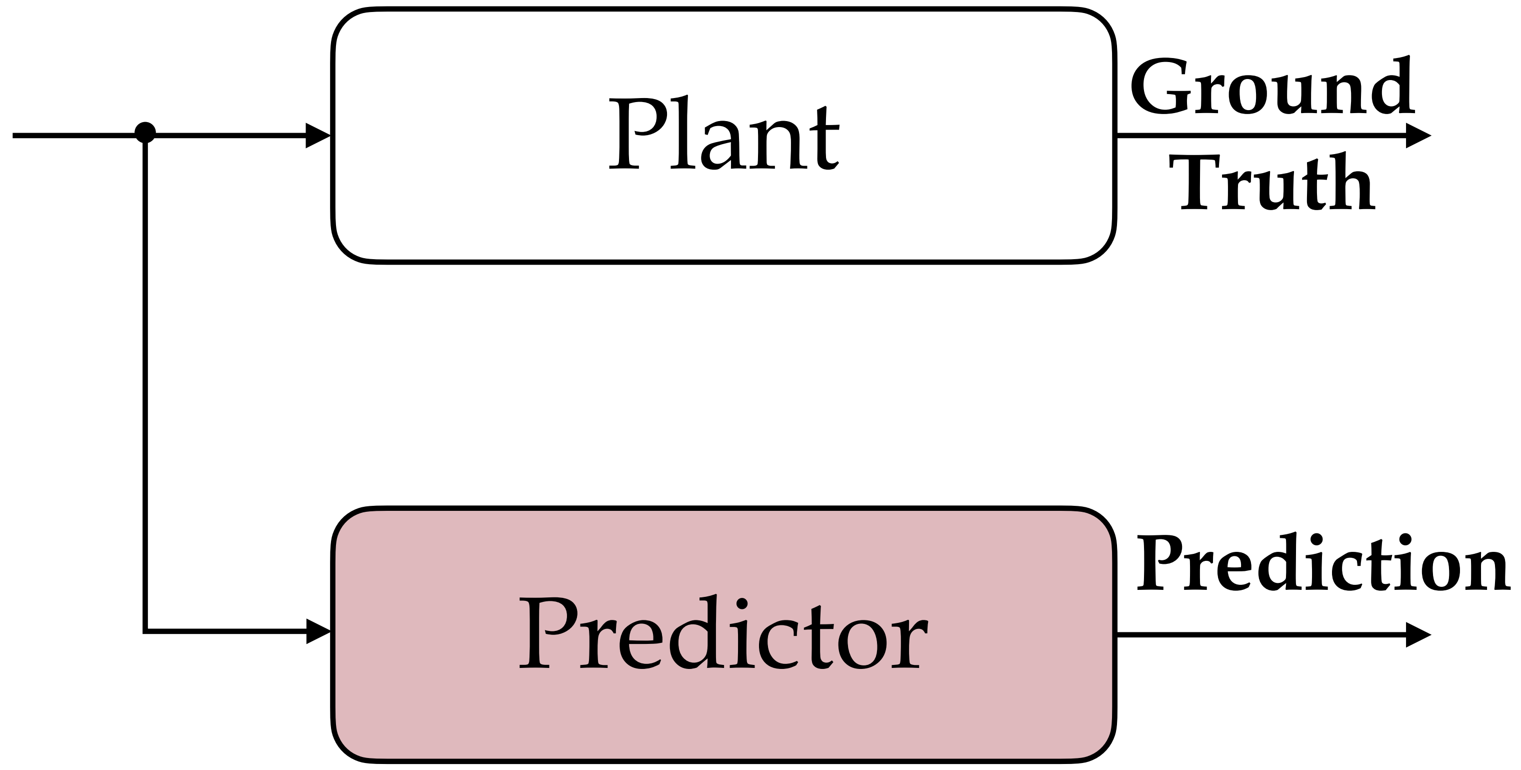
Example

- Recall: Infinite time LQR
  - System dynamics $$x_{k+1} = Ax_k + Bu_k$$
  - Cost function $$J = \frac{1}{2}\sum_{k=0}^{\infty}[x_k^T Q x_k + u_k^T R u_k]$$
  - Control law $$u = -[B^T P B + R]^{-1} B^T P A x$$

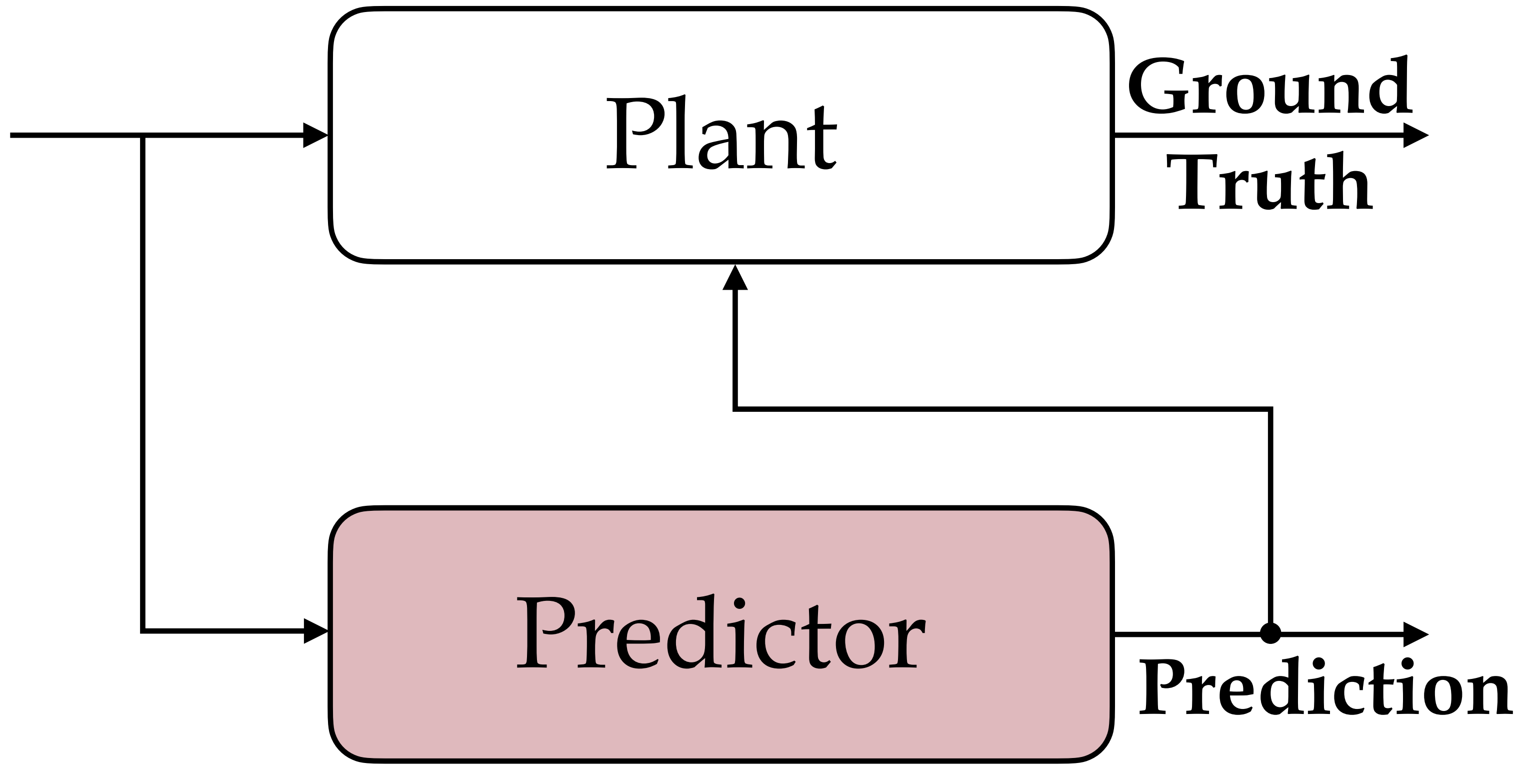
$$P = Q + A^T P A - A^T P B [B^T P B + R]^{-1} B^T P A$$

- The control law depends on the system parameters.

Control Problem vs Prediction Problem



Prediction problem



Control problem

- The predictor does not affect the plant.
- We only care about parameter convergence.

- The predictor affects the controller, hence the plant.
- Not only the parameter convergence, but also the closed-loop stability is important.

Offline Model Learning

$x_{k+1} = f(x_k)$

Historical Trajectory

$x_0^{(1)}, x_1^{(1)}, \ldots, x_T^{(1)},$

$x_0^{(2)}, \ldots,$

$\vdots$

→

Model Learning

$$\min_\theta \sum_{k,i} \mathcal{L}(f_\theta(x_k^{(i)}), x_{k+1}^{(i)})$$

→

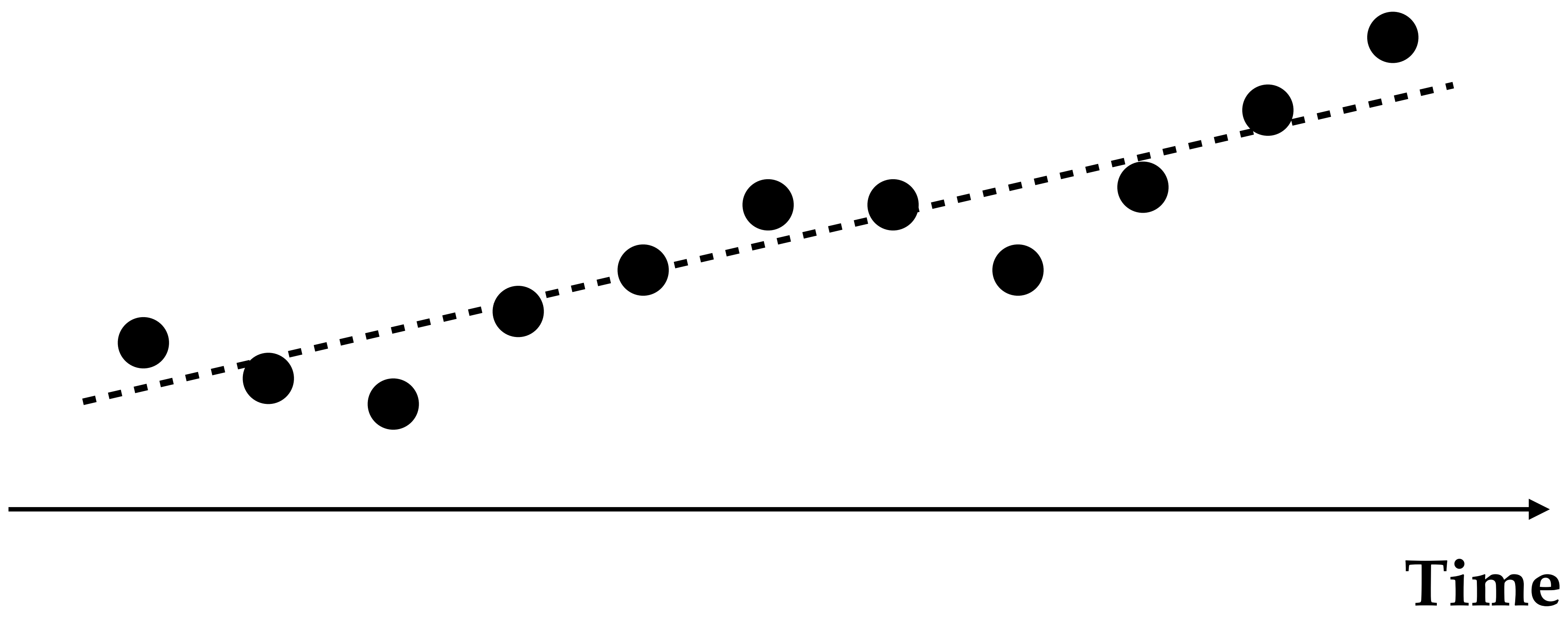
Prediction Model

f_θ

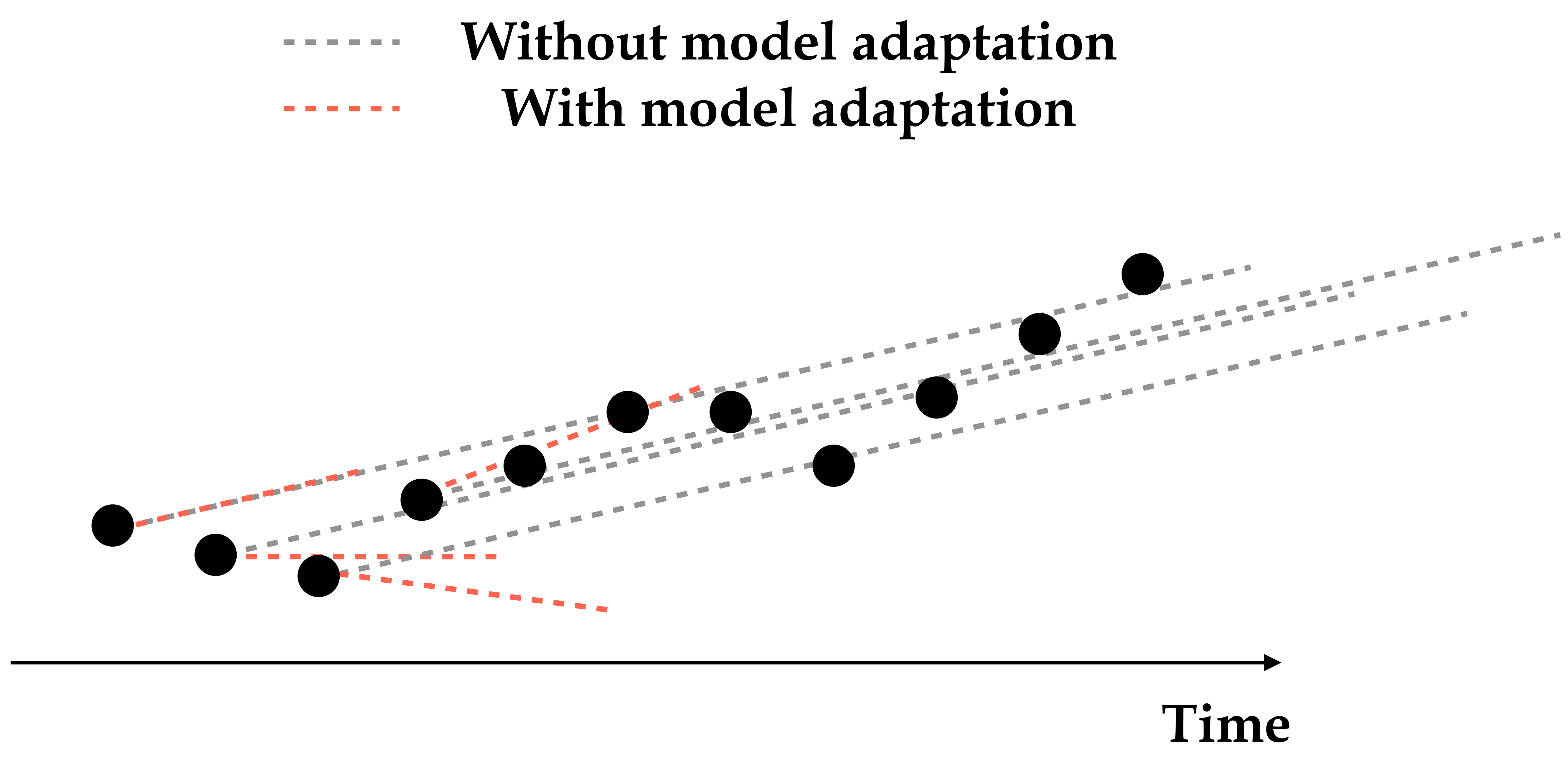
Model Learning vs. Model Adaptation

	Model Learning	**Model Adaptation**
Objective	Minimize error on batch	Minimize real-time prediction error
Purpose	Emphasize generalizability	Exploit local overfitting
Underlying assumptions	Homogeneity in data	Slow-changing dynamics

Model Learning vs. Model Adaptation



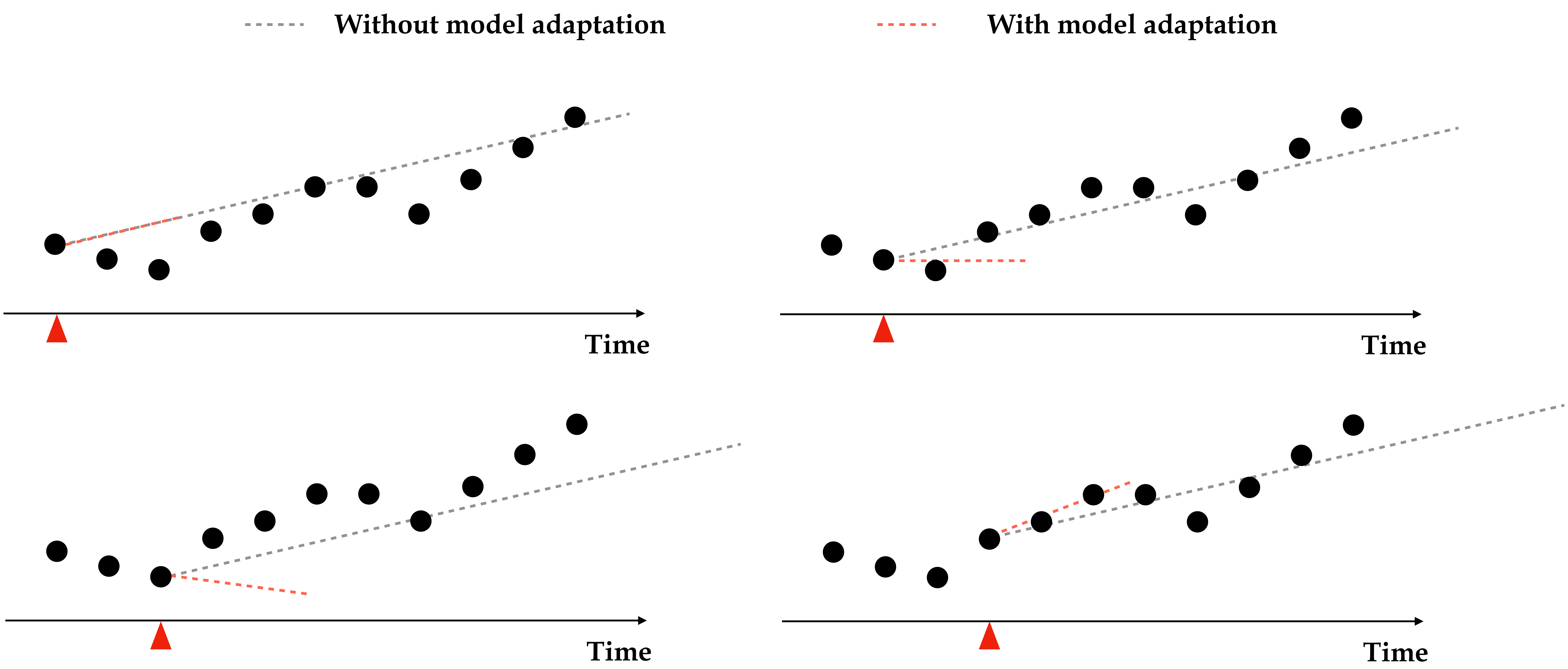
Model Learning



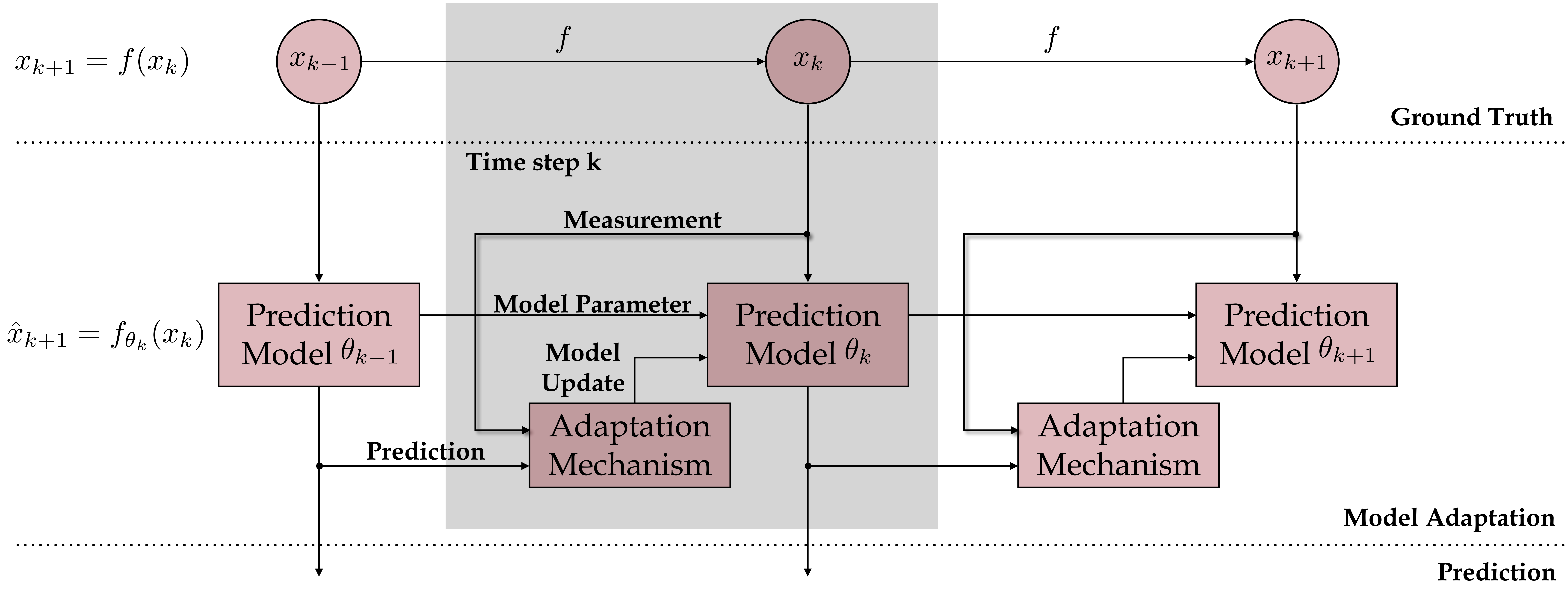
Model Adaptation

*** It is important to filter out noise during model adaptation**

Model Learning vs. Model Adaptation



Model Adaptation



Online Adaptation

- Problem Formulation
- Recursive Least Square
- Similarity with Kalman Filters

Recursive Least Square

- At time k, we want to have a minimum variance prediction (least square)

$$\min_{\theta} \mathbb{E}_{x_{k+1}} \|x_{k+1} - f_\theta(x_k)\|^2$$

- With slow dynamics assumption $\dot{\theta} \approx 0$, the parameter that results in smallest estimation error can be

$$\theta_k = \min_{\theta} L_k$$

$$L_k = \frac{1}{2}\|x_k - f_\theta(x_{k-1})\|^2 + \lambda L_{k-1} = \frac{1}{2}\sum_{i=0}^{k} \lambda^{k-i} \|x_i - f_\theta(x_{i-1})\|^2$$

- $\lambda \in [0, 1]$ is a forgetting factor.

Forgetting factor

- $\lambda = 0$, forget all past information.
  - Stability and robustness issues.
- $\lambda = 1$, memorize all past information.
  - Saturation problem.

Recursive Least Square

- Recursively compute the parameter estimate

$$\theta_k = \min_\theta L_k = \min_\theta \left\{ \frac{1}{2} \| x_k - f_\theta(x_{k-1}) \|^2 + \lambda L_{k-1} \right\}$$

- Need to find the parameter such that

$$\nabla_\theta L_k = -\nabla_\theta f_\theta(x_{k-1})^T [x_k - f_\theta(x_{k-1})] + \lambda \nabla_\theta L_{k-1} = 0$$

- Idea: find a relationship between θ_k and θ_{k-1}.

Recursive Least Square

$$\nabla_\theta L_k = -\underline{\nabla_\theta f_\theta(x_{k-1})}^T \underline{[x_k - f_\theta(x_{k-1})]} + \lambda \underline{\nabla_\theta L_{k-1}} = 0$$

- $$\begin{aligned} & x_k - f_{\theta_k}(x_{k-1}) \\ =& \underline{x_k - f_{\theta_{k-1}}(x_{k-1})} + \underline{f_{\theta_{k-1}}(x_{k-1}) - f_{\theta_k}(x_{k-1})} \\ \approx& \underline{e_k} + \underline{\underbrace{\nabla_{\theta|\theta_{k-1}} f_\theta(x_{k-1})}_{G_{k-1}}[\theta_{k-1} - \theta_k]} \end{aligned}$$

- $$\nabla_{\theta|\theta_k} f_\theta(x_{k-1}) \approx \underbrace{\nabla_{\theta|\theta_{k-1}} f_\theta(x_{k-1})}_{G_{k-1}}$$

  * First order approximation; similar to EKF

- $$\nabla_{\theta|\theta_k} L_{k-1} \approx \nabla_{\theta|\theta_{k-1}} L_{k-1} + \underbrace{\nabla^2_{\theta|\theta_{k-1}} L_{k-1}}_{H_{k-1}} \cdot [\theta_k - \theta_{k-1}] = H_{k-1}[\theta_k - \theta_{k-1}]$$

  Q: why $\nabla_{\theta|\theta_{k-1}} L_{k-1} = 0$ **?**

Recursive Parameter Update

$$\nabla_\theta L_k = -\nabla_\theta f_\theta(x_{k-1})^T [x_k - f_\theta(x_{k-1})] + \lambda \nabla_\theta L_{k-1} = 0$$

$$G_{k-1}^T [G_{k-1}(\theta_k - \theta_{k-1}) - e_k] + \lambda H_{k-1}(\theta_k - \theta_{k-1}) = 0$$

$$[G_{k-1}^T G_{k-1} + \lambda H_{k-1}](\theta_k - \theta_{k-1}) = G_{k-1}^T e_k$$

$$\theta_k = \theta_{k-1} + \underline{[G_{k-1}^T G_{k-1} + \lambda H_{k-1}]}^{-1} G_{k-1}^T \underline{e_k}$$

a priori prediction error

Update on Hessian

$$L_k = \frac{1}{2}\|x_k - f_\theta(x_{k-1})\|^2 + \lambda L_{k-1}$$

$$\begin{aligned}\nabla^2_{\theta|\theta_k} L_k &= \frac{1}{2}\nabla^2_{\theta|\theta_k}\|x_k - f_\theta(x_{k-1})\|^2 + \lambda \nabla^2_{\theta|\theta_k} L_{k-1} \\ &\approx [\nabla_{\theta|\theta_k} f_\theta(x_{k-1})]^T \nabla_{\theta|\theta_k} f_\theta(x_{k-1}) + \lambda \nabla^2_{\theta|\theta_k} L_{k-1} \\ &\approx [\nabla_{\theta|\theta_{k-1}} f_\theta(x_{k-1})]^T \nabla_{\theta|\theta_{k-1}} f_\theta(x_{k-1}) + \lambda \nabla^2_{\theta|\theta_{k-1}} L_{k-1}\end{aligned}$$

$$\Downarrow$$

$$H_k \approx G_{k-1}^T G_{k-1} + \lambda H_{k-1}$$

Recursive Least Square: Summary

Parameter update law $\theta_k = \theta_{k-1} + [H_k]^{-1} G_{k-1}^T e_k$

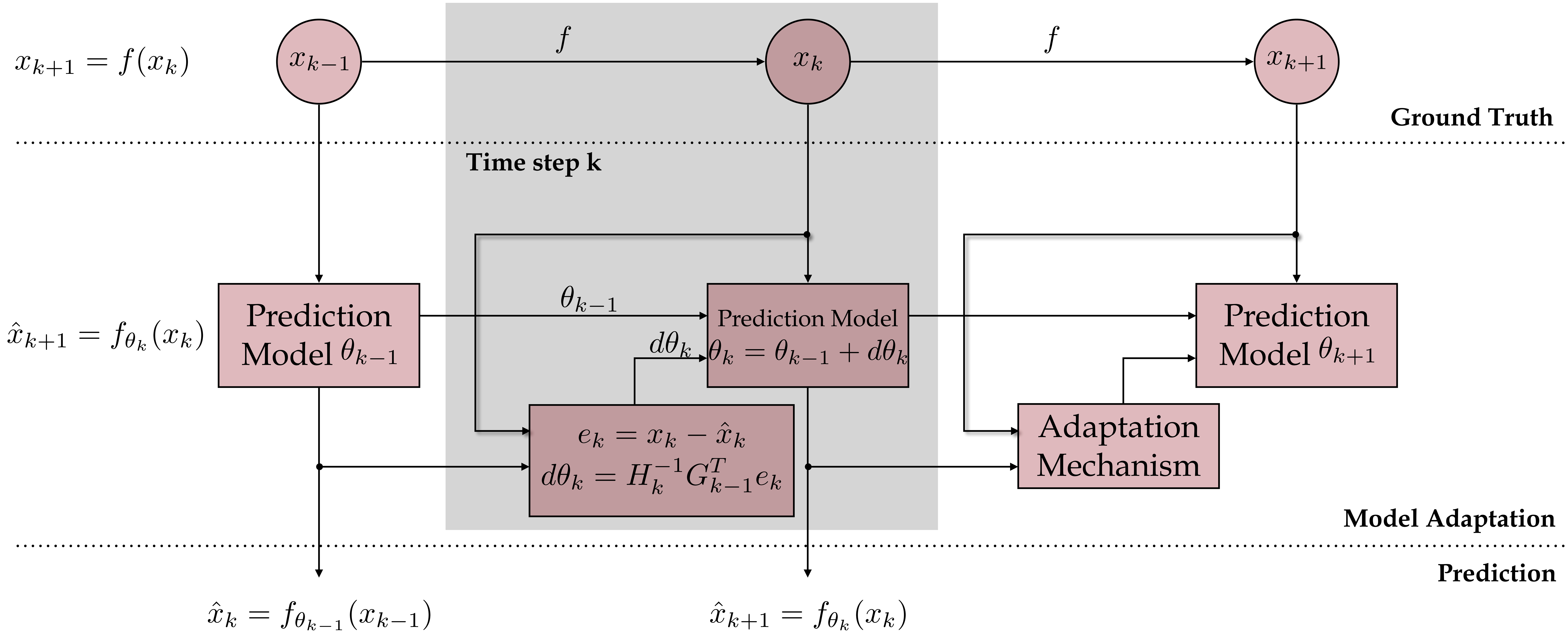
A priori prediction error $e_k = x_k - \hat{x}_k \qquad \hat{x}_k = f_{\theta_{k-1}}(x_{k-1})$

Gradient $G_{k-1} = \nabla_{\theta|\theta_{k-1}} f_\theta(x_{k-1})$

Hessian $H_k = \nabla^2_{\theta|\theta_k} L_k \qquad H_k = G_{k-1}^T G_{k-1} + \lambda H_{k-1}$

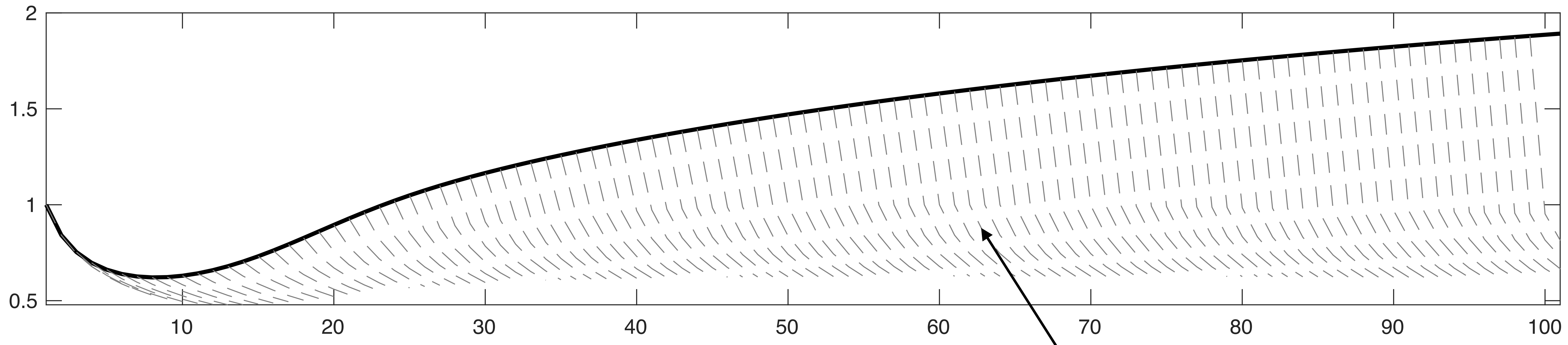
*** Note the similarity to Newton's update rule in optimization**

Model Adaptation



Example

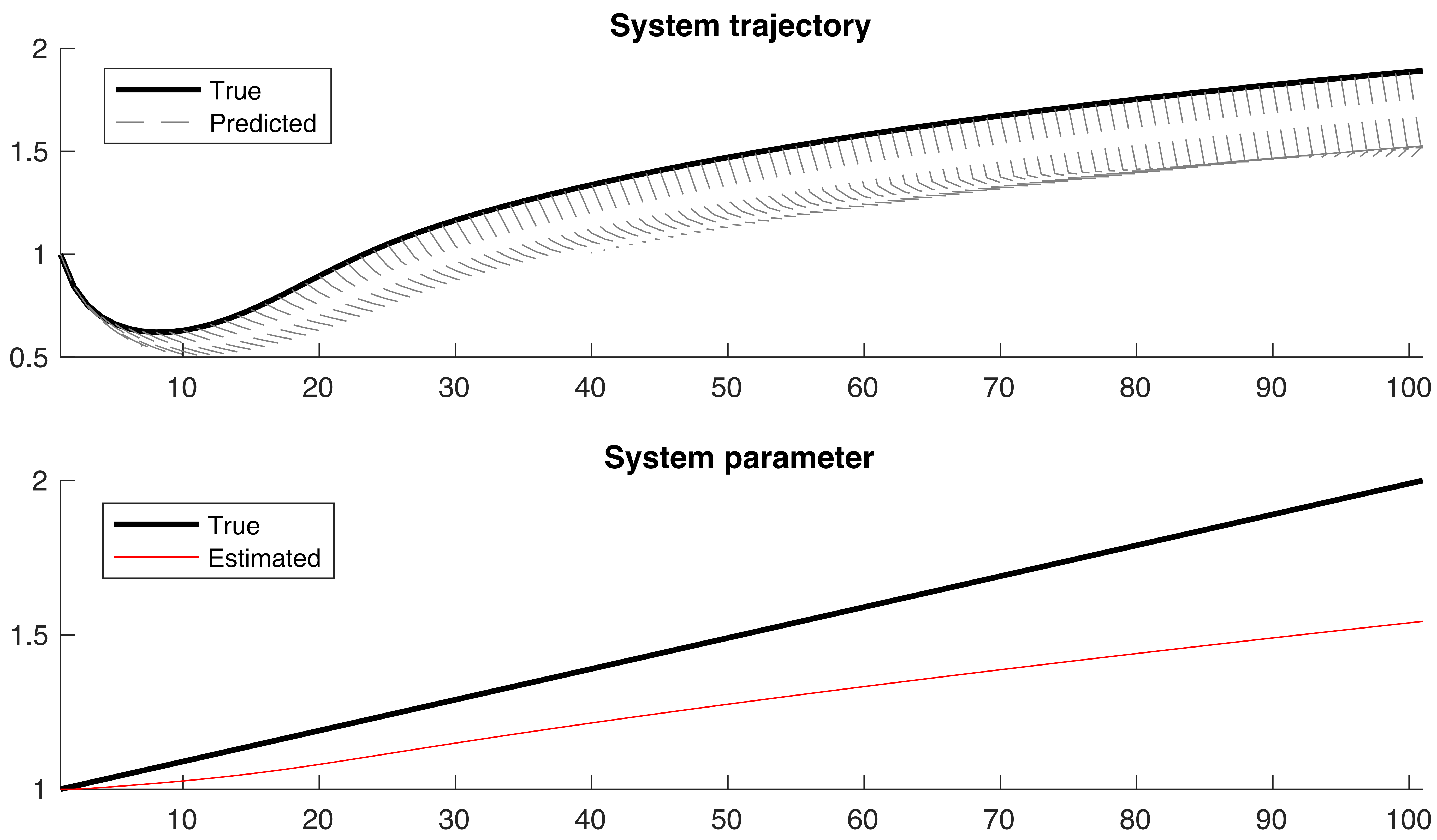
- Dynamics $x_{k+1} = a(k)\sin(x_k)$
- Time varying component $a(k) = 0.99 + 0.01k$



Predicted trajectory at different step assuming a = 1

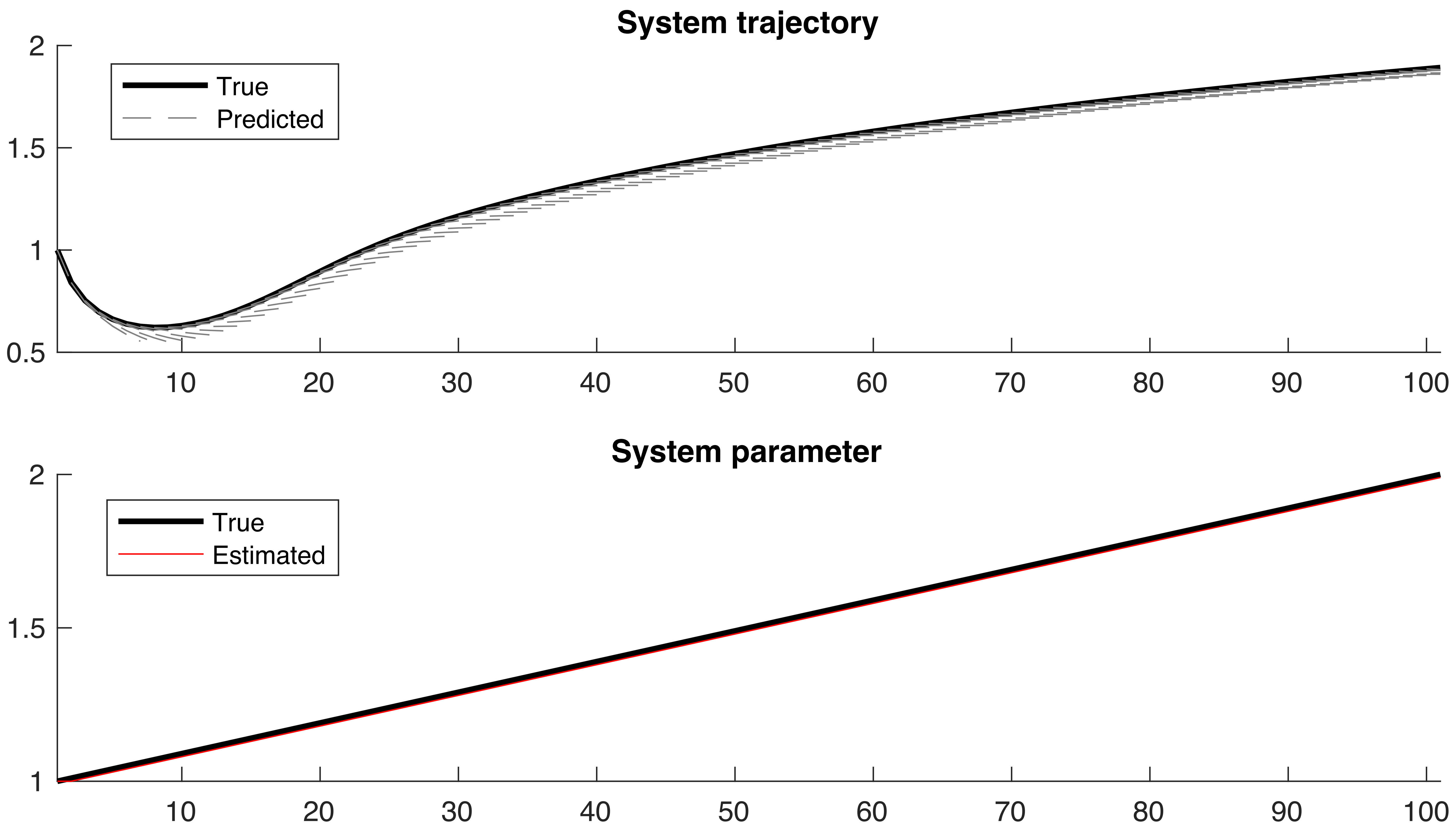
https://github.com/changliuliu/AdaptablePrediction

Example - With Adaptation 1



Forgetting factor = 1

Example - With Adaptation 2



Forgetting factor = 0

Remark

- The algorithm does not require that the model input and output are from the same source. The model can be generalized to be $y_{k+1} = f_\theta(x_k)$.
- There is no notion of "noise" in the derivation, unlike the Kalman filter.
- However, in order for the estimated parameter to converge to the true value (not stuck in local optima), it is necessary for the ground truth system to have persistent disturbances across all spectrum, e.g., white noise.

Recursive Least Square Linear Case

Nonlinear RLS	Linear RLS
$y_{k+1} = f_\theta(x_k)$	$y_{k+1} = \theta^T x_k$
$H_k = \nabla^2_{\theta\vert\theta_k} L_k$	$H_k = \sum_{i=0}^{k-1} \lambda^{k-1-i} x_i x_i^T$
$G_{k-1} = \nabla_{\theta\vert\theta_{k-1}} f_\theta(x_{k-1})$	$G_{k-1} = x_{k-1}^T$
$\theta_k = \theta_{k-1} + [H_k]^{-1} G_{k-1}^T e_k$	$\theta_k = \theta_{k-1} + \left[\sum_{i=0}^{k-1} \lambda^{k-1-i} x_i x_i^T\right]^{-1} x_{k-1} e_k$

For simplicity, assume

$y_{k+1} \in \mathbb{R}; \quad \theta, x_k \in \mathbb{R}^n$

Not an approximation!

Matrix Inverse Identity*

- Matrix Inverse Identity

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

- Inverse of Hessian $F_k := H_k^{-1}$

$$\begin{aligned} H_k^{-1} &= (G_{k-1}^T G_{k-1} + \lambda H_{k-1})^{-1} \\ &= [\lambda H_{k-1}]^{-1} - [\lambda H_{k-1}]^{-1} G_{k-1}^T [I + G_{k-1}[\lambda H_{k-1}]^{-1} G_{k-1}^T]^{-1} G_{k-1} [\lambda H_{k-1}]^{-1} \end{aligned}$$

$$\begin{aligned} F_k &= \frac{1}{\lambda}\left\{F_{k-1} - F_{k-1}G_{k-1}^T[\lambda I + G_{k-1}F_{k-1}G_{k-1}^T]^{-1}G_{k-1}F_{k-1}\right\} \\ &= \frac{1}{\lambda}\left\{F_{k-1} - \frac{F_{k-1}x_{k-1}x_{k-1}^T F_{k-1}}{\lambda + x_{k-1}^T F_{k-1} x_{k-1}}\right\} \end{aligned}$$

For linear case

Recursive Least Square Linear Case

$$y_{k+1} = \theta^T x_k$$

$$F_k = \left[\sum_{i=0}^{k-1} \lambda^{k-1-i} x_i x_i^T\right]^{-1}$$

Parameter update law $\quad \theta_k = \theta_{k-1} + F_k x_{k-1} e_k$

A priori prediction error $\quad e_k = y_k - \hat{y}_k \qquad \hat{y}_k = \theta_{k-1}^T x_{k-1}$

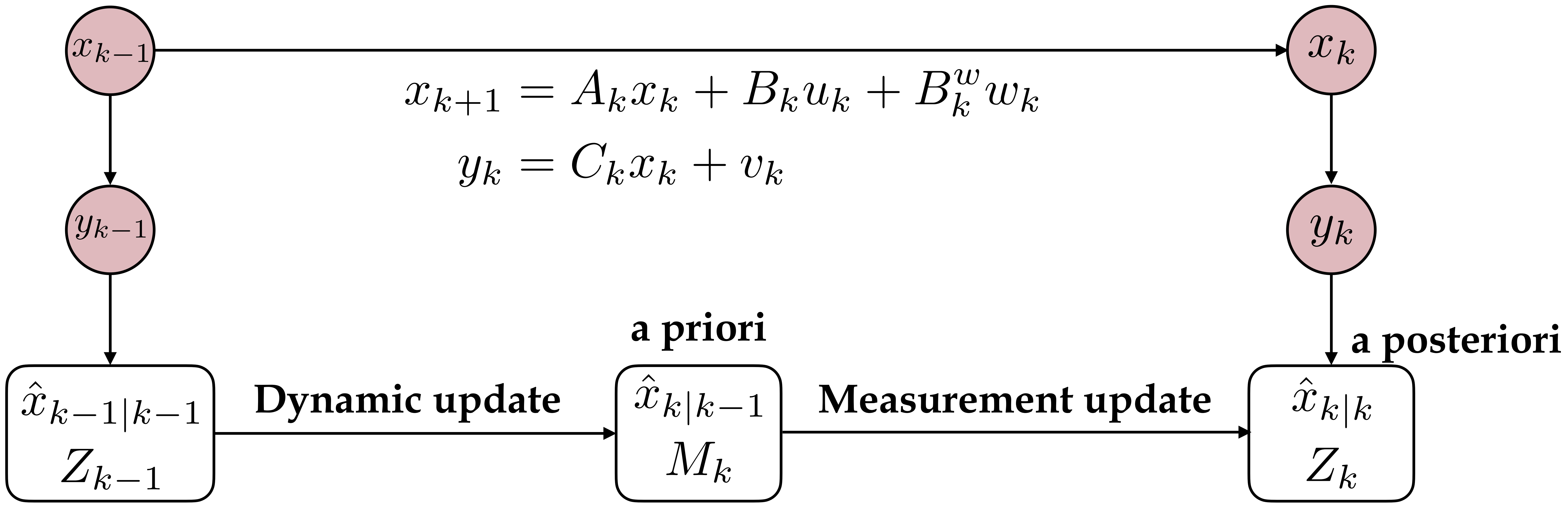
Learning gain $\quad F_k = \frac{1}{\lambda}\left\{F_{k-1} - \frac{F_{k-1} x_{k-1} x_{k-1}^T F_{k-1}}{\underline{\lambda + x_{k-1}^T F_{k-1} x_{k-1}}}\right\}$

This is easier to compute than the inverse of Hessian.

Online Adaptation

- Problem Formulation
- Recursive Least Square
- Similarity with Kalman Filters

Kalman Filter: Summary



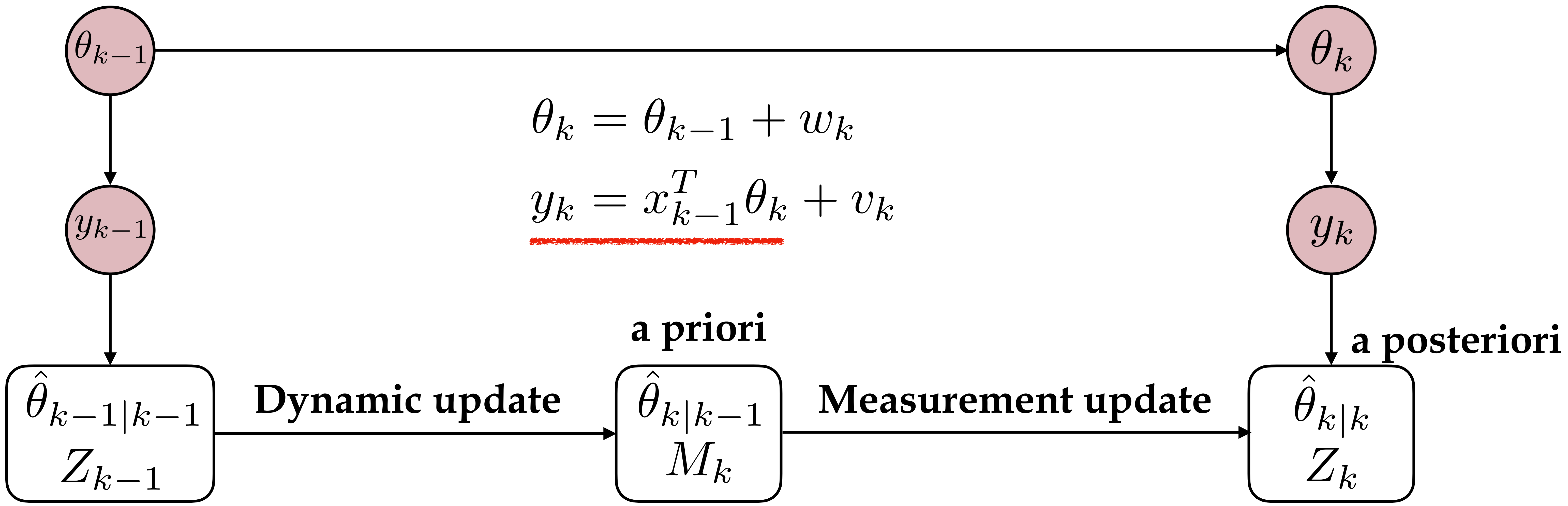
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \underbrace{M_k C_k^T [C_k M_k C_k^T + V_k]^{-1}}_{F_k}[y_k - C_k \hat{x}_{k|k-1}]$$

$$M_k = A_{k-1} Z_{k-1} A_{k-1}^T + B_{k-1}^w W_{k-1} (B_{k-1}^w)^T$$

$$Z_k = M_k - M_k C_k^T (C_k M_k C_k^T + V_k)^{-1} C_k M_k$$

Parameter Estimation with KF



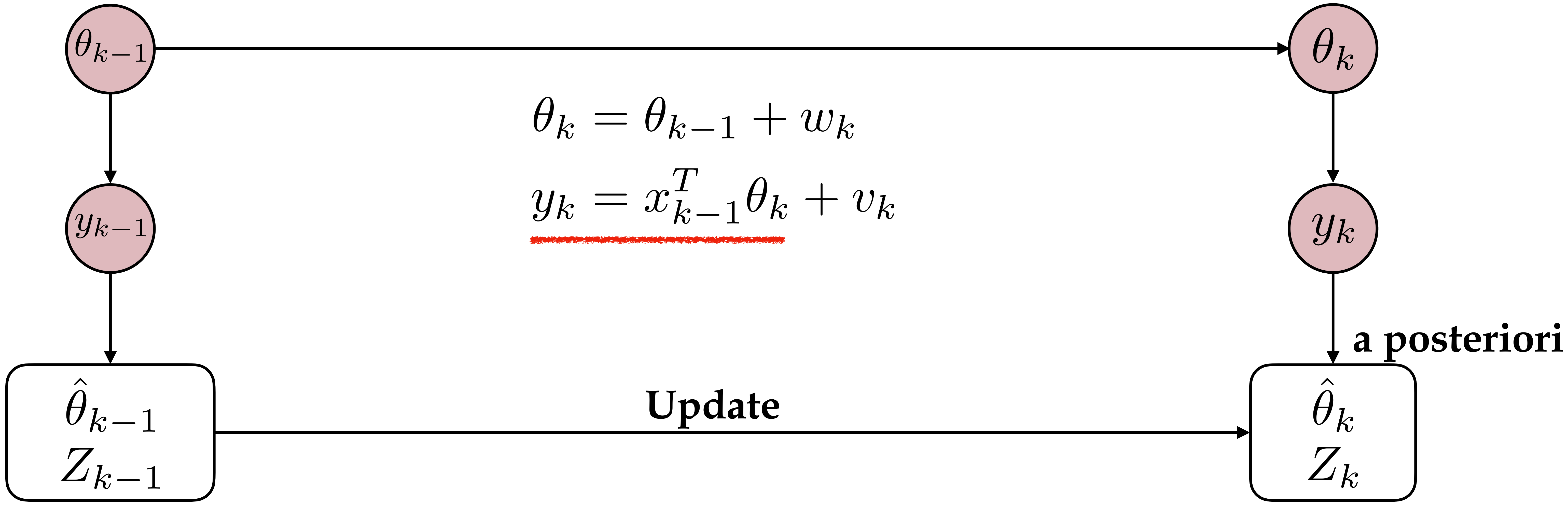
$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1}$$

$$\hat{\theta}_{k|k} = \hat{\theta}_{k|k-1} + M_k x_{k-1}[x_{k-1}^T M_k x_{k-1} + V_k]^{-1}[y_k - x_{k-1}^T \hat{\theta}_{k|k-1}]$$

$$M_k = Z_{k-1} + W_k$$

$$Z_k = M_k - M_k x_{k-1}(x_{k-1}^T M_k x_{k-1} + V_k)^{-1} x_{k-1}^T M_k$$

Parameter Estimation with KF



$$\hat{\theta}_k = \hat{\theta}_{k-1} + (Z_{k-1} + W_k) x_{k-1} [x_{k-1}^T (Z_{k-1} + W_k) x_{k-1} + V_k]^{-1} [y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$Z_k = (Z_{k-1} + W_k) - (Z_{k-1} + W_k) x_{k-1} (x_{k-1}^T (Z_{k-1} + W_k) x_{k-1} + V_k)^{-1} x_{k-1}^T (Z_{k-1} + W_k)$$

How is it related to the RLS algorithm?

Parameter Estimation with KF

KF Update $\hat{\theta}_k = \hat{\theta}_{k-1} + \underline{(Z_{k-1} + W_k)x_{k-1}[x_{k-1}^T(Z_{k-1} + W_k)x_{k-1} + V_k]^{-1}}[y_k - x_{k-1}^T\hat{\theta}_{k-1}]$

$$Z_k = (Z_{k-1} + W_k) - (Z_{k-1} + W_k)x_{k-1}(x_{k-1}^T(Z_{k-1} + W_k)x_{k-1} + V_k)^{-1}x_{k-1}^T(Z_{k-1} + W_k)$$

$$\begin{aligned}
Z_k x_{k-1} &= \underline{(Z_{k-1} + W_k)x_{k-1}} - \underline{(Z_{k-1} + W_k)x_{k-1}}(x_{k-1}^T(Z_{k-1} + W_k)x_{k-1} + V_k)^{-1}x_{k-1}^T(Z_{k-1} + W_k)x_{k-1} \\
&= (Z_{k-1} + W_k)x_{k-1}\left[I - (\underline{x_{k-1}^T(Z_{k-1} + W_k)x_{k-1}} + V_k)^{-1}\underline{x_{k-1}^T(Z_{k-1} + W_k)x_{k-1}}\right] \\
&= \underline{(Z_{k-1} + W_k)x_{k-1}(x_{k-1}^T(Z_{k-1} + W_k)x_{k-1} + V_k)^{-1}}V_k
\end{aligned}$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + Z_k x_{k-1} V_k^{-1}[y_k - x_{k-1}^T\hat{\theta}_{k-1}]$$

Parameter Estimation with KF

$$\hat{\theta}_k = \hat{\theta}_{k-1} + Z_k x_{k-1} V_k^{-1} [y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$Z_k = (Z_{k-1} + W_k) - (Z_{k-1} + W_k) x_{k-1} (x_{k-1}^T (Z_{k-1} + W_k) x_{k-1} + V_k)^{-1} x_{k-1}^T (Z_{k-1} + W_k)$$

Let $W_k = (\lambda_1 - 1) Z_{k-1}, V_k = \lambda_2 I$

$$Z_k = \lambda_1 Z_{k-1} - \lambda_1^2 Z_{k-1} x_{k-1} (\lambda_1 x_{k-1}^T Z_{k-1} x_{k-1} + \lambda_2 I)^{-1} x_{k-1}^T Z_{k-1}$$

Define $\bar{Z}_k = \dfrac{Z_k}{\lambda_2}$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k x_{k-1} [y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \bar{Z}_{k-1} x_{k-1} (x_{k-1}^T \bar{Z}_{k-1} x_{k-1} + \frac{1}{\lambda_1} I)^{-1} x_{k-1}^T \bar{Z}_{k-1} \right\}$$

KF vs RLS

Kalman Filter

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \frac{\bar{Z}_{k-1} x_{k-1} x_{k-1}^T \bar{Z}_{k-1}}{x_{k-1}^T \bar{Z}_{k-1} x_{k-1} + \frac{1}{\lambda_1}} \right\}$$

Recursive Least Square

$$\hat{\theta}_k = \hat{\theta}_{k-1} + F_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$F_k = \frac{1}{\lambda} \left\{ F_{k-1} - \frac{F_{k-1} x_{k-1} x_{k-1}^T F_{k-1}}{\lambda + x_{k-1}^T F_{k-1} x_{k-1}} \right\}$$

Conclusion 1:

The measurement noise will not affect the update rule of the learning gain if its covariance matrix is proportional to the identity matrix.

The measurement noise affects the initialization of the learning gain. The smaller the noise, the higher the gain.

KF vs RLS

Kalman Filter

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \frac{\bar{Z}_{k-1} x_{k-1} x_{k-1}^T \bar{Z}_{k-1}}{x_{k-1}^T \bar{Z}_{k-1} x_{k-1} + \frac{1}{\lambda_1}} \right\}$$

Recursive Least Square

$$\hat{\theta}_k = \hat{\theta}_{k-1} + F_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$F_k = \frac{1}{\lambda} \left\{ F_{k-1} - \frac{F_{k-1} x_{k-1} x_{k-1}^T F_{k-1}}{\lambda + x_{k-1}^T F_{k-1} x_{k-1}} \right\}$$

Conclusion 2:

The dynamic noise affects the update rule of the learning gain.

Moreover, the bigger the dynamic noise is, the smaller the forgetting factor should be, because previous estimates are less reliable.

KF vs RLS

Kalman Filter

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \frac{\bar{Z}_{k-1} x_{k-1} x_{k-1}^T \bar{Z}_{k-1}}{x_{k-1}^T \bar{Z}_{k-1} x_{k-1} + \frac{1}{\lambda_1}} \right\}$$

Recursive Least Square

$$\hat{\theta}_k = \hat{\theta}_{k-1} + F_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$F_k = \frac{1}{\lambda} \left\{ F_{k-1} - \frac{F_{k-1} x_{k-1} x_{k-1}^T F_{k-1}}{\lambda + x_{k-1}^T F_{k-1} x_{k-1}} \right\}$$

Conclusion 3:

We have the following equivalence between the design parameters of KF and the design parameters of RLS:

$$W_k V_k^{-1} = (\lambda^{-1} - 1) F_{k-1}.$$

Under the following assumptions:

$$W_k = (\lambda_1 - 1) Z_{k-1} \quad V_k = \lambda_2 I \quad \bar{Z}_k = \frac{Z_k}{\lambda_2}$$

KF vs RLS

Kalman Filter

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \frac{\bar{Z}_{k-1} x_{k-1} x_{k-1}^T \bar{Z}_{k-1}}{x_{k-1}^T \bar{Z}_{k-1} x_{k-1} + \frac{1}{\lambda_1}} \right\}$$

Recursive Least Square

$$\hat{\theta}_k = \hat{\theta}_{k-1} + F_k x_{k-1}[y_k - x_{k-1}^T \hat{\theta}_{k-1}]$$

$$F_k = \frac{1}{\lambda} \left\{ F_{k-1} - \frac{F_{k-1} x_{k-1} x_{k-1}^T F_{k-1}}{\lambda + x_{k-1}^T F_{k-1} x_{k-1}} \right\}$$

- Interpretation of the learning gain:
  - KF: the ratio between the error covariance of the a posteriori estimate and the covariance of the measurement noise
  - RLS: the inverse of the Hessian of the objective function

KF vs RLS

- Both KF and RLS are least square estimators.
- In general, KF offers more design freedoms than RLS
  - The two covariance matrices can be non-diagonal and can encode prior information.

EKF vs NRLS

- For general nonlinear cases, we just need to replace x_k with gradient G_k^T of the measurement function $y_{k+1} = f_\theta(x_k)$

Kalman Filter

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{Z}_k G_{k-1}^T [y_k - f_{\hat{\theta}_{k-1}}(x_{k-1})]$$

$$\bar{Z}_k = \lambda_1 \left\{ \bar{Z}_{k-1} - \bar{Z}_{k-1} G_{k-1}^T (G_{k-1} \bar{Z}_{k-1} G_{k-1}^T + \frac{1}{\lambda_1} I)^{-1} G_{k-1} \bar{Z}_{k-1} \right\}$$

Recursive Least Square

$$\hat{\theta}_k = \hat{\theta}_{k-1} + F_k G_{k-1}^T [y_k - f_{\hat{\theta}_{k-1}}(x_{k-1})]$$

$$F_k = \frac{1}{\lambda} \{ F_{k-1} - F_{k-1} G_{k-1}^T [\lambda I + G_{k-1} F_{k-1} G_{k-1}^T]^{-1} G_{k-1} F_{k-1} \}$$

Algorithm Initialization

- Initial distribution $\theta_0 \sim \mathcal{N}(\bar{\theta}, \Sigma)$
- KF:

$$\hat{\theta}_0 = \bar{\theta}$$
$$Z_0 = \Sigma$$

- RLS:

$$\hat{\theta}_0 = \bar{\theta}$$
$$F_0 \to \infty$$

Online Adaptation

- Problem Formulation
- Recursive Least Square
- Similarity with Kalman Filters