Asst. Prof. Tobias K. S. Ritschel
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Prof. John Wyller
Department of Mathematics, Faculty of Science and Technology
Norwegian University of Life Sciences

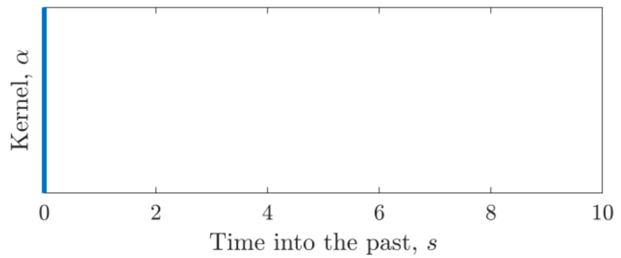
Estimation of absolute and distributed time delays in differential equations based on the linear chain trick

11.30 – 11.50, Friday, August 18th, 2023
24th Nordic Process Control Workshop
Norwegian University of Science and Technology

What are distributed delay differential equations?

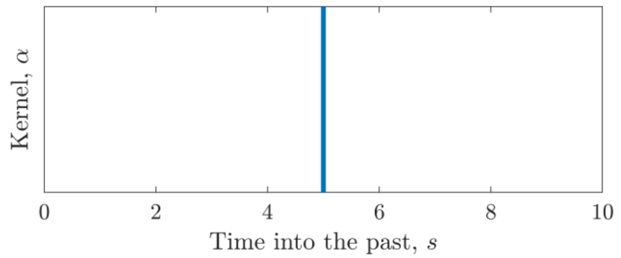
Ordinary differential equation (ODE)

$$\dot{x}(t) = f(x(t), p)$$



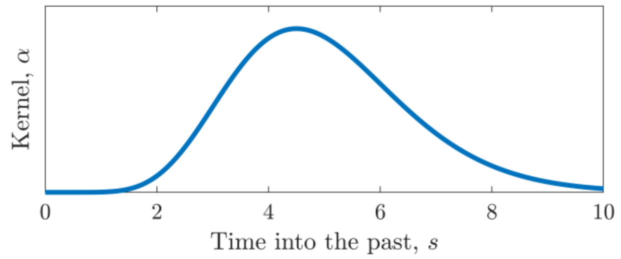
Delay differential equation (DDE) w. absolute/discrete delay

$$\dot{x}(t) = f(x(t), z(t), p)$$
$$z(t) = x(t - \tau)$$



Distributed delay differential equation (DDDE), i.e., w. distributed/continuous delay

$$\dot{x}(t) = f(x(t), z(t), p)$$
$$z(t) = \int_{-\infty}^{t} \alpha(t - s) x(s) \, \mathrm{d}s$$



System

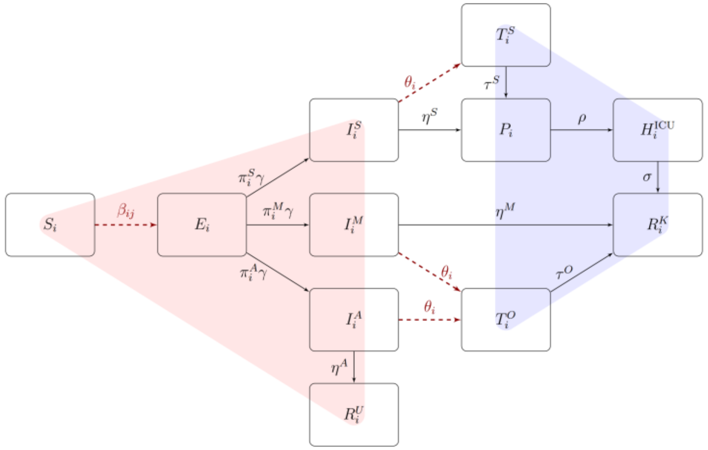
Distributed delay differential equation (DDDE), i.e., w. distributed/continuous delay

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), z(t), p) \\
z(t) &= \int_{-\infty}^{t} \alpha(t-s) r(s) \, \mathrm{d}s \\
r(t) &= h(x(t), p) \\
y(t_k) &= g(x(t_k), p)
\end{aligned}
$$

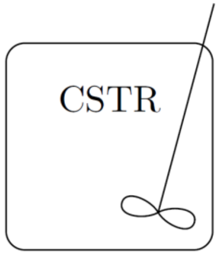
Why are we interested in delays?

Transport through pipes

Epidemiology (latency period)



Slow mixing



Why are we interested in distributed delays?

1. approximate the kernel

$$\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{z}(t), p)$$
$$\hat{z}(t) = \int_{-\infty}^{t} \hat{\alpha}(t-s)\hat{r}(s)\,\mathrm{d}s$$
$$\hat{r}(s) = h(\hat{x}(s), p)$$

$$\hat{\alpha}(t) = \sum_{m=0}^{M} c_m \hat{\alpha}_m(t)$$
$$\hat{\alpha}_m(t) = b_m t^m e^{-at},$$
$$b_m = \frac{a^{m+1}}{m!}$$

2. transform to a set of ODEs (linear chain trick)

$$\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{z}(t), p)$$
$$\dot{\hat{Z}}(t) = A\hat{Z}(t) + B\hat{r}(t)$$
$$\hat{z}(t) = C\hat{Z}(t)$$
$$\hat{r}(t) = h(\hat{x}(t), p)$$

$$A = a\begin{bmatrix} -I & & & \\ I & -I & & \\ & \ddots & \ddots & \\ & & I & -I \end{bmatrix} \qquad B = a\begin{bmatrix} I \\ \\ \\ \\ \end{bmatrix}$$

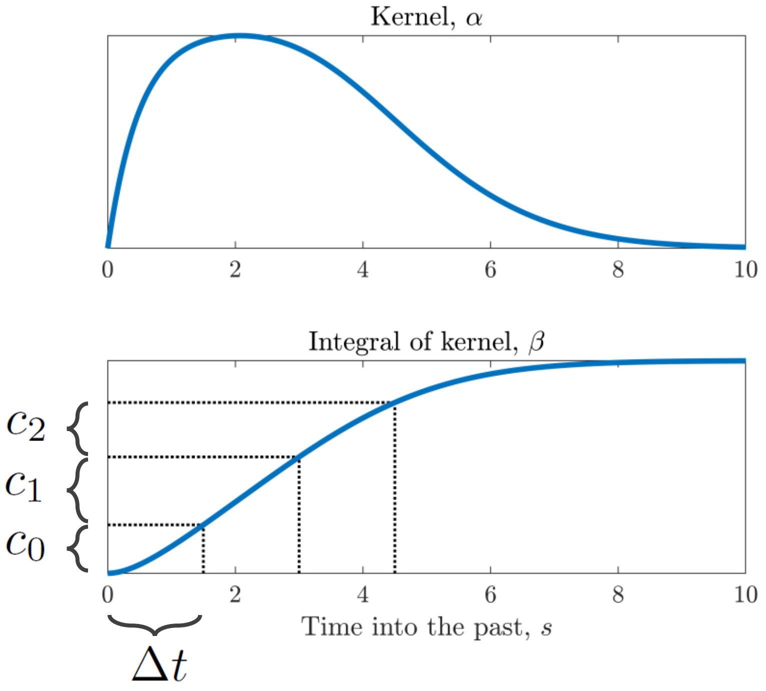
$$C = \begin{bmatrix} c_0 I & c_1 I & \dots & c_M I \end{bmatrix}$$

Is it justified?

PDF and CDF of Erlang distribution

$$\hat{\alpha}_m(t) = b_m t^m e^{-at},$$

$$\hat{\beta}_m(t) = \int_0^t \hat{\alpha}_m(s)\,\mathrm{d}s = 1 - \frac{1}{a}\sum_{n=0}^{m} \hat{\alpha}_n(t)$$



Theorem 1 *Let $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ be a kernel that satisfies (3), and define*

$$\beta(t) = \int_0^t \alpha(s)\,\mathrm{d}s \tag{6}$$

as well as

$$\hat{\beta}^{(M)}(t) = \sum_{m=0}^{M} c_m \hat{\beta}_m(t), \tag{7}$$

where $t_{m+1} - t_m = \Delta t$, $a = 1/\Delta t$, and $c_m = \beta(t_{m+1}) - \beta(t_m)$. Note that $\hat{\beta}_m$ depends on the rate parameter, a. Then,

$$\lim_{\Delta t \to 0} \hat{\beta}^{(\infty)}(t) = \beta(t) \tag{8}$$

for any continuity point, $t \in \mathbb{R}_+$, of β.

H. C. Tijms. *Stochastic models: An algorithmic approach.* Wiley Series in Probability and Statistics. Wiley, 1995.

H. Cossette, D. Landriault, E. Marceau, and K. Moutanabbir. Moment-based approximation with mixed Erlang distributions. *Variance*, 10(1):166–182, 2016.

Estimation of model and time-delay parameters

Dynamical least-squares problem

$$\min_{p,q} \quad \phi = \frac{1}{2}\sum_{k=0}^{N}(y(t_k) - \hat{y}(t_k))^T(y(t_k) - \hat{y}(t_k))$$

subject to

$$\begin{aligned}
\hat{x}(t_0) &= \hat{x}_0(p), \\
\hat{Z}(t_0) &= \hat{Z}_0(p), \\
\dot{\hat{x}}(t) &= f(\hat{x}(t), \hat{z}(t), p), && t \in [t_0, t_N], \\
\dot{\hat{Z}}(t) &= A(q)\hat{Z}(t) + B(q)\hat{r}(t), && t \in [t_0, t_N], \\
\hat{z}(t) &= C(q)\hat{Z}(t), && t \in [t_0, t_N], \\
\hat{r}(t) &= h(\hat{x}(t), p), && t \in [t_0, t_N], \\
\hat{y}(t_k) &= g(\hat{x}(t_k), p), && k = 0, \ldots, N, \\
e^T q &= 1, \\
p_{\min} &\leq p \leq p_{\max}, \\
q_{\min} &\leq q \leq q_{\max}
\end{aligned}$$

Assumption on initial condition

$$\hat{Z}(t_0) = \left[\hat{z}_0(t_0); \; \cdots \; ; \; \hat{z}_M(t_0)\right]$$

$$\begin{aligned}
\hat{z}_m(t_0) &= \int_{-\infty}^{t_0} \hat{\alpha}_m(t_0 - s)\hat{r}(s)\,\mathrm{d}s \\
&= \hat{r}(t_0)
\end{aligned}$$

$$\hat{r}(t_0) = h(\hat{x}_0, p)$$

Single-shooting approach

Single-shooting objective function

$$\psi = \psi(p,q)$$
$$= \left\{ \phi = \frac{1}{2} \sum_{k=0}^{N} (y(t_k) - \hat{y}(t_k))^T (y(t_k) - \hat{y}(t_k)) : \right.$$

$$\left. \begin{aligned}
\hat{x}(t_0) &= x_0(p), \\
\hat{Z}(t_0) &= \hat{Z}_0(p), \\
\dot{\hat{x}}(t) &= f(\hat{x}(t), \hat{z}(t), p), \quad t \in [t_0, t_N], \\
\dot{\hat{Z}}(t) &= A(q)\hat{Z}(t) + B(q)\hat{r}(t), \quad t \in [t_0, t_N], \\
\hat{z}(t) &= C(q)\hat{Z}(t), \quad t \in [t_0, t_N], \\
\hat{r}(t) &= h(\hat{x}(t), p), \quad t \in [t_0, t_N], \\
\hat{y}(t_k) &= g(\hat{x}(t_k), p), \quad k = 0, \ldots, N
\end{aligned} \right\}$$

Single-shooting nonlinear program (NLP)

$$\begin{aligned}
\min_{p,q} \quad & \psi(p,q), \\
\text{subject to} \quad & e^T q = 1, \\
& p_{\min} \leq p \leq p_{\max}, \\
& q_{\min} \leq q \leq q_{\max}
\end{aligned}$$

- We use fmincon to solve this
- We provide gradients (but not Hessian)

- We use ode45 for nonstiff systems
- We use ode15s for stiff systems

Sensitivities

1. We derive the Jacobian of the right-hand side function
2. Can be parallellized ✗
3. **Sparsity can be exploited** ✓

$$\frac{\partial \psi}{\partial p_i}(p,q) = \sum_{k=0}^{N} (y(t_k) - \hat{y}(t_k)) S_{yp,i}(t_k)$$

$$\frac{\partial \psi}{\partial q_i}(p,q) = \sum_{k=0}^{N} (y(t_k) - \hat{y}(t_k)) S_{yq,i}(t_k)$$

$$\begin{aligned}
S_{xp,i}(t_0) &= \frac{\partial \hat{x}_0}{\partial p_i}(p), \\
S_{Zp,i}(t_0) &= \frac{\partial \hat{Z}_0}{\partial p_i}(p), \\
\dot{S}_{xp,i}(t) &= \frac{\partial f}{\partial x} S_{xp,i}(t) + \frac{\partial f}{\partial z} S_{zp,i}(t) + \frac{\partial f}{\partial p}, \\
\dot{S}_{Zp,i}(t) &= A(q) S_{Zp,i}(t) + B(q) S_{rp,i}(t), \\
S_{zp,i}(t) &= C(q) S_{Zp,i}(t), \\
S_{rp,i}(t) &= \frac{\partial h}{\partial x} S_{xp,i}(t) + \frac{\partial h}{\partial p}, \\
S_{yp,i}(t_k) &= \frac{\partial g}{\partial x} S_{xp,i}(t_k) + \frac{\partial g}{\partial p}
\end{aligned}$$

$$\begin{aligned}
S_{xq,i}(t_0) &= 0, \\
S_{Zq,i}(t_0) &= 0, \\
\dot{S}_{xq,i}(t) &= \frac{\partial f}{\partial x} S_{xq,i}(t) + \frac{\partial f}{\partial z} S_{zq,i}(t), \\
\dot{S}_{Zq,i}(t) &= \frac{\partial A}{\partial q_i} \hat{Z}(t) + A(q) S_{Zq,i}(t) \\
&\quad + \frac{\partial B}{\partial q_i} \hat{r}(t) + B(q) S_{rq,i}(t), \\
S_{zq,i}(t) &= \frac{\partial C}{\partial q_i} \hat{Z}(t) + C(q) S_{Zq,i}(t), \\
S_{rq,i}(t) &= \frac{\partial h}{\partial x} S_{xq,i}(t), \\
S_{yq,i}(t_k) &= \frac{\partial g}{\partial x} S_{xq,i}(t_k)
\end{aligned}$$

Point reactor kinetics model (nuclear fission)

$$\dot{n}(t) = \frac{\rho(t) - \beta}{\Lambda} n(t) + \sum_{i=1}^{N_p} \lambda_i C_i,$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} n(t) - \lambda_i C_i(t) - \frac{C_i}{\tau_c} + \frac{\tilde{C}_i(t) e^{-\lambda_i \tau_\ell}}{\tau_c},$$

$$\dot{\rho}(t) = -\alpha H n(t)$$

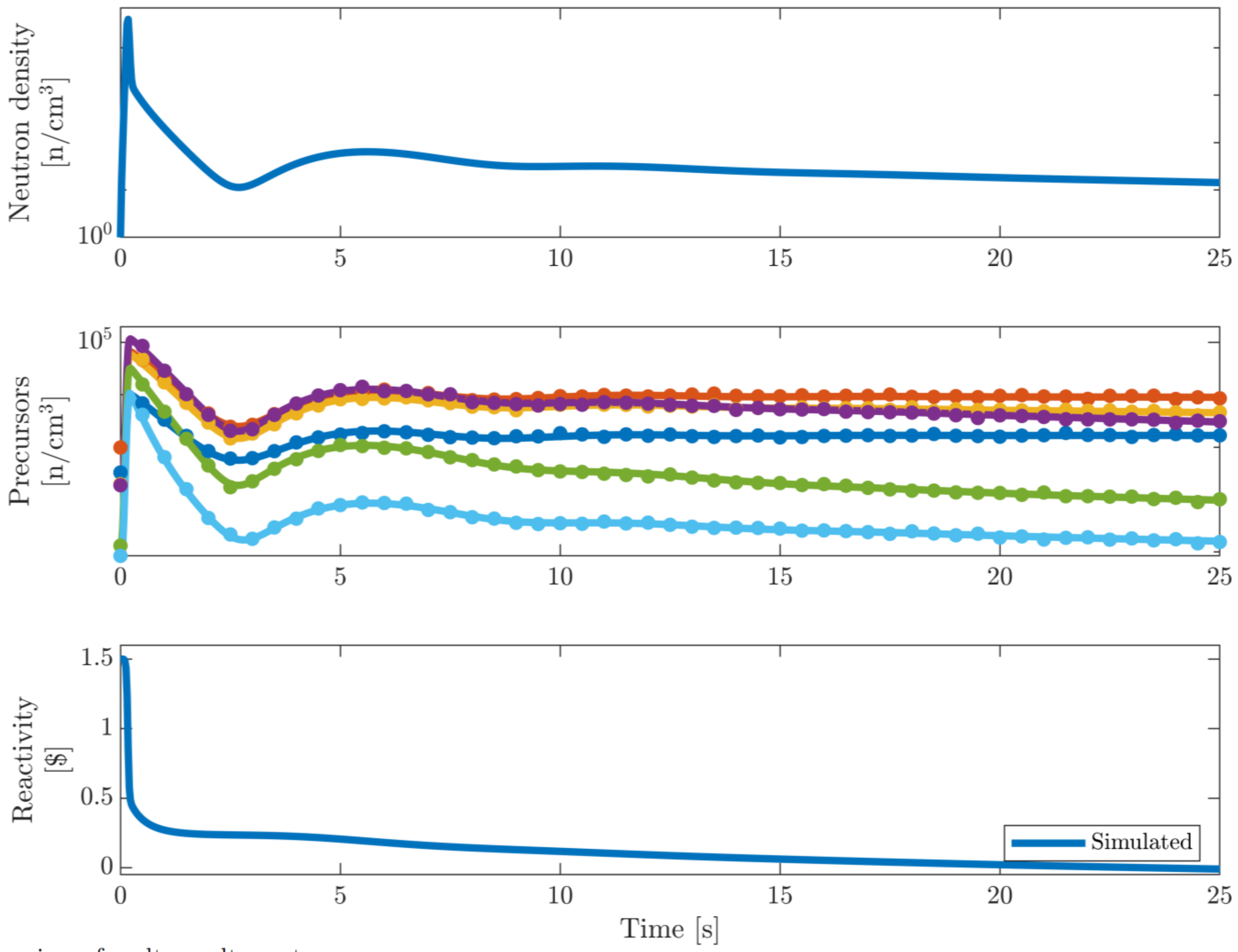
$$\tilde{C}_i(t) = \int_{-\infty}^{t} \alpha(t-s) C_i(s) \, \mathrm{d}s$$

$$N_p = 6$$

$$\beta = \textstyle\sum_{i=1}^{N_p} \beta_i$$

- Estimate model parameter α and kernel α

(U235-graphite reactor)

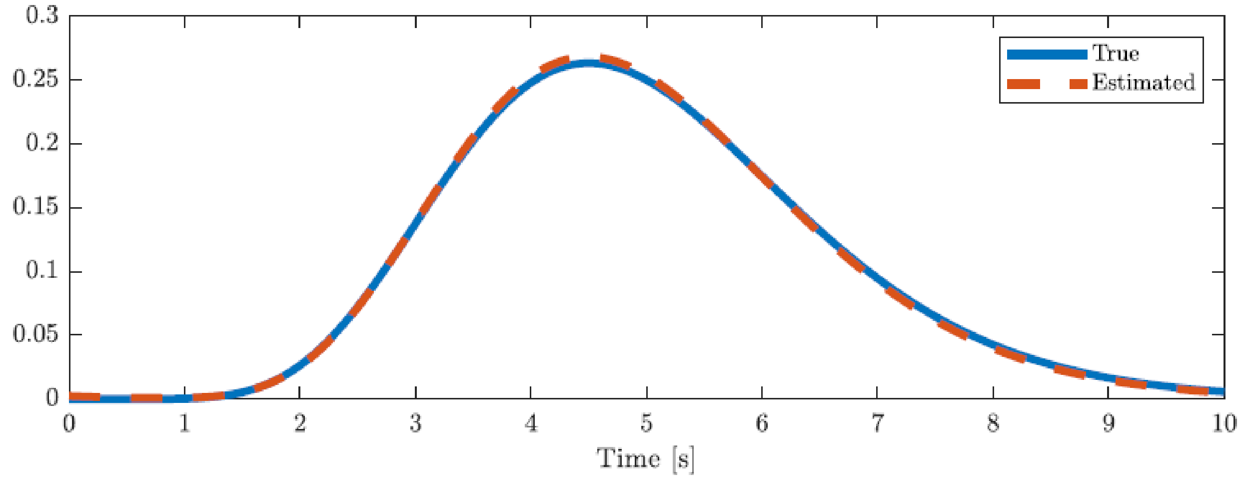


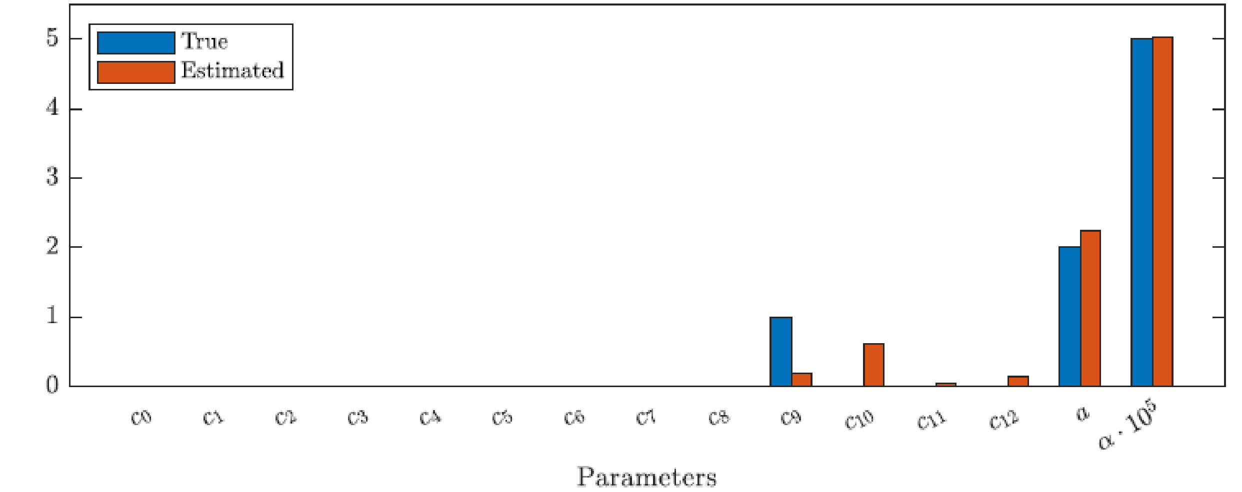
W. M. Stacey. *Nuclear reactor physics.* Wiley, 2nd edition, 2007.

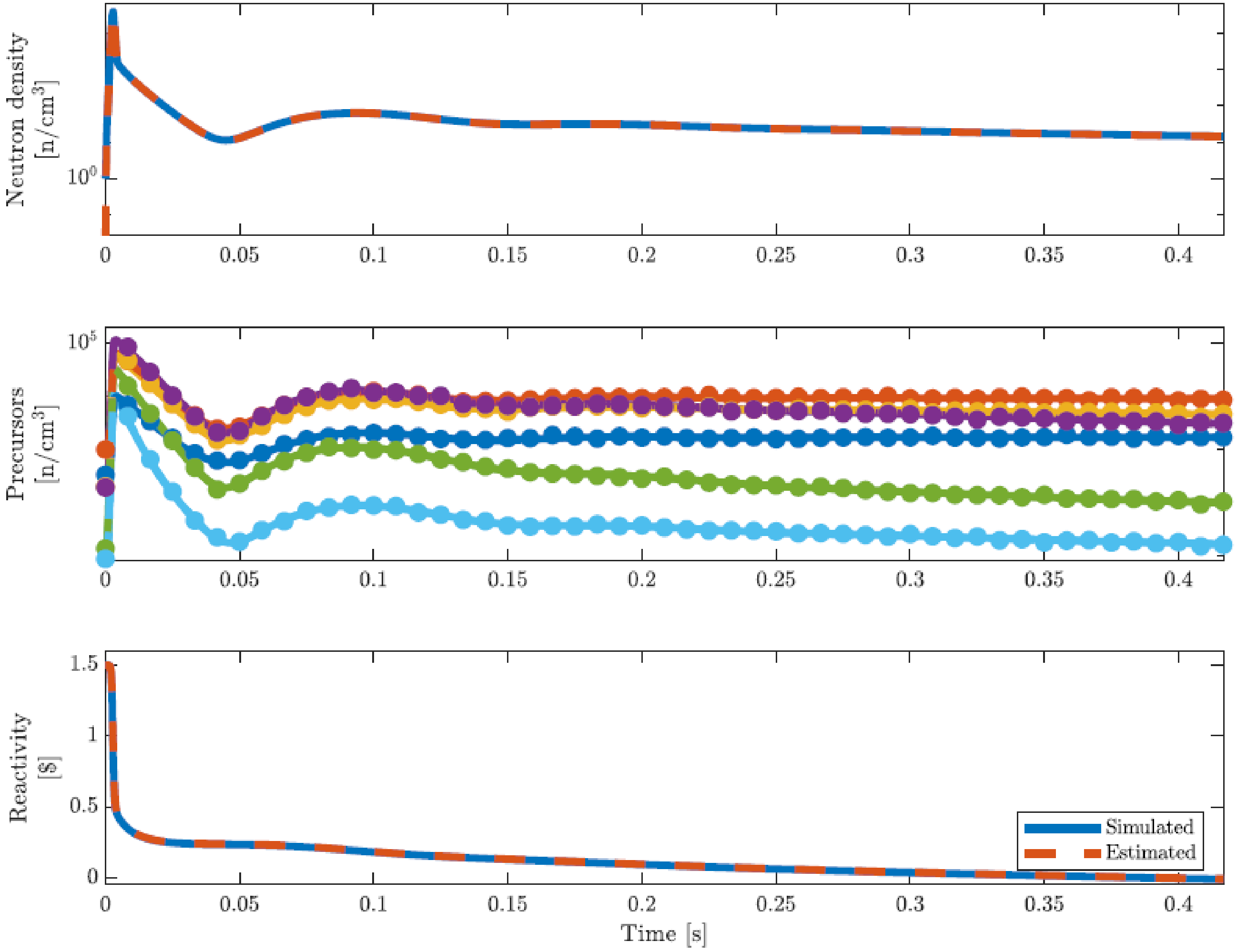
S. Q. B. Leite, M. T. de Vilhena, and B. E. J. Bodmann. Solution of the point reactor kinetics equations with temperature feedback by the ITS2 method. *Progress in Nuclear Energy*, 91:240–249, 2016.

D. Wooten and J. J. Powers. A review of molten salt reactor kinetics models. *Nuclear Science and Engineering*, 191:203–230, 2018.

Point reactor kinetics – Noisy measurements

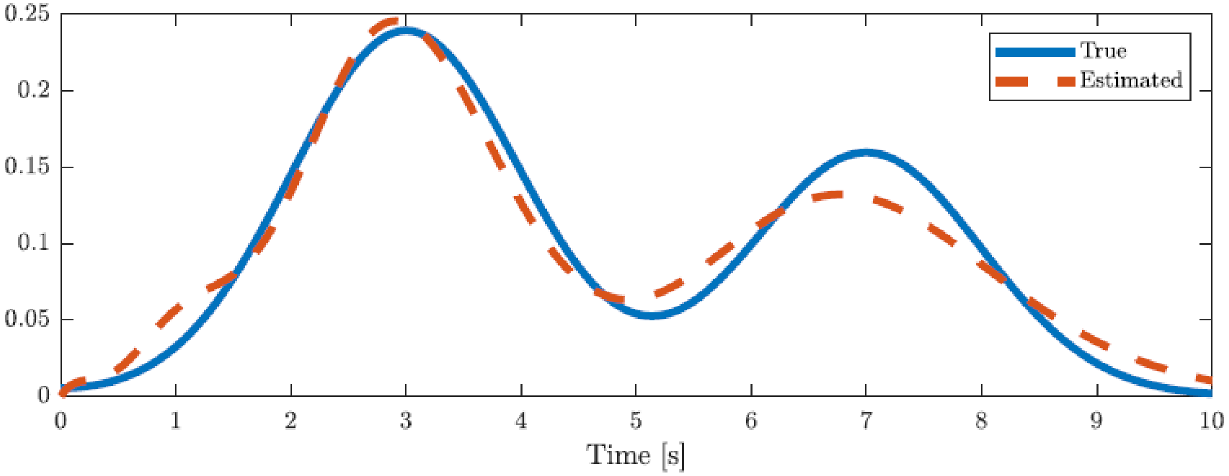


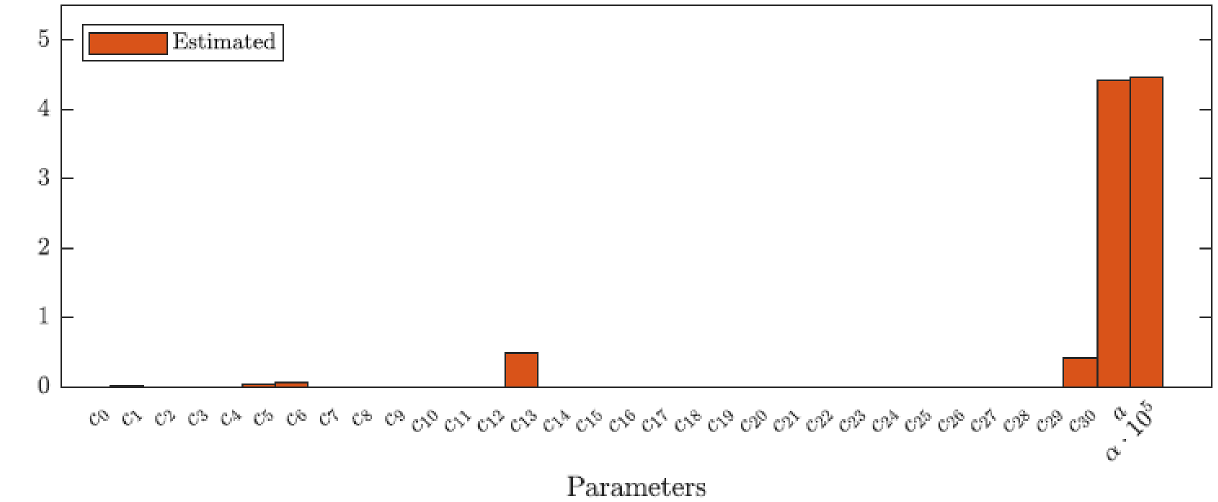


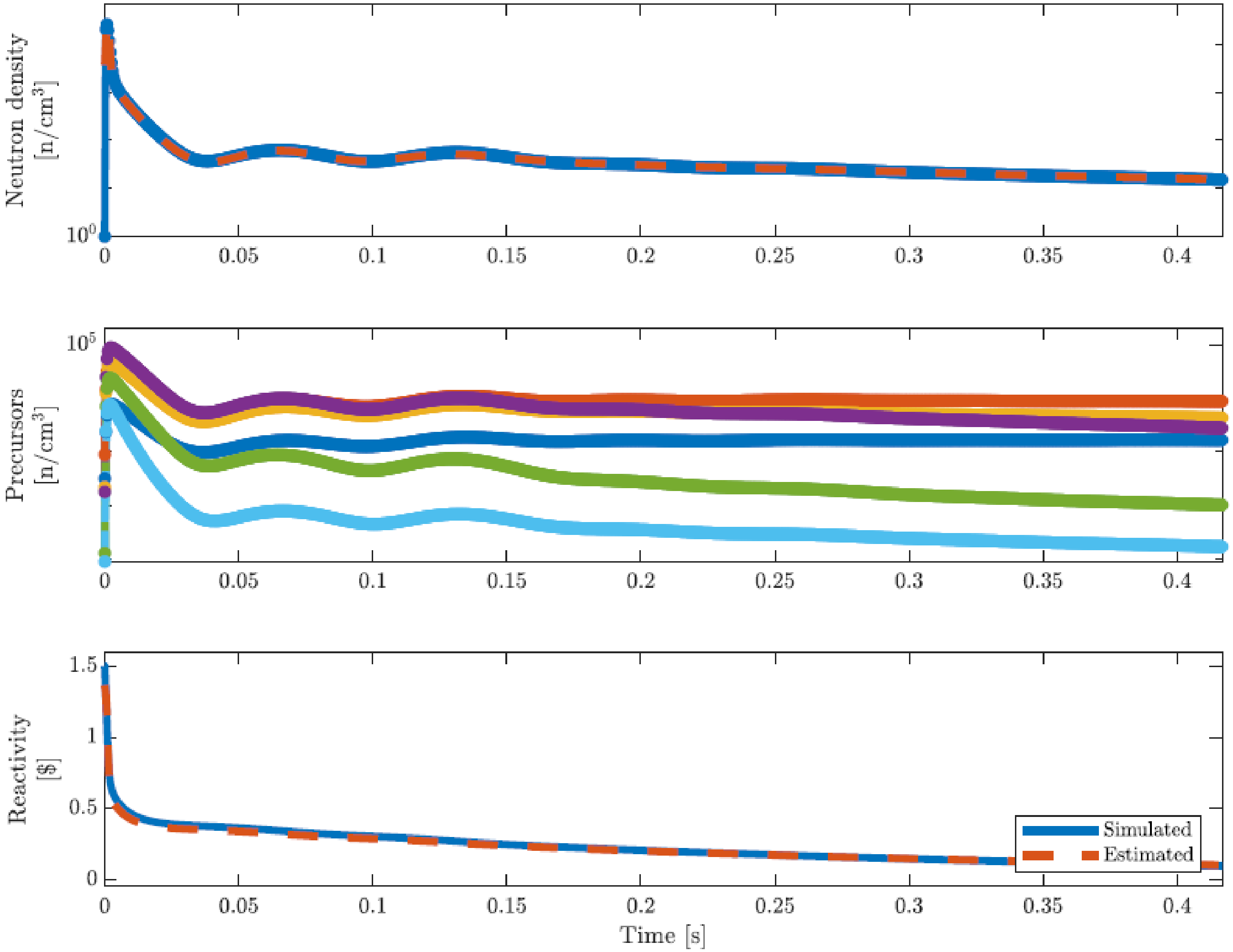


Top left: True (blue solid) and estimated (red dashed) kernel. **Bottom left**: The coefficients, c_m and a, for the true (blue) and estimated (red) kernel. **Right (top, middle, and bottom):** Simulation based on the true (solid) and the estimated (dashed) kernel as well as the measurements (circles).

Point reactor kinetics – Exact meas., non-Erlang kernel







Top left: True (blue solid) and estimated (red dashed) kernel. **Bottom left**: The coefficients, c_m and a, for the estimated (red) kernel. **Right (top, middle, and bottom):** Simulation based on the true (solid) and the estimated (dashed) kernel as well as the measurements (circles).

Summary

Approximate delay using a mixed Erlang PDF

Convert to ODEs

Estimate time delay parameters as ordinary parameters

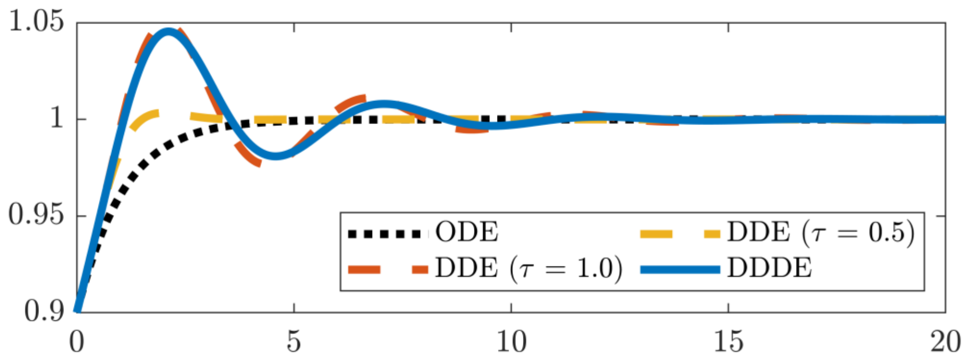
Thank you

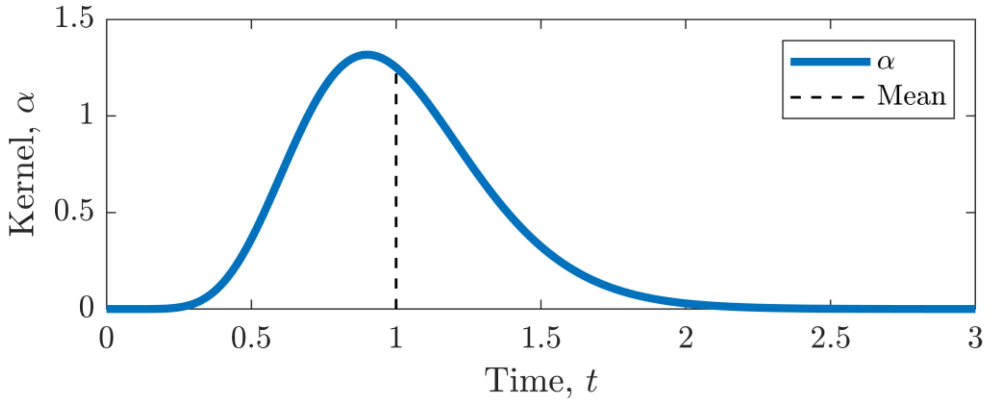
Logistic equation

$$\dot{N}(t) = \sigma N(t)\left(1 - \frac{\tilde{N}(t)}{K}\right)$$

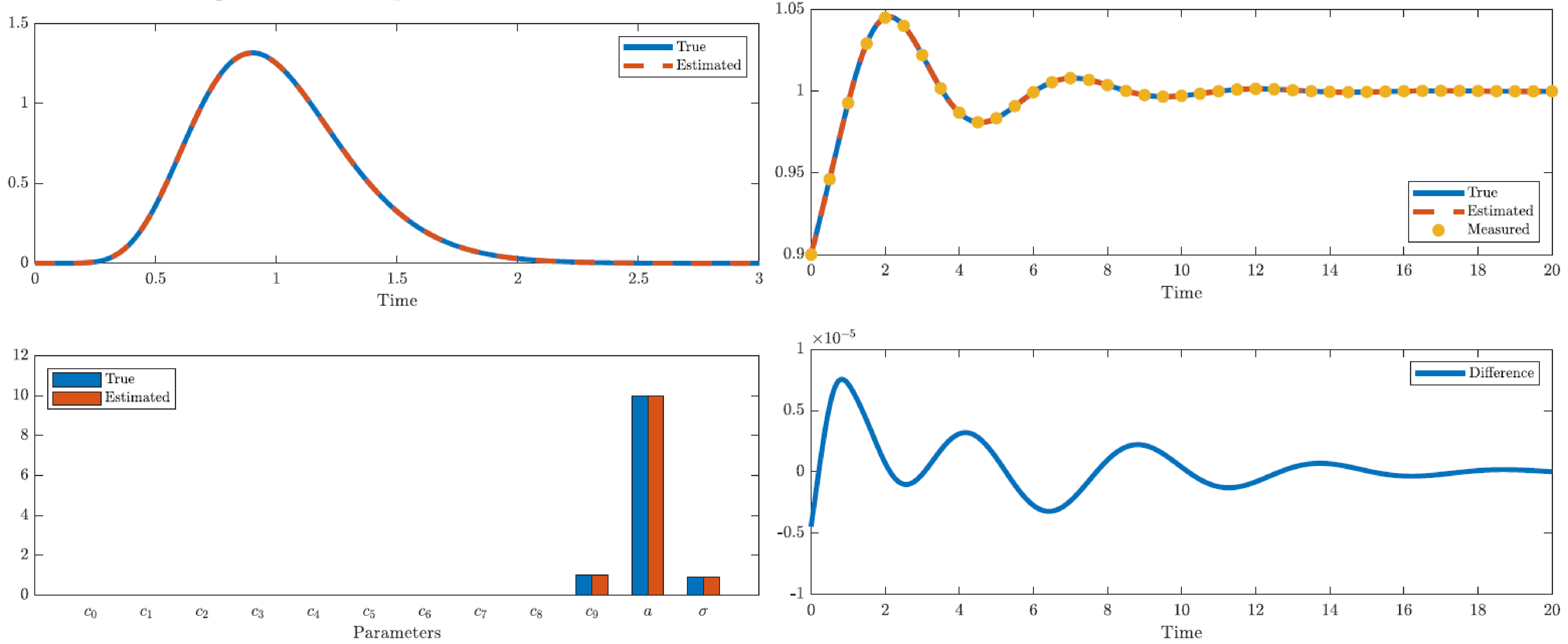
$$\tilde{N}(t) = \int_{-\infty}^{t} \alpha(t-s)N(s)\,\mathrm{d}s$$

- Estimate σ and α



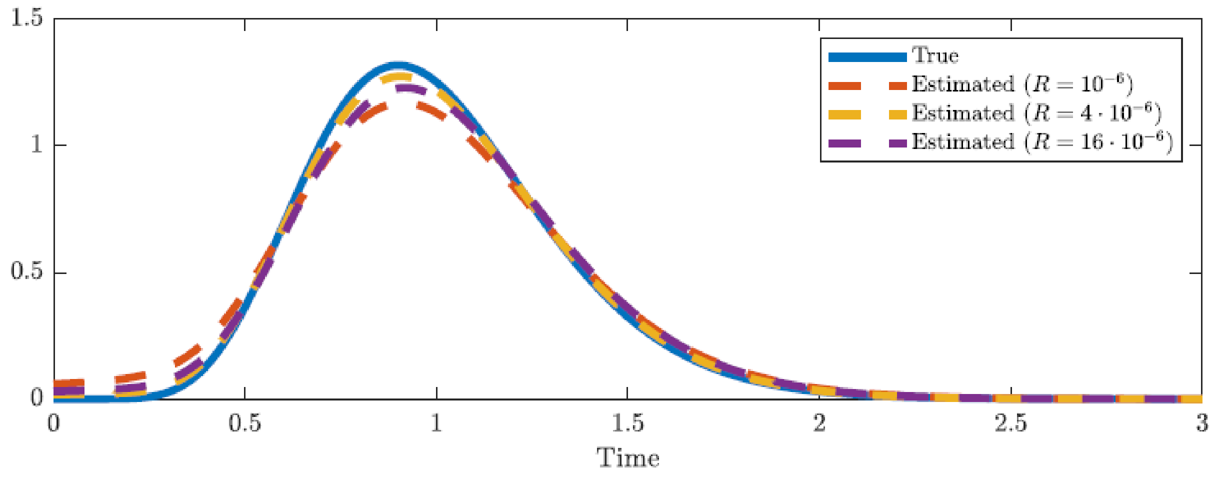


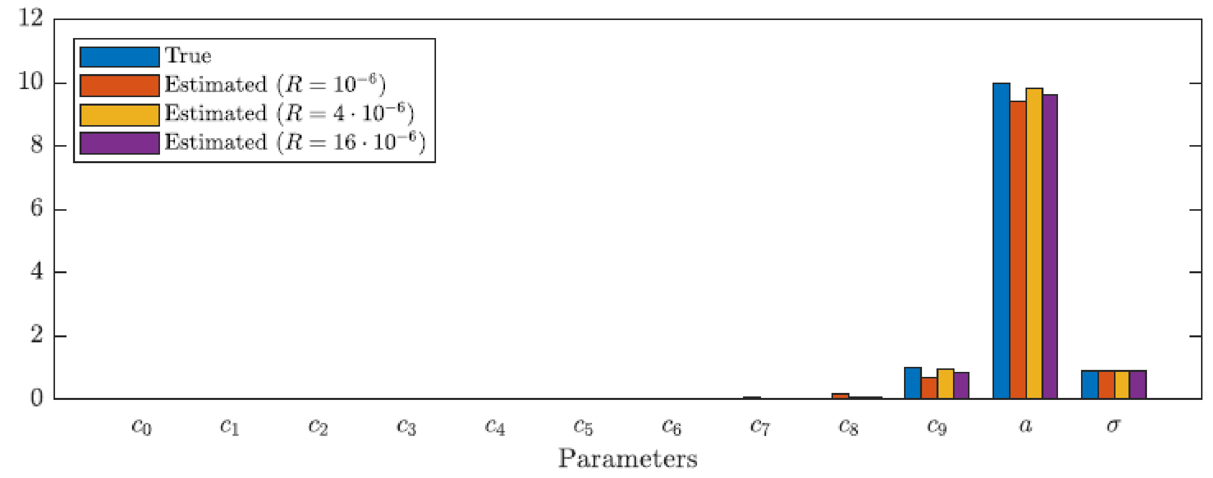
Logistic equation – Exact measurements, correct M



Top left: Identified kernel based on 21 measurements without noise (i.e., $R = 0$). **Bottom left**: The true (blue) and estimated (red) time-delay and model parameters, c_m, a, and σ. **Top right**: Simulation based on the true (blue solid) and estimated (red dashed) parameters as well as the measurements (yellow circles). **Bottom right**: Difference between simulations with true and estimated parameters.

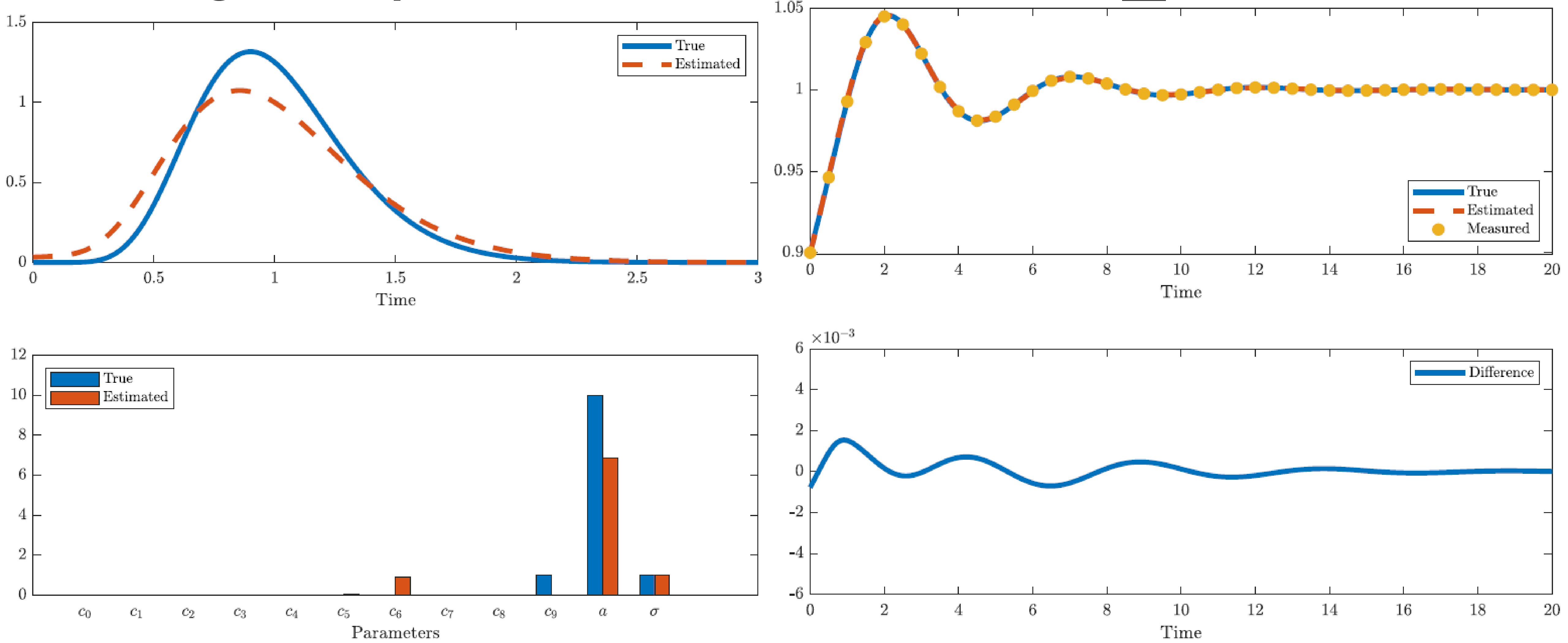
Logistic equation – Noisy measurements, correct M





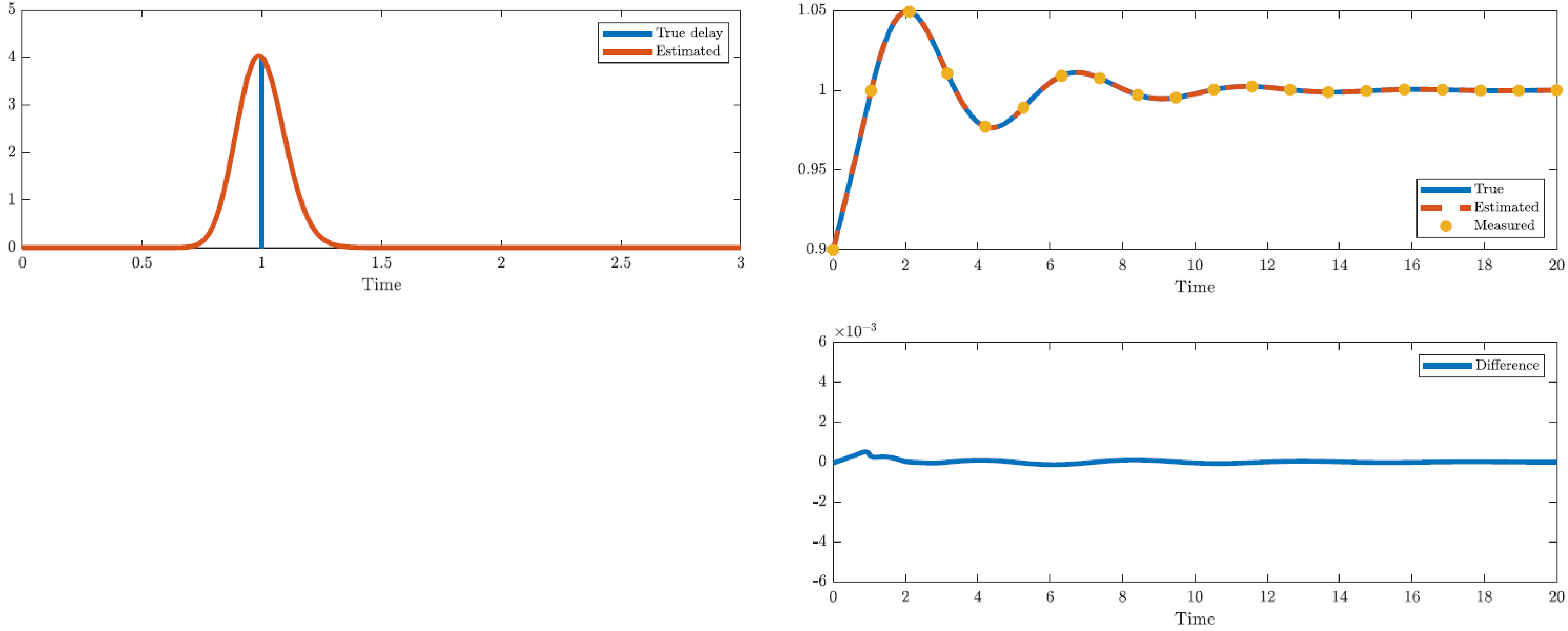
Identified kernels based on 161 measurements with varying values of the measurement noise variance, *R*.

Logistic equation – Exact measurements, incorrect M



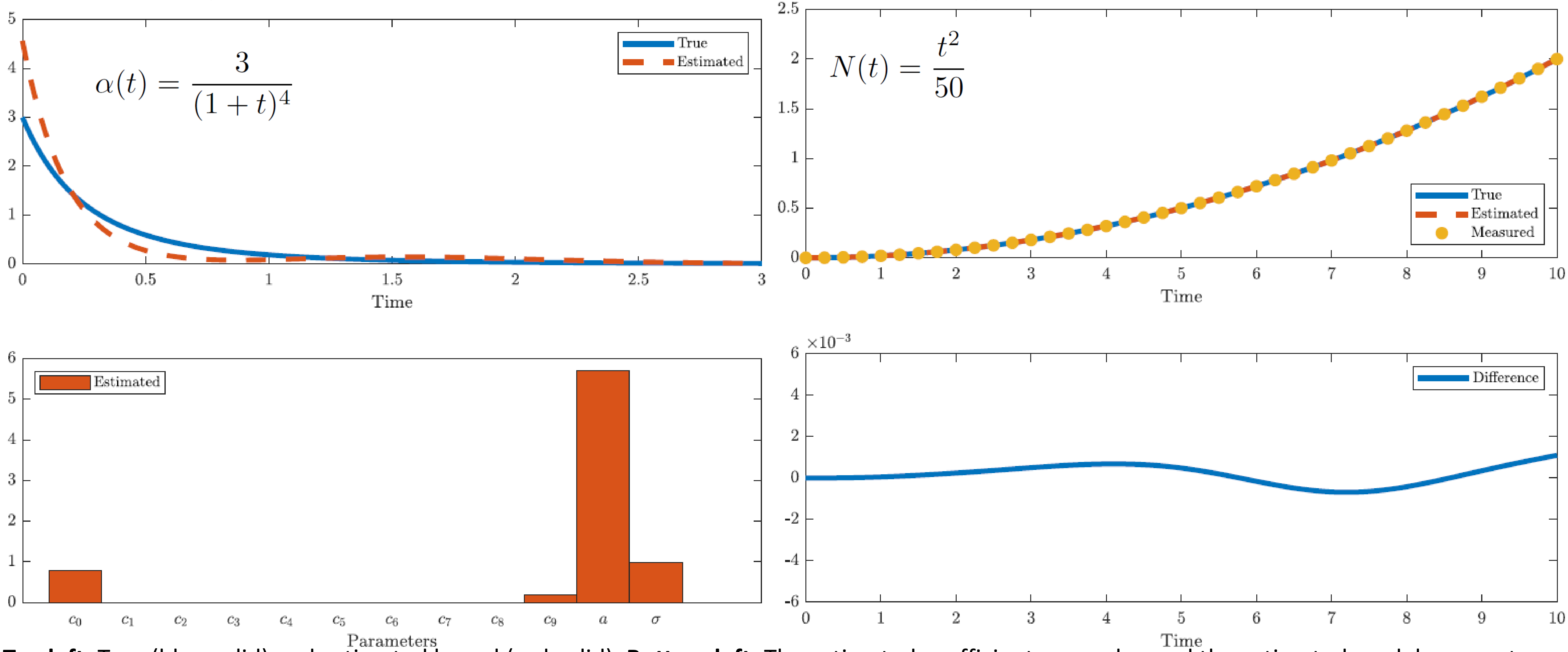
Top left: True (blue solid) and estimated (red dashed) kernel. **Bottom left**: The coefficients, c_m and a, for the true (blue) and estimated (red) kernel. **Top right**: Simulation based on the true (blue solid) and the estimated (red dashed) kernel as well as the measurements (yellow circles). **Bottom right**: Difference between simulations with true and estimated parameters.

Logistic equation – Absolute delay



Top left: True delay (blue solid) and estimated kernel (red solid). **Top right**: Simulation based on the true absolute delay (blue solid) and the estimated (red dashed) kernel as well as the measurements (yellow circles). **Bottom right**: Difference between simulations with true and estimated parameters. Bottom:

Logistic equation – ”Nonconforming” kernel



Top left: True (blue solid) and estimated kernel (red solid). **Bottom left**: The estimated coefficients, c_m and a, and the estimated model parameter σ. **Top right**: Simulation based on the true (blue solid) and the estimated (red dashed) kernel as well as the measurements (yellow circles). **Bottom right**: Difference between simulations with the true and estimated parameters.