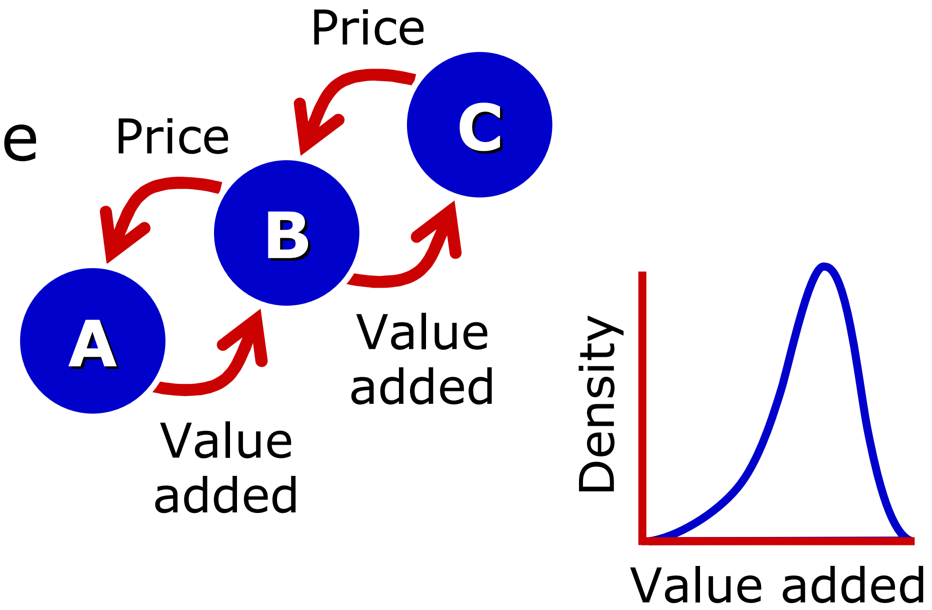
Bayesian Analysis of Stochastic System Dynamics

Rudolf Kulhavý

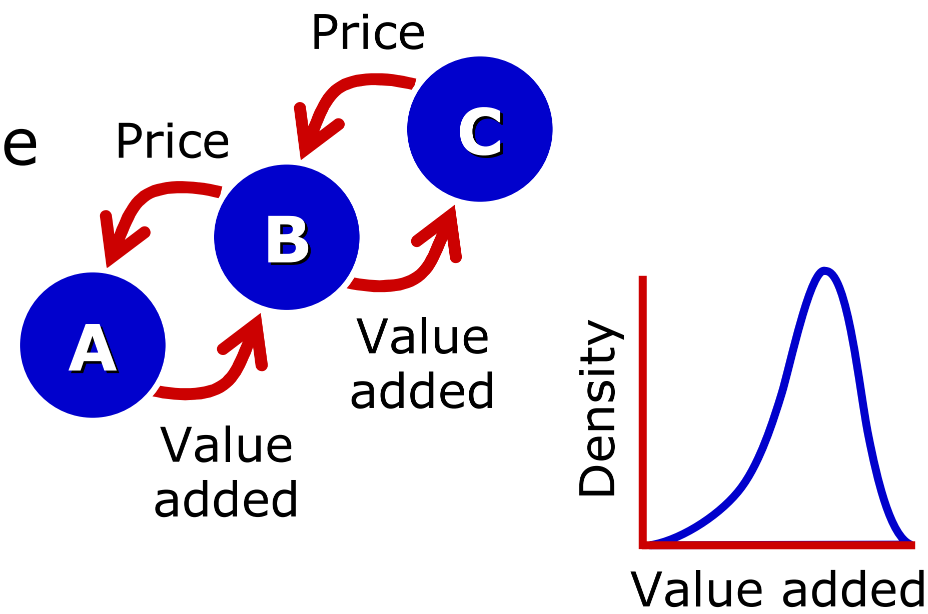
Why to study stochastic systems?

- Dynamic modeling of the overall performance of
  - value chains
  - value networks
  - virtual enterprises

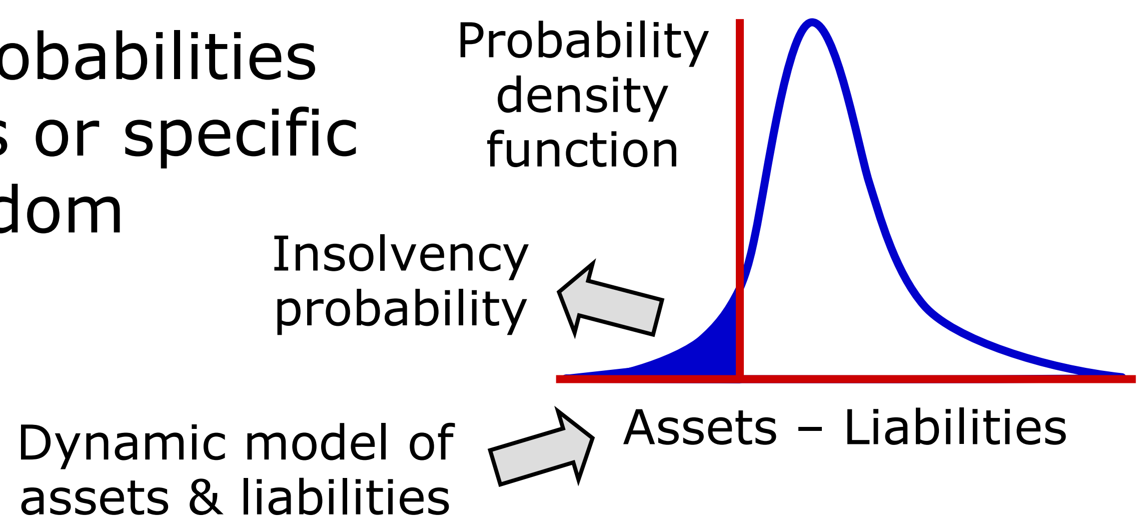


Why to study stochastic systems?

- Dynamic modeling of the overall performance of
  - value chains
  - value networks
  - virtual enterprises



- Estimation of probabilities of critical events or specific quantiles of random variables



Stochastic dynamic model

Generalization A

$$\begin{aligned} \dot{x}_t &= f(x_t, u_t) \\ y_t &= g(x_t, u_t) \end{aligned}$$

Generalization B

Time index

Sampling period

$$\begin{aligned} x_{k+1} &= x_k + \tau_s f(x_k, u_k, w_k) \\ y_k &= g(x_k, u_k, v_k) \end{aligned}$$

$$p(x_{k+1} | x_k, u_k)$$

$$p(y_k | x_k, u_k)$$

Discrete-time values

State and measurement "noise"

Conditional probability density functions

Stochastic dynamic model

$$\begin{cases} \dot{x}_t = f(x_t, u_t) \\ y_t = g(x_t, u_t) \end{cases}$$

Stochastic differential equation representation ↓

$$x_{k+1} = x_k + \tau_s f(x_k, u_k, w_k)$$
$$y_k = g(x_k, u_k, v_k)$$

Conditional probability representation ↓

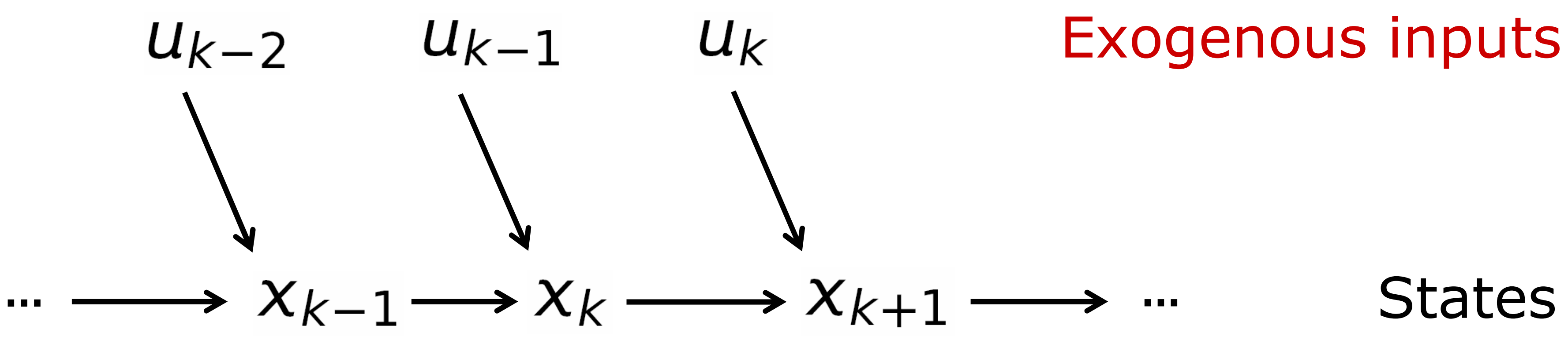
$$p(x_{k+1} | x_k, u_k)$$
$$p(y_k | x_k, u_k)$$

Markov chain

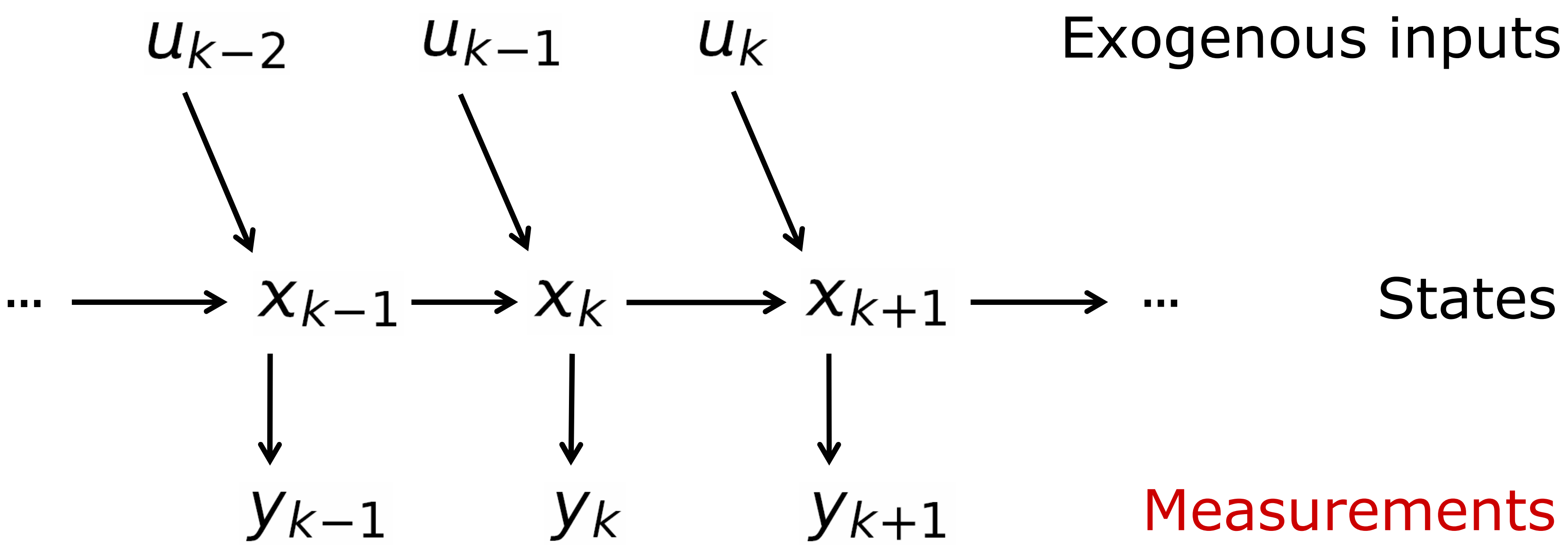
$$\cdots \longrightarrow x_{k-1} \longrightarrow x_k \longrightarrow x_{k+1} \longrightarrow \cdots$$

States

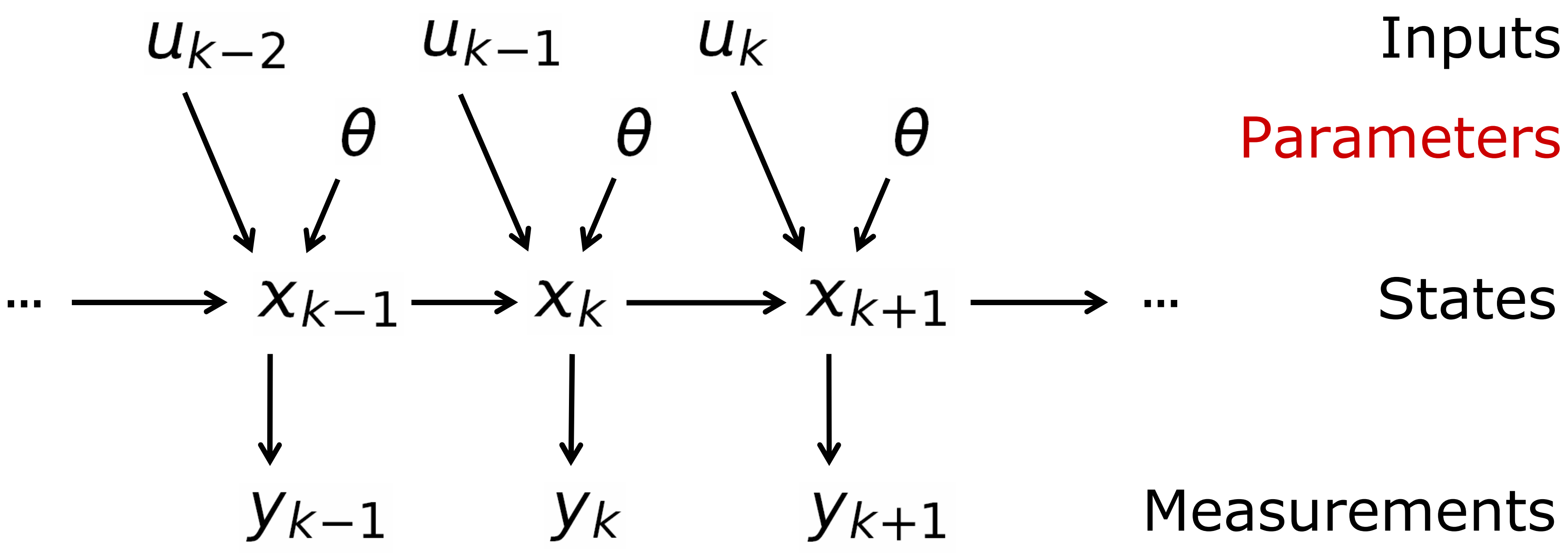
Controlled Markov chain



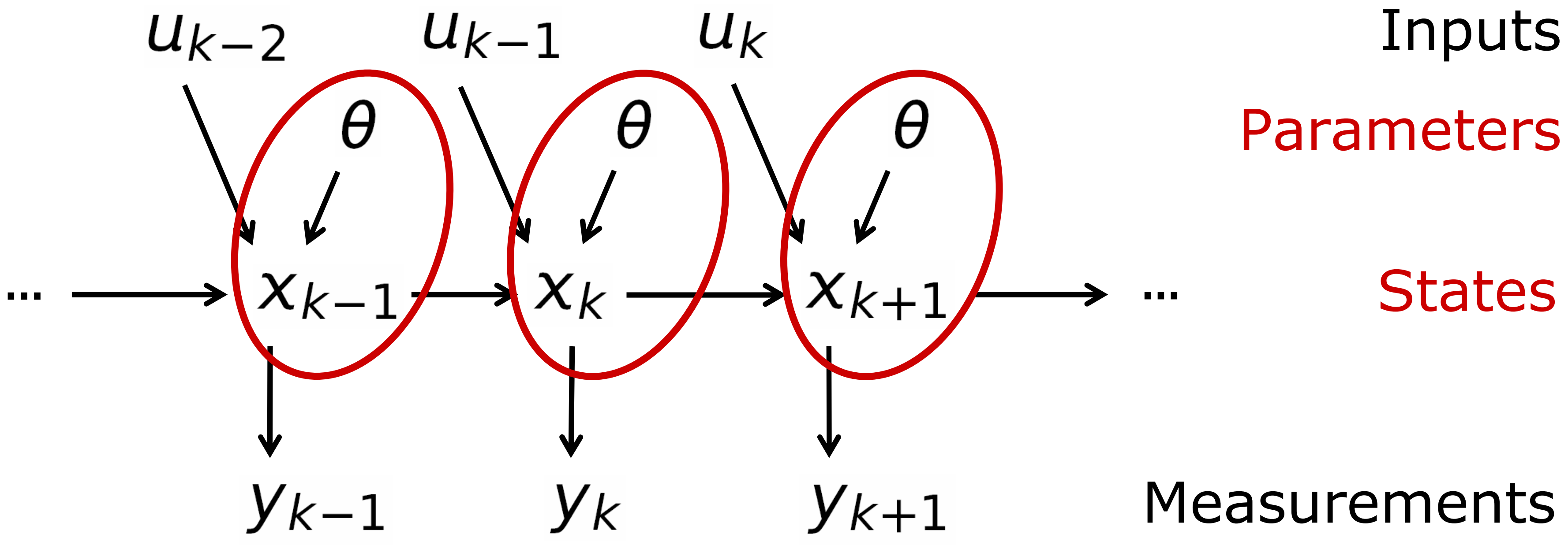
Partially observed, controlled Markov chain



Partially observed, controlled Markov chain, with unknown parameters



Partially observed, controlled Markov chain, with unknown parameters



Unknown model parameters can be treated as extra states

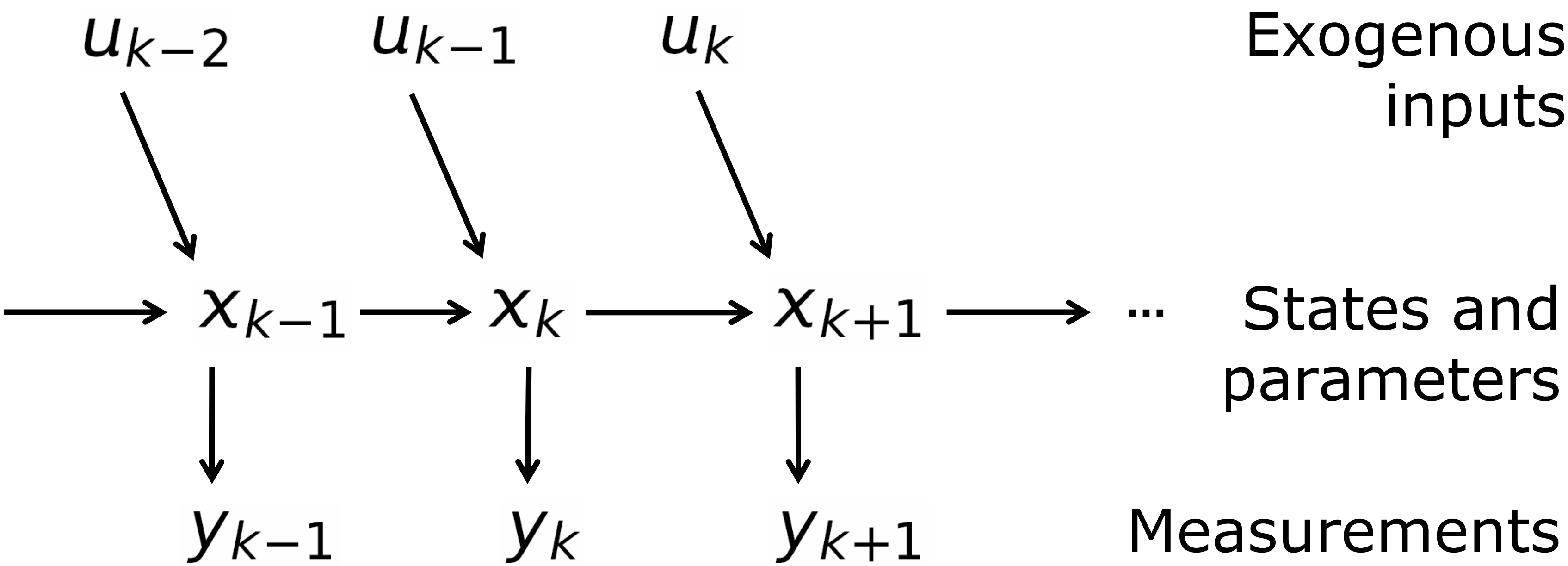
The augmentation of the state vector

- increases the dimensionality of the problem (and, thereby, uncertainty of the original states);
- adds additional nonlinearities.

$$x_{1,k+1} = \theta\, x_{1,k} \xrightarrow{x_{2,k} = \theta} \begin{aligned} x_{1,k+1} &= x_{2,k}\, x_{1,k} \\ x_{2,k+1} &= x_{2,k} \end{aligned}$$

On the other hand, it allows for explicit modeling of parameter variations.

Summary of model



$$\begin{aligned} x_{k+1} &= x_k + \tau_s f(x_k, u_k, w_k) \\ y_k &= g(x_k, u_k, v_k) \end{aligned}$$

$$p(x_{k+1} | x_k, u_k)$$
$$p(y_k | x_k, u_k)$$

Estimation of states (and parameters)

Past data sequences

$$p(x_k | y^{k-1}, u^{k-1})$$

↓ Measurement update

Observations

u_k, y_k

$$p(x_k | y^k, u^k)$$

↓ Time update

$$p(x_{k+1} | y^k, u^k)$$

Bayesian inference

Quantification of all uncertainty via <u>probability</u>

Functional recursions

Measurement update

$$\underbrace{p(x_k|y^k, u^k)}_{\text{Posterior}} \propto \underbrace{p(y_k|x_k, u_k)}_{\text{Likelihood}} \underbrace{p(x_k|y^{k-1}, u^{k-1})}_{\text{Prior}}$$

Product rule

Probability Theory

Sum rule

Time update

$$\underbrace{p(x_{k+1}|y^k, u^k)}_{\text{Next-step prior}} = \int \underbrace{p(x_{k+1}|x_k, u_k)}_{\text{Transition probability}} \underbrace{p(x_k|y^k, u^k)}_{\text{Posterior}} \, \mathrm{d}x_k$$

Sequential Monte Carlo approximation

Replacing probabilities with samples

$$p(x_k | y^{k-1}, u^{k-1}) \rightarrow \{x_{k|k-1}^{(i)}\}$$

Measurement update

$$p(x_k | y^k, u^k) \qquad \{x_{k|k}^{(i)}\}$$

Time update

$$p(x_{k+1} | y^k, u^k) \leftarrow \{x_{k+1|k}^{(i)}\}$$

Particle filter (step 1)

1. *Initialization:* Draw M samples from the prior distribution

$$x_{1|0}^{(i)} \sim p(x_1),\ i = 1, \ldots, M$$

and set $k := 1$.

Particle filter (step 2)

2. *Data Update:* Collect the data u_k, y_k. If the output y_k is measured at time k, then evaluate the importance weights

$$\pi_i = \frac{p(y_k | x_{k|k-1}^{(i)}, u_k)}{\sum_{j=1}^{M} p(y_k | x_{k|k-1}^{(j)}, u_k)}, \quad i = 1, \ldots, M$$

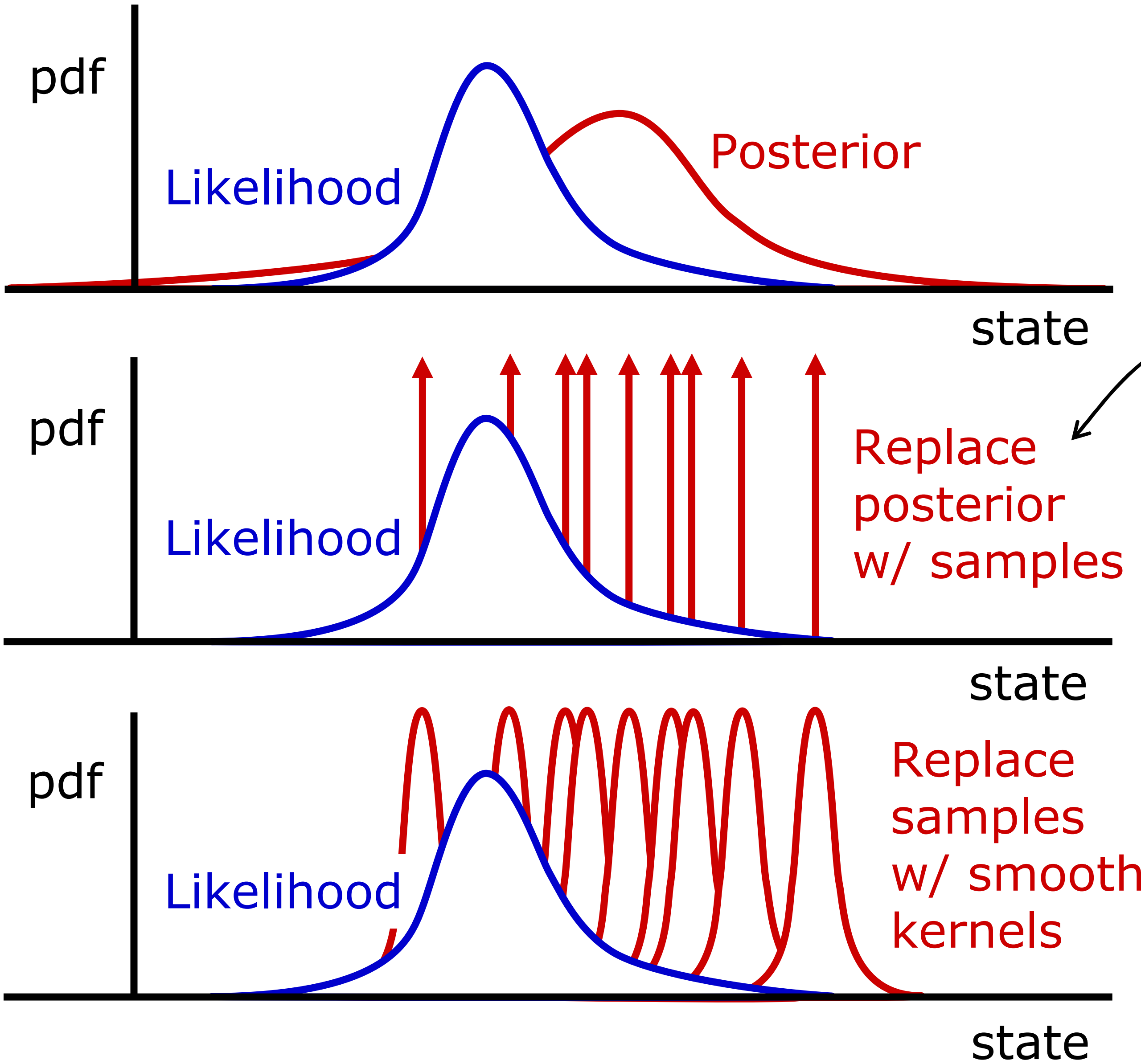
and draw M samples from a kernel approximation to the posterior distribution

$$x_{k|k}^{(i)} \sim \sum_{j=1}^{M} \pi_j K(x_k - x_{k|k-1}^{(j)}), \quad i = 1, \ldots, M.$$

If the output y_k is not measured, set

$$x_{k|k}^{(i)} = x_{k|k-1}^{(i)}, \quad i = 1, \ldots, M.$$

Data update



No need for explicit sampling, except Step 1

Samples come from the preceding iteration step

Algorithm:

Resample from posterior samples with probabilities proportional to the likelihood values, then draw a sample from the corresponding kernel density

Particle filter (steps 3, 4)

3. *Time Update:* Draw M samples from a mixture approximation to the predictive distribution

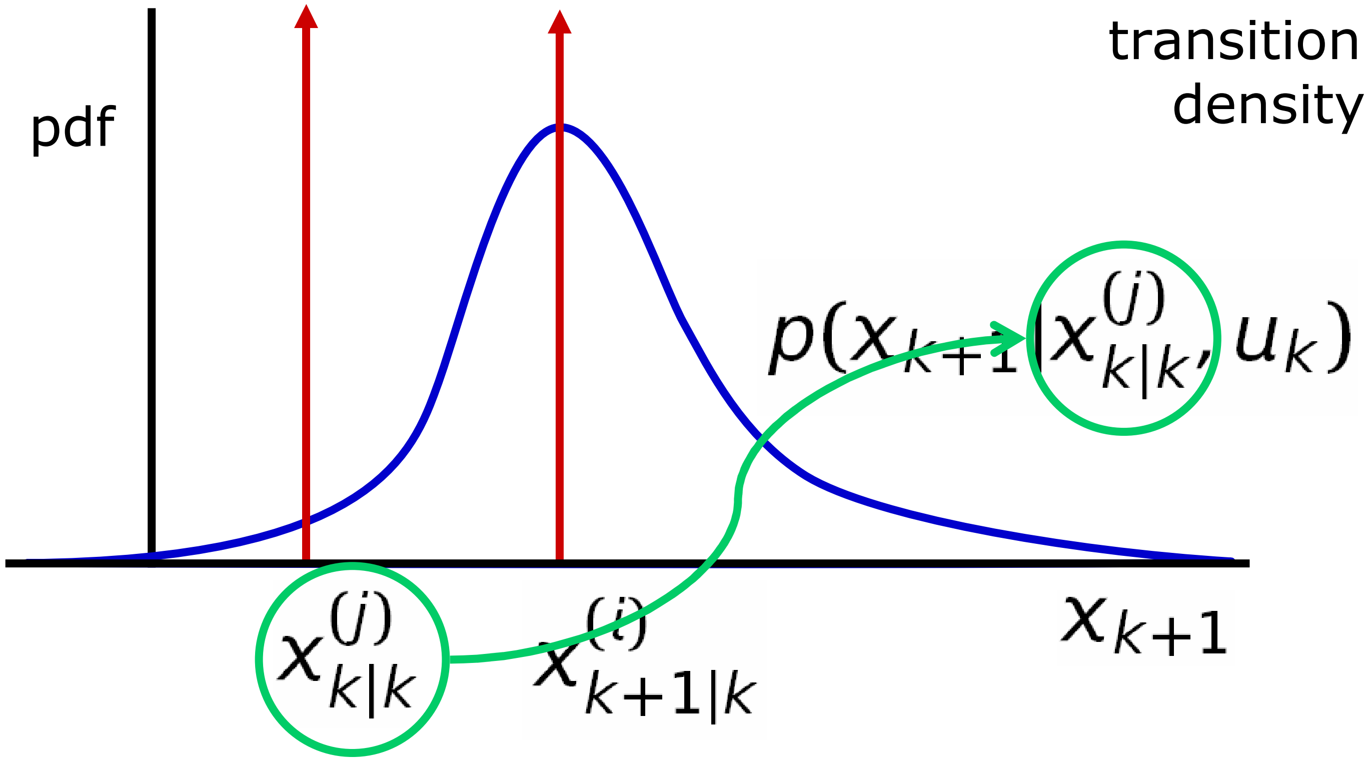
$$x_{k+1|k}^{(i)} \sim \frac{1}{M}\sum_{j=1}^{M} p(x_{k+1}|x_{k|k}^{(j)}, u_k), \; i = 1, \ldots, M.$$

4. *Iteration:* Increment $k := k+1$ and iterate from data update.

Time update

Algorithm:

Pick up randomly one of the posterior samples, then draw a new sample from the corresponding transition probability density function

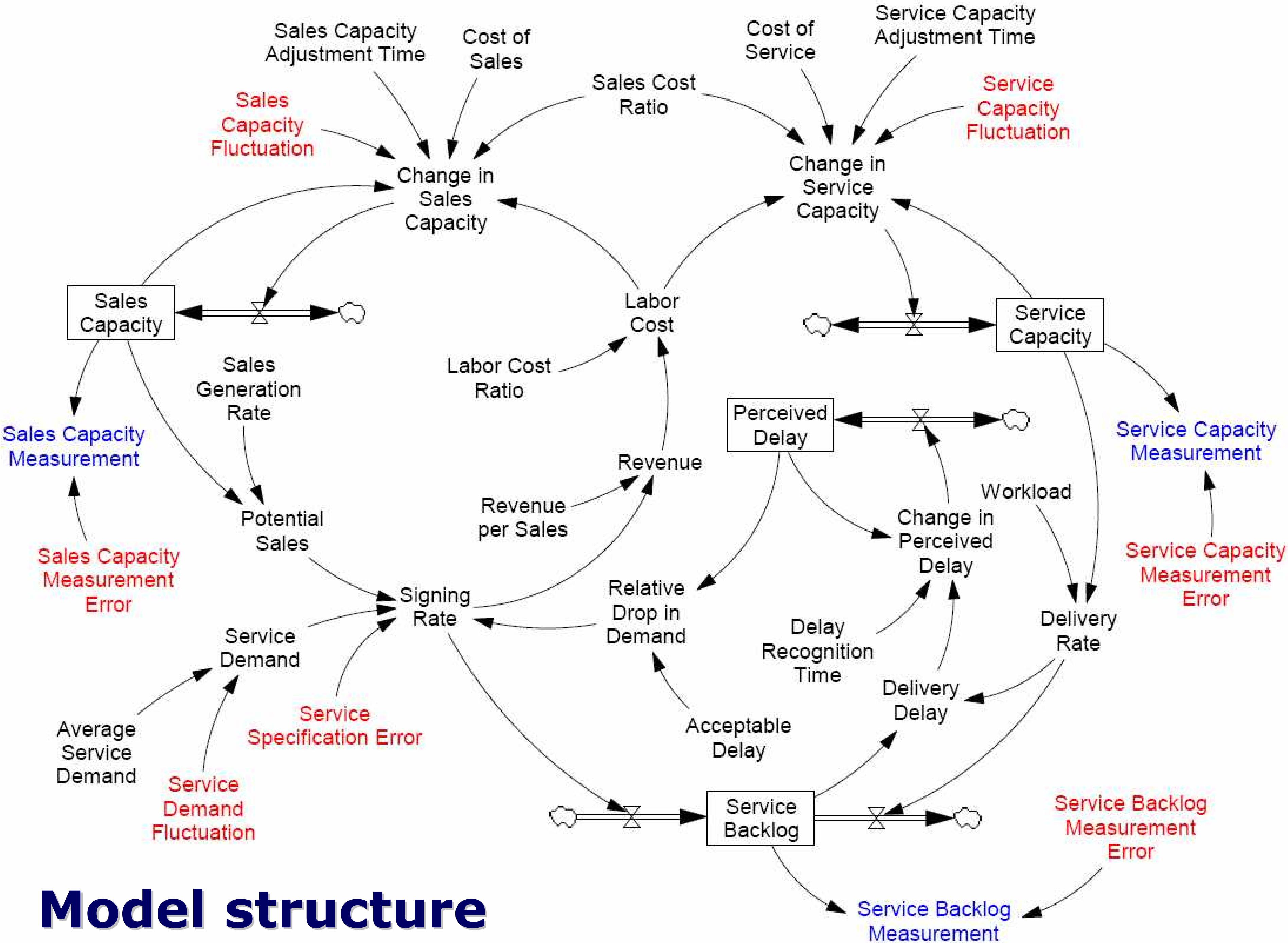


- Automated target recognition
  Anuj Srivastava, Michael Miller , Ulf Grenander
- Bayesian networks
  Daphne Koller; Kevin Murphy
- Computational anatomy
  Ulf Grenander, Michael Miller
- Mobile robotics
  Dieter Fox, Wolfram Burgard, Sebastian Thrun
- Neural networks
  Nando de Freitas
- Signal processing
  Petar Djurić
- Tracking and guidance
  David Salmond, Neil Gordon
- Visual shape and motion
  Andrew Blake, Michael Isard, John MacCormick

Particle filter has been applied success-fully in many areas

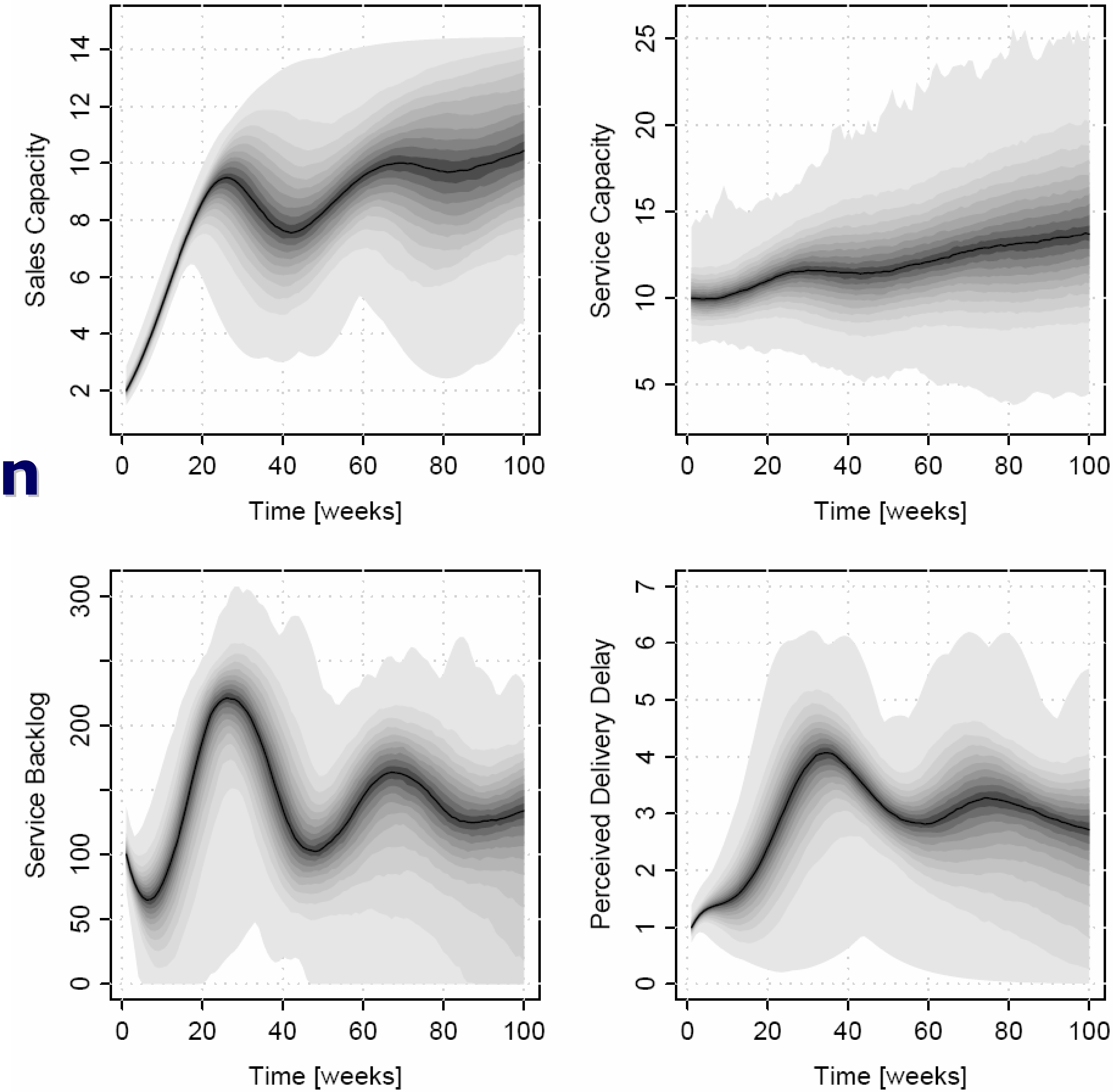
Illustrative example

- Consider a service company whose economic results depend critically on the performance of both the sales and service staff.
- The stocks of Sales Capacity and Service Capacity are measured in multiples of full-time equivalents (FTE) of an average sales or service person.
  - This relates the labor capacity to the total performance of a team rather than the number of physical persons.
  - Thus, hiring an additional person can increase the stock by more or less than one, depending on the actual person’s productivity.



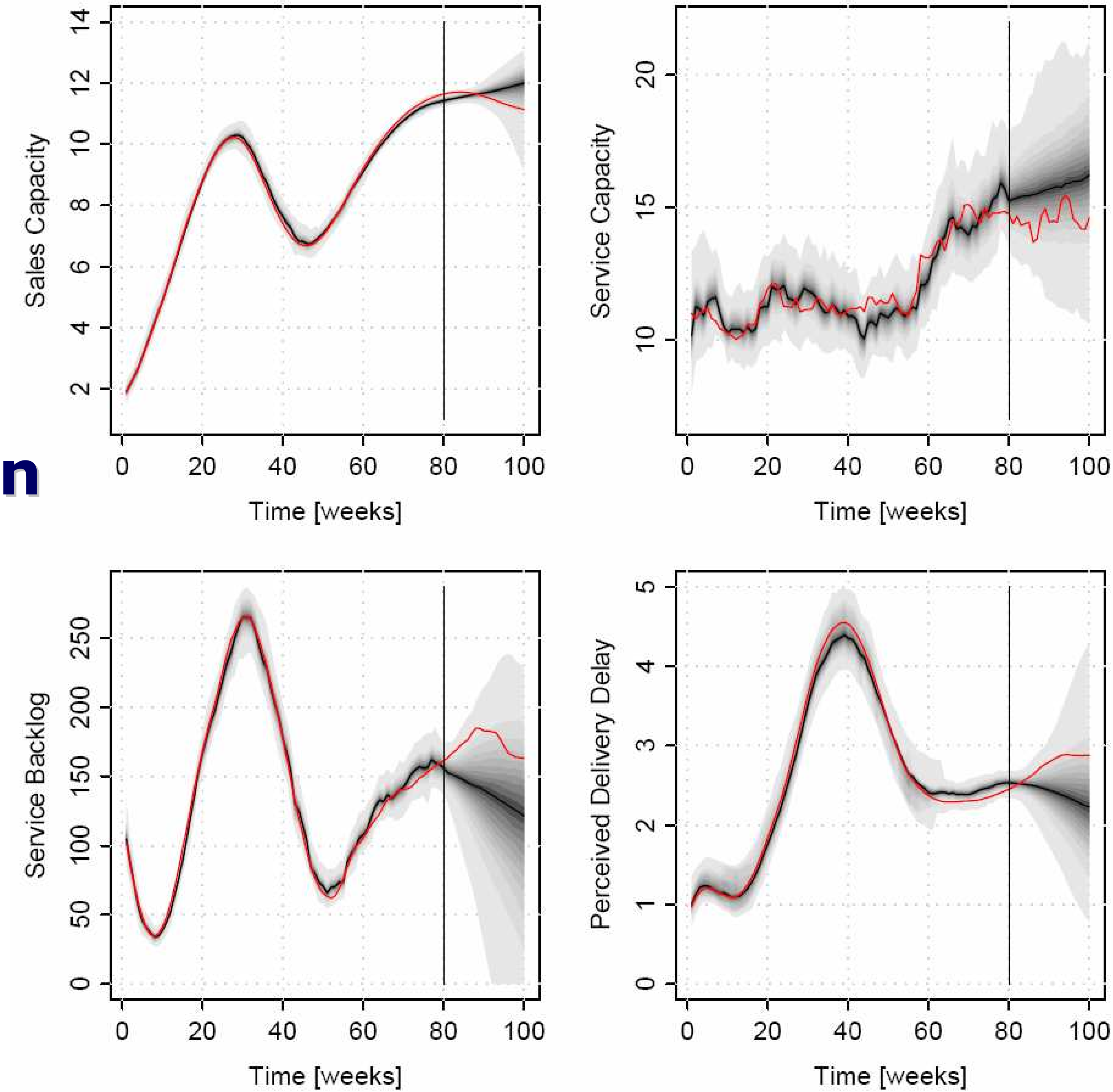
Model structure

Simulation results



— estimate

Estimation results



— measurement

— estimate

Model comparison

Posterior probability function

Model class index

$$p(l|y^N, u^N) \propto p(l) \prod_{k=1}^{N} p(y_k|y^{k-1}, u^k, l)$$

Predictive density function

$$p(y_k|y^{k-1}, u^k, l)$$

$$= \int p(y_k|x_{k,l}, u_k, l)\, p(x_{k,l}|y^{k-1}, u^{k-1}, l)\, \mathrm{d}x_{k,l}$$

Model class index

Approximate model comparison

Monte Carlo approximation

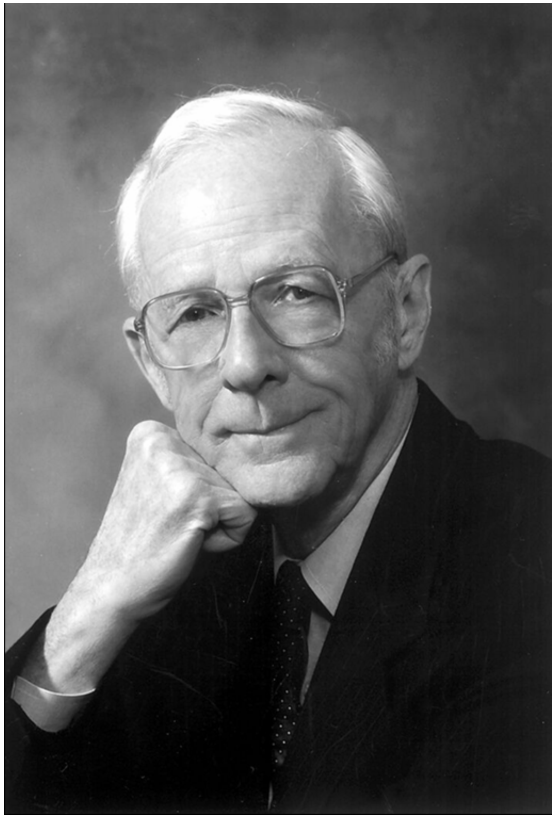
$$p(y_k|y^{k-1}, u^k, l) \approx \frac{1}{M}\sum_{i=1}^{M} p(y_k|x_{k,l}^{(i)}, u_k, l)$$

Samples from posterior pdf

$$x_{k,l}^{(1)}, \ldots, x_{k,l}^{(M)} \sim p(x_{k,l}|y^{k-1}, u^{k-1}, l)$$

So, what has *Bayesian Inference* to do with System Dynamics?

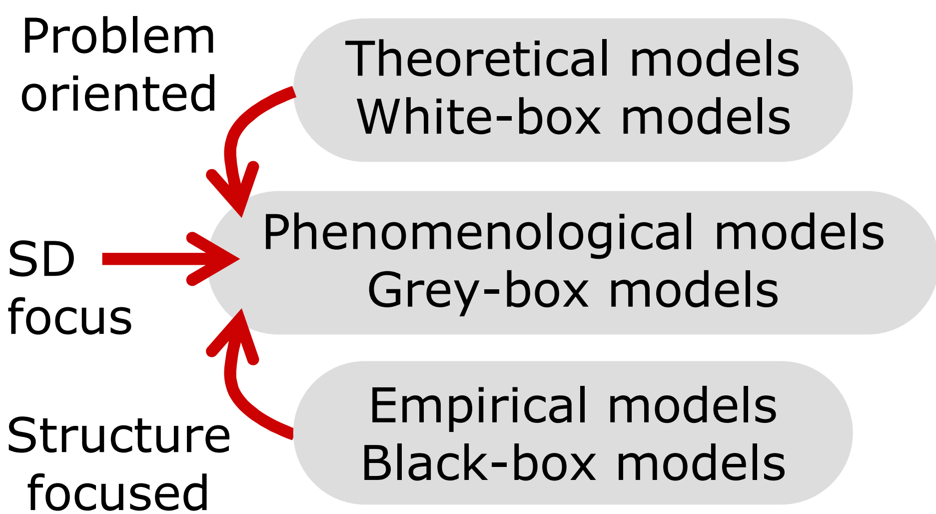
Pierre-Simon,
Marquis de Laplace



Jay Wright
Forrester

System Dynamics x Bayesian Inference

- System dynamics provides the modeler with practical methodology for converting prior information into a dynamic model structure (*highly informative priors*).



Bayesian inference

- gives precise meaning to all modeling concepts;
- yields a coherent framework for consistently updating the prior state of knowledge with numerical evidence at hand;
- captures and combines all manifestations of uncertainty (stochastic fluctuations, measurement errors, unknown model parameters, unknown model structure).

$$
\begin{aligned}
p(l|y^N, u^N) &\propto p(l)\, p(y^N, u^N|l) \\
&\propto p(l) \int p(y^N, u^N|\theta_l, l)\, p(\theta_l|l)\, \mathrm{d}\theta_l \\
&\propto p(l)\, p(y^N, u^N|\hat{\theta}_{l,\max}, l)\, P_l
\end{aligned}
$$

The maximum likelihood value

The amount of prior probability contained in the high likelihood region of parameter space

Occam's razor

"[T]here is a loose connection between simplicity and plausibility, because the more complicated a set of possible hypotheses, the larger the manifold of conceivable alternatives, and so the smaller must be the prior probability of any particular hypothesis in the set."

Edwin Jaynes, Probability Theory: The Logic of Science

Thus, among models of comparable predictive power, Bayesian inference assigns higher posterior probability to "simpler" ones.

Conclusion

- The progress made in sequential Monte Carlo methods has made Bayesian inference an attractive option for system dynamics modeling, especially for problems where quantification of the state (and parameter) uncertainty is critical.