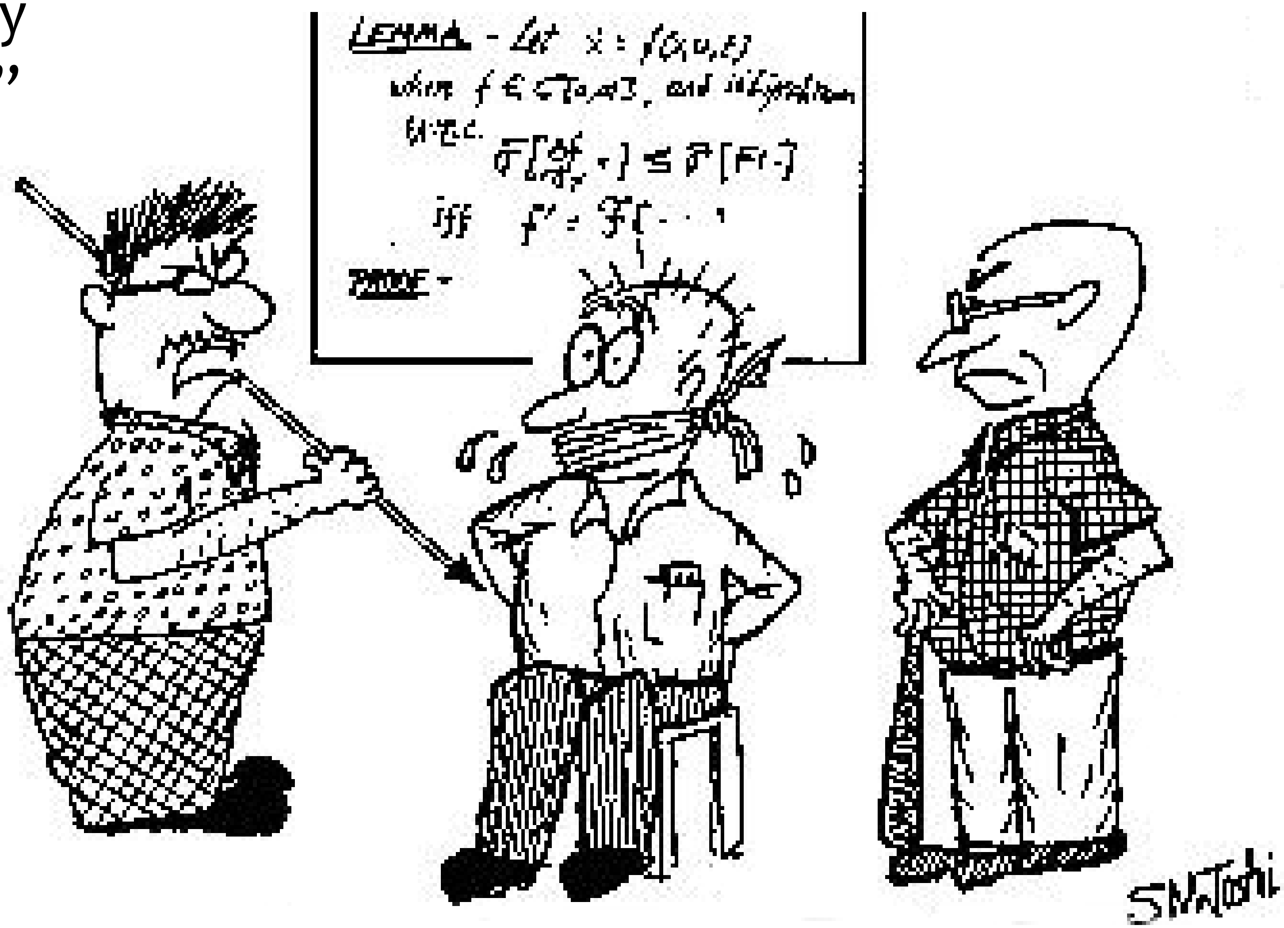
AS-74.4192 Elementary Cybernetics

Lecture 8:
Practical Experiments

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology
Cybernetics Group

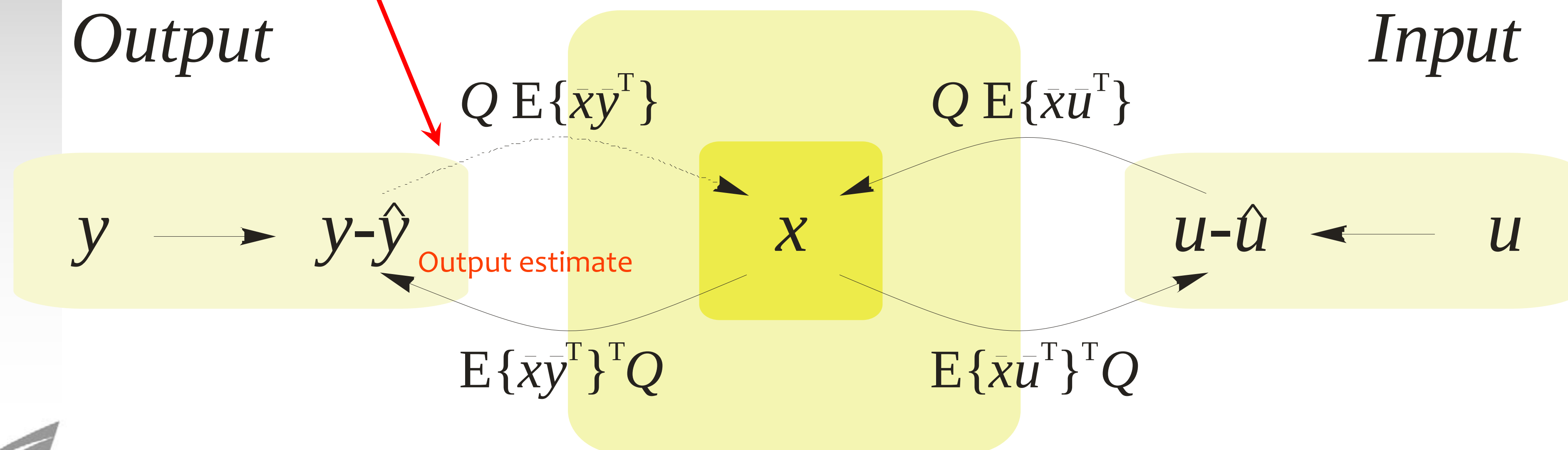
- Key point to note: In practice, theory has to be “tuned” appropriately
- Luckily, the neocybernetic framework is very versatile!

“After we beat the proof out of him, let’s dump him in the theory–practice gap!”



Some applications are evident...

- Some ways of theory application are straightforward...
- Easy: Iterative ways of implementing multivariate regression
- For example, "PCR style" and "PLS style" regression (when the feedback from the output is omitted and when it is not)



"PLS" – alternate applying *u* and *y* as input

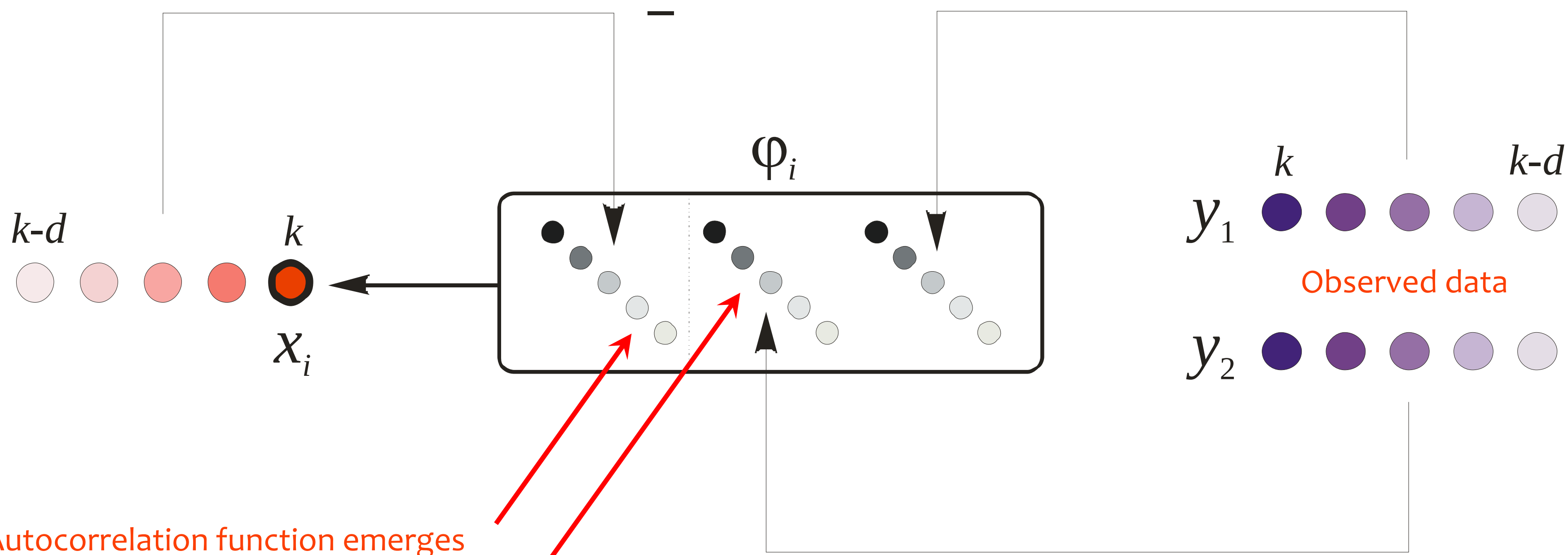
Apply cybernetic intuitions in general data-based modeling?

- Depending on the case:

  No mean-centering? Logarithmic variables applied? …
- Scaling either in the form $v_i \leftarrow v_i / \overline{v}_i$, meaning that relative changes matter most, or standard normalization of variables (variances = 1) due to the *equalization of variables*!
- *Cybernetic semantics* studied during Lecture 10: Data must contain not only the system state z but also the "tensions" = actions (controls) that push the system towards balance

$$v = \begin{pmatrix} z \\ \hline \frac{dz}{dt} \end{pmatrix}$$

- Form of "implicit subspace identification" can be implemented when variables are grouped as time series

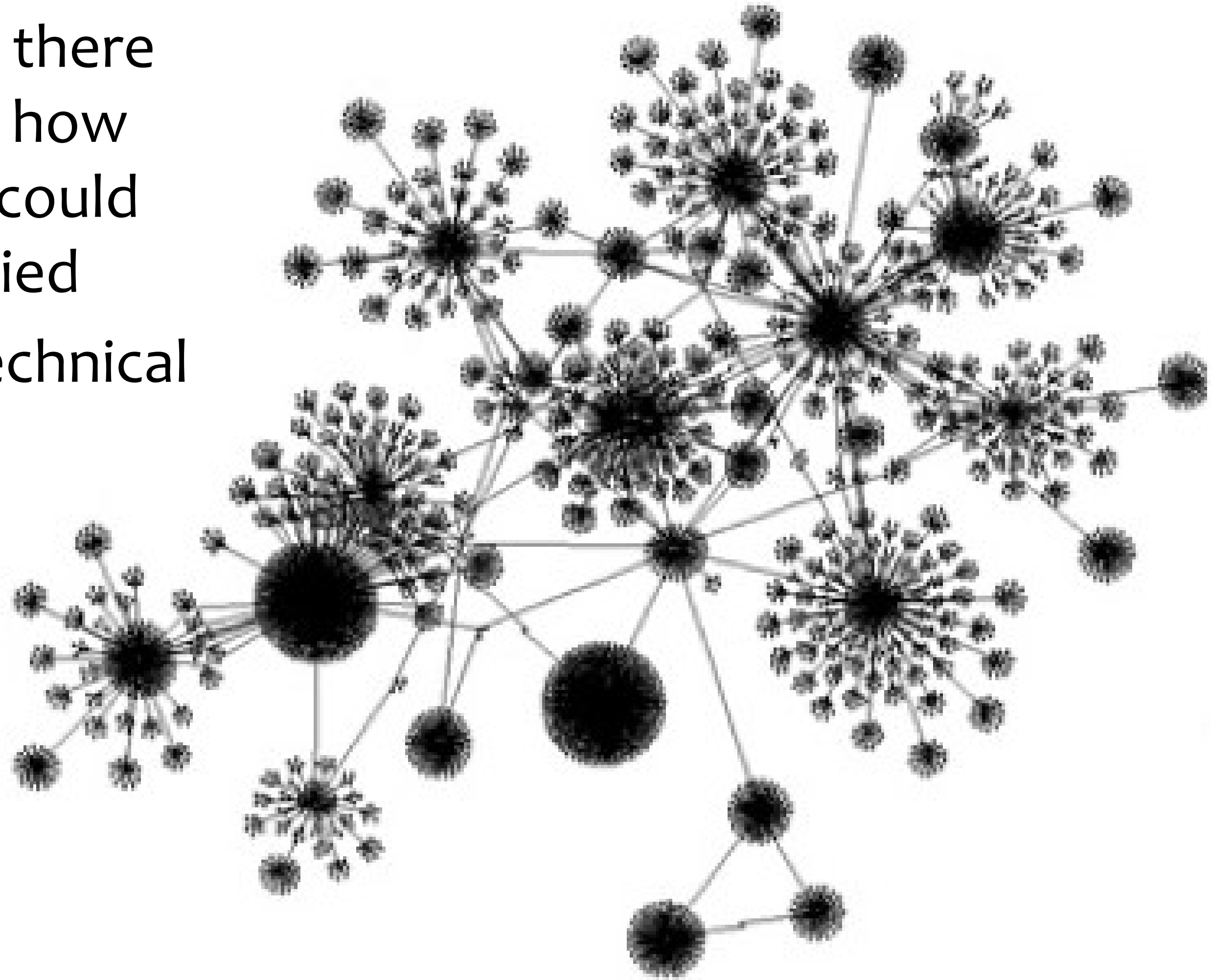


Autocorrelation function emerges

Cross-correlation function emerges – direct relation to *power spectra* (after being Fourier transformed)

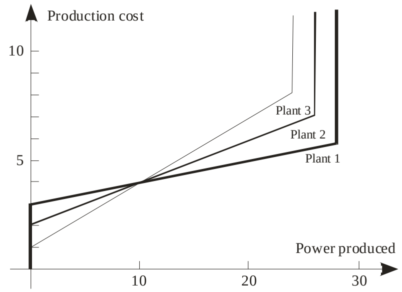
… Some applications are less evident

- In what follows, there are examples of how neocybernetics could perhaps be applied
- So, what are “technical populations”?



Design of networks: Case *energy production*

- Strict optimality:



- Predetermined profiles:

$$J' = (u - \varphi x)^T (u - \varphi x)$$

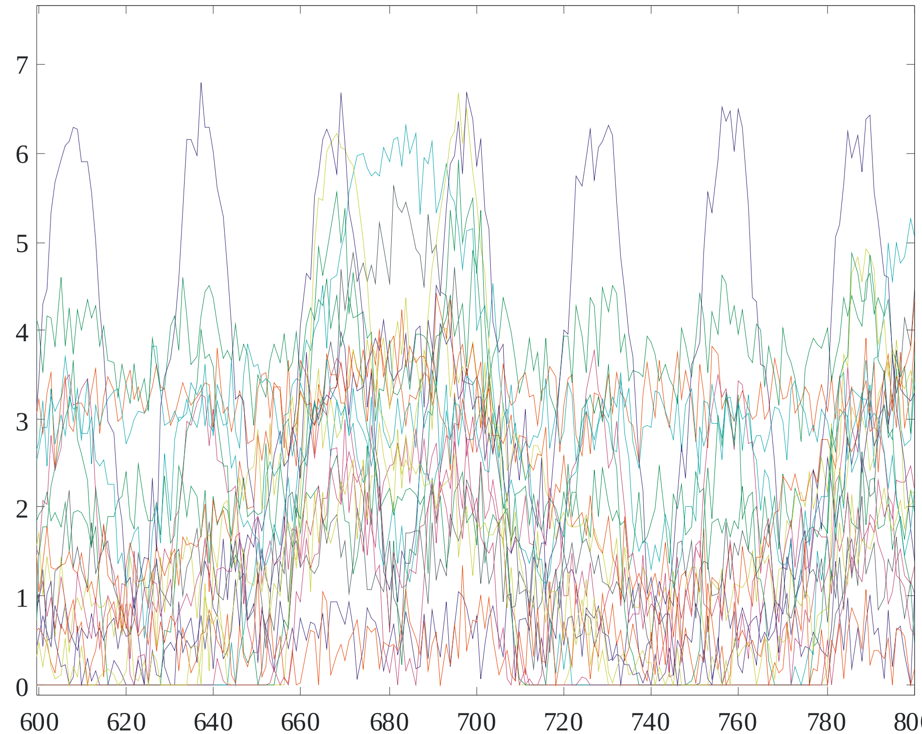
- Cybernetic cost:

$$J'' = (u - \phi x)^T \, \mathrm{E}\{uu^T\} (u - \phi x)$$

- Additional constraint:

$$\sum_{i=1}^{3} \bar{x}_i = \sum_{j=1}^{20} u_j$$

Behaviors of 20 consumers (u_j)



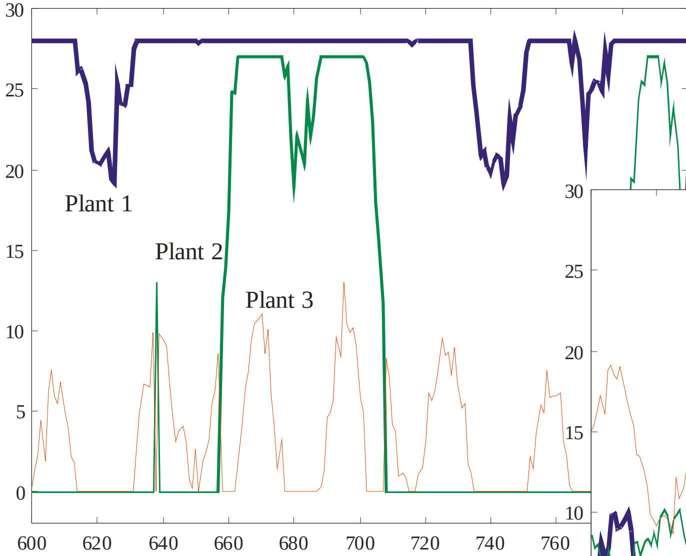
Goal: Optimize production of the three production units!

- Static minimization of the criterion separately for each time instant – three strategies experimented:

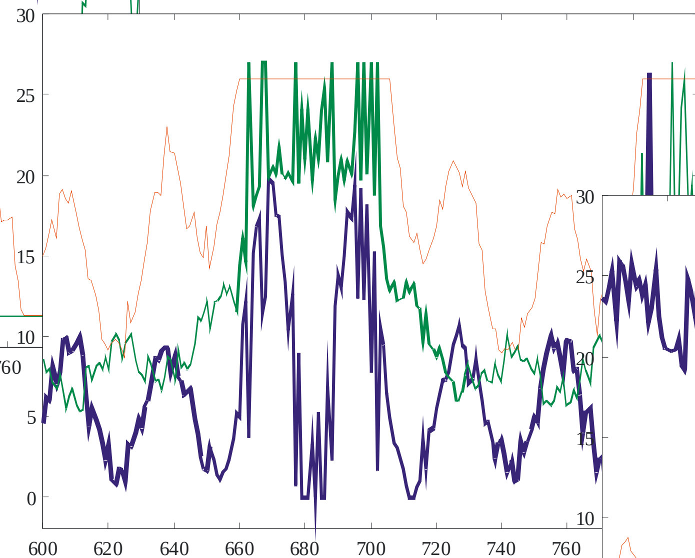
- Explicit optimization: Piecewise linear cost criterion means that only one of the producers is active at a time, others being in either of the extreme values (zero or maximum)
- Explicit distribution: Profiles φ define (randomly) preferred consumers for each producer; further, some plants can be "spare plants" to substitute malfunctioning master plants
- Cybernetic strategy: Profiles ϕ are determined by the correlation structures among consumers; because of the nonlinearities, there exist various minima to choose from

Quadratic optimum is rather flat – solutions in the vicinity of optimum are not much worse

Behaviors of 3 producers $(\bar{x}_i)$

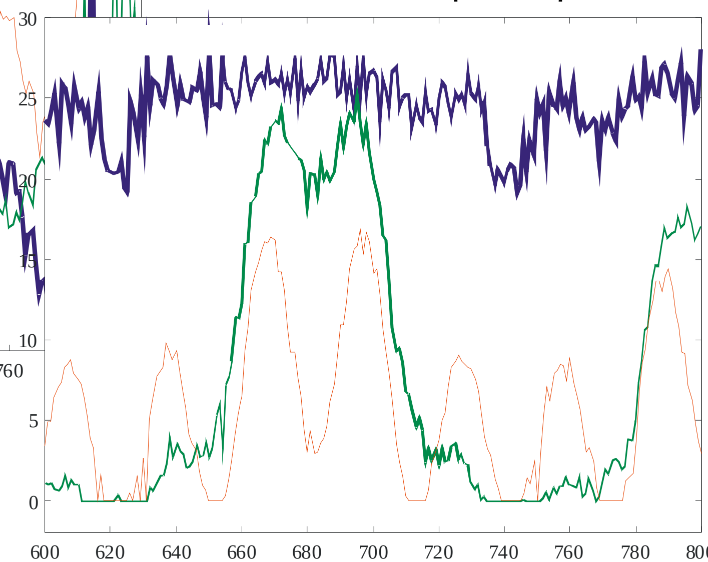


"Optimized"



Distributed

Cybernetic local balance
near the explicit optimum



Cybernetic

- There is a plenty of variation in the cybernetic case, but the variations are small = **robust?**

Technical networks in general

- Typically, the nodes in practical networks are not identical – they can have different roles, and these roles have to be taken into account in modeling
- The networks themselves are also very different:
  - In Internet, the "raw material" can be produced and copied indefinitely, restrictions and costs coming from transfer capacity
  - In power production, on the other hand, energy transfer is no problem, capacity restrictions and costs being caused in production
  - Still, the same modeling approaches can be applicable in both cases applying the idea of *dual graphs*?
- Possible applications: steam (pressure) pipelines in paper mills; design of electric networks with varying loads, etc.

“Agents”

- Traditionally, *agents* are software constructs
- Only computation is distributed, while control of operation is still centralized
- Higher-level coordination
- Nothing “unanticipated” can emerge here?

- Agents are the mainstream conceptual framework in AI today – something new now?

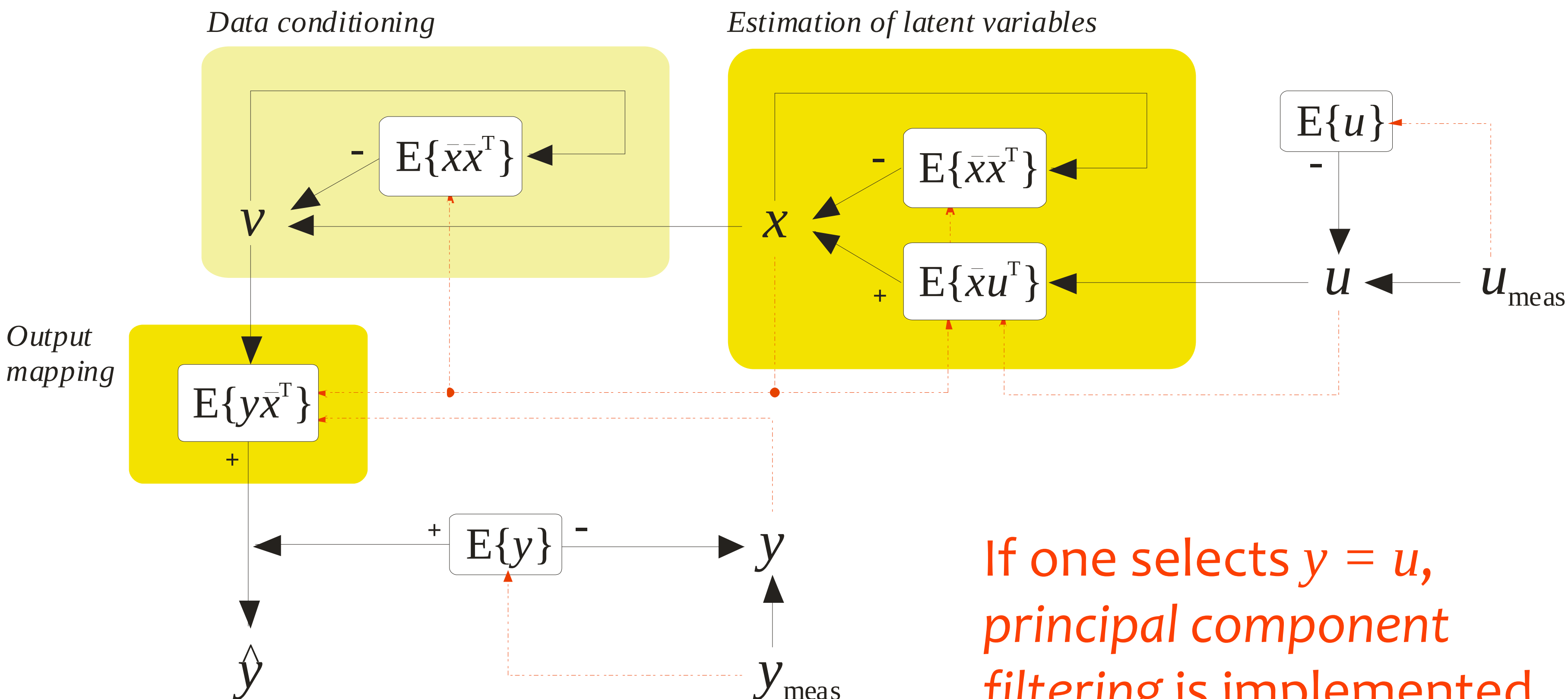
Distributed filtering

- How to apply the cybernetic ideas in modeling of agents?
- Now: direct application of the neural intuitions
- Example: *Filtering of noisy measurements delivered by a network of distributed "social" sensors*

- Easy case: The variation-orientation can be naturally motivated in the "sum-of-squared-errors" oriented framework
- No coordination is needed: High-level functionality (filtering) emerges from low-level actions
- "Data-based data reconciliation" implemented in distributed manner: Correlations among sensors are utilized to construct a system model and state estimator to eliminate noise

"Emergent coordination in distributed sensor networks", presented during *STeP 2004 – Web Intelligence*

- Implementation of *principal component regression* (PCR):



If one selects $y = u$, *principal component filtering* is implemented

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology
Cybernetics Group

$$\hat{u} = \hat{E}\left\{xu^T\right\}^T \hat{E}\left\{xx^T\right\}^{-1} \hat{E}\left\{xx^T\right\}^{-1} \hat{E}\left\{xu^T\right\}\left(u - \cancel{\hat{E}\{u\}}\right) + \cancel{\hat{E}\{u\}}$$

- Principal component filtering can be implemented as

$$\frac{dx}{dt} = -Ax + B\left(u - \cancel{\hat{E}\{u\}}\right) \qquad \frac{dv}{dt} = -Av + x$$

where

$$\frac{dA}{dt} = -\lambda A + \lambda x x^T \qquad \frac{dB}{dt} = -\lambda B + \lambda x \left(u - \cancel{\hat{E}\{u\}}\right)^T$$

variables x and v having much faster dynamics than u, variables u behaving much faster than the covariances

- After these are known one can calculate

$$\hat{u} = B^T v + \cancel{\hat{E}\{u\}} = B^T A^{-1} x + \cancel{\hat{E}\{u\}}$$

Luckily, variance (error-squared) criterion is natural in applications

- In practice, it is reasonable to only model *changes*, not absolute values
- To reach this, differentiated signals can be applied – but to avoid noise, there has to be cut-off frequency:

$$G(s) = \frac{\tau s}{1 + \tau s} \qquad \text{so that} \qquad u(s) = G(s)\, u_{\text{meas}}(s)$$

- To reconstruct final estimates, integration is needed as the final step
- To avoid bias problems, "leaking integrator" can be applied:

$$\frac{d\, \hat{u}_{\text{meas}}}{dt} = \hat{u} + \varepsilon\left(u_{\text{meas}} - \hat{u}_{\text{meas}}\right)$$

- In practice, it is of great importance to know also about the validity of the estimates
- Can condition monitoring be distributed?
- The validity of the PCA model can be measured in terms of two quantities:

1. T^2 statistic, measuring the fit “inside” the model:

$$T^2 = x^T \mathrm{E}\left\{\overline{xx^T}\right\}^{-1} x = x^T v = x_1 v_1 + \cdots + x_n v_n$$

2. Q statistic, measuring the fit “outside” the model:

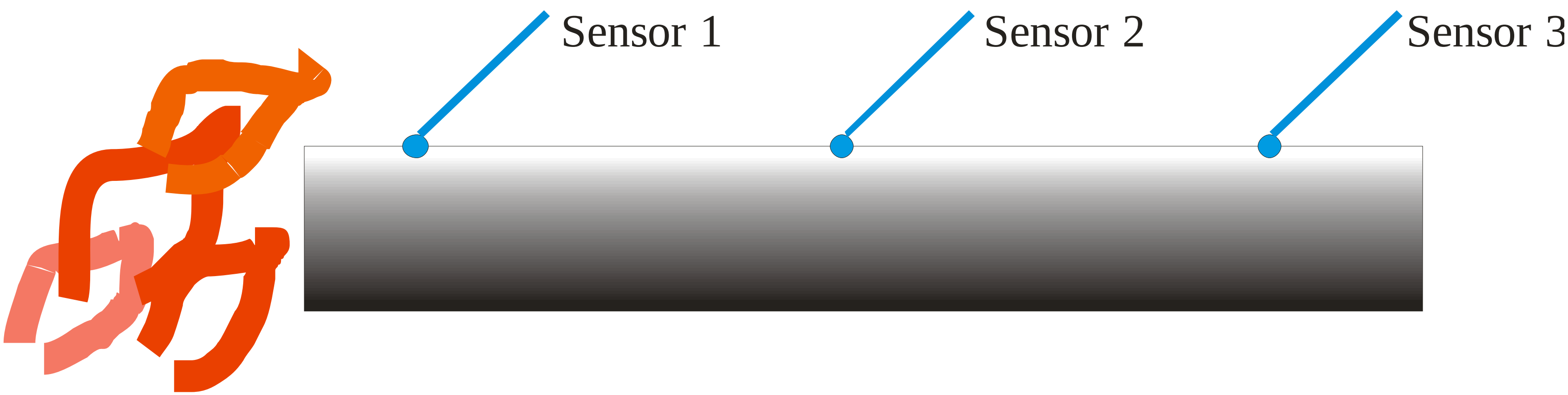
$$Q = (u - \hat{u})^T (u - \hat{u}) = (u_1 - \hat{u}_1)^2 + \cdots + (u_m - \hat{u}_m)^2$$

- It seems that both of these can be calculated nodewise.

Simulation: Heat transfer

- Typical processes are infinite-dimensional, cannot be modeled exactly applying finite-dimensional models

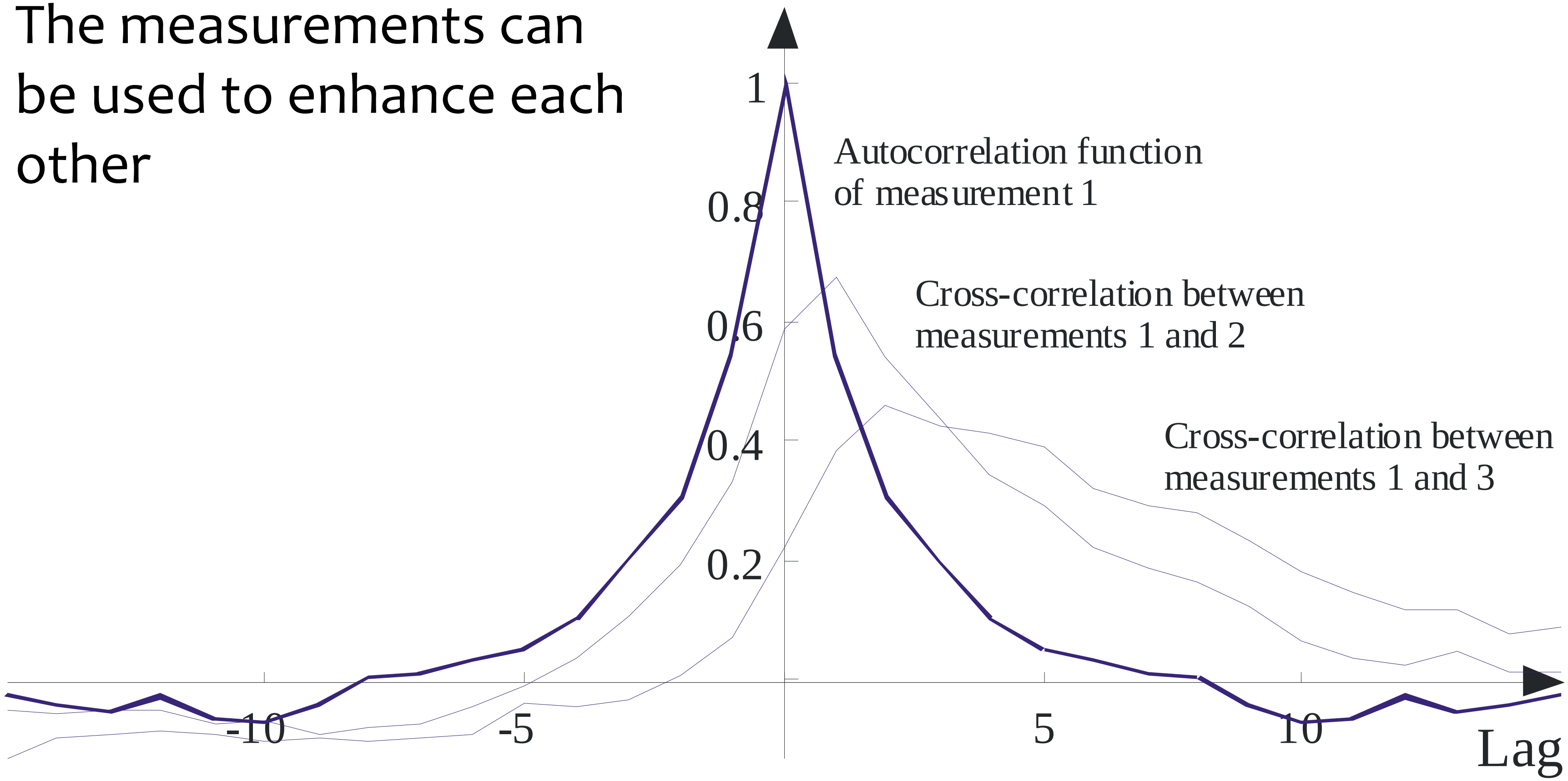
- How to determine the process state appropriately?
- How to enhance the measurements utilizing that state?



Correlations between measurements

- There exists correlation:

  The measurements can be used to enhance each other



Incomplete correlation structures

- Computations can be distributed in nodes
- Nodes transmit “corrections” to each other

$$
A = \left(\begin{array}{c|c|c}
\bullet & {\scriptstyle\bullet} & \cdot \\ \hline
{\scriptstyle\bullet} & \bullet & {\scriptstyle\bullet} \\ \hline
\cdot & {\scriptstyle\bullet} & \bullet
\end{array}\right)
$$

- $\bullet$ Local model
- ${\scriptstyle\bullet}$ Chain model also
- $\cdot$ Global model only

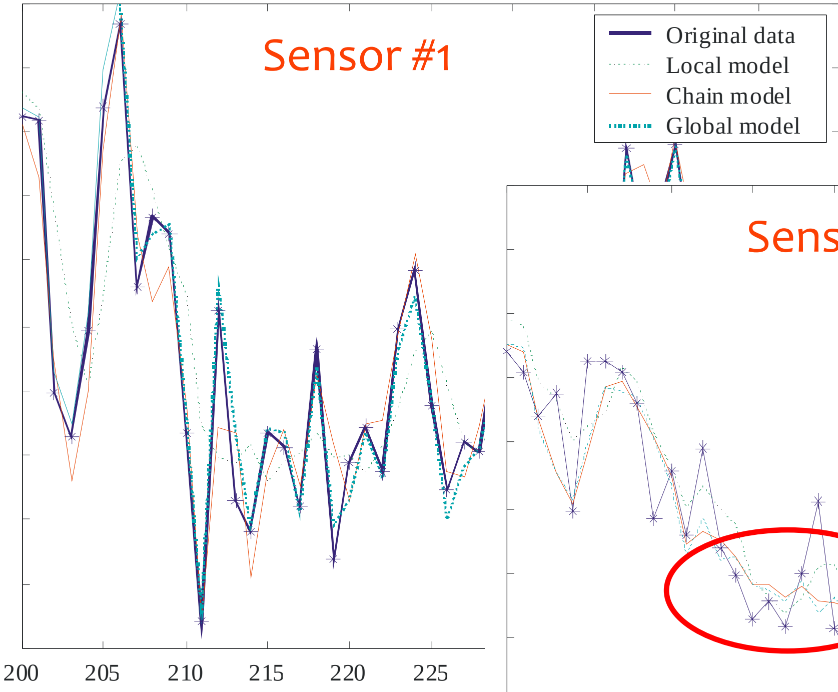
Non-zero non-diagonal blocks = communication among nodes needed

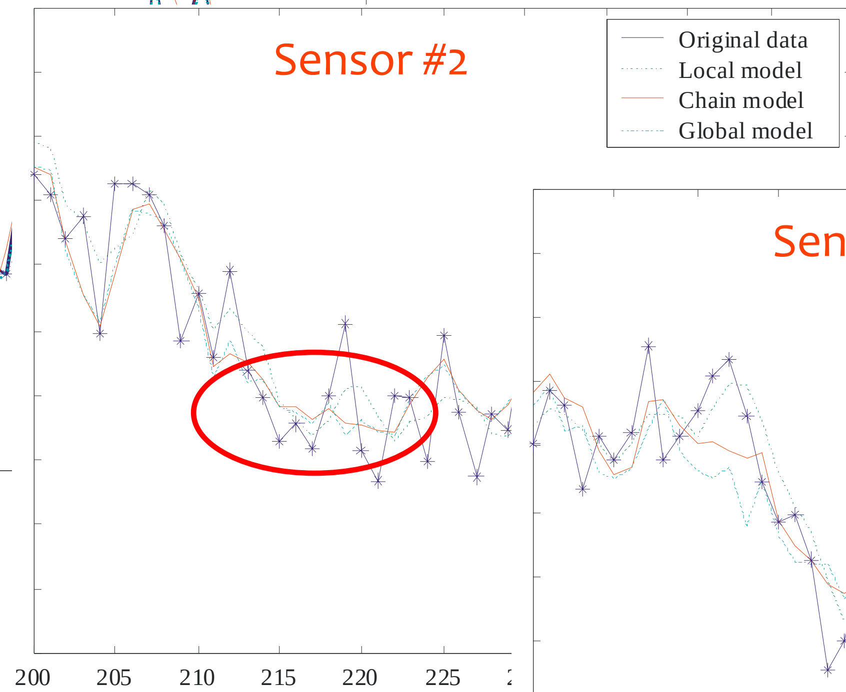
$$
B = \left(\begin{array}{ccc|ccc|ccc}
\bullet & \bullet & \bullet & {\scriptstyle\bullet} & {\scriptstyle\bullet} & {\scriptstyle\bullet} & \cdot & \cdot & \cdot \\ \hline
{\scriptstyle\bullet} & {\scriptstyle\bullet} & {\scriptstyle\bullet} & \bullet & \bullet & \bullet & {\scriptstyle\bullet} & {\scriptstyle\bullet} & {\scriptstyle\bullet} \\ \hline
\cdot & \cdot & \cdot & {\scriptstyle\bullet} & {\scriptstyle\bullet} & {\scriptstyle\bullet} & \bullet & \bullet & \bullet
\end{array}\right)
$$

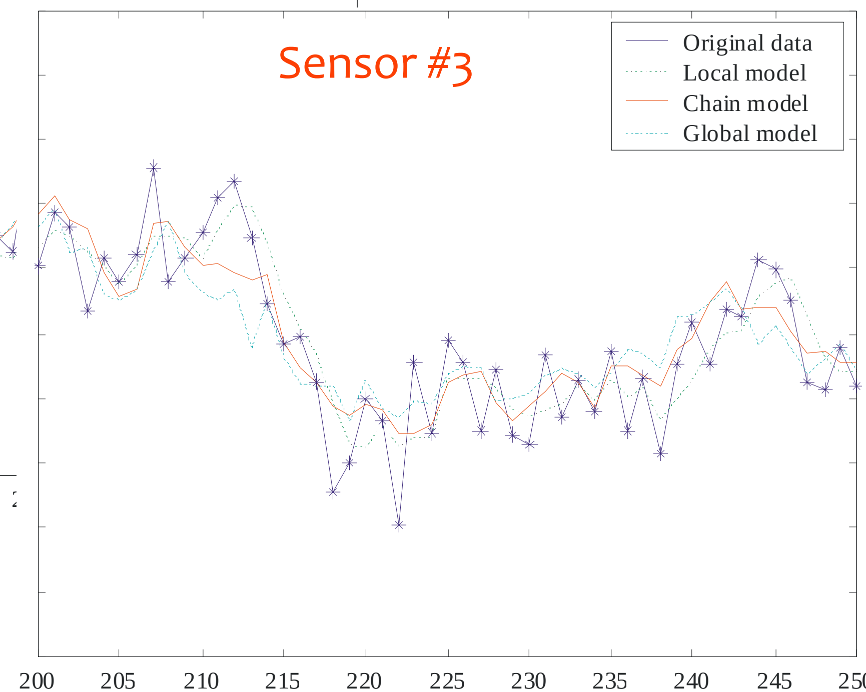
$$
u_{\mathrm{in}}(t) = \left(\begin{array}{c}
u_1(t) \\ u_1(t-k) \\ u_1(t-2k) \\ \hline
u_2(t) \\ u_2(t-k) \\ u_2(t-2k) \\ \hline
u_3(t) \\ u_3(t-k) \\ u_3(t-2k)
\end{array}\right)
$$

Nodes far apart do not necessarily know about each other – resulting in implicit structuring of the model

- Implemented in discrete time to avoid bandwidth problems







- Global model – clumsy and high-dimensional
- Local model – simple filtering of data – smooth but always late if there are trends in behaviors

Experiences

- If sensors are fully connected, "trivial" (distributed) principal component regression functionality is obtained
- More interesting results are reached if the network is not fully connected ("chain" structure above)
- Incomplete, localized information results in *better* estimates
- The local models are lower-dimensional: Only appropriate information is present, resulting in fast adaptation, and enhanced robustness
- As compared to mainstream approaches to distributed sensors, now one has overlapping "fuzzy" clusters of sensors
- There does not exist global-level sparse optimality criterion – distributed structure has theoretic, not only practical interest

Conclusion this far

- Intuitions on "natural robustness":
  - Uncorrelatedness of profiles prevents cascading failures in networks?
  - Only "real" changes in signal patterns are reacted to
  - Noise in the system attenuated, rapid variations reduced
  - Model behaviors are *natural*
- Intuitions on distributed agents:
  - Localized information, "fuzzy clusters", low-dimensional data
  - No global design criterion exists!?
  - The above applied to "social" agents with complicated (twofold) communication structure among nodes; clearly, population of "selfish" agents would make it simpler?
  - However, there are complications ...

System-wide fractal structure of stabilities means *robustness?*

More sophisticated cases: Actuation added

- Remember that applying the *selfish agent* feedback can be implemented implicitly through the environment
- If measurements affect the environment directly, this makes it possible to implement complete locality in control

  - **Scenario 1:** There is a tank of unevenly stirred liquid; there exist various sensor/actuator units (controlling concentrations / temperatures)
  - Learning of action / reaction dependency results in PCR-type control
  - Problem above: Effects of action follow only after a delay when diffusion has taken place; there are also more straightforward application examples –
  - **Scenario 2:** There is a thin flexible (steel) plate whose deformations are to be actively compensated (after all, passive components suffice)
  - Combined local measurement and control: Now the control effects become effective in a delayless manner – *stiffness* increases (see Lecture 11)

- In some applications, one can readily apply the cybernetic “selfishness”
- For example, assume that there are cellular phones connected to a link
- The goal is to utilize the available spectrum (resource) so that the total power would be minimized
- There is negative feedback through the environment as the neighboring cellulars cause interference; the stronger one transmits, the more it worsens others’ signal-to-noise ratio.

Embedded neocybernetic controls

- Before, control theory was applied for understanding of cybernetic systems; now, cybernetic understanding is applied to study explicit, agent-based control systems

- The traditional ways to attack complex control systems are either based on SISO techniques (for example, RGA tries to couple inputs and outputs appropriately), or on sophisticated multivariate techniques, where explicit models are needed
- Now, study the possibilities of constructing cybernetics-type data-based distributed controllers and actuators … after all, neocybernetic systems are control systems, one just has to employ these effects so as to reach one's objectives

Challenges

1. In a control system individual signals and actions are relevant; transients are important, one cannot simply study the system in the statistical way, concentrating on the emergent phenomena like power spectra, etc., alone
2. The agents are not identical; they have their individual predestinated roles as determined by the system structure. What is more, the controllers are not explicitly connected to each other, interactions take place through the environment
3. Explicit knowledge of control effects are needed, containing the actual system dynamics; this means that there are problems related to *causality* that need to be explicitly attacked.

About causality

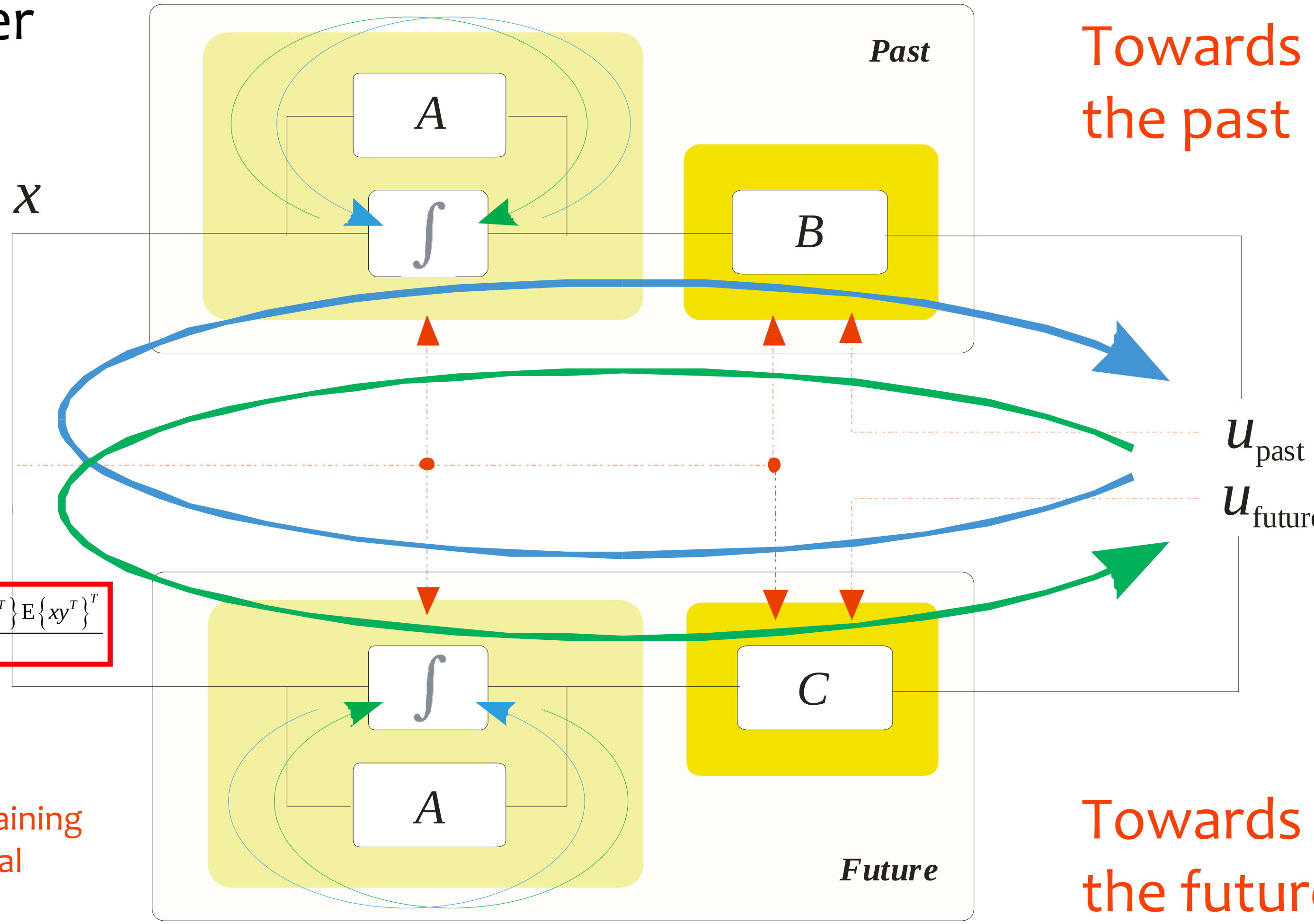
- Above, the measurement agent system was rather simple: The actuators affected mainly corresponding measurements
- In control systems, one tries to affect behaviors farther away: There needs to exist knowledge of action/reaction dependencies – causal structure is needed
- Then one faces the old Humean reality: *Causalities cannot be seen in data, only knowing correlations is not enough*
- The causal structure has to be given a priori, determining the temporal or spatial precedences between signals
- One needs to distinguish (local!) input and output variables
- The models need not only tract what has been observed but also *what can be done* – how to reach this?

Vision

- Cybernetic systems implement balance on various levels – what a tempting control view!?
- Now correlations between input and output are modeled, controller implementing the latent intermediate variable
- The model predicts behaviors in the output based on input; when a "–" is added, one implements negative action, the *model trying to suppress expected deviations*
- The model is a balance between the past and the future
- The model is a mirror image between the past and the future (however, now *antisymmetric* image)
- Intuitively appealing idea: *Symmetry pursuit*

Determination of an input/output model

- "PLS" rather than PCR?



$$\boxed{\mathrm{E}\{xx^T\}^3 = \frac{\mathrm{E}\{xu^T\}\mathrm{E}\{uu^T\}\mathrm{E}\{xu^T\}^T + \mathrm{E}\{xy^T\}\mathrm{E}\{yy^T\}\mathrm{E}\{xy^T\}^T}{2}}$$

Towards the past

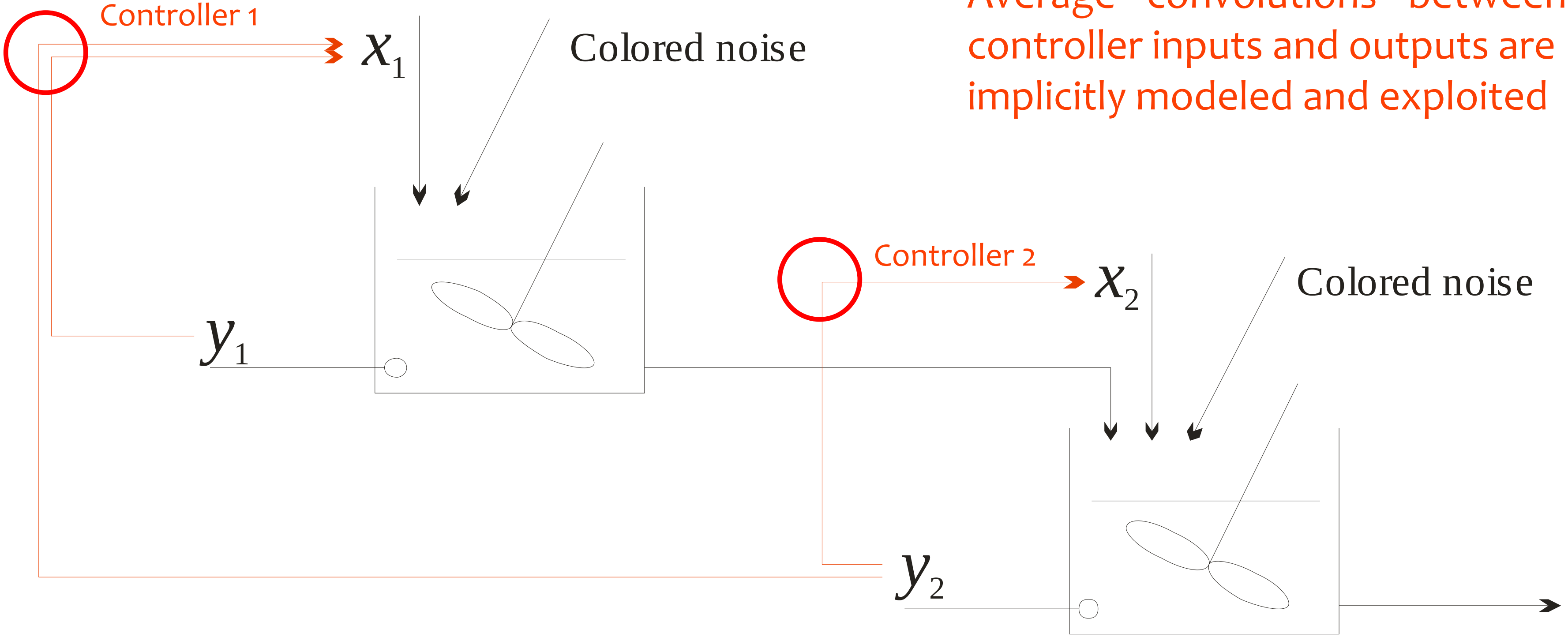
Towards the future

Two-directional in training phase, one-directional when being applied

Symmetricity exploited here!

Example process

- Two multivariate controllers trying to eliminate the disturbances entering the system



Average “convolutions” between controller inputs and outputs are implicitly modeled and exploited

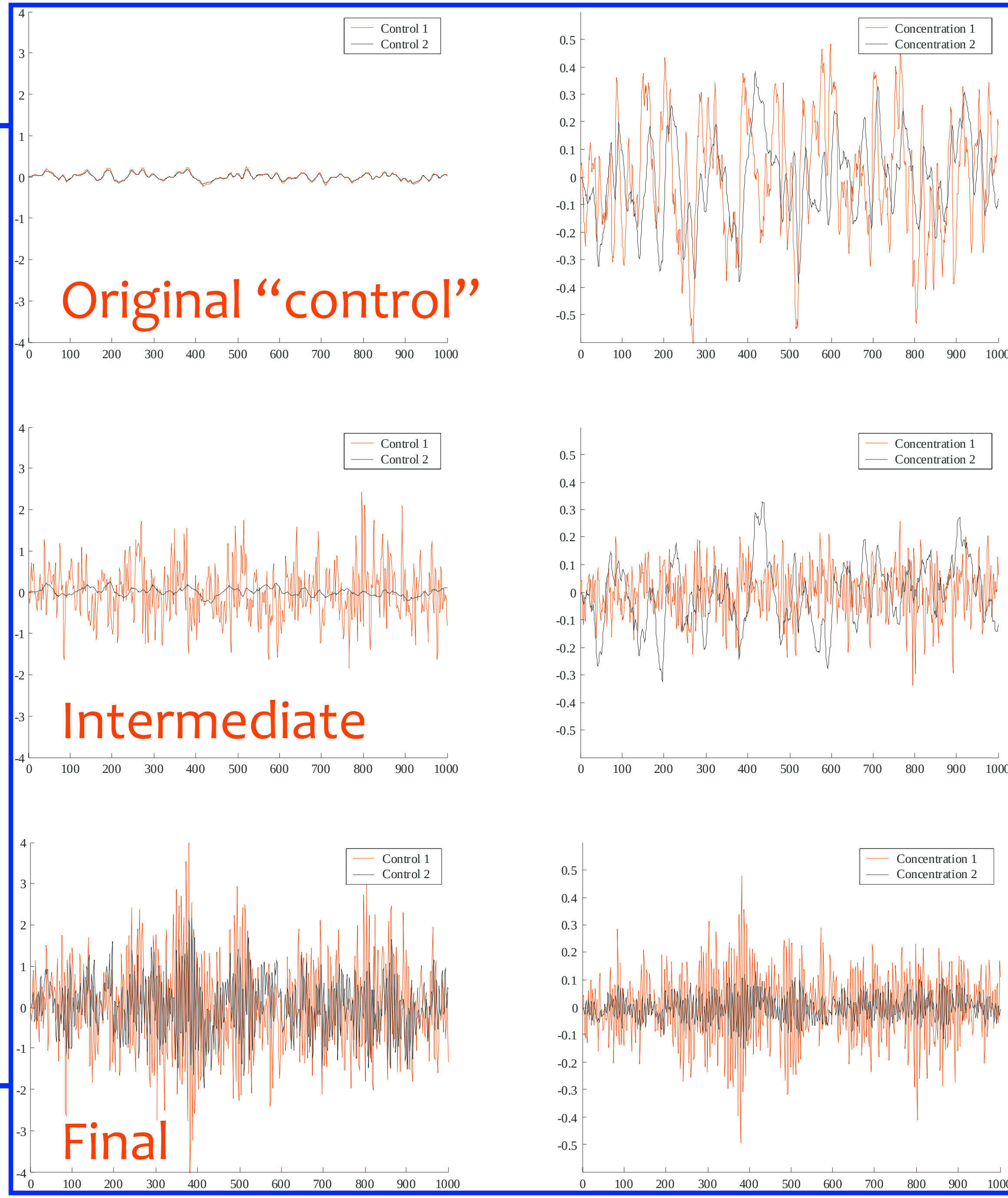
- Control signal = latent variable between the past and the future
- Assume causality is OK = Correct signals + correct signs (otherwise explosion takes place during adaptation)

$$\begin{pmatrix} y_1(k+3)\\ y_1(k+2)\\ y_1(k+1)\end{pmatrix} \leftarrow u_1(k) \leftarrow \left(\begin{array}{c} y_1(k)\\ y_1(k-1)\\ y_1(k-2)\\ \hline y_2(k)\\ y_2(k-1)\\ y_2(k-2)\end{array}\right)$$

$$\begin{pmatrix} y_2(k+3)\\ y_2(k+2)\\ y_2(k+1)\end{pmatrix} \leftarrow u_2(k) \leftarrow \begin{pmatrix} y_2(k)\\ y_2(k-1)\\ y_2(k-2)\end{pmatrix}$$

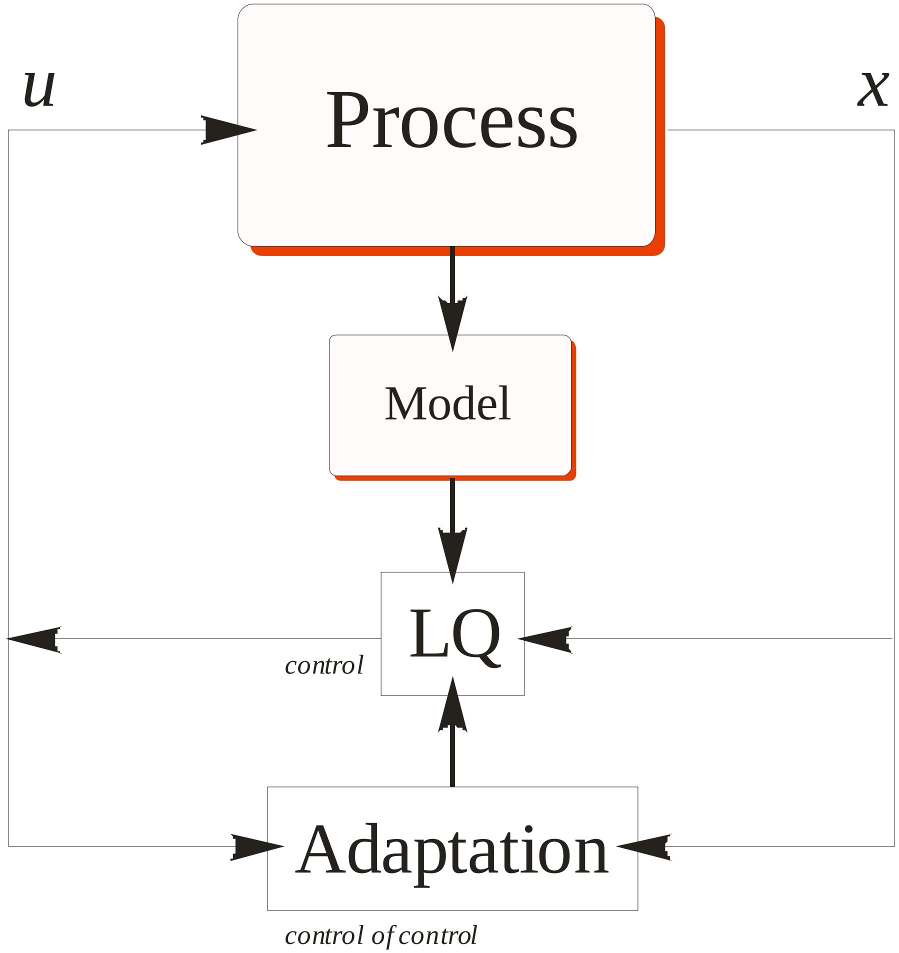
Results

- Advantage (?): Unique to an extent, there are no weighting factors, etc.
- Highly iterative adaptation
- Final behaviors very (too) fast
  - Compare to *dead-beat control*, where, too, deviations are similarly eliminated "instantly" (brute f.)



“Second level” control?

- In neocybernetics, it is assumed that the underlying system is already in balance, there are inner control loops in action
- If this cannot be assured for all neocybernetic adaptations, the closed loop system can explode
- But one can implement neocybernetic strategy over some existing control structure, *adapting the control parameters*
- New adaptive control scheme: Assume there is criterion with weighting matrices Q and R (and perhaps cross-term S)
- Implement negative feedback from signal variances to new, diagonal weighting factors Q_{ii} and R_{jj} + implement LQ control
- *Robust control* is reached where signal properties in varying operating conditions are utilized to map the parameter space



- Adaptation of the control consists of the neocybernetic formula:

$$u' = \left(\frac{\sqrt{Q}\ \ \mathrm{E}\{|x|\}}{\sqrt{R}\ \ \mathrm{E}\{|u|\}} \right)$$

$$x' = q\ \mathrm{E}\left\{x' u'^T\right\} u'$$

$$\begin{pmatrix} Q_{11}' \\ \vdots \\ Q_{nn}' \\ \hline R_{11}' \\ \vdots \\ R_{nn}' \end{pmatrix} = x'$$

To be scaled to unit length (only relative weights relevant)

"Stiffness regulation" of the neocybernetic strategy equalizes all $Q_{ii}\mathrm{E}\{x_i x_i\}$ etc.!

Generalization

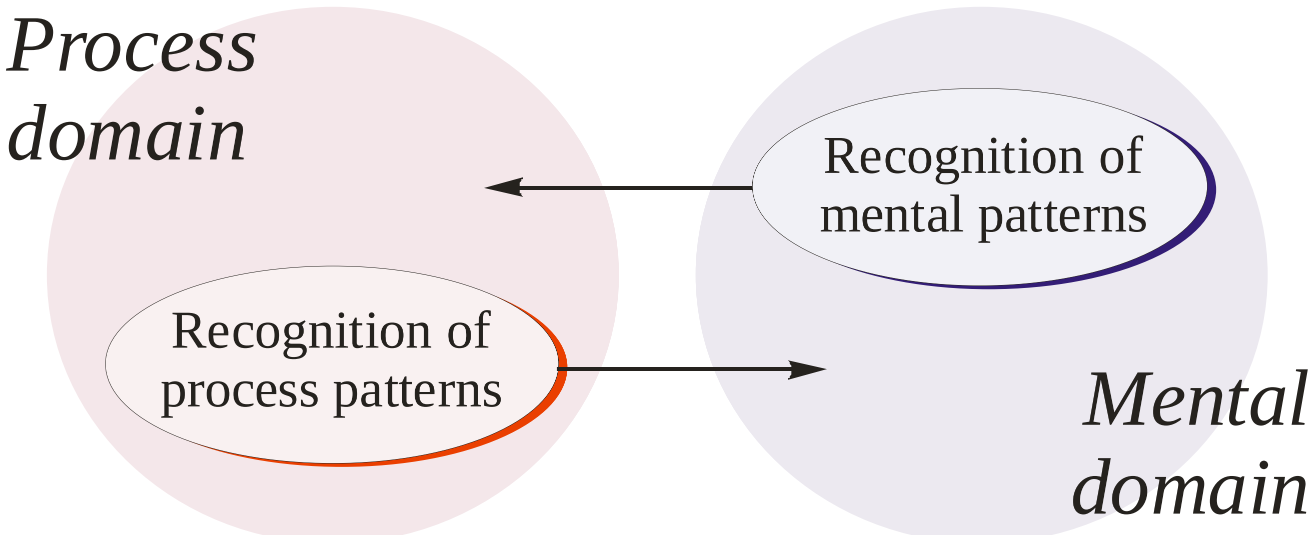
- The above case is an example of a more general principle ...
- If only there exists some kind of negative feedback from the "investments" to the "losses" (above, from LQ parameters to variation levels), no matter how complicated the dependency is, or how it is implemented, applying the adaptation principle

$$\bar{x} = q\mathrm{E}\left\{\overline{xu}^T\right\}\bar{u}$$

between the losses u and investments x, the resulting closed-loop behaviors are finally forced to follow the linear subspace model: *Losses are equalized*, and within the investment space, a linear DOF structure emerges that reflects the (smooth) "operating regimes" of the system.

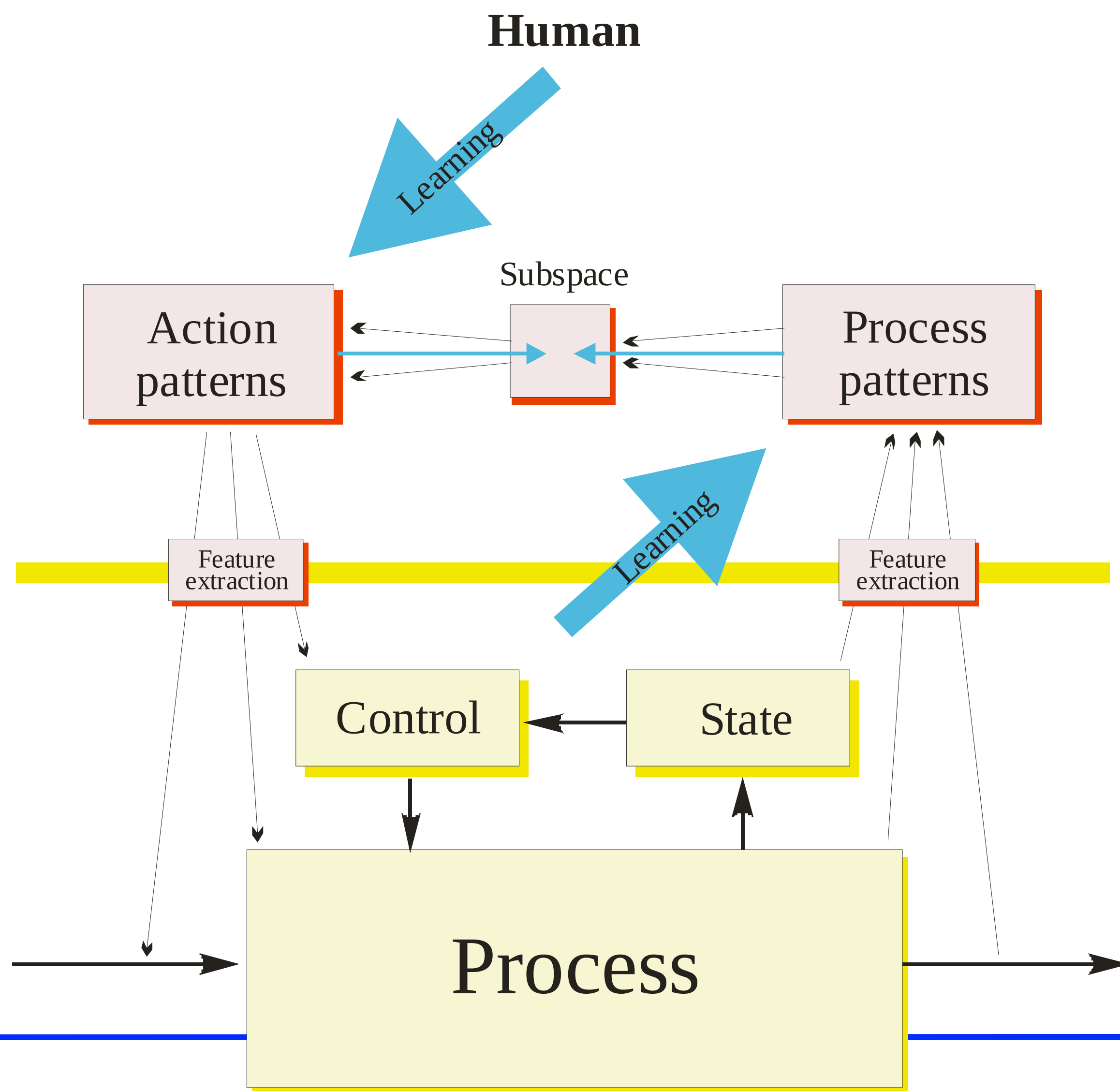
“Second-Order Control”

- Often in complex automation systems, it is *humans* who implement the highest-level controls
- One has to apply “second-order neocybernetic” thinking



Routines can be automated

- In larger plants, there are various “patterns of operation” – can be done using sparse coded features
- How to implement “expert model” – see Lec. 10



- Above, degrees of freedom were employed to find the “controller controller” models
- In general, DOF’s seem to offer many approaches for applications in control engineering
- DOF based pattern recognition and subsequent system control in a high-dimensional measurement space would be, of course, a straightforward application ...

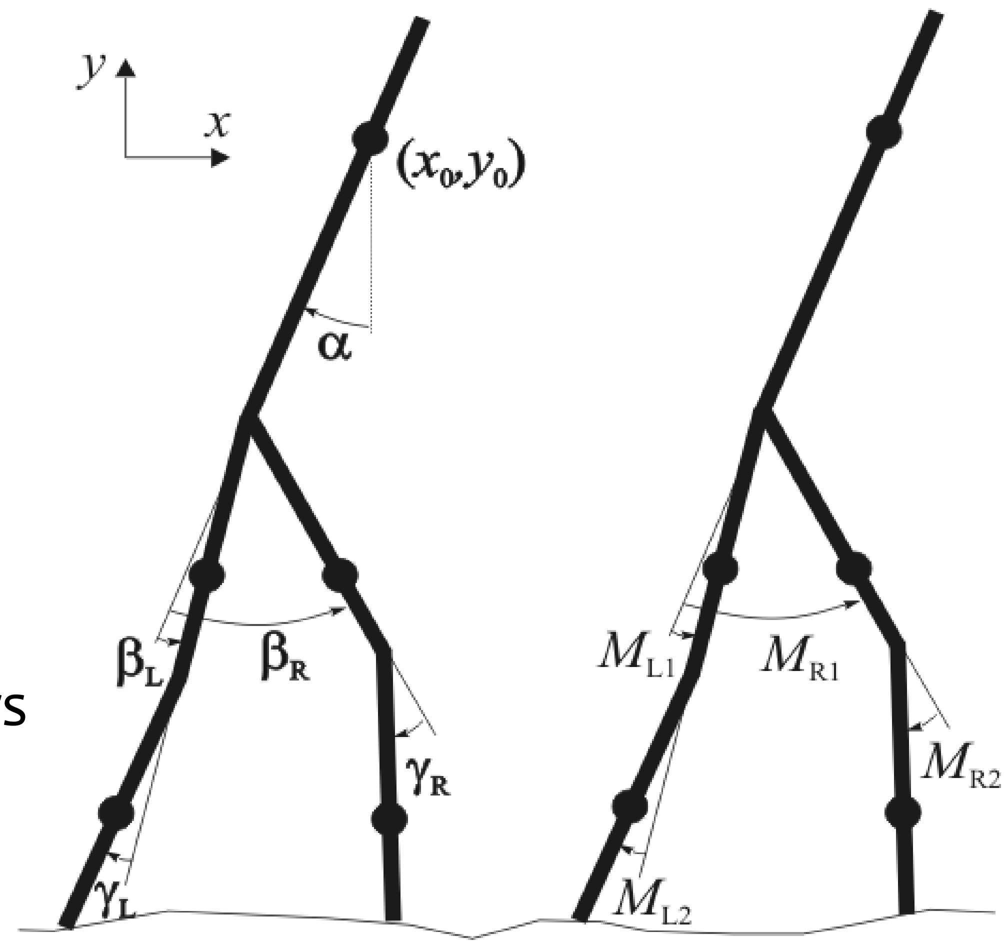
- Below, possibilities of DOF based piecewise linear control are studied in a truly challenging case – controlling of the walking of a two-legged unstable “robot”
- This was part of Olli Haavisto’s diploma work.

Clustered regression structure

- Piecewise linear regression model
- Data is divided into *clusters*, each belonging to an *operating point*
- Local *principal component PCR regression* model attached to each cluster
- Total model estimate is a combination of the relevant (nearest) local model estimates

Robot simulation model

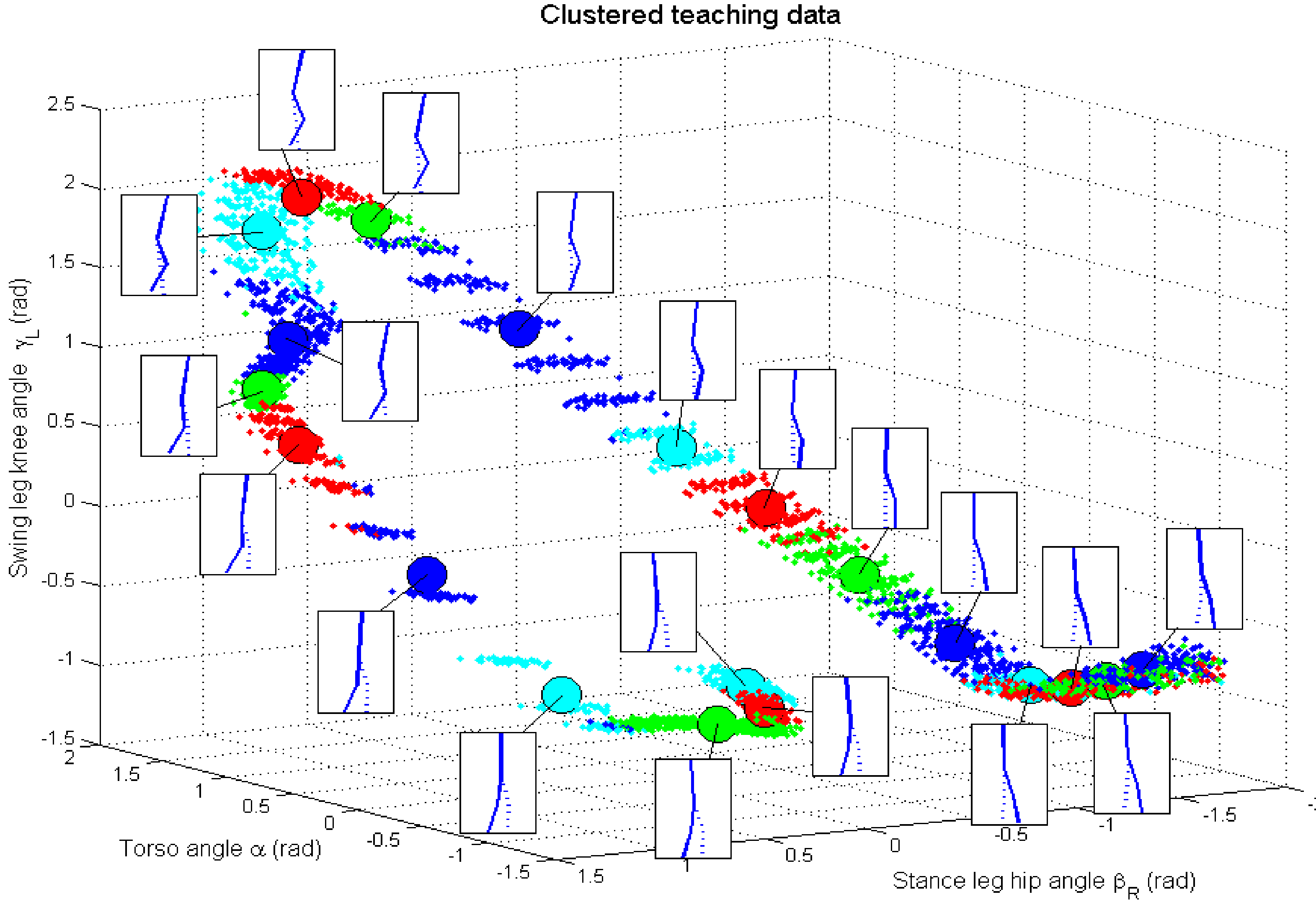
- Walking biped robot in 2D plane
- Input: moments M
- Ouput: state of the system
- Simulated in Matlab/Simulink using the exact dynamic equations
- Sample gait produced by four ordinary PD controllers



Modeling the inverse dynamics

- Goal was to form a regression mapping from the current biped state to the next control (moment) vector
- Clustered regression was applied to the sample data collected from the PD controlled gait
- 20 clusters were used
- State dimension reduction to 8 principal components in each local model
    - only relevant information in the data modeled
    - model simplification
    - noise removal

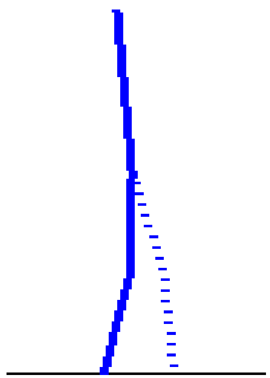
Locally linear submodels



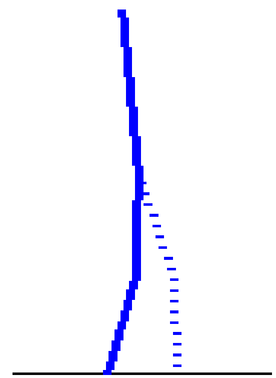
Clustered regression control

- The moment estimate corresponding to the current state was used as a new control value
  - No reference signals or additional controllers needed
  - The gait is stored inside the model
  - Sample gait can be reproduced quite well
- Comparing the behaviours (animation):

Sample gait



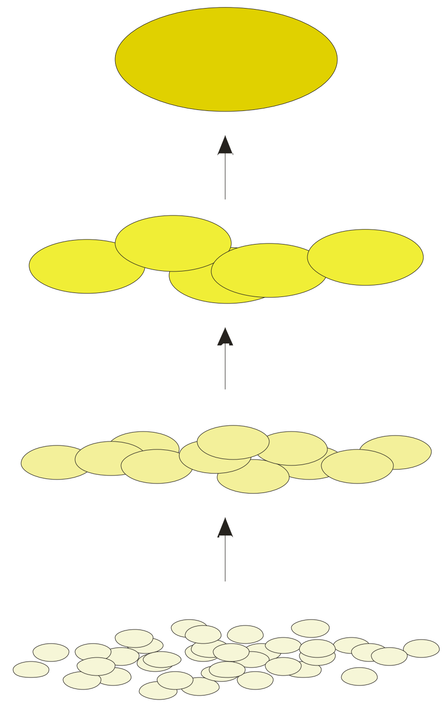
Learned gait



Towards more challenging views

- **Semantic webs** are today's hot topic in Internet: that is, WWW is extended so that the *semantics* of information and services on the web is defined
- These *ontologies* are defined by humans … distantly, one is reminded of expert systems and what happened with them …
- In special environments, being based on mathematics, semantics ("good behavior", etc.) can often be quantified
- What is more, *relevance* issues can be addressed:
  - For example, theory says that identification algorithms generally converge to the correct parameter values; this means that very much research is done on such methods, even though *practical identifiability* is a very different thing
  - On the other hand, there are many soft computing methods that cannot be proven; however, they seem to work, thus offering a platform for applications

- In control systems, semantics (behavioral qualities) can be reduced to numbers
- Simulation – a consistent way upwards in the hierarchy
- “Trust” can be reached?



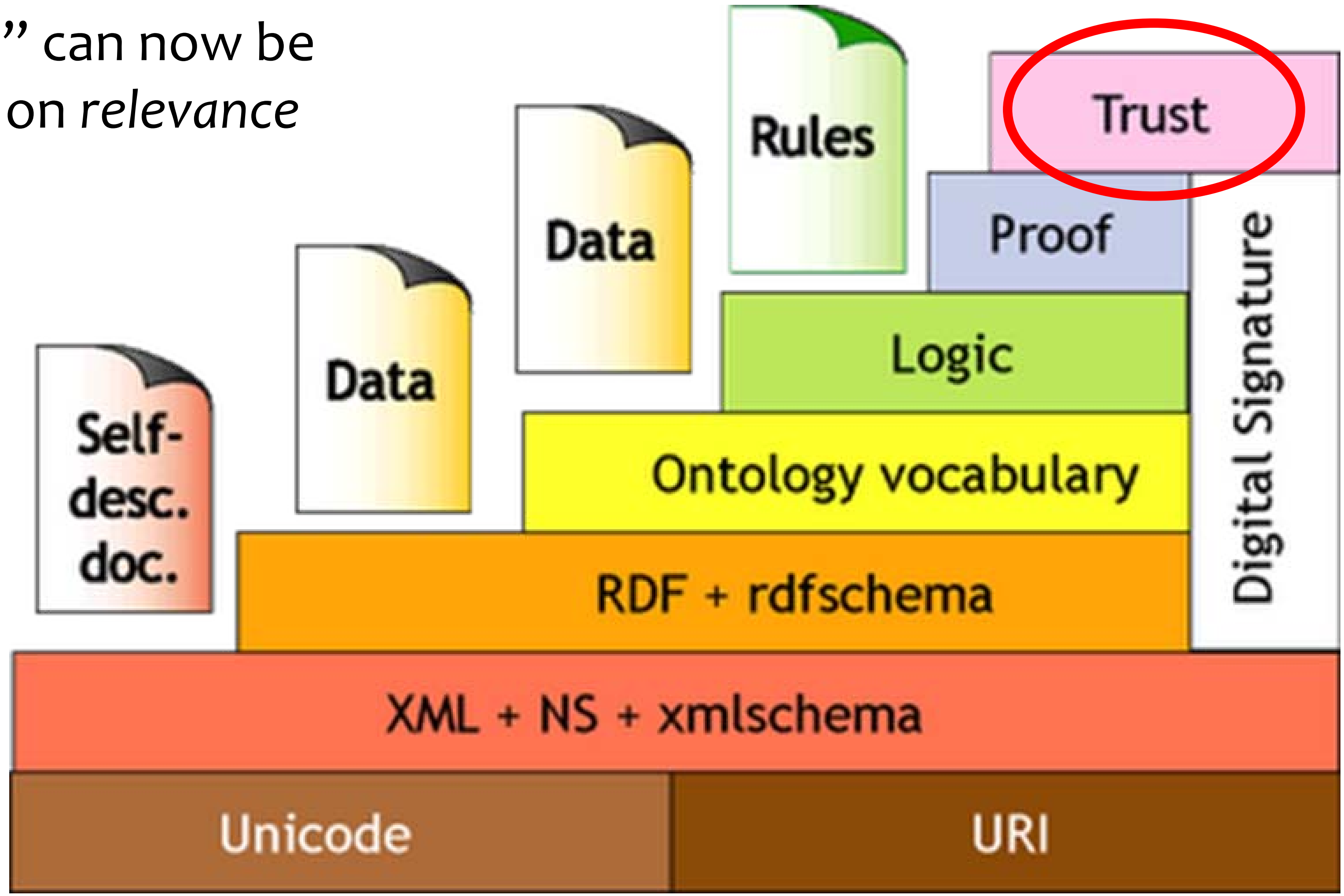
System behavior
Plant properties

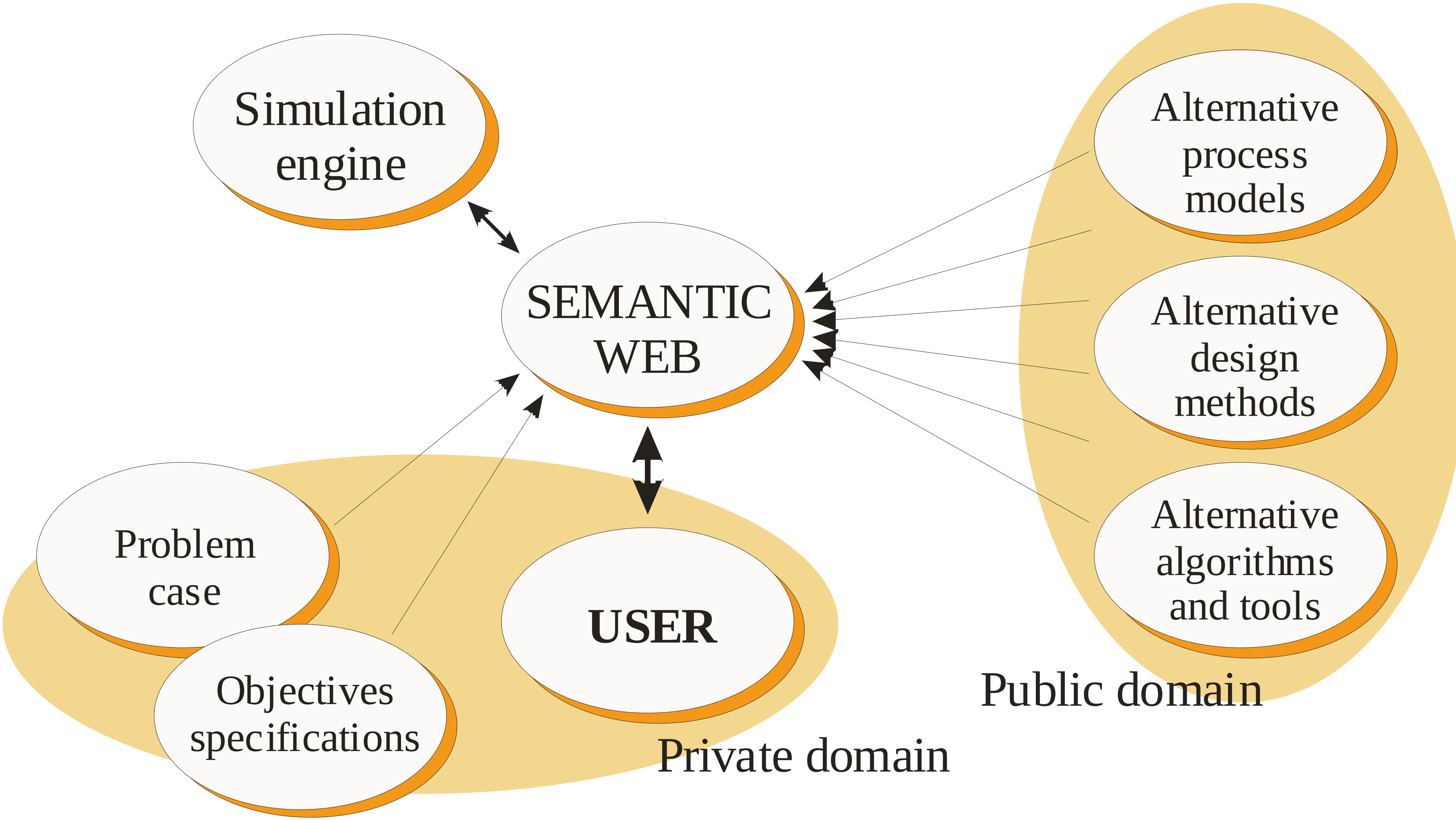
Stability
Gain Margins

Poles
Eigenvalues

Data structures
Numbers

- “Trust” can now be based on *relevance*




Simulation engine
SEMANTIC WEB
Alternative process models
Alternative design methods
Alternative algorithms and tools
Problem case
Objectives specifications
USER
Public domain
Private domain

Innovations through DOF's

- What can such "truly semantic" web perhaps do then?
- Again, in the space of semantics-loaded parameters, it can help finding the degrees of freedom, thus compressing and structuring the search space when doing design
- One does not have to blindly believe some "gurus" when one can test different approaches with one's own problem cases and specifications
- The Web can "extend the mind" (see Lectures 10 and 12!)

- Can such views be put in practice? – Yes, as seen next time: in narrow cases where "good behaviors" can be defined; this has already been done (doctoral thesis of Kalle Halmevaara)

Today’s challenges in industrial automation

- Earlier in the course: Populations, agents, networks, etc., were discussed – perhaps interesting in future automation?
- Today, the reality is still very different
  - Systems to be studied *are* structured and hierarchic
  - Underlying system/model structures are non-homogeneous
  - Methodologies and tools are deterministic and of SISO type
- What is more ...
  - Humans are to be discussed with and convinced in all phases
  - Pragmatical orientedness rather than theoretical innovativeness is valued
- Goal: Try to understand complex system dynamics – new methods have to match the old practices and *extend* them

Complex systems cannot be put up once and for all

Role of humans

- Humans are parts of control systems in many ways
  - They constitute the highest level of feedback controls
  - They implement the new systems
- Systems including humans are *automatically* complex

Holistic understanding of new model structures is needed

1. Methods must be capable of capturing heterogeneous subsystems (like humans) in the same framework
2. Methods must be understandable and related to old technologies to become accepted

- Homogeneity is needed: To become accepted, fluent transition between old and new modeling practices needed

Example

- Still today, almost all of the industrial controllers are of the *classical* PID type
- Why the *modern control* theory that outperforms classical did not flourish and never really became to factory floor level?
- Why the "*postmodern*" ideas (neural networks, fuzzy systems) are today so popular instead?
  - Promises: They "need no model", they "are simple and understandable"
- The operators, system developers, etc., consist of humans, constituting *cybernetic communities*
- Approval goes through understanding
- Learning is a constructivistic process – new things must be related to old practices

Conclusion

- To put it boldly –
  - *Mathematics* – is a strictly formal play within the world of constraints
  - *Arts* – is strictly intuitive, accepting no constraints whatsoever
  - *Engineering* – is to find the degrees of freedom when all real life constraints are taken into account, thus fulfilling the cybernetic evolutionary ideal
- Next time – about *evolutionary DOF's*