Wireless Control Systems: Scientific Challenges and Emerging Applications

Karl Henrik Johansson
School of Electrical Engineering
Royal Institute of Technology, Stockholm, Sweden

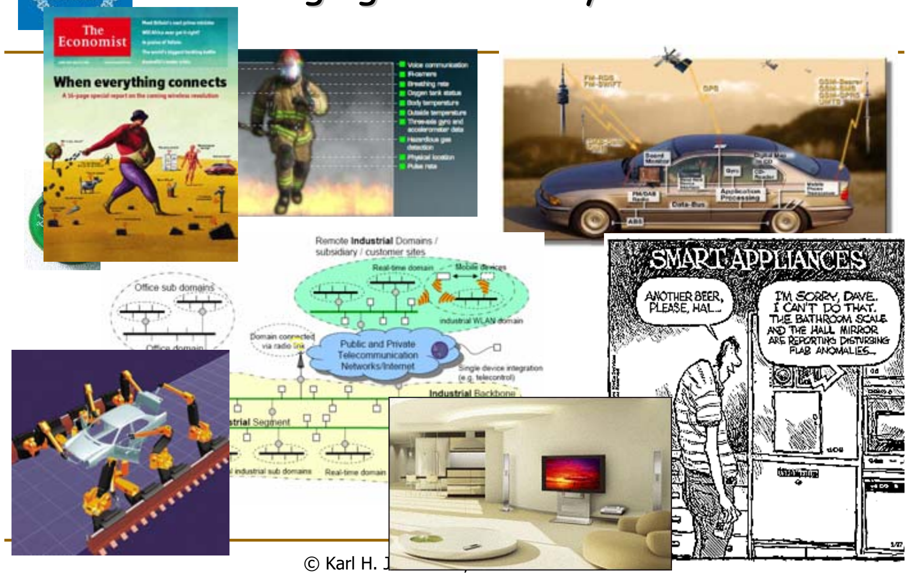
15th Mediterranean Conference on Control and Automation
Athens, Greece
27-29 Jun, 2007

Acknowledgements

With
- Dimos Dimarogonas, Carlo Fischione, Håkan Hjalmarsson, Krister Jacobsson, Henrik Jansson, Björn Johansson, Mikael Johansson, Magnus Lindhé, Niels Möller, Pan Gun Park, Niklas Pettersson, Maben Rabi, Per Sahlholm, Alberto Speranzon, Emmanuel Witrant, Sandro Zampieri etc.

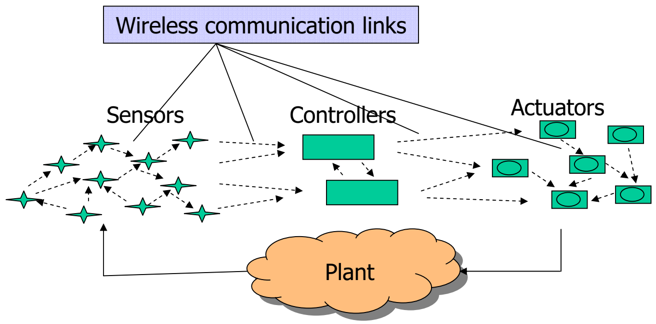
Financially supported by
- Swedish Research Council
- Swedish Foundation for Strategic Research
- Swedish Governmental Agency for Innovation Systems
- Swedish Programme Council for Vehicle Research
- European Commission IST 6th Framework Programme
  - RUNES Integrated Project
  - SOCRADES Integrated Project
  - HYCON Network of Excellence

An emerging world of dynamic networks



Wireless control system

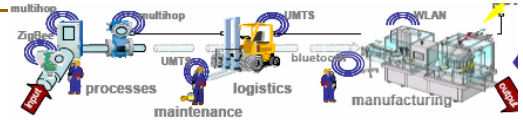
Distributed control system with sensor, actuator and controller communication supported by wireless network



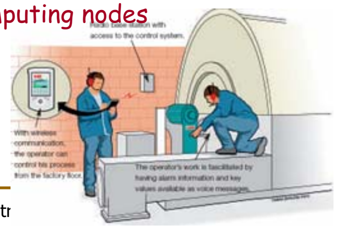
Outline

- Why wireless control
- Aspects of design
  - Layered **models** of abstraction
  - Global objective from local **controls**
  - Component-based **implementation**
- Conclusions

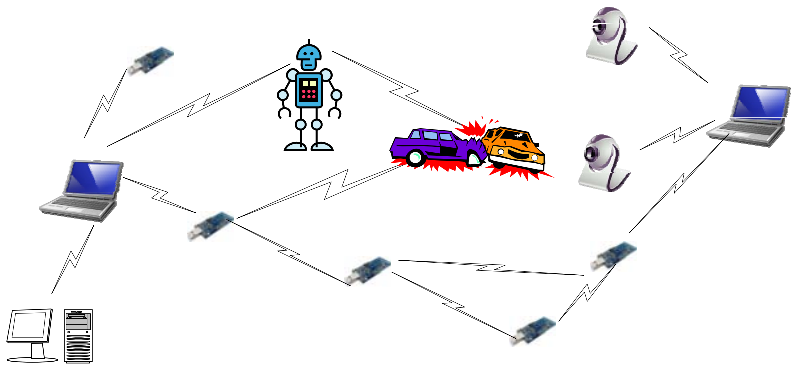
Benefits of wireless networking in industrial control



- Reduced installation and maintenance costs
  - Less cabling
  - Efficient monitoring and diagnosis
  - Fault-resilient connectivity
- Added flexibility
  - Better placed sensor, actuators and computing nodes
  - Enhanced maneuvers and control actions
  - Easier to reconfigure

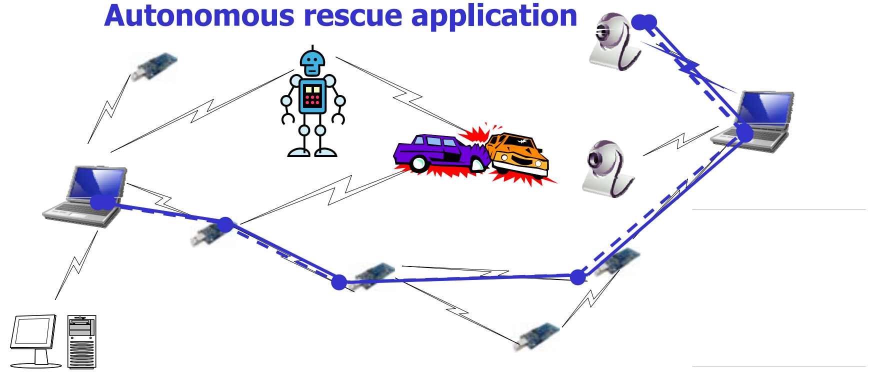


Layered models of abstraction

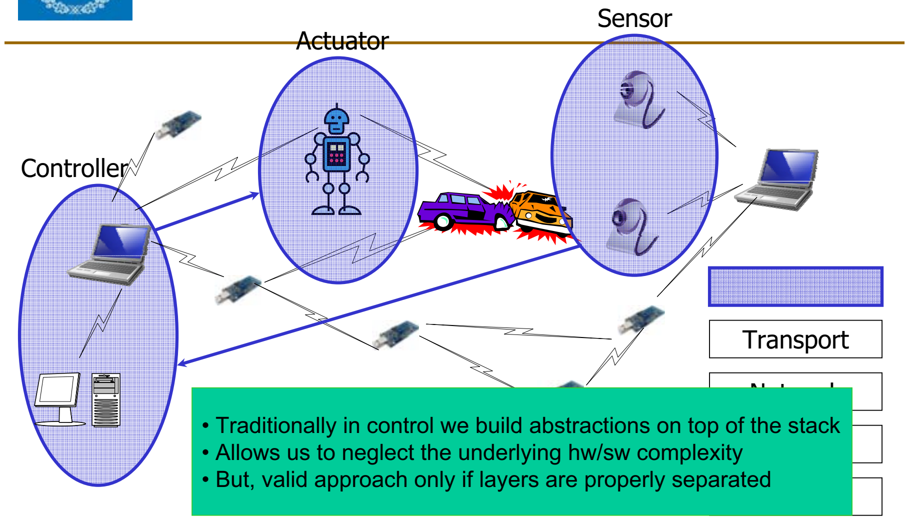


Communication abstractions

Autonomous rescue application



Control abstractions



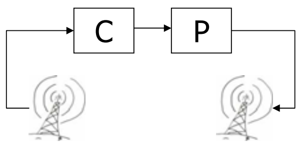
- Traditionally in control we build abstractions on top of the stack
- Allows us to neglect the underlying hw/sw complexity
- But, valid approach only if layers are properly separated

© Karl H. Johansson, Wireless control

What's the difference between wireless and wired control?

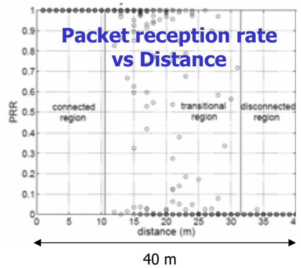
Wireless communication is more challenging than **wired**:

- Larger variations in connectivity, bandwidth, delays etc
- More explicit energy constraints
- Potential for mobility
- Less well-developed communication theory

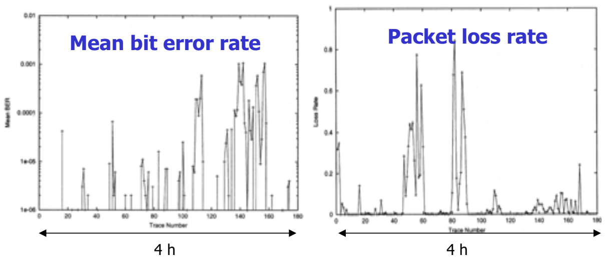


© Karl H. Johansson, Wireless control

Respect imperfect layer separation in wireless networks



Zuniga & Krishnamachari, *SECON*, 2004

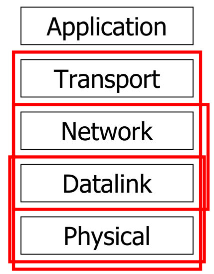


Willig et al., *IEEE Trans. Ind. Electron.*, 49, 6, 2002

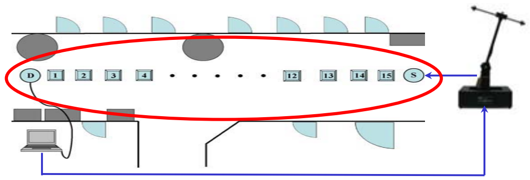
Park et al., KTH, 2007

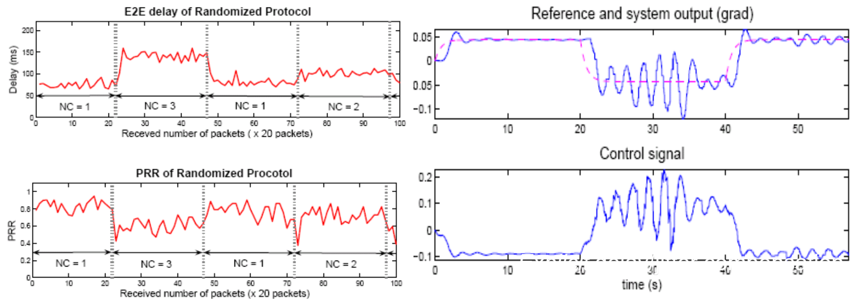
Large variations in

- Connectivity
- Bit and packet delivery
- End-to-end delivery



Influence on control performance



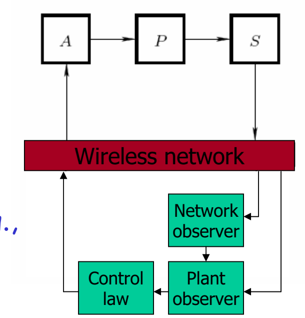


Witrant et al., *IEEE CCA,* 2007

A simple network-aware control architecture

Modify controller to cope with communication imperfections

- Control under varying **network delay**
- Control under varying **data loss**
- Control under **bandwidth limitations**
- Control under **topology variations**



Measure or estimate network state, e.g.,

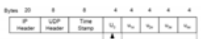
- Network delay
- Data loss probability
- Bandwidth

and adjust controller

A slightly more complex network-aware control architecture

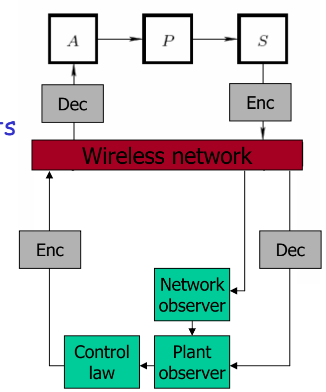
- Coding-control codesign
- Employs communication and control theories

- Large variation of implementation constraints
  - Processing capabilities in nodes
  - Time- or event-triggered nodes
  - Data delivery format



  - Cross-layer signaling

- Major recent progress, but lack of unifying theory

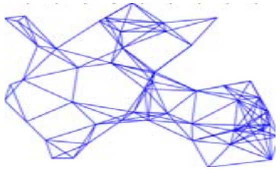


Bushnell, *IEEE CSM,* 21-1, 2001
Antsaklis & Baillieul, *IEEE TAC,* 49-9, 2004
Baillieul & Antsaklis, *Proc. IEEE,* 95-1, 2007

Global objective from local controls

Large-scale wireless control networks

- Wireless control networks are potentially very large
- Need architecture and protocols that scale well
  - Distributed implementation (control, resource allocation etc)
  - Limited information exchange

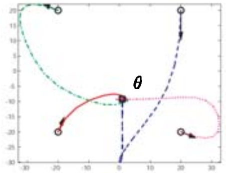


- How obtain a global objective from local actions?
- How is performance affected by network architecture?

A prototype problem on "global objective from local controls"

$$\underset{u\ \ ,\theta}{\text{minimize}} \quad \sum_{i=1}^{N}\sum_{t=1}^{T} \underbrace{(y_i(t)-\theta)^{\mathsf T} Q_i\,(y_i(t)-\theta) + u_i(t)^{\mathsf T} R_i u_i(t)}_{V_i(y_i(t),u_i(t),\theta)}$$

s.t.	$x_i(t+1) = A_i x_i(t) + B_i u_i(t)$	Linear dynamics
	$y_i(t) = C_i x_i(t)$	
	$y_i(T) = \theta$	Finite time consensus
	$\theta \in \Theta$	Consensus point constraint
	$u_i(t) \in \mathcal{U}_i$	Input constraints
	$x_i(0) = x_i^0$	Initial state

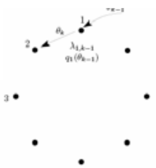


- Decentralized negotiation on Θ among agents
- Based on primal decomposition and incremental subgradient method

Johansson et al., *MTNS*, 2006

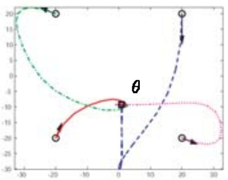
Sequential negotiation and control

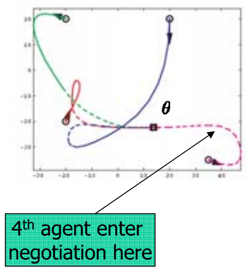
- Negotiation based on subgradient algorithm



1: Initialize θ_0 and α_0
2: $k := 0$
3: **loop**
4: **for** $i := 1$ to N **do**
5: Compute a subgradient, $\lambda_{i,k}$, corresponding to $q_i(\theta_k)$
6: $\theta_{k+1} := \mathcal{P}_\Theta\{\theta_k - \alpha_k \lambda_{i,k}\}$
7: $k := k+1$
8: $\alpha_k := \alpha_0 / k$
9: **end for**
10: **end loop**

- Distributed receding horizon control





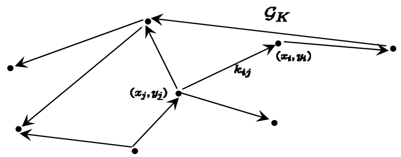
Johansson et al., *MTNS*, 2006

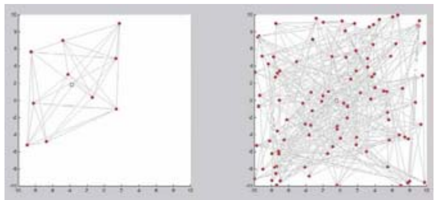
How does network size and connectivity influence performance?

Simplest local control dynamics

- consensus algorithm

$$x_i^+ = x_i + \frac{1}{\operatorname{indeg}(i)} \sum_{\substack{j \neq i \\ (j,i)\in\mathcal{E}}} (x_j - x_i).$$





Here we will discuss scalability and connectivity

Speranzon, PhD Thesis, KTH, 2006
Carli et al., *Automatica*, 2007

Hence, original compared to

Tsitsiklis et al., *IEEE TAC*, 31(9), 1986
Jadbabaie et al., *IEEE TAC*, 48(6), 2003
Olfati-Saber & Murray, *IEEE TAC*, 49(9), 2004
Etc.

Cayley graphs

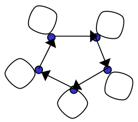
Cayley graph (G, S) is a graph with $G \supset S$ finite Abelian group:

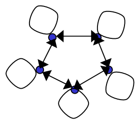
Vertex set G

Edge set $E = \{(x, y) \in G \times G : \ \exists s \in S, y = x + s\}$

Examples

$G = \mathbb{Z}_5,\ S = \{0, 1\}$ $\qquad$ $G = \mathbb{Z}_5,\ S = \{-1, 0, 1\}$





Abelian group $(G, +)$ is a set G with binary operation $+$:

- $a, b \in G \Rightarrow a + b = b + a \in G$
- $a, b, c \in G \Rightarrow (a + b) + c = a + (b + c)$
- $\exists e \in G$: $a \in G \Rightarrow e + a = a + e = a$
- $a \in G \Rightarrow \exists b \in G$: $a + b = b + a = e$

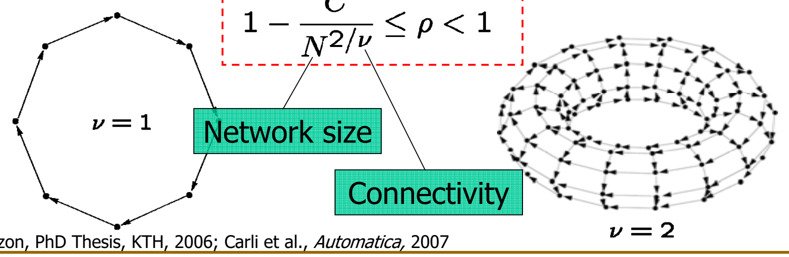
Consensus under symmetries

$$x(t+1) = (I + K)x(t), \qquad I + K \text{ doubly stochastic}$$

$$\rho = \inf_K \max\{|\lambda| : \lambda \in \sigma(I + K), \lambda \neq 1\}$$

If communication network is described by Cayley graph $\mathcal{G}_K$ and $\nu > 0$ is the in-degree of each vertex, then

$$1 - \frac{C}{N^{2/\nu}} \leq \rho < 1$$

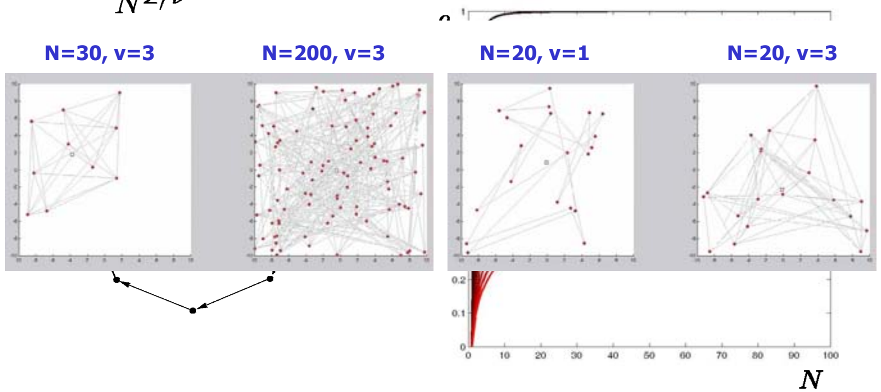


Speranzon, PhD Thesis, KTH, 2006; Carli et al., *Automatica,* 2007

Consensus under symmetries

$$1 - \frac{C}{N^{2/\nu}} \leq \rho < 1$$

If ν is fixed and $N \to +\infty$ then $\rho \to 1$



Speranzon, PhD Thesis, KTH, 2006; Carli et al., *Automatica,* 2007

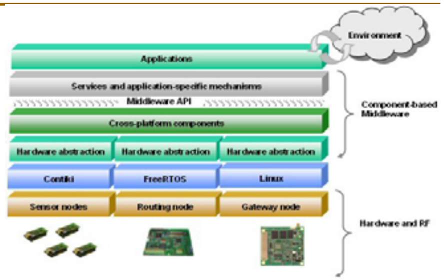
Component-based implementation

RUNES component-based programming model

A component is a reusable software objects offering a service and able to communicate with other components

Shield application level software from details of the hardware and low-level software implementation and vice versa

Component-based approach to networked embedded control problems



- Component model implemented w/ Contiki: lightweight OS for tiny network nodes
- Provides µIP stack for TCP and UDP, µAODV routing protocol
- Components can be replaced during runtime

Zhang et al., *Int. Conf. on Distr. Comp. Systems*, 2002
Heck et al., *IEEE CSM*, 23(1), 2003
Baliga et al., *IEEE Distributed Systems Online*, 5(8), 2004
Baras & Huang, *Army Science Conf.* 2006

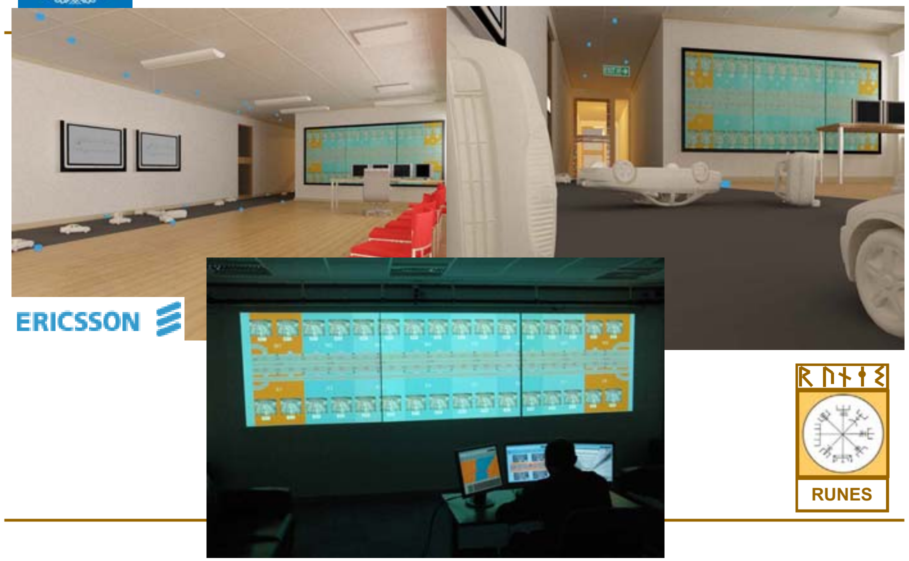
Dunkels et al., *IEEE Workshop Embedded Networked Sensors*, 2004

RUNES tunnel disaster relief scenario

Wireless sensor networks to support rescue operation at tunnel accident

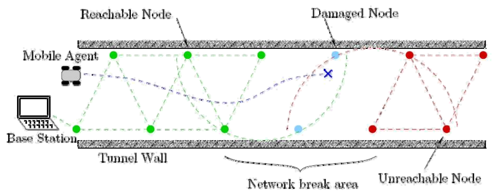
RUNES video

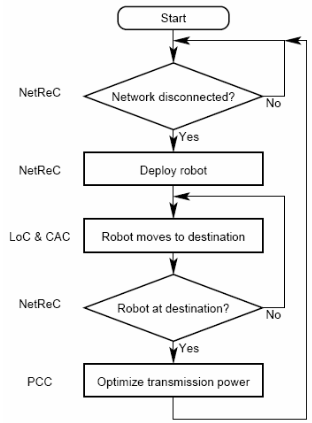
Demonstration in Stockholm 25-27 July 2007



Network reconfiguration scenario and control components

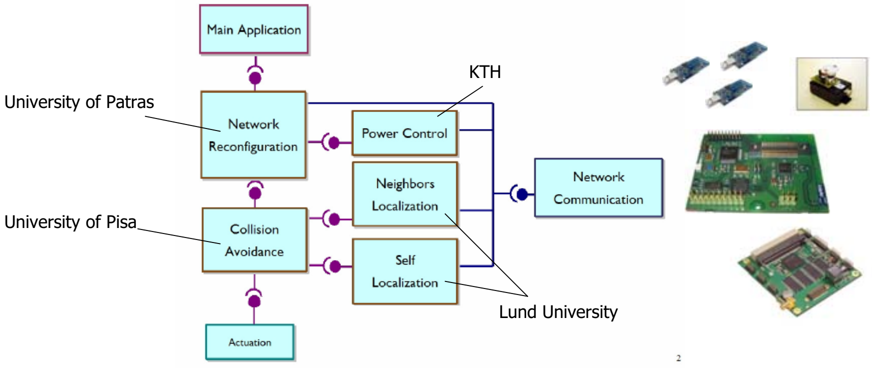
Components on each mobile agent
- Localization Component (LoC)
- Collision Avoidance Component (CAC)
- Power Control Component (PCC)
- Network Reconfiguration Component (NetReC)

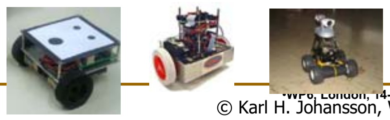




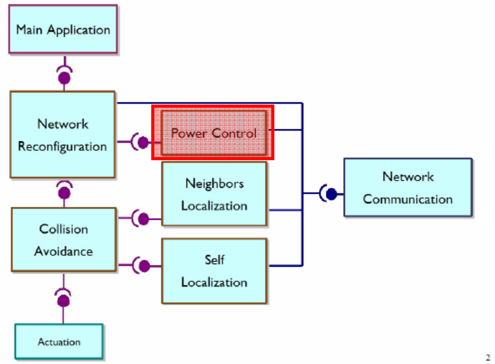
Årzen et al., *EJC*, 2-3, 2007; Johansson & Lygeros, *ECC*, 2007

Software structure

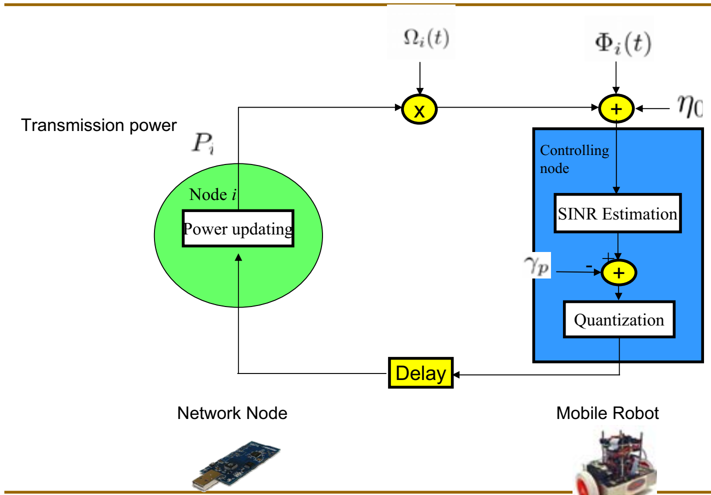




Power control component

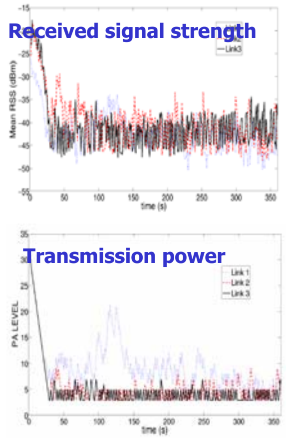


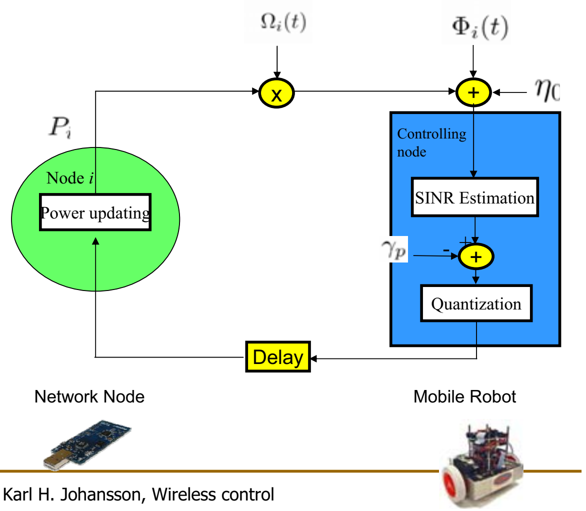
Power control component



Zurita Ares et al., *ECC*, 2007

Power control component

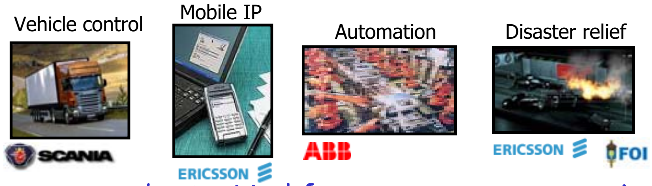




Zurita Ares et al., *ECC*, 2007

Conclusions

- Some emerging wireless control applications



- Challenges can be tackled from common perspectives
  - Layered models of abstraction
  - Global objectives from local controls
  - Component-based implementation
- Lack of theory, not technology

http://www.ee.kth.se/~kallej