NICCI: Network-Informed Control – Control-Informed Network

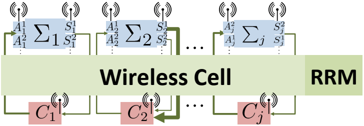
Arunselvan Ramaswamy

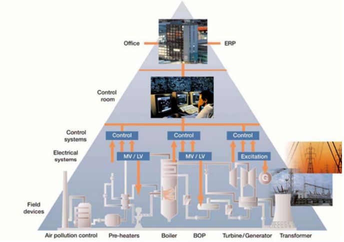
Dept. of Electrical Engineering and Information Technology,
Paderborn University
arunselvan.ramaswamy@upb.de

October 18, 2017

Overview

Networked control systems





Source: library.abb.com

Figure: Figure illustrating modularity

- Large scale, modular, distributed systems.
- Components share resources such as communication channels, computational resources.
- Controllers use both local (immediate) and global (delayed) information.
- Controllers act asynchronously (run on their own clocks).

Resource aware control – control aware resource

- Due to the large scale of systems being considered, resources such as communication channels and processors may be constrained.
- We assume the existence of a radio resource manager (RRM) who is responsible for co-ordinating communication.
- RRM is responsible for sequential decisions in regards to communication.
- 'Quality of control' and 'RRM decisions' are mutually influential.
- There may be other resource managers, eg. computational resources.
- Goal: Develop and analyze algorithms for distributed control of large-scale systems (process and radio) under resource constraints.

Chapter I: Radio Resource Management (RRM)

RRM for short-term dependability

- Let's start with the RRM-side.
- Issue: Dependable communication
  - Over fading channels.
  - With short deadlines.
  - And limited diversity (related to number of stochastically independent channels).
  - But with some limited prediction (deadline longer than coherence time).
- Bad combination – but what is achievable?

First approach: Markov model for fading channel

- Markov chain extracted from Clark's model
  - States correspond to SNR regions
  - which in turn correspond to Modulation/Coding Scheme (MCS) performance at negligible transmission error rate.

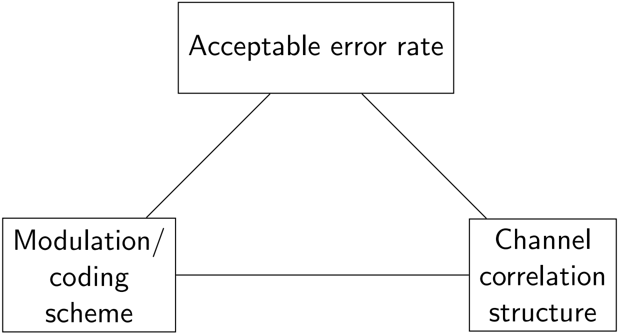
- Under these assumptions:
  - What is the *time-limited capacity*, for an acceptable model prediction. error?
    - In a sense: Real-time version of outage capacity.
  - Can it be communicated to a control system as options?

- Formulate as Markov reward (no. of bits that can be transmitted) model.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

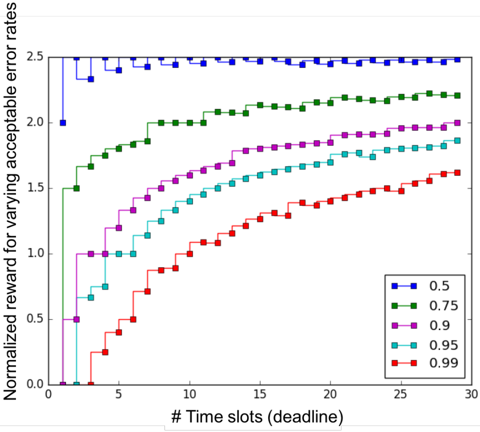
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Time-limited capacity: Influencing factors



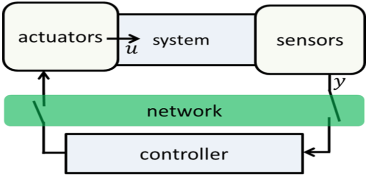
Time-limited capacity: First hints at results

- Over increasing number of time slots
- What is the (normalized, per time slot) available capacity
- For different acceptable error rates
- **Observation**: Rate of increase depends on acceptable error rate



Chapter II: Constrained MPC for a given communication schedule

MPC under reliable communication schedule



- Dynamics: (i) Unknown but bounded uncertainties (ii) Discrete-time linear systems (iii) noisy measurements.
- Communications networks: (i) Reliable (ii) Single link available: either **to actuator** *or* **from sensor**.
- Challenge: When/what/where to send/receive to maximize control performance.

Different possible communication choices provided by the RRM

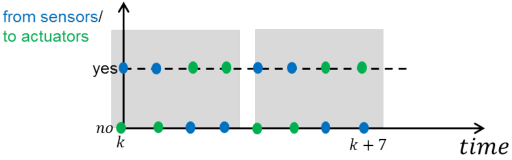


Figure: Fixed reliable communication schedule

- What to communicate (how much data):
  - **Naive approach:** Sensors send y_k while controllers send u_k.
    If no data is received at time k, then the actuators use $u_k = 0$ or $u_k = u_{k-1}$.
  - **General Case:** Multiple measurements/commands in each packet
    (a) Piggy back old measurements, e.g., send y_k and y_{k-1} at k.
    (b) Send input sequences, e.g., send u_k and u_{k+1} at k.
- Requirement: Simple performance evaluation and avoid influence of tuning.

Exploit robust output feedback MPC

- We start with the basics of robust output feedback MPC.
- **Framework:** (i) LTI system with bounded process noise, $x_{k+1} = Ax_k + Bu_k + w_k$, where $x_k \in \mathbb{X}$, $u_k \in \mathbb{U}$ and $w_k \in \mathbb{W}$.
  (ii) There exist constraints on x_k and u_k.
  (iii) Sensor measurements, bounded noise: $y_k = Cx_k + v_k, \; v_k \in \mathbb{V}$.

- **Robust output feedback MPC:**
  - **Tube approach:** Decompose dynamics into (a) **nominal system** (no noise) + MPC and (b) **error systems + fixed linear controller**.
  - **Control objective: Worst case optimal set-point tracking:**
    - Minimize back-off from constraints (set $\mathbb{Z}$) - tune "tube controller".

Our contribution

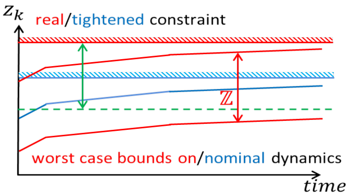


Figure: "Economic performance" $Z_k = Fx_k + Gu_k$

- Automated optimal controller design for a given communication schedule.
- Allows for comparison between different communication schedules.
- Guarantees on closed-loop behavior (constraint satisfaction, robust stability).
- Computationally efficient.

Chapter III: Reinforcement learning approach

Multi-agent systems perspective

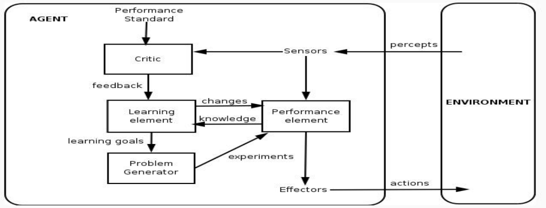


Figure: Point-of-view of one agent, one of many! (source: wikipedia)

- RRM and controllers are viewed as autonomous agents.
- These agents interact with a stochastic environment.
- Model of this environment is unknown.
- Requirements for offline algorithms: availability of labeled data.
- In case of online algorithms: learn the model on-the-go.

Motivation: Dynamic programming approach

- Develop and analyze approximate, asynchronous value iteration and policy gradient algorithms.
- Value iteration is given by $J_{n+1} = TJ_n$, where $TJ(i) := \min_{\pi} E_{\pi}\left[g(i, \pi(i), j) + J(j)\right]$ is the Bellman operator.
- We mutate its stochastic iterative counterpart:

$$J_{n+1} = J_n + a(n)\left(TJ_n - J_n + M_{n+1}\right).$$

- We also mutate policy gradient descent given by:

$$\theta_{n+1} = \theta_n - a(n)\left[\nabla_{\theta}\pi(\theta) + M_{n+1}\right].$$

- Summary: Re-engineer for large-scale distributed asynchronous systems with lossy, delay-prone communication systems.

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Approximate asynchronous Value Iteration

Re-engineered value iteration algorithm

$J_{n+1}(i) =$

$J_n(i) + a(\nu(i,n)) I\{i \in Y_n\} \times \left(\mathcal{A} T(J_{n-\tau_{1i}(n)}(1), \ldots, J_{n-\tau_{di}(n)}(d))(i) - J_n(i) + M_{n+1}(i) \right).$

- $\nu(i,n)$ is the number of times that agent i was active till time n.
- $Y_n \subseteq \{1, \ldots, d\}$ denotes the active agents at time n.
- $0 \leq \tau_{ij}(n) \leq n$. Delay faced by agent j, at time n, in receiving information from agent i.
- $\mathcal{A}$ is the approximation operator (obtained as a consequence of a supervised learning algorithm).
- M_{n+1} is the Martingale difference noise.

Approximate asynchronous Policy Gradient Descent (AAPGD)

- Basic idea of PGD is the existence of a parameterization $\pi(\theta)$ (policy objective function), Sutton et. al. [5]. The goal is to find a local minimizer $\hat{\theta}$.

Re-engineered policy gradient descent

$$\theta_{n+1}(i) = \theta_n(i) - a(\nu(i,n))I\{i \in Y_n\}\left(\mathcal{A}\nabla_\theta \pi_i(\theta_{n-\tau_{1i}(n)}(1), \ldots, \theta_{n-\tau_{di}(n)}(d)) + M_{n+1}\right).$$

- It allows for "finite-difference" implementations of GD using SPSA or SPSA-C (simultaneous perturbations stochastic approximations).
- In the above equation, we may replace the gradient terms with sub-gradients, our analysis still carries through, verbatim.
- While approximate asynchronous value iteration is offline, AAPGD is online.

Our contribution

- Developed sufficient conditions for convergence and stability (boundedness of the iterates).
- We completely characterize the limiting sets of the algorithms. For example, $\{J \mid \|TJ - J\|_{\omega,p} \leq \epsilon\}$ is the limiting set of the re-engineered value iteration scheme.
- Stability ($\sup_{n\geq 0}\|x_n\| < \infty$ a.s.) is always a concern in approximate dynamic programming, reinforcement learning approaches.
- Our stability conditions are Lyapunov function based, extending [4].

Summary

- We studied the Markov reward model for communication networks, currently being used in ML approaches.
- Constrained MPCs that allow for comparison between given communication schedules.
- Value iteration and policy gradient algorithms, re-engineered for our setting.
- Proofs of convergence and boundedness of the algorithm.

Current and Future Work

- RRM-side:
  - Use Markovian rewards (previously studied) to develop multi-arm bandit based algorithms.
  - For more general rewards we are currently looking at deep reinforcement learning algorithms.
- Controller-side: Value iteration networks and deep Q networks in the setting of distributed systems with lossy delayed communications.
- Experimental aspects of the project: Run an intelligent RRM and model-free controller in tandem.
- Final goal: The RRM learns the optimal scheduling taking into account the control issues. The controller is distributed and accounts for dynamic resource availabilities. ONLINE!!

The End

References

M. Kögel and R. Findeisen 2017. Low latency output feedback model predictive control for constrained linear systems. CDC

M. Kögel and R. Findeisen 2017. Robust output feedback MPC for uncertain linear systems with reduced conservatism IFAC WC

E.G. Peters, D.E. Quevedo and M. Fu 2016. Controller and Scheduler Codesign for Feedback Control Over IEEE 802.15. 4 Networks. IEEE Transactions on Control Systems Technology, 24(6).

A. Ramaswamy and S. Bhatnagar (2017). Conditions for Stability and Convergence of Set-Valued Stochastic Approximations: Applications to Approximate Value and Fixed point Iterations with Noise. Submitted in September, 2017.

References

- M. Kögel and R. Findeisen 2016. Output feedback MPC with send-on-delta measurements for uncertain systems. NECSYS.
- M. Kögel and R. Findeisen 2016. Sampled-data, output feedback predictive control for uncertain, nonlinear systems. NOLCOS.
- V. Mnih, et. al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.
- V. Mnih et. al. (2016, June). Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning (pp. 1928-1937).
- R.S. Sutton et al. "Policy gradient methods for reinforcement learning with function approximation." Advances in neural information processing systems. 2000.