Energy-Efficient Data Converters for 5G

Ameya Bhide, Tai Duong, A. Ojani, Vishnu Unnikrishnan, Mark Vesterbacka, Atila Alvandpour
Division of Integrated Circuits and Systems, Dept. of Electrical Engineering, Linköping University

VCO-based ADCs

- Signal represented with time instead of voltage/current.
- Ring VCO based noise-shaping all-digital architecture.
- Built with vendor supplied standard cells.
- Reported lowest power and best FoM in class.

High-speed time-interleaved \(\Sigma\Delta\)-DAC

- First reported \(\Sigma\Delta\) DAC with >1 GHz bandwidth and 11 GHz sampling rate.
- Highly digital and scalable solution for advanced CMOS technologies.

Performance metric

<table>
<thead>
<tr>
<th>OSR = 1</th>
<th>Sample rate (MHz)</th>
<th>Bandwidth (MHz)</th>
<th>ENOB</th>
<th>SNDR (dB)</th>
<th>SNR (dB)</th>
<th>SFDR (dB)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>205</td>
<td>25</td>
<td>8.1</td>
<td>50.3</td>
<td>52.8</td>
<td>55.3</td>
<td>3.3</td>
</tr>
</tbody>
</table>

ADCs fully described in HDL.
- Synthesized towards standard cells.
- Signal aware synthesis and placement.

High-Speed Switched-Capacitor DACs

- Fastest reported SC DAC with >1 GHz bandwidth.
- Smaller capacitor mismatch and reduced voltage headroom in switches lead to higher design scalability in advanced CMOS technologies.

Comparison with other SC DACs

<table>
<thead>
<tr>
<th>Ref</th>
<th>Usage</th>
<th>Fclk (GHz)</th>
<th>BW (GHz)</th>
<th>SNDR (dB)</th>
<th>Power (mW)</th>
<th>BFOM</th>
<th>SFDR (dB)</th>
<th>SFOM (dB)</th>
<th>FOM2 [10^12]</th>
<th>FOM3 [10^12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSC</td>
<td>Wireline</td>
<td>0.8</td>
<td>0.4</td>
<td>53</td>
<td>103</td>
<td>12</td>
<td>15.9</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISSC</td>
<td>Comm. SoC</td>
<td>0.2</td>
<td>0.034</td>
<td>70</td>
<td>693</td>
<td>10</td>
<td>0.05</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSSC</td>
<td>60-GHz Radio</td>
<td>5</td>
<td>0.6/1</td>
<td>60/44</td>
<td>44/50</td>
<td>12</td>
<td>75/82</td>
<td>1.7/1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Algorithm for sampling multi-phase outputs.
- Based on two complementary ones counters with modular operation.
- Designed to suppress bubble errors.

Comparison with other DACs in 60-GHz radio applications

<table>
<thead>
<tr>
<th>Ref</th>
<th>Usage</th>
<th>Fclk (GHz)</th>
<th>BW (GHz)</th>
<th>SNDR (dB)</th>
<th>Power (mW)</th>
<th>BFOM</th>
<th>SFDR (dB)</th>
<th>SFOM (dB)</th>
<th>FOM2 [10^12]</th>
<th>FOM3 [10^12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSSC</td>
<td>Wireline</td>
<td>11</td>
<td>1.1</td>
<td>53</td>
<td>117</td>
<td>8</td>
<td>39</td>
<td>2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISSC</td>
<td>Comm. SoC</td>
<td>3.456</td>
<td>0.4</td>
<td>40</td>
<td>21</td>
<td>8</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSSC</td>
<td>60-GHz Radio</td>
<td>5</td>
<td>0.8/1</td>
<td>50/44</td>
<td>44/50</td>
<td>12</td>
<td>75/82</td>
<td>1.7/1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


V. Unnikrishnan and M. Vesterbacka, “Mitigation of sampling errors in VCO-based ADCs”, IEEE Transaction on Circuits and Systems, accepted 2017

Cellular Terminal Antenna Impedance Tuners in CMOS-SOI

J. Lindstrand, H. Sjöland, I. Vasilev, and B.K. Lau, Department of EIT, Lund University, Sweden

**Without Tuner**
- Impedance mismatch $Z_{ant}$ and $Z_{transceiver}$
- Reflection loss
- Degraded RX sensitivity
- Degraded TX output power and efficiency

**Match Domain**
- Impedances that can be matched
- Frequency dependent
- Should be large
- Trade-off with loss

**With Tuner**
- Most reflection loss eliminated
- Absorption loss in the tuner
- Improvement depends on initial mismatch

**Tuner System**
- Voltage handling: Off state RF voltage distributed over series devices ($W_{1-8}$)
- Matching domain: $C_{on}/C_{off}$ ratio
- Losses: $Q_{on}$ and $Q_{off}$
- 8 Switches, 2.5V each
- In on-state Cap must take this voltage, Custom MOM needed
- SiO$_2$ voltage handle of 100V/µm

**Measurements**
- Freq: 700MHz
  - Loss: 0.76dB
  - Losses at 845MHz
  - OIP3 at 845MHz
  - Spurious Emission
- 845MHz
  - 0.93dB
- 900MHz
  - 1.03dB

**Terminal Prototype**
- Dual antennas with tuners

**Terminal Measurements**
- Different grips, persons and environments

**Multiplexing Efficiency Gain**
- Effect of power and correlation on MIMO capacity
- Up to 3dB improvement including tuner loss

**Publications**
- ESSCIRC 2014
- Part of a Workshop @ IMS 2015: Technologies for Tunable and Reconfigurable RF/Microwave Filters
- MTT May 2016

Email: jonas.lindstrand@eit.lth.se
Optimizing Hardware for High Rate Communications

Daniel Verenzuela, Emil Björnson, LIU

Introduction
- Wireless data traffic [bit/s/km²] is continuously rising
- Power consumption also grows

- How to design more energy-efficient networks?
  maximize data rate
  subject to power consumption ≤ threshold

- What are the optimization variables?
  Traditionally: Power allocation, beamforming
  New idea: Hardware configuration

Tradeoff:
- Many antennas
  Higher rates
  More hardware is needed
- Low-resolution hardware
  Reduced power
  Creates distortion

System Setup
- Uplink of single-cell multi-user, multi-antenna system
  - $K$ single-antenna users, $M$ antennas at base station
  - Channel to user $k$: $\mathbf{h}_k \sim CN(\mathbf{0}, \beta \mathbf{I}_M)$
  - Same channel realization for $S$ symbols
  - Bandwidth $B_W$: transmit power of user $k$: $p_k$
  - Received signal at BS:
    $$y = \sqrt{1-\epsilon^2} \left( \sum_{k=1}^{K} \sqrt{\frac{p_k}{B_W}} \mathbf{h}_k x_k + n \right) + e \quad \in \mathbb{C}^{M \times 1}$$
    - Reduction of received energy by distortion
    - Transmission symbols
    - Thermal noise
    - Additive hardware distortion

- Basic hardware distortion model
  - Distortion is proportional to received power:
    $$|e|^2 = \epsilon ^2 \left( \sum_{k=1}^{K} \sqrt{\frac{p_k}{B_W}} \mathbf{h}_k x_k \right)^2 + \sigma^2$$
    - Low distortion: $\epsilon \to 0$
    - High distortion: $\epsilon \to 1$
    - Interpretation: $\epsilon^2 = \zeta 2^{-2b}$ with $b$-bit quantization

Achievable Sum Rate
$$R_{\text{SUM}} = K B_W \left( 1 - \frac{\epsilon}{2} \right) \log_2 (1 + \text{SINR}) \, [\text{bit/s}]$$
$$\text{SINR} = \frac{(1-\epsilon^2)^2 M}{\epsilon^2 (1-\epsilon^2)} \left( \frac{2\epsilon^2 - (1-\epsilon^2)^2}{2 - \epsilon^2} \right) + K \left( 1 + \frac{\sigma^2}{\epsilon^2} \right)^2$$
- Accounts for imperfect channel estimation
- Assumes inverse power control: $p_k = \rho / \beta_k$

Rate increases with $M$ and decreases with $\epsilon$

Power Consumption Model
$$P(\epsilon, M) = \frac{1}{M} \sum_{k=1}^{K} p_k + C_0 + C_1 K + D_0 M + \hat{P}(\epsilon)$$
- Transmission power
- Fixed power
- Independent of $\epsilon$
- Decreasing function of $\epsilon$

Power increases with $M$ and decreases with $\epsilon$

Sum Rate Optimization Problem
$$\max_{\epsilon \in [0,1], \epsilon \in \mathbb{Z}^+} R_{\text{SUM}}(\epsilon, M),$$
subject to
$$P(\epsilon, M) \leq \gamma$$
- Optimal point: $M(\epsilon) = \frac{\gamma - \frac{1}{M} \sum_{k=1}^{K} p_k + C_0 + C_1 K}{D_0 M + \hat{P}(\epsilon) / 2 K \log_2 (\gamma)}$

Numerical Illustration
- $b = 4$ or $5$ [bits]
- $K = 10, \rho = 0.3162 \sigma^2$ (SNR = -5dB)
- $\gamma = \{22, 26, 30\}$ [W]

Conclusion
Communication systems with many antennas and low-resolution hardware is most energy efficient

Project: 5G Wireless
Intelligent input generation for security testing

Ulf Kargén*, Nahid Shahmehri*, Ben Smeets†

*Department of Computer and Information Science, Linköping University
†Department of Electrical and Information Technology, Lund University

Automatic Input Generation – Fundamental Approaches

Black Box – Treat software as black box and “blindly” generate inputs.
- **Mutational Fuzzing**: Apply random mutations to corpus of valid seed inputs
- **Generational Fuzzing**: Generate semi-valid input based on (formal) input specification

White Box – Observe program control-flow and feed to constraint solver to generate new input.

Grey Box – Random input generation (fuzzing) guided by lightweight coverage feedback

Evolution and Current Trends

**Early 2000’s**: Simple fuzzing found to be remarkably effective at uncovering unknown security bugs, but...
- 😞 Very labor intensive to craft generational fuzzers
- 😞 Mutational fuzzers often break structure of inputs, leading to poor and "shallow" code coverage

**Late 2000’s**: Increasing interest in white box methods (symbolic/concolic execution) to systematically explore program state space, but...
- 😞 Constraint satisfaction is NP-complete → Highly limited scalability in practice

**Recently**: Renewed interest in "dumb" fuzzing. Carefully designed grey box fuzzers using lightweight coverage feedback often outperform more sophisticated white box methods in practice.

Our Contributions

In view of the scalability problems of whitebox methods, we focus on making “dumb” fuzzing smarter, while still retaining the efficiency of traditional fuzzing.

**MutaGen**

We seek to marry the low human effort of mutational fuzzing with the good code coverage of generational fuzzing. Instead of blindly mutating valid inputs directly, we mutate a correct implementation of a *program* that can generate inputs of the given format.

**Essentially, we turn an off-the-shelf program binary into a generational fuzzer** by injecting “bugs” in carefully chosen parts of the binary, by means of detailed dynamic data-flow analysis.

Traditional Mutational Fuzzing

Mutational fuzzers often break the structure of inputs, causing premature input rejection by the software under test.

Our approach largely retains the correct structure of inputs, while introducing subtle mutations. This results in an order of magnitude higher code coverage than traditional mutational fuzzers. MutaGen also **found several previously unknown bugs** in popular Linux software.

Project: 5G Wireless
In our work we present a solution on how to migrate keys from the older TPM 1.2 to the newer TPM 2.0, while still maintaining the same functionality. The conversion is performed by a conversion authority as below.

Some differences between the standards, related to migration, are:

- Authorization of a key migration
- Certifiable migratable keys (CMKs)
- Owner authorization

To maintain backwards compatibility between the TPM versions, we introduce the concept of sibling keys, which ensures that identical migration authorization is kept.

Intel Software Guard Extensions (SGX) is a technology available in newer Intel processors. It allows for both isolation and attestation.

During recent years, Software Defined Networking (SDN), has become a key part of the cloud infrastructure. SDN is also expected to be a key technology in the design of modern 5G networks.

The concept of a central controller communicating with Virtual Network Functions (VNFs) on a remote host over the northbound interface is flexible. However, it raises security concerns, since the controller must be sure that the applications are not under the control of an attacker. We propose a solution depicted in the overview below:

It has the following properties:

- Places VNFs in Docker containers for easy deployment
- Uses SGX to attest the integrity of the remote VNFs
- Uses a TPM and SGX to attest the virtualization host
- Provision the (attested) application with a key, such that it can securely communicate with the controller.
Massive MIMO in Mobile Channels: Real Measurement with LuMaMi

Steffen Malkowsky, Paul Harris, Liang Liu et al.

I Measurement Setup

LuMaMi Testbed placed on rooftop

Pedestrians emulated on bike carts

High mobility users on cars

II Measurement Scenario

- Distance from testbed to users about 60-70m
- 4 users on cars and 4 users on bike carts
- Cars moving at a speed of up to 50km/h.

III Measurement Results

- Fig. A: Resilience to fading for a 3 second car-based user path indicated above. Relative channel magnitude for both a single antenna and the composite MIMO channel to the depicted user over the 3 second period.
- Fig. B: Correlation of the signal power on one antenna versus 100 antennas for a pedestrian and car UE.
- Fig. C: Correlation of the composite channel over time of car 2 at a speed of 29 km/h. 100 antenna and 8 antenna cases are shown in 4x25 and 2x4 configurations.
- Fig. D: CDF of uncoded BERs for static and mobile scenarios using QPSK and ZF. 0.5 ms coherence interval, mobility up to 50 km/h.
Positioning with Large Surfaces

S. Hu, F. Rusek, and O. Edfors

We envision a future where man-made structures will be electronically active, enabling a transition from massive MIMO arrays to large radiating surfaces.

A terminal in front of a surface sends a narrowband pilot signal, and we consider the ability of the surface to position the terminal. We do this via the Cramér-Rao lower bound. For simplicity, we consider a circular surface of radius \( R \).

The signal received at a point \((x,y,0)\) at the surface is

\[
s_{x_0,y_0,z_0}(x,y) = \frac{\sqrt{z_0}}{2\sqrt{\pi}\eta^{1/4}} \exp\left(-\frac{2\pi j \sqrt{\eta}}{\lambda} - j\theta\right) + n(x,y)
\]

where \( \eta = z_0^2 + (y - y_0)^2 + (x - x_0)^2 \)

and where \( \theta \) is an unknown phase modeling non-ideal hardware, reflections, etc.

Based on this, we compute the CRLB as a function of deployed surface area \( A = \pi R^2 \)

**Results with known phase \( \theta \)**

(more precise formulas in paper)

The CRLB for this case equals (\( d \) being the distance from the terminal to the center of the surface)

\[
CRLB \approx \frac{4\lambda^2 d^4}{A^2}
\]

Thus, the positioning precision is inversely proportional to the square of surface area.

**Results with unknown phase \( \theta \)**

(more precise formulas in paper)

\[
A \gg 2\sqrt{3}\pi\lambda d \quad CRLB \approx \frac{48\pi\lambda^2 d^6}{A^3}
\]

\[
A \ll 2\sqrt{3}\pi\lambda d \quad CRLB \approx \frac{4\pi d^4}{A}
\]

Thus, with sufficiently large surface, precision inversely proportional to the cube of surface area, otherwise inversely proportional to surface area.

**Example**

Large lecture hall, surface in the roof, \( d = 10\)m 2GHz carrier. (Results scale with 1/SNR).
On the Impact of Sybil Attacks in Cooperative Driving Scenarios
Felipe Boeira, Marinho Barcellos, Edison Pignaton, Alexey Vinel, Mikael Asplund

Platoons use Inter-Vehicular Communication (IVC) to reduce the distance (headway time) between them while traveling on a highway.

- beaconing: platoon members periodically broadcast a message that conveys information about the physical state of the vehicle.
- Network communication introduces cyberattacks to vehicles' threat surface.

Attack Model
- In a Sybil attack, a malicious entity may present itself via multiple identities to control a substantial part of a system.
- The Sybil attack in the platoon context may be conducted by introducing falsified vehicle identities to the platoon formation.
- Malicious nodes may perform message falsification to interfere with the platoon vehicle controller.

Attack Simulations
- Sybil nodes that collude in a message falsification attack may compromise the platoon’s string stability if governed mainly by IVC-based information.
- Combining Sybil and message forgery attacks with position falsification can directly affect the longitudinal control algorithm and may result in the violation of the control law.
- As can be observed in Fig 3 and Fig 4, results indicate that, in a pure IVC-based platoon, vehicle accidents at high speed may be caused by nodes that collude with carefully crafted beacons.

Ongoing Work
This knowledge is important as an input when making a dependability assessment on the entire platoon logic. In particular, it demonstrates the need to study the effects of combination of normal sensor uncertainty and noise in adverse conditions together with an IVC-based attack.
1. Problem
- Location of vehicles inside a vehicular network must be known for safety reasons
  - GNSS (GPS, GALILEO)
  - Radar
  - Computer vision
- Location can be extracted from received signal
  - DOA estimation can be used
  - Fully passive method
  - Estimation based on physical layer parameters
- Advantages
  - Fully passive method, alleviates network load
  - Very precise when distances involved are not large
  - Available in dense urban environments, where GNSS fails
  - Can be used to avoid location spoofing

2. Data Model
We consider a set of \( d \) wavefronts impinging onto an antenna array composed of \( M \) antenna elements. The received baseband signal can be expressed in matrix form as
\[
X = AS + N \in \mathbb{C}^{M \times N},
\]
where \( S \in \mathbb{C}^{d \times N} \) is the matrix containing the \( N \) symbols transmitted by each of the \( d \) sources, \( N \in \mathbb{C}^{M \times N} \) is the noise matrix with its entries drawn from \( \mathcal{N}(0, \sigma_n^2) \), and
\[
A = [a(\theta_1), a(\theta_2), ..., a(\theta_D)] \in \mathbb{C}^{M \times d},
\]
where \( \theta_i \) is the direction of arrival of the \( i \)-th signal and \( a(\theta_i) \in \mathbb{C}^{M \times 1} \) is the array response. An estimate of the signal covariance matrix can be obtained by \( \hat{R}_{xx} = X^H X / N \).

3. Scenario
This work assumes that the vehicles are equipped with two linear antenna arrays at two distinct locations on their frames. Assuming the 5.8 GHz frequency band and that DOA estimation is to be done at carrier frequency the antenna elements can be placed as far as 2.5 cm apart.

4. Location Estimation
![Diagram of location estimation](image)
- Location Estimation
  - DOA can be estimated using a wide selection of methods such as ESPRIT, SAGE, EM, and MUSIC.
  - After DOA estimation, the location of a transmitting antenna can be obtained with
  \[
  x = \frac{\tan(\theta_2)x_{Rx_2} - \tan(\theta_1)x_{Rx_1}}{\tan(\theta_2) - \tan(\theta_1)},
  \]
  \[
  y = \frac{\tan(\theta_2)x_{Rx_2} - x_{Rx_1}}{\tan(\theta_2) - \tan(\theta_1)}.
  \]
- Obtaining the position of one of the transmitting arrays can be sufficient for some applications. For instance, to increase the accuracy of a dead reckoning method in the temporary absence of a GNSS lock.

5. Direction Estimation
- The direction of a vehicle can be estimated when two transmitting antennas are present
  - A line that cuts both antennas can be found by
  \[
  y_{T_{x_1}} = \frac{y_{T_{x_2}} - y_{T_{x_1}}}{x_{T_{x_2}} - x_{T_{x_1}}} (x - x_{T_{x_1}}).
  \]
  - The angle \( \theta_{direction} \) that this line forms defines the direction that the transmitting vehicle is facing, and is given by
  \[
  \theta_{direction} = \arctan \left( \frac{x_{Rx_2} - x_{Rx_1}}{y_{Rx_2} - y_{Rx_1}} \right).
  \]

6. Simulations Results
- Antenna arrays at the wing mirrors are considered to be an ULA composed of \( M = 3 \) antennas with inner element spacing of \( \frac{\lambda}{2} \).
- For obtaining \( \hat{R}_{xx} \) we use \( N = 100 \) snapshots.
- At a SNR of 7 dB the proposed method is capable of a 1 m precision, superior to the nominal accuracy of GPS.
- At the 7 dB SNR threshold the direction error is smaller than one degree.
- Communication ranges at VANETs are expected to be relatively small, therefore the proposed method can be used as a safety mechanisms or standalone localization method.

Project: 5G Wireless, WP6: Vehicular communication

Eastlink Center for Linköping – Lund in Information Technology

Antenna Array Based Localization Scheme for Vehicular Networks
Marco A. M. Marinho, Alexey Vinel, Felix Antreich, João Paulo C. L. da Costa, and Edison Pignatoni de Freitas
Resource Allocation with Service Availability and QoS constraints in Mobile Fog Networks
Nader Daneshfar, Di Yuan, Nikolaos Pappas, and Vangelis Angelakis
Department of Science and Technology, Linköping University

1. Introduction
- With the recent evolutions in the IoT era and developments of Cloud based technologies and applications, the reachability of resources and services for end users are becoming more vital.
- Fog networking is being presented as a promising approach to address the recent issues by utilizing the unused resources at the edge of the networks.
- Similar concepts to Fog computing:
  1. Mobile Edge Computing (MEC): Bringing the required computational power from core to the edge of network.
  2. Mobile Cloud Computing (MCC): Sending all the computational (or storage) tasks to be carried out at the core of network.
- Potential advantages of Fog over traditional Cloud:
  1. Local availability and distribution
  2. Distributed architecture
  3. Lower latency and quick resource access
  4. Smart bandwidth usage resulting in lower backhaul load
  5. Dynamic scalability and implementation

2. Optimizing Fog Network Operational Cost
   Motivation:
   - To date, there is no significant research done focusing on the mobile nature of the Fog network elements.
   - The effect of diversity among Fog server set elements (that are providing services) has not been sufficiently investigated.
   - Applying mobility to network elements will impose uncertainty in their role as service providers. How will this phenomena affect QoS?
   Contribution:
   - An MILP model is proposed to minimize the total cost by optimizing the number of connections needed for service allocation.
   - Servers are coupled with “probability of availability” variable to demonstrate mobility.
   - Quality of Service per each user’s demand is assured by allowing demand duplication and multicasting.

3. Mathematical Model
   Objective: \( \min \sum_{u} \sum_{s} d_u w_s x_{us} \)

   Constraints:
   1. \( \sum_{s} x_{us} \leq M, \forall u \in U \) (1)
   2. \( \sum_{u} d_u w_s x_{us} \leq B_s, \forall u \in U \) (2)
   3. \( \sum_{u} d_u x_{us} \leq D_s, \forall s \in S \) (3)
   4. \( x_{us} \ln(1-pu) \leq \ln(1-lu), \forall u \in U \) (4)

   The main objective is minimizing the total cost of providing services.

   Constraints:
   1. Each user is limited by a factor \( M \) for excessive demands (flooding).
   2. Users are prevented from exceeding their budget \( (\mu_u) \) with respect to their demands \( d_u \) and connected servers \( w_s \).
   3. Server capacity \( D_s \) overflow is controlled considering all users’ demands.
   4. Insuring each user receives its requested minimum QoS \( l_u \) by considering the probability of availability for each server \( p_u \).

4. System Model
   - A simplified system model with basic components.
   - Servers with different availability construct a service set.
   - Availabilities of different servers are independent.
   - Set of users demand services from the service set.
   - A centralized controller has complete and correct information of all network elements.

5. Scenarios & Results
   Scenario 1: Different Server Capacity
   - Fixed demands per users
   - Models for cost of services
     - Fixed (10 units)
     - Normally distributed with low standard deviation (std=1)
     - Normally distributed with moderate standard deviation (std=3)
     - Uniformly distributed (\( \mu=10 \))
   - Results:
     - Worst case scenario when fixed set of services is implemented (basic allocation).
     - The result binds two normal distributions with different standard deviations.
     - Highest diversity with normal distribution results in least increase in total cost.
     - Saturating the system with more users, all scenarios tend to converge to maximum.

   Scenario 2: Mobility and QoS
   - Fixed demands and cost of services for users and servers
   - Fixed resources (number of servers and server capacity)
   - 4 combinations of server set availability and user’s level of service request
   - Cost optimization is then compared in fixed gap (30%) between requested QoSs and service availability values
   - Results:
     - Basic scenario (green): equal request and availability level
     - No demand duplication.
     - Identical increase in requested QoS level does not result in the same relation with respect to the required number of service request duplication.
     - Level of requested QoS is in a direct relation with the multiplication factor of duplication in case the availability is altered.

6. Conclusion
   - Diversity in Fog servers’ distribution is the key factor in maximizing optimization:
     - Higher diversity → Lower increment rate of total cost
     - Randomizing demand is ineffective to cost optimization
   - Desired QoS can be achieved by duplicating the demand to the Fog server set.
   - The total cost does not change equally while keeping the gap between the server set availability and users QoS request at various scenarios.
System-Level Modelling of the Platooning Application in Plexe

Christian Nelson, Nikita Lyamin, Fredrik Tufvesson, Alexey Vinel

- PHY Layer: A Measurement Based Multilink Shadowing Model

MAC Layer: Cooperative Awareness Messages (CAM) Model

- Generate CAMs in accordance with the ETSI EN 302 637-2 specification
- Transmit CAMs on a dedicated channel in accordance with the IEEE 802.11p MAC specification

System Level: Heterogeneous Platooning Model

Volvo XC70, S60, XC90 and V70
New Networking Solutions for LTE and WLAN
B. Chen1,2, Z. Chen3, N. Pappas3, J. Zhang1, and D. Yuan2
Farnaz Moradi3, Mehmet Karaca3, Emma Fitzgerald3, Michal Pioro3,4, Rickard Ljung5, Bjorn Landfeldt3
1: The University of Sheffield, 2: Linköping University, 3: Lund University, 4: Warsaw University of Technology, 5: Sony Mobile Communication AB, Lund

Motivation and Analysis of MPTCP Proxy-based LTE-WLAN Path Aggregation

Background and Motivation
- There is a great interest in offloading cellular traffic to unlicensed spectrum.
- LTE-WLAN Path Aggregation (LWPA) based on Multi-path Transmission Control Protocol (MPTCP) is a promising solution.
- A fundamental understanding of the performance of LWPA is needed.

Contributions
- A practical model to analyze LWPA network based on stochastic geometry;
- A tractable analysis of the impact of various parameters on LWPA;
- A practical protocol to determine under which conditions a Wi-Fi AP can work under LWPA-mode.

System Model
- \( p \): closed-access Wi-Fi AP probability among all Wi-Fi APs
- LWPA-mode Wi-Fi AP activation conditions
  1. Open-access;
  2. At least one LTE user is inside the service range \( R \) of each AP;
  3. Any closed-access Wi-Fi AP must be at a minimum distance of \( \delta \);
  4. Two active LWPA mode Wi-Fi APs cannot be closer than \( \delta \).

Cellular Rate Improvement Analysis
- \( R_{C} = \frac{R_{C,Fi} - \epsilon_{Fi} \cdot N_{Fi}}{1 - \epsilon_{Fi} \cdot N_{Fi}} \) (a)
- \( R_{LTE} = \int_{e_{LTE}}^{1} \frac{P_{LTE}(\epsilon_{LTE} | \lambda_{LTE})}{\lambda_{LTE}} \, d\lambda_{LTE} \) (b)
- NW = \( \frac{\lambda_{LTE}}{\lambda_{LTE} - \epsilon_{LTE} \cdot N_{LTE}} \) (c)

Observations:
- With smaller \( \rho \), the potential to improve cellular rate improvement is greater.
- LWPA can significantly improve the rate of LTE users without causing much interference to closed-access Wi-Fi users.

WLAN Area Spectral Efficiency Analysis
- \( P_{W} = \frac{T}{T} \) (d)
- \( T = (\lambda_{W} \cdot \delta)^{-1} \int_{e_{W}}^{1} \frac{P_{W}(e_{W} | \lambda_{W}, \delta)}{\lambda_{W} \cdot \delta} \, de_{W} \) (e)

Observations:
- With smaller \( \rho \), cellular rate can be better improved.
- The guard zone radius can be optimized.


Optimizing DRX for Video delivery over LTE: Utilizing Channel Prediction and In-network Caching

Motivation and Background
- By 2021, 4G will account for 75 percent of total mobile data traffic and more than 78 percent of this traffic will be video.
- Prediction of future channel conditions in cellular networks is well established.
- Static Discontinuous Reception (DRX) scheme results in delay and/or buffer underflow.

Contributions
- Utilizing channel prediction and in-network caching to jointly optimize DRX and LTE scheduling;
- Guaranteed zero buffer underflow through Strict buffer constraint;
- Two novel DRX schemes based on future channel conditions and buffer status;

Proposed DRX schemes and LTE scheduling

Optimization problem
- \( \text{min } K = \sum_{j=1}^{M} \sum_{i=1}^{L} c_j x_{ij} + \sum_{j=1}^{M} \sum_{i=1}^{L} c_i x_{ij} \) (b)
- \( \text{subject to } \sum_{j=1}^{M} x_{ij} = \gamma_i \) (c)
- \( \sum_{i=1}^{L} x_{ij} = \alpha_j \) (d)
- \( x_{ij} \leq 1 \) (e)
- \( x_{ij} \geq 0 \) (f)

- DRX cycles in VDRX can vary in accordance with the future channel conditions and buffer status.
- DRX cycles in DRXSet are chosen from a set of possible values and cannot be changed later.

Future Work
- Considering channel prediction errors
- Intelligent buffer management according to the user’s watching behavior
- Content placement optimization and inter-eNB cooperation in the presence of channel prediction

Reference
RF electronics for 5G wireless
Henrik Sjöland and Waqas Ahmad, Lund University

Transceiver architecture
The targeted system is massive MIMO with hundreds of antenna elements, see transceiver architecture depicted below. Each antenna radio frequency (RF) signal is handled by an antenna unit.

Remote Antenna Units
The figures below summarize our results on fully integrated CMOS remote antenna units for fiber fed distributed antennas, published in IEEE Transactions on Microwave Theory and Techniques, Jan. 2017.

Antenna Unit
The antenna unit will be implemented as a CMOS chip for small size, low cost and power. This will feature radio receiver + transmitter, and will communicate and synchronize with the baseband units over an electrical interface.

We have in a previous project (SSF Distrant) designed remote antenna units for distributed antenna systems fed by optical fibers. We will now build on that experience, but using it in a completely new system.

Measurements results
Harmonic Scheduling and Control Co-Design

Yang Xu, Anton Cervin, Karl-Erik Årzén

Motivation
- Demands on resource efficiency make it increasingly common with multiple (control) applications sharing the same computing platform
- Real-time task scheduling issues, e.g., priorities and preemption, affect the timing of controller tasks and, hence, the control performance
- Increased interest in control and scheduling co-design

Problem setup
- Set of independent controller tasks
- Uniprocessor with fixed-priority scheduling
- Sampling at job arrival (no sampling jitter)
- Actuation at end of the job
  - High priority controllers preempt low priority controllers, causing jitter in the delay
- Goal: Minimize the global LQG cost

Example
Non-harmonic periods (upper):
- Task $\tau_1$ (high prio): $T_1 = 3$, $C_1 = 1 \rightarrow R_1 = 1$
- Task $\tau_2$ (low prio): $T_2 = 5$, $C_2 = 3 \rightarrow R_2 = 5, 4, 4, 5, 4, 4, ...$

Harmonic periods (lower):
- $T_2$ changed to 6 $\rightarrow$ harmonic periods
- $R_1 = 1$, $R_2 = 5$ [constant]
- Adding an offset of 1 to $\tau_2$ $\rightarrow$ $R_2 = 4$ [constant]

Advantages of harmonic task periods
- Task set always schedulable if utilization $\leq 1$
- Response times are constant if execution times are constant $\rightarrow$ no jitter $\rightarrow$ better control performance
- Possible to assign release offsets to minimize delays

Co-design basic idea
- Assume that initial task periods that give good control performance are available
- Find the closest sets of harmonic periods:
  \[ T_{k+1} = m_k T_k, m_k \in \mathbb{N}, k \in \{1, 2, \cdots, n-1\} \]
  - Formulated in Theorem 1 of [2]
- Redesign the LQG controllers accordingly

Co-design procedure
- Calculate non-harmonic periods using Stochastic LQG design [IFAC 2014]
- Uses Bini & Cervin method [RTSS 08] to calculate initial periods
- Harmonize the periods using Theorem 1
- $2^{n-1}$ candidate harmonic period sets obtained
- Redesign controllers for all the candidate sets
- Optionally add task offsets
- Evaluate the control performance using TrueTime for all the candidate sets
- Real-time scheduling and control simulation toolbox based on Matlab/Simulink
- Select the candidate with the lowest cost

Results
Average total cost for 20 randomly selected sets of three plants from three different families:

<table>
<thead>
<tr>
<th>Family</th>
<th>I</th>
<th>II</th>
<th>III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-harmonic tasks</td>
<td>2.92</td>
<td>8.61</td>
<td>29.46</td>
</tr>
<tr>
<td>Harmonic tasks</td>
<td>2.53</td>
<td>5.17</td>
<td>17.76</td>
</tr>
<tr>
<td>Harmonic tasks with offsets</td>
<td>2.03</td>
<td>3.91</td>
<td>15.43</td>
</tr>
</tbody>
</table>

References
Intrusion-Damage Assessment and Mitigation in Cyber-Physical Systems for Control Applications

Rouhollah Mahfouzi¹, Amir Aminifar¹,², Petru Eles¹, Zebo Peng¹, Mattias Villani²

¹ Embedded Systems Laboratory, Linköping University, Sweden
² Embedded Systems Laboratory, Ecole Polytechnique Federale de Lausanne, Switzerland
³ Statistics and Machine Learning Division, Linköping University, Sweden

Control System

\[ \dot{x} = Ax + Bu + v \]
\[ y = Cx + e \]

- \( y \): expected state
- \( \hat{y} \): observed state
- \( h \): controller period

Resource – Quality trade-off

Decreasing sampling period leads to lower control cost (better control quality) but needs more resource

Attack Model

- Performance-observable (attack is observable by controller)
- Resource-controllable (allocating more resource can compensate)
  e.g. manipulation or discard messages from/to the controller

Problem Formulation

Uniprocessor platform

\[ f_i = \frac{1}{h_i} \]

Goals:
Assess the intrusion-generated damage and mitigate the impact of the attacks

Approach

Intrusion-Damage Assessment

a. Offline Learning
- model \rightarrow\) expected (\( y \))
- sensor \rightarrow real (\( \hat{y} \))
- history of states (\( \hat{y} \))

\[ X \sim N(\mu, \Sigma) \]

b. Online Evaluation
- Apply the state to Multi-Variate Normal Distribution \( X \)
- Compute metric \( \Delta \)

\[ 0 \leq \Delta \leq 1 \] is called Intrusion generated damage

Intrusion-Damage Mitigation

Idea: allocating more resource \rightarrow higher quality of control

- Previous frequency \( f^{(j-1)} \)
- Damage generated \( \Delta^{(j)} \)

Desired frequency \( f^{(j)} \)

Resource Management
Solve an optimization problem to find best frequencies based on desired frequencies.

Experimental Results

Attacker discards 10 messages from the green control system starting at time 10

Conclusions

- Proposing notion of observable and controllable attacks
- Proposing continuous intrusion damage assessment
- Learning the normal behaviour of the control system
- Mitigating attack using resource management

Bibliography

Aegis: Reliable Execution over the Mobile Cloud

Shubhabrata Sen, Jörn W. Janneck

Generalization: supporting a broader class of dataflow

The current failure detection and repair approach assumes that actors are deterministic and prefix-monotonic. That allows efficient local failure detection and simplifies repair and checkpointing. It also means that any time-dependent or reactive actor cannot be replicated in this system.

Time-dependent actors require special treatment in order to be executed in a replicated, distributed fashion. In essence, the nodes executing their copies need to achieve consensus over the execution path. Since that path can depend on input timing, they also need to achieve consensus over (at least) the order in which input arrives.

Distributed consensus is costly both in terms of time and communication overhead. We might get some leverage from exploiting that any one of the various alternatives is permissible, and by tuning the frequency with which consensus is maintained.

Generalization: supporting a more dynamic actor model

Our current model assumes that the structure of the dataflow program, i.e., the connections between actors, remain static throughout the execution of the system.

This is a model that works well for signal processing etc., but not being able to dynamically create, destroy, and connect actors is a significant limitation for more general applications.

If actors remain prefix-monotonic, the main challenge is to synchronize on creating actors, and on connection/disconnection events.

Future work: programming tools

Traditional implementation of dataflow programs can be guided by various tools for, for example, profiling, scheduling, and timing/throughput analysis that take advantage of the structure and degree of freedom provided by the dataflow model.

A mobile cloud presents an implementation substrate with novel characteristics. We need to study and understand, and then build tools for developers that allow them to anticipate the performance (and failure) characteristics of their application on that platform.

Persistence and catastrophic failure

Catastrophic failure happens when node or communication failures occur more quickly than the system is able to recover from them, and wind up overwhelming the ability of the system to repair itself.

The only way the system can recover from a catastrophic failure is through regular checkpointing. This could employ a hierarchy of storage, from local machines updated frequently to servers and ultimately cloud storage.

Update frequency would decrease along that persistence chain.

In recovering from catastrophic failure, the main challenge is consistency: if the checkpoints only consist of the state of actors, recovery would be impossible, since these states are gathered asynchronously, so there is no way to tell how they relate to each other.

Also, all in-flight communication would be lost.

The solution is to checkpoint actor state and enough information about the communication (buffers, current sequence numbers) to allow recovery. Actors are prefix-monotonic and determinate, so they will always reproduce the same results given the same inputs.

This work is a collaboration with Björn Landfeldt (EIT and MAPCI) and has been funded by MAPCI and ELLIIT.

Stream Computing Infrastructures
**Test charters as a vehicle to guide exploratory testing sessions**

**Presenter:** Kai Petersen (BTH)  
**Collaborators:** Ahmad Nauman Ghazi (BTH), Elizabeth Bjarnason (LTH), Per Runeson (LTH)

**Definition**

Exploratory software testing (ET) is a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project.

**One size does not fit all**

The exploration space is determined by the test charter. Which degree of exploration to strive for is an open question. We defined five levels of exploration and corresponding test charter types.

**A Decision Support Method**

Reflective tool taking input from practitioners to discuss the distribution of available testing time between the levels of exploration. Solution based on the Repertory Grid Technique, an approach for decision making in groups.

**Steps:**

1. **Element elicitation:** Identify the decision alternatives (levels of exploration and corresponding test charter types)
2. **Construct elicitation:** Identify the criteria for the decision (key differences between the options)
3. **Prioritize the criteria (distribute 100 points)**
4. **Assign effect value for each decision option in relation to the criteria (negative effect [=1], neutral [=2], and positive effect [=3])**
5. **Calculate recommendation taking effect value and priorities into consideration.**
Delta-Oriented FSM-Based Testing

M. Varshosaz and M. R. Mousavi

Motivation
- **Product lines**
  - Advantages: mass production, reduced price and time to market
- **Testing software product lines**
  - More challenging compared to testing single systems
  - Due to potentially large number of the products efficient testing is important

Approach
- Adapting the Harmonized State Identification (HSI) method for testing software product lines
- Using Finite state machines (FSMs) as test models with a structure based on DeltaJava
  - DeltaJava: a framework for delta-oriented programming of software product lines in Java

Delta-oriented FSM-Based Testing

1. **Delta-oriented FSM Modeling**
   - **Core Model**: FSM including abstraction of state valuations as states and the method calls as transitions

   ![Delta-oriented FSM Diagram](image)

2. **Delta-oriented HSI Test Case Generation**
   - Generating test cases for the core module
   - Automatically generating the test cases for products by modifying the core test cases with regards to the applied changes in the model

3. **Case Study**
   - A software system from healthcare domain
   - The core module includes 6 Java classes
   - Applied two types of delta modules

4. **Experimental Results**

<table>
<thead>
<tr>
<th>Core Model</th>
<th>After Applying Delta</th>
<th>Delta-Oriented Test Case Generation Steps</th>
<th>HSI Test Case Generation Steps</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>74</td>
<td>11</td>
<td>85</td>
<td>194</td>
</tr>
<tr>
<td>11</td>
<td>74</td>
<td>21</td>
<td>89</td>
<td>253</td>
</tr>
<tr>
<td>21</td>
<td>167</td>
<td>21</td>
<td>188</td>
<td>419</td>
</tr>
<tr>
<td>21</td>
<td>167</td>
<td>37</td>
<td>215</td>
<td>731</td>
</tr>
</tbody>
</table>

Publication

References
Gray-Box Conformance Testing for Symbolic Reactive State Machines

M. Taromirad and M.R. Mousavi

Motivation
- Model-Based Testing (MBT) is a black-box testing technique.
- Implementation under test (IUT) is accessible through its interfaces.
- Test data is selected based on the specification.
- Generated test suites may leave untested gaps in a given IUT and/or redundantly cover the same logical path several times.

Aim
- Enrich test models and test case generation processes with structural/behavioural information extracted from the IUT.
- Using learning-based approaches inferring models from software and providing an abstraction of the implementation based on its observable behaviour.

Gray-Box Conformance Testing

Symbolic Reactive State Machines (SRSM)
Symbolic representation of the state-based behaviour of a system, with input/output variables.

Transition Composition of SRSMs [TC]
A (sub-)product of the models in that the transitions are defined based on the intersection of transitions of the underlying models.

Test Generation [TG]
Abstract test sequences are generated based on the transition composition and then, the test inputs are selected based on ‘n-uniformity’.

n-Uniforinity
A special case of the uniformity hypothesis [2], the theoretical foundation for testing with a finite subset of values.

References

Effectiveness - Experimental Results
Case: Ceiling Speed Monitor (CSM); Railway Domain; IECP[3]

<table>
<thead>
<tr>
<th>IUT</th>
<th>Random Testing</th>
<th>IECP</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Killed</td>
<td>No. TCs</td>
<td>Killed</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>24</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>30</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>25</td>
<td>✔</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>37</td>
<td>✔</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>24</td>
<td>✔</td>
</tr>
<tr>
<td>6</td>
<td>✔</td>
<td>21</td>
<td>✔</td>
</tr>
</tbody>
</table>

Coverage
All the paths in both models are covered.

Soundness
None of the test cases fails, if the implementation conforms to the specification.

Relative Exhausivness
For any non-conformant behaviour in the implementation, there is a specific test case discovering the deviating behaviour.
One common feature request for OpenModelica has been generating real-time controllers for embedded systems. The previously existing code generators were too heavy-weight (requiring a processor with megabytes of memory due to the way the code was structured), limiting controllers to platforms such as the Raspberry Pi (which does not have analog input, but does not run OpenModelica simulations, but also the entire toolchain). The work to perform real-time simulations or controllers based on Modelica models using embedded systems with limited memory is the combination of two new developments:

- **OpenModelica has been extended with a code-generator that generates C-code which has minimal dependencies on libraries or OS functionality (mainly math functions, but only if they are used by the model).**
- The **Modelica_DeviceDrivers** (MDD) library has been extended with a sublibrary for implementing access to embedded hardware.

### OpenModelica Embedded C-Code Generator

One of the goals of the new C-code generator for embedded systems is to work on as many systems as possible. This is due to the desire to require as few modifications to the code generator as possible for each supported platform. The code generator supports few Modelica constructs (and no linear or nonlinear systems at the moment). In the future, it is expected to generate code for linear systems only if the model requires it. When compiling for AVR ATmega microcontrollers, including the Arduino Uno (Figure 1), code like BLAS cannot be included in the generated controller since the code is too complex (the microcontroller runs out of registers and available program code when compiling some BLAS routines).

The code generator only generates code that it knows will behave well on any given system. One side-effect is that the code has a small footprint (the SBHS real-time PID and LCD display controller in Figure 3 on the AVR ATmega16 microcontroller fits in 151 bytes of RAM and 4kB of flash memory, which is then using roughly 25% of the available resources).

To try out the new code generator, the current method is to use --simCodeTarget=ExperimentalEmbeddedC and then calling the appropriate C-compiler for the embedded target for the model itself does not use the commands to upload the controller to the board via the programmer is a working controller for the given system. One side-effect is that the code has a small footprint (the SBHS real-time PID and LCD display controller in Figure 3 on the AVR ATmega16 microcontroller fits in 151 bytes of RAM and 4kB of flash memory, which is then using roughly 25% of the available resources).

To try out the new code generator, the current method is to use --simCodeTarget=ExperimentalEmbeddedC and then calling the appropriate C-compiler for the embedded target for the code generator would work on pretty much any embedded target supporting C.

---

### Example: Single Board Heater System

One of the AVR examples included in MDD is the **Single Board Heater System** (SBHS). The SBHS consists of a heater assembly, fan, temperature sensor, AVR ATmega16 microcontroller and associated circuitry. It was developed by IIT Bombay and is used for teaching and learning control systems (Arora et al. 2010). The MDD SBHS example uses pulse width modulation (PWM) blocks to control the heater and fan, and an analog-to-digital converter (ADC) block to read the temperature. It combines these elements with a PID controller with the goal to control the fan such that the temperature settles at a setpoint of 45°C while the heater assembly is fed by a constant voltage. The board is modelled in MDD using drag-and-drop of blocks onto a diagram. Behind the scenes, there are external objects (C-code) which set the constants to make the AVR microcontroller run in real-time, set PWM outputs, control the LCD, and so on.

The result, when compiled with the OpenModelica code generator for embedded targets and uploaded to the board via the programmer is a working controller for the fan on the board:

**Figure 2**: The MDD example model of the single board heater system running at 125 Hz together with a real-time PID controller trying to control the fan such that the temperature is at the setpoint of 45°C while the heat source is fed by a constant voltage. The figure to the left is the model of the SBHS board with the fan and heat signal inputs and the measured temperature as output. The figure to the right is the model of the PID controller.

**Figure 3**: Single Board Heater System running a real-time control application using OpenModelica and Modelica_DeviceDrivers. There is a programmer attached to the board to upload new firmware, but the code runs without any computer connected to the SBHS.

---

### References


Single Board Heating System. URL: http://sbhs.fossee.in/.
**Problem**

Automation programs often need many different variants of the same basic computations. Each variant can be thought of as a combination of features. Example features for a control loop: master, slave, filters, etc. Example features for a tank: heating, agitation, filling, etc.

Programming each feature manually can be cumbersome and error-prone: the right blocks have to be selected, and wiring needs to be added in the right way. How can libraries capture these features, and support easier programming by automation engineers?

**Our solution**

New language constructs allow features to be programmed in libraries. The editor interprets the library to generate feature wizards. The automation engineer can simply select features in the wizard instead of adding and wiring blocks manually.

**Next step**

Develop larger example libraries. Compare to previous solutions:

- Cost reduction for development and testing
- Improvement of program quality

---

**Example: Create cascade controller for steam tank**

1) Add sensors, actuators and operator input

```
operatorSP
temperature
operatorGV
pressure
```

2) Use wizard to add controller with selected features

3) Controller blocks are automatically added and wired

5) The user can easily change the selected features

For example, replace the slave filter by the newly added MinZeroFilter

4) Example addition of feature alternative

Filter

```
in  noiseRemover  out
```

MinZeroFilter extends Filter

A new feature alternative can be added simply by adding a new subtype. In this case we create a MinZeroFilter that extends the existing Filter type. The new alternative appears automatically in the wizard.

---

**Example specification of optional features**

```
 recommendation Loop {}
    slave: ControllerPart[CV, slavePV: Int];

 recommendation ControllerPart {
    filter: Filter[controller.PV];
 }
```

More optional features can be added modularly, simply by adding new recommendations.
LinkBoard - Advanced Flight Control System for Micro Unmanned Aerial Vehicles

Mariusz Wzorek, Piotr Rudol, Gianpaolo Conte, Patrick Doherty

Artificial Intelligence and Integrated Computer Systems Division
Department of Computer and Information Science, Linköping University

Flight control systems (i.e. autopilots) play a crucial role in supporting autonomy in Unmanned Aerial Vehicles (UAVs). Larger platforms with larger payloads have the capacity to carry and use advanced computer systems with considerable computational capabilities. However, in small UAV designs custom-made autopilots have to be used due to limited payload and computational power capacities. The idea behind the LinkBoard is to design and develop a small flight control system which is capable of running basic control modes and sensor fusion but also the advanced algorithms and functionalities needed when operating autonomously. This is achieved by maximizing the available computational power and minimizing the physical dimensions and weight. The system is designed to be highly configurable, flexible and easy to extend with new sensors, algorithms, platforms and applications.

The LinkBoard has a modular design and this allows for adjusting the required computational power depending on mission requirements. Due to the available onboard computational power, it has been used for computationally demanding applications such as the implementation of an autonomous indoor vision-based navigation system with all computation performed on-board.

A newly developed attitude estimation filter has been implemented on the LinkBoard which combines the information from an accelerometer, a magnetometer and an orthogonally mounted triad of gyroscopes. The estimation is independent from the GPS measurement which has two implications. First, the attitude angles are not affected by GPS outages, which makes it useful for indoor navigation operations. Second, the pitch and roll estimations depend on the sensed gravity field acceleration. The consequence of this fact is that in case of accelerated flight the estimated angles will be affected by a bias error. The AHRS filter structure is depicted in the figure and it is implemented using the quaternion attitude representation.

The LinkBoard has been integrated and tested on a number of UAV platforms in various applications presented in multiple publications:

In this work we address the issue of connecting abstract task definitions at a mission level with control functionalities for the purpose of performing autonomous robotic missions using multiple heterogeneous platforms. The heterogeneity is handled by the use of a common vocabulary which consists of parametrized tasks such as fly-to, take-off, scan-area, or land. Each of the platforms participating in a mission supports a subset of the tasks by providing their platform-specific implementations.

At the mission level, we take advantage of a general and expressive task specification language - Task Specification Trees (TSTs). Inner nodes in a TST can specify standardized control structures such as sequences (S), concurrent execution (C), conditionals (IF) and loops (WHILE). Leaf nodes declaratively specify potentially domain-specific tasks to be executed, typically corresponding to high-level actions such as taking off or scanning an area.

In order to execute a particular node, a platform requires an executor providing a procedural implementation of the declarative specification encapsulated in a TST node. Platforms can share implementations of some executors, such as those corresponding to general control structures. However, most executors must be platform-specific in order to call the proper platform-specific functionalities. These executors must satisfy the general definition of the node type in question, such as fly-to.

We propose an approach for implementing such platform-specific executors. It takes advantage of a flight-command based control interface with setpoint generation abstraction layer for vertical take-off and landing platforms. We show that by using this highly expressive and easily parameterizable way of specifying and executing flight behaviors it is straightforward to implement a wide range of task executors.

Flight commands are a central component of the method for interfacing with an underlying control system. A flight command consists of three main components managing the control channels, namely horizontal, vertical, and heading, as well as a miscellaneous component dealing with aspects such as an end condition of a command.

![Diagram of Flight Commands](image)

The system presented has been fully implemented and used in numerous autonomous missions where TST executors were implemented using the flight-command concept.

**Flight command (s)**

- Target parameters \( P_t \)
- Setpoint generation
- Desired parameters \( P_d \)
- Control loops
- Actuators

An overview of the process of calculating control signals based on flight commands.
Model-Predictive Control with Stochastic Collision Avoidance using Bayesian Optimization

Olov Andersson, Mariusz Wzorek, Piotr Rudol, Patrick Doherty

Artificial Intelligence and Integrated Computer Systems Division
Department of Computer and Information Science, Linköping University

Summary

Want: Collision avoidance without prior coordination.

Requirements: Dynamics, uncertainty in obstacle movement, safety guarantees and real-time performance.

Problem: Stochastic MPC difficult due to recourse and constraints.

Contributions: (See [1])
- Novel parameterized soft-constrained approximations via probabilistically constrained policy search
- Leverage data-efficient Bayesian policy optimization learn safe policy parameters
- We demonstrate an MPC controller with integrated probabilistically safe collision avoidance with a real quadcopter system.

MPC with Probabilistic Safety Constraints

- Want the benefits of MPC, but with probabilistic safety constraint:
  \[ p(\text{dist}(p_{t+1}, p_o)) > 0, \forall t, \forall o > p \]
- A common simplification is to use the predictive distribution

Pedestrian Obstacle Models

- Humans interacting with robots without prior coordination can be inattentive and unpredictable
- The obstacle predictive uncertainty quickly grows very large for a human acceleration profile [2]

Bayesian Policy Optimization

- Need to optimize policy with probabilistic safety constraint
- We use recently proposed constrained Bayesian optimization [3]
- Objective and constraint are both approximated by data-efficient GP surfaces, where smoothness allows points to share strength

Warehouse Scenario

- Humans and UAV in small workspace
- UAV wants to pick up green packages
- 3 non-cooperative moving obstacles given destinations randomly

MPC Implementation with Real UAV

- Obstacle constraints are non-convex, but instead of costly MIP we compute local solutions via a simplified SQP based on [5]
- Since our notion of safety is validated empirically, we do not require global optima and can use random restarts
- The online MPC solution also includes task constraints on velocities, input saturation and input delta:

\[
\begin{align*}
\arg \min_{u_0, \ldots, u_{T-1}, x_1, \ldots, x_T} & \sum_{t=1}^{T} c(x_t, u_{t-1}) \\
\text{subject to} & \\text{dist}(p_{t+1}, p_{o}) > 0, \forall t, \forall o \Rightarrow \ \text{p} \\
& \text{arg min} \ \mathbb{E} \left[ \sum_{t=1}^{T} c(x_t, p_{o}) \right] \\
& \text{subject to} \ p(\text{dist}(p_{t+1}, P_{o})) > 0, \forall t, \forall o \Rightarrow \ \text{p} \\
& \text{We propose "MPC-policies", where} \ \text{probabilistic constraints are replaced by parameterized soft safety-margin feature functions:} \\
& \text{dist}(p_{t+1}, p_{o}) > m(\theta, x_t) \\
& \text{Want the benefits of MPC, but with probabilistic safety constraint:} \\
& \text{A common simplification is to use the predictive distribution} \\
\end{align*}
\]

Results

- Intended and measured safety levels

Approximations via Policy Search

- Stochastic and recourse are easier to handle with a global policy, but suitable policies can be difficult to construct

References


Deep Vision: Multiple Object Tracking

Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking

Martin Danelljan, Andreas Robinson, Fahad Khan, Michael Felsberg
Computer Vision Laboratory, Linköping University, Sweden

Introduction

Discriminative Correlation Filters (DCF):

- Single-resolution feature map
- Learns a set of discrete filters for target localization
- Outputs discrete detection scores

Our Approach:
Posing the learning problem in the continuous spatial domain

- Multi-resolution (deep) feature map
- Learns continuous filters
- Outputs continuous detection scores

Advantages
- Integration of multi-resolution (deep) features
- Accurate sub-pixel (or sub-grid) localization
- Sub-pixel supervision in the learning
- Efficient processing of all available information
- Avoids artefacts caused by explicit resampling

Applications
1) Object tracking 2) Feature point tracking

Convolution Operator Learning

Training loss

\[ E(f) = \sum_{j=1}^{m} \alpha_j \| S_f \{ x_j \} - y_j \|^2 + \sum_{d=1}^{D} \| \omega f_d \|^2 \]

- Training sample
- \( g(t) = \frac{1}{T} \int_0^T |g(t)|^2 \, dt \)
- Desired continuous output scores (labels)

Fourier Domain

Assumption: finitely many non-zero Fourier coefficients. Gives normal equations: \( (A^H F A + W^H W)^{-1} F = A^H Y \)

Object Tracking Framework
- Features: VGG network (pre-trained on ImageNet)
- Optimization: Conjugate Gradient

Feature Point Tracking Framework

Grayscale:

\[ \hat{f}[k] = \sum_{j=1}^{m} \alpha_j X_j[k] \hat{b}[k] \hat{y}_j[k] + \beta \sum_{j=1}^{m} \alpha_j X_j[k] \hat{b}[k] \hat{y}_j[k] \]

- Uniform regularisation
- \( w(t) = \beta \)

Continuous Convolution Operators

Interpolation Operator \( J_d : \mathbb{R}^N_d \rightarrow L^2(T) \)

\[ J_d \{ x^d \} (t) = \sum_{n=0}^{N_d-1} x^d[n] b_d \left( t - \frac{T}{N_d} n \right) \]

Convolution Operator

\[ S_f \{ x \} = \sum_{d=1}^{D} f^d \ast J_d \{ x^d \} \]

- Notation
  - \( g[k] \) - Fourier coefficients of \( g \in L^2(T) \)
  - \( X^d_j[k] \) - discrete Fourier transform of \( x^d_j \in \mathbb{R}^N_d \)

Experiments

Object Tracking: Layer fusion on OTB (100 videos)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Layer 0</th>
<th>Layer 1</th>
<th>Layer 2</th>
<th>Layer 3</th>
<th>Layer 4</th>
<th>Layer 5</th>
<th>Layer 6</th>
<th>Layer 7</th>
<th>Layer 8</th>
<th>Layer 9</th>
<th>Layer 10</th>
<th>Layer 11</th>
<th>Layer 12</th>
<th>Layer 13</th>
<th>Layer 14</th>
<th>Layer 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDA AUC</td>
<td>58.3</td>
<td>78.3</td>
<td>67.1</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
<td>67.3</td>
<td>77.3</td>
</tr>
</tbody>
</table>

OTB dataset (100 videos)

Temple-Color (128 videos)

VOT2016 challenge results (top 3) [Mate et al., VOT workshop 2016]

<table>
<thead>
<tr>
<th>Tracker</th>
<th>EAO A</th>
<th>R A</th>
<th>R_rank R_rank AO</th>
<th>EFO Impl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. C-SVEN</td>
<td>0.331</td>
<td>0.590</td>
<td>0.298</td>
<td>0.298</td>
</tr>
<tr>
<td>2. S-CNN</td>
<td>0.253</td>
<td>0.544</td>
<td>0.400</td>
<td>0.400</td>
</tr>
<tr>
<td>3. SSAT</td>
<td>0.237</td>
<td>0.577</td>
<td>0.297</td>
<td>0.297</td>
</tr>
</tbody>
</table>

Feature Point Tracking: The Sintel dataset

- Error distribution plot
- Precision plot
- Recall plot
Topological controlled tensor field simplification

Jochen Jankowai, Ingrid Hotz
Linköping University

Background

Tensor as linear operator

$T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}$

- Eigenvalues $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$
- Anisotropy $D = T - 0.5 \times \text{trace}(T) \times I$
- Isotropic part $0.5 \times \text{trace}(T) \times I$
- Deviator $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$

Tensor field topology

Segmentation of the field into regions of uniform tensorline behavior.

Tensorlines: lines following eigenvectors.

Texture highlighting eigenvector directions
Skeleton: detail and simplification (sketch)

Degenerate points $D = 0$

Structural distinguished points
- Winding number (degree) of degenerate points (DP) are half integers $\pm n \frac{1}{2}$
- First order DP

Goal and concept

Multi-scale structural simplification of the tensor field respecting the stability of topological features.
Quantification of stability of degenerate points with respect to field perturbations.
Hierarchical cancelation of pairs of critical points.

Measuring perturbation of tensor fields

Given a continuous tensor field $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2, f(x) = T$
A continuous tensor field $h$ is an $r$-perturbation of $f$, if $d(f, h) \leq r$

$$d(f, h) = \| f - h \| = \sup_{x} \| f(x) - h(x) \| \leq r$$

- What is an appropriate metric in $\mathbb{R}^2$ and $\mathbb{R}^2$?
- How is it related to the existence of degenerate points?
  - Parameterize tensor space in terms of eigenvectors and anisotropy

Robustness of degenerate points

Robustness quantifies the stability of critical points with respect to the minimum amount of perturbation in the fields required to remove them.

$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ continuous tensor field with a finite number of isolated degenerate points.

Robustness diagram

All possible $r$-perturbations

tracks changes of the $0$-dim homology group, number of connected components in $F_r$ with $\text{deg} r > 0$

Illustrative example

Simple collection of degenerate points

Variant 1

Variant 2

Project: Feature based visualization
Tensor volume exploration
Jochen Jankowai, Robin Skånberg, Anders Ynnerman, Ingrid Hotz
Linköping University

Tensors and tensor fields
Tensors are widely used in many scientific fields as descriptors for anisotropic physical entities.

Tensor data result e.g. from numerical simulations of material stresses or from diffusion tensor imaging of the human brain.

Concept
Volume rendering of tensor data that combines

1. Intuitive interaction interface
   Domain experts can navigate the feature space (FS) in their familiar context to design a tensor transfer function (TTF)
   - FS parametrization by few representatives
   - Glyphs as intuitive interaction widgets

2. Tensor field rendering
   - Textured volume rendering to convey tensor characteristics
   - Legend using glyphs to encode the complex tensor information
   - Feature level-set allow analysis of critical regions

Pipeline

Goal
Cross-application systematic approach to volumetric tensor visualization maintaining domain specific choices of glyphs and other design parameters.

Challenges
- Complexity of the tensor data
- High dimensionality of the interaction space
- Diversity of the applications

Results

DTI Brain data set
Clustering in feature space (Fig. top left) reveals five major clusters, each represented by one glyph. Left image: Volume rendering of three clusters. The largest cluster that is related to image noise has been removed from the rendering. For texturing, tensor lines have been seeded in the red segment whose representative is also highlighted. Right image: Segment volume in an atlas like stile, sliced along the y-axis.

Stresses in a solid block with two applied forces
Manually selected representatives from one slice, one outlier exhibiting high stresses (green) and a typical low stress state (purple). The rendering shows a volume combining two radial fall-off transfer function and a level-set surface at a fixed distance to the outlier in feature space to highlight critical regions in the material. Short tensor lines seeded throughout the volume to convey directional information of the stresses. The feature space used was defined by shear stress, shape factor, and mean stress.

Simulation of a rotating neutron star
The feature space (figure below) of this data set does not show strongly expressed clusters. Clustering defines an almost uniform segmentation of a 3D subspace of the feature space on which the tensors of this simulation live. The renderings show volumes for different clusters also used for seeding the tensor lines which forms the texture.

Project: Feature based visualization
Distributed Robustness Analysis

Anders Hansson, Sina Khoshfetrat Pakazad
Anders Rantzer and Martin Andersen

**Table:**

<table>
<thead>
<tr>
<th>Solver</th>
<th>Avg. CPU time [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPT3 (lumped)</td>
<td>5640</td>
</tr>
<tr>
<td>SeDuMi (lumped)</td>
<td>2760</td>
</tr>
<tr>
<td>DSDP (sparse)</td>
<td>167</td>
</tr>
<tr>
<td>SMCP (sparse)</td>
<td>33</td>
</tr>
<tr>
<td>ALM (sparse)</td>
<td>1623</td>
</tr>
</tbody>
</table>

Scalable Optimization for Control Systems
The travelling salesman

The proposed genetic algorithm is based on a population of \( p_{\text{max}} \) feasible solutions, \( S = \{ s_1, s_2, ..., s_n \} \), a set of permutation vectors, \( A, B \), and a set of control parameters (see [5]). On each iteration, \( i \), subsets of the population, \( S_i \subseteq S \), are shuffled by these same permutation vectors. The only fit limit involves with respect to the selection criteria survive to the next generation. The approach is similar to the work in [2], but employs a different path representation, additional permutation laws and a new convergence criteria which is later extended to take point subset priorities into account. We define a close-to-optimal solution when the criteria

\[
1 - \frac{J_{\text{GA}}(s)}{J_{\text{BnB}}(s)} < \epsilon,
\]

is met. In addition, by allowing restricting permutation laws to set on subsets of the path vectors which share priority, feasibility conditions of priorities are satisfied at a slightly decreased computational cost (see Fig. 1).

\[\text{Path representation}, \ C \equiv \{ C_0, C_1, ..., C_n \}, \text{ with a solution vector } \{ 0, 1, 0, 1 \}. \]

**Figure 1:** Path vector representation.

**Figure 2:** TSP solution generated (blue) and obstacle (black), implementing two priority subsets. Beginning in a starting point (black), high priority points \( 1, 11, 20 \) (red) and low priority points \( 2, 10, 21, 30 \) (green) are traversed before ending up in the start point.

**TSP-GA planning with priority subsets**

For the simplest possible exact brute force method of checking all possible solutions, the worst-case complexity is \( \mathcal{O}(N! \cdot \sum_{i} a_i) \). When adopting the BnB algorithm or the dynamical programming approach of Held and Karp as described in [8], the worst-case complexity is slightly better but still exponential, as shown in [6]. The derived GA cannot be examined in the traditional sense, but we may relate it to the worst-case complexity of the Held-Karp algorithm. Computationally by comparing the 95% confidence intervals \( \mu(N), \sigma(N) \) of the number of iterations required to converge, \( \mu(N), \sigma(N) \), to a close-to-optimal solution for [1]. By fitting the costs of various degrees, a simple quadratic polynomial was found to be a satisfactory fit of the upper confidence bound (see Fig. 4). This upper bound is scaled linearly with \( N \), leading to the total time complexity \( \Omega(N^3) \) (see Fig. 4).

**Results I: TSP-GA benchmark**

To demonstrate the presented algorithm in a real-time context, the Cayenne 2.0 UAV is used. The control system is designed and implemented in the UAV-W framework (see [9]). Some means of positional information is required to control the well known plane. Part of the navigation system is integrated in the flight test. The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.

The control system is a self-contained unit.
Real-Time Trajectory Generation Using Model Predictive Control

M. Mahdi Ghazaei A., Björn Olofsson, Anders Robertsson, Rolf Johansson

Introduction

We address the problem of generating movements to transfer a robot from its current state to a new position and velocity at a certain time, when the target state and the final time may require correction at a high rate.

Model Predictive Control Approach

Given the initial state of the system $x(k)$, solve at each sample $k$

- $V_k(x, u)$
- Optimizer
- Model

Minimize

subject to

- $x(k+1) = f(x(k), u(k))$
- $z(k) = h(x(k))$
- $Fu(k) \leq f, GU(k) \leq u$
- $z(k+N) = r_f(k)$ (optional)

Real-Time Considerations

- Linear interpolation
- Convex optimization problems
  - unique optimum
  - efficient solvers (CVXGen)
- Improving the resolution sequentially
  - a terminal constraint in the prediction horizon
  - successively reducing the sampling period

Fixed-Time Point-to-Point Trajectory Generation

- Examples of initial and final states

  - Initial state
  - Final state

  - Sparsely-defined reference (soft or hard constraint)
  - Set of indices with desired values $\Psi(k)$

  $V_k(\mathcal{U}) = \sum_{i=k+1}^{k+N} \|z(i) - r(i)\|_Q^2 + \sum_{i=k}^{k+N-1} \|u(i)\|_R^2$

  $= \sum_{i=k}^{k+N-1} \|z(i) - r(i)\|_Q^2 + \sum_{i=k+1}^{k+N-1} \|z(i)\|_Q^2$

  $+ \sum_{i=k}^{k+N-1} \|u(i)\|_R^2$

  where $\bar{Q}(i) = Q(i)$ if $i \notin \Psi(k)$ and $\bar{Q}(i) = 0$ if $i \in \Psi(k)$.

Ball-Catching Example

- Replanning in real-time because of improved estimation accuracy and disturbances
- A model including kinematic variables
  - constraints on joint jerk, acceleration, and velocity
  - multiple triple integrators
  - resonance modes

Conclusions

- Trajectory generation as a controller design problem
  - online trajectory generation
- Adapting Model Predictive Control (MPC) formulation
  - point-to-point fixed-time trajectory generation
  - coordination and reduction of energy consumption
- Real-time implementation of a subset of the problems


Optimal Maneuvers & Collaborative Robotic Systems
**Background & Motivation**

Automated driving systems are on the horizon of driver-assistance systems. In an automated driving system as defined by the SAE [1] and adopted by NHTSA [2], the human driver is not required to monitor the driving environment and is at most only conditionally required to intervene. The need to be able to handle time-critical situations is evident. Many traffic situations, such as a pedestrian unexpectedly running out in front of the car leaves little time to react. The ability to perform maneuvers that push close to the limits of the vehicle’s capabilities is thus necessary. Automation of vehicles and the addition of sensors create new possibilities for the operation of active-safety systems. Traditional yaw-control systems such as electronic stability control (ESC) stabilize the vehicle given the steering input supplied by the driver [3]. For an automated driving system, such a yaw-control may provide suboptimal braking torques to the wheels. In [4], a control strategy based on vehicle speed, road curvature, and available lateral acceleration was derived and shown to outperform yaw control for a lane-keeping scenario.

**Methods & Preliminary Results**

Optimal control may be used as a tool to analyze the validity of vehicle models and study optimal trajectories during critical maneuvers [5]. Figures 2-4 show optimal braking maneuvers when entering a curve with too high speed. The trajectories are calculated using JModelica.org [6] for modeling and optimization. In this case, the entry speed is a free parameter for the optimizer and the vehicle is constrained to stay within a lane. The optimization problem is stated mathematically as:

\[
\begin{align*}
\text{minimize} & \quad -\eta \omega_0 - (1-\eta) v_f \\
\text{subject to} & \quad T_{u,i,\text{min}} \leq T_{u,i} \leq 0, \quad i \in \{1, 2, 3, 4\}, \\
& \quad |\delta| \leq \delta_{\text{max}}, \quad |\dot{\delta}| \leq \dot{\delta}_{\text{max}}, \\
& \quad |\epsilon| \leq \epsilon_{\text{max}}, \quad \dot{\epsilon}(t_f) \leq 0, \\
& \quad f(X_p, Y_p) \leq 0, \\
& \quad \dot{x} = G(x, y, u), \quad h(x, y, u) = 0.
\end{align*}
\]

Figure 2 shows one of the resulting paths. Depending on the optimization criterion, different velocity profiles are obtained as seen in Figure 3. Corresponding moment \( \Delta M \) due to applied braking torques are seen in Figure 4. It can be seen that the optimal braking strategy varies greatly when adjusting the optimization criterion. This work is accepted for publication [7].

**Future Research**

The goal is to further develop and improve knowledge of vehicle control systems that perform close to physical limits. Furthermore, in the intended applications such as safety systems, there are real-time constraints, nonlinear dynamics, and significant model uncertainty to account for. For optimization-based systems two research challenges are: How to define suitable optimization criteria in dynamically changing environments, and how to adhere to real-time computational constraints with complex nonlinear models.

**Bibliography**


