FRAMEWORKS AND TOOLS FOR HIGH-CONFIDENCE OF ADAPTIVE DISTRIBUTED EMBEDDED CONTROL SYSTEMS

Janos Sztipanovits and Gabor Karsai
ISIS-Vanderbilt University
Claire Tomlin, Edward Lee and Shankar Sastry
UC Berkley
Bruce Krogh and Edmund Clarke, CMU
Stephen Boyd, Stanford
Air Force (FA9550-06-1-0312)

July 16, 2009
Team

- Vanderbilt
  - Sztipanovits (PI), Karsai, Kottenstette, Porter, Hemingway, Thibodeaux
- UC Berkeley
  - Tomlin (PI), Lee, Sastry, Gonzales, Zhou, Leung, Hsia, Ding, Brooks
- CMU
  - Krogh (PI), Clarke, Kapinski, Rahjans, Jain, Lerda, Bhave, Donze
- Stanford
  - Boyd (PI), Skaf, Joshi, Mutapcic, Kim
Project Overview

Develop a comprehensive approach to the model-based design of high-confidence embedded control systems.

- Hybrid Control Systems (aircraft collision avoidance): Tomlin, Krogh, Sastry
- Integrated Software Model Checking (bisimulation functions) and new widening operator for verification on numerical programs: Clarke, Krogh
- Networked-digital-control of (cascaded) continuous-time conic (passive) systems which are $L^m_{2}$-stable: Kottenstette
- Working prototype of an end-to-end tool-chain for model-based design of networked-control systems: Sztipanovits, Karsai, Kottenstette
- Ptolemy-II extended to build code generators based on synchronous dataflow, FSM, ..., models of computation which can target micros, microkernels, and RTOS’s: Lee
- Convex (optimal, affine, model-predictive, ...) control: Boyd
Outline

- Networked-digital-control of (cascaded) continuous-time conic (passive) systems which are $L^m_2$-stable: Kottenstette
  - Nested controllers for cascades of (passive) conic systems
    - Quad-rotor control problem
  - Digital Control of (non-linear) continuous-time systems – Fundamental limitations
    - use wave-variables, a constructive solution
    - networked control of passive cont.-time systems
- Polyhedral Domains and Widening for Verification of Numerical Programs: Krogh
- Working prototype of an end-to-end tool-chain for model-based design of networked-control systems: Sztipanovits, Karsai, Kottenstette
Conic systems

\[
\int_{0}^{NT_s} y_p^T(t)y_p(t)dt - (a_p + b_p) \int_{0}^{NT_s} y_p^T(t)u_p(t)dt + a_p b_p \int_{0}^{NT_s} u_p^T(t)u_p(t)dt \leq 0
\]

\[
\| (y_p)_{NT_s} \|_2^2 - (a_p + b_p) \left\langle y_p, u_p \right\rangle_{NT_s} + a_p b_p \| (u_p)_{NT_s} \|_2^2 \leq 0
\]

\[
\sum_{i=0}^{N-1} y_p^T(i)y_p(i) - (a_p + b_p) \sum_{i=0}^{N-1} y_p^T(i)u_p(i) + a_p b_p \sum_{i=0}^{N-1} u_p^T(i)u_p(i) \leq 0
\]

\[
\| (y_p)_N \|_2^2 - (a_p + b_p) \left\langle y_p, u_p \right\rangle_N + a_p b_p \| (u_p)_N \|_2^2 \leq 0
\]

inside the sector \([a_p, b_p]\),

\(-\infty < a_p \leq b_p, 0 < b_p \leq \infty\)

Digital Passive Attitude and Altitude Control Schemes for Quadrotor Aircraft (Kottenstette, Porter: to appear ICIA’09)
**Conic systems - Properties**

\[ H_1 : u_1 \to y_1 \]

**Properties**

1. **Strictly inside** \([0, 1 + \epsilon] \) \(\epsilon > 0\), \([\epsilon, 1] - 1 \leq \epsilon \leq 1\)
Conic systems – Feed-back Properties

\[ H : u \rightarrow y \]

\[ [0, \infty], 0 < k_i < \infty \]

\[ [0, 1] \]

\[ [0, k_o] \]

\[ [0, 0] \]

\[ [0, b_1] \]

\[ [0, b_2] \]

\[ e_1 \]

\[ k I \]

\[ u \]

\[ y \]

\[ \begin{align*}
let \quad & y^T = [y_1^T, y_2^T], u^T = [u_1^T, u_2^T]: \quad H : u \rightarrow y \quad \text{is inside} \quad [0, \max\{b_2, k_b\}] 
\end{align*} \]
**Conic systems – Feed-back Properties**

Assume \( u_2 = 0 \) and \( b_2 = \infty \) then \( H : u_1 \rightarrow y_1 \) is inside \([0, kb_1]\), in addition for many cases \( H_{12} : u_1 \rightarrow y_2 \) is inside \([a_{12}, b_{12}]\), \( a_{12} < 0, 0 < b_{12} < \infty \)

Assume \( H_1 : e_1 \rightarrow y_1 \) is inside \([a_1, b_1]\) and \( H_2 : e_2 \rightarrow y_2 \) is strictly inside \([0, 1 + \epsilon]\), \( \epsilon > 0 \) then \( H : u \rightarrow y \) is bounded if

\[
0 < k < -\frac{1}{a_1} \quad \text{if} \quad a_1 < 0, \quad 0 < k < \infty \quad \text{if} \quad a_1 = 0,
\]

\[-\frac{1}{a_1} < k \leq \infty \quad \text{otherwise}.\]
**Four Integrators – No Saturation**

Step Response $K_4 = 157.0796$

Nyquist Diagram

- $k_4$ (outside)
- $\frac{k_4}{8}$ (outside)
- $\frac{k_4}{4}$ (outside)
- $\frac{k_4}{2}$ (outside)
- $\frac{1}{s}$ (outside)

[-1/2, ∞]

[-1/2, ∞]
Quad-Rotor Cont. Subj. To Actuator Saturation

\[ \dot{\zeta} = v_I \]

\[ m \dot{v}_I = f_I = mge_D - TR^\top (\eta)e_z \]

\[ I \ddot{\omega} = -\omega \times I \omega + \Gamma \]

\[ \dot{\eta} = J(\eta)\omega \]
IPESH-Transform used to synthesize lead-compensators.

\[
T = -f_{lcz}, \\
\begin{bmatrix}
\phi_{set} \\
\theta_{set}
\end{bmatrix} = \begin{bmatrix}
s_\psi & -c_\psi \\
c_\psi & s_\psi
\end{bmatrix} \begin{bmatrix}
f_{lcx} \\
f_{lcz}
\end{bmatrix}, \\
\begin{bmatrix}
T_1 \\
T_2 \\
T_3 \\
T_4
\end{bmatrix} = \begin{bmatrix}
0 & -\delta & 0 & \delta \\
\delta & 0 & -\delta & 0 \\
-K_t & K_t & -K_t & K_t \\
1 & 1 & 1 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
\gamma_x \\
\gamma_y \\
\gamma_z \\
T
\end{bmatrix}
\]
Attitude PD-Controller

\[ H_{\eta} : \omega \rightarrow \eta \quad [-0.004, \infty] \]

\[ H_{\omega c} : \Gamma_c \rightarrow \omega \quad [0, \infty] \]

\[ H_{k\omega} : k_{\eta} e_{\eta} \rightarrow \omega \]
Inertial PD-Controller

IPESH-Transform used

\[
\text{diag} \left\{ \frac{s}{\tau_v s + 1} \right\}
\]

\[
\begin{bmatrix}
0 \\
0 \\
mg
\end{bmatrix}
\]
**Advanced Quad-Rotor System Model**

- Used Extensively
- OpenC2WT
- Integrated seamlessly with autonomous target trackers

\[
\begin{align*}
\zeta (m) \\
\dot{\zeta} (m/s) \\
\psi (rad) \\
\text{error (m)}
\end{align*}
\]
Quad-Rotor System Model

Robostix (attitude)

Gumstix (inertial)

Inertial, $T_s = 0.01s, 16$-bit, GPS, $T_s = 0.2s$
UAV Dynamics

Blade Flapping Effects

Rotor Thrust Model

Wind Disturbances

- Highly detailed and refined model
- UC-Berkley, CMU, Vanderbilt
Recall Passivity Theorem

If \( H_p : u_p \to y_p \) is inside the sector \([0,b_p]\)

and \( H_c : u_c \to y_c \) is inside the sector \([0,b_c]\)

then

\[
\langle y_p, r_p \rangle_{NT_s} + \langle y_c, r_c \rangle_{NT_s} \geq \frac{1}{b_p} \left\| (y_p)_{NT_s} \right\|_2^2 + \frac{1}{b_c} \left\| (y_c)_{NT_s} \right\|_2^2
\]

let \( y^T = [y_p^T, y_c^T], r^T = [r_p^T, r_c^T] \)

\( \langle y, r \rangle_{NT_s} \geq \frac{1}{b} \left\| (y)_{NT_s} \right\|_2^2 \), \( b = \max b_c, b_p \)

\( H : u \to y \) is inside the sector \([0,b]\)
Res. Controller Has Infinite $L^m_2$ - Gain

Let $H_{dc} : u_{dc} \rightarrow y_{dc}$ be a constant gain $k_{dc} = 1$ which is inside the sector $[1,1]$.

$$u_c(t) = \begin{cases} 
1, & t = i T_s \\
-\delta & (\delta > 0), \text{ otherwise.}
\end{cases} \quad \rightarrow \quad y_c(t) = 1$$

correlated noise BAD

$$\langle y_c, u_c \rangle_{NT_s} = -\delta NT_s < 0 \rightarrow H_c \text{ is not passive}$$

$$\| (y_c)_{NT_s} \|_2^2 = NT_s \rightarrow \gamma^2 < \frac{1}{\delta^2} = \lim_{\delta \to 0}$$

therefore $H_c$ can only be bounded by the sector $[-\infty, \infty]$. 

\[ \text{Diagram of system with ZOH, } H_{dc}, \text{ and control signal} \]
Not all is lost, can handle many inputs.

clear path for some special definitions...

If $H_{dc} : u_{dc} \rightarrow y_{dc}$ is inside $[0, b_{dc}]$ and

$$\int_{iT_s}^{(i+1)T_s} u_c^T(t)u_c(t)dt \geq T_s u_{dc}^T(i)u_{dc}(i)$$

then

$H_c : u_c \rightarrow y_c$ stays inside $[-b_{dc}, b_{dc}]$. In addition, if

$$\int_{iT_s}^{(i+1)T_s} u_{cj}(t)dt \geq T_s u_{dcj}(i), \quad y_{dcj}(i) > 0$$

and

$$\int_{iT_s}^{(i+1)T_s} u_{cj}(t)dt \leq T_s u_{dcj}(i), \quad y_{dcj}(i) < 0$$

then $H_c$ stays inside $[0, b_{dc}]$. 

Anti-aliasing filters don't necessarily solve the problem.
**$L^m_2$– stable networks w/ digital control**

If $G_{p_2} : e_{p_2}(t) \to f_{p_2}(t)$ is inside $[0, b_{p_2}]$ and $G_{c_1} : f_{c_1}(i) \to e_{c_1}(i)$ is inside $[0, b_{c_1}]$ and delays... then denoting $y^T = [f_{p_2}^T, e_{c_2}^T]$, $r^T = [r_{p_2}^T, r_{c_1}^T]$ $H : r \to y$ is inside $[0, b]$ ($b = \max\{b_{p_2}, Ts^2k_{s1}^2b_{c_1}\}$).

**Design of Networked Control Systems Using Passivity**
(Kottenstette, Hall, Koutsoukos, Sztipanovits, Antsaklis, under review TPDS)

(extra bonus) delays essentially do not matter
Bilinear Transform w/ Wave Variables

\[
\begin{bmatrix}
    u_{p2}(t) \\
    e_{dc2}(t)
\end{bmatrix} = \begin{bmatrix}
    -I & \sqrt{2}bI \\
    -\sqrt{2}bI & bI
\end{bmatrix}\begin{bmatrix}
    v_{p2}(t) \\
    f_{p2}(t)
\end{bmatrix}
\]

\[
\begin{bmatrix}
    v_{c1}(i) \\
    f_{dp1}(i)
\end{bmatrix} = \begin{bmatrix}
    I & -\frac{\sqrt{2}}{b}I \\
    \sqrt{\frac{2}{b}}I & -\frac{1}{b}I
\end{bmatrix}\begin{bmatrix}
    u_{c1}(i) \\
    e_{c1}(i)
\end{bmatrix}
\]

\[
\frac{1}{2}u_{p2}^T(t)u_{p2}(t) - v_{p2}^T(t)v_{p2}(t) = f_{p2}^T(t)e_{dc2}(t)
\]

\[
\frac{1}{2}u_{c1}^T(i)u_{c1}(i) - v_{c1}^T(i)v_{c1}(i) = f_{dp1}^T(i)e_{c1}(i)
\]
Passive Sampler (PS) and Passive Hold (PH) provide a causal tool to transform a continuous-time signal to a discrete-time signal:

\[ \|(u_{p2}(i))_N\|^2 \leq \|(u_{p2}(t))_{NT_s}\|^2, \quad \|(v_{p2}(t))_{NT_s}\|^2 \leq \|(v_{p2}(i))_N\|^2 \]

straight forward to implement digital-anti-aliasing-filters

Basic power junction network \( u_1(i) = u_2(i), \) \( v_2(i) = v_1(i) \) in which total-power-in \( \geq \) total-power-out.

\[ u_2^T u_2 - v_2^T v_2 \geq u_1^T u_1 - v_1^T v_1 \]
L^m_2 – stable digital control networks

Single Digital Controller, Multiple Continuous Plants [1]
(Kottenstette, Chopra: to appear Necsys 2009)

Digital Control of Multiple Discrete Passive Plants Over Networks (Kottenstette, Hall, Koutsoukos, Antsaklis, and Sztipanovits: under review IJSCC)
$L^m_2$ -- stable digital control networks

- Both consensus-like networks and (even better) parallel networks can be created which allow:
  - passive-plants (different dynamics) with similar steady-state gains to be led by a single SISO PID-controller.
  - used to coordinate multiple robotic arms
  - Can be used to coordinate aircraft
  - resilient networks to be created which allow redundant controllers to control a single passive plant.
Polyhedral Domains and Widening for Verification of Numerical Programs

- code generation
- target processor
- compiler

model-based development

source code implementation

Goran Frehse
Verimag

Hitashyam Maka and Bruce H. Krogh
Carnegie Mellon University

platform implementation
Verification of numerical programs

Problem: simulation tools don’t execute numerical code exactly as it will be executed on the target processor (e.g., because of different round off properties due to different word-lengths)
Verification of numerical programs

To achieve verification of the target code, the binary is disassembled into a control flow graph – similar to what is done in compilers.
Verification of **linear** numerical programs

CFG is augmented with a model of the numerical error propagation (IEEE 754 standard), leading to a linear hybrid automaton (LHA) model.

A LHA reachability tool which calculates error bounds on critical variables by using polyhedra in order to, for example, over-approximate piecewise affine dynamics.
Polyhedral domains

- convex polyhedron: conjunction of linear predicates
- polyhedron: disjunction of convex polyhedra
- Parma Polyhedra Library (PPL): performs exact computations with non-convex polyhedra
- PHAVer: performs reachability for LHA
  - exact and robust arithmetic with unlimited precision (PPL)
  - bit-constrained over-approximations for termination heuristics
  - on-the-fly over-approximation of piecewise affine dynamics
  - support for compositional and assume-guarantee reasoning.
Finite resources require over approximation
Semi-bounded exact arithmetic
- exact computations (rational arith.) that result in finite precision

1. generate time-derivative polyhedron

2. * compute conservative over-approximation (bits bounded!)

* Managing the complexity by over-approximation

1. limit the number of bits of coefficients

2. limit the number of constraints

bounding the number of bits guarantees convergence of reachability computations
Standard widening: example (Cousot)

Program 1

1: initially: $x \in [4, 8], y \in [1, 2]$
2: while true do
3: \hspace{1em} $t_1 = 0.25 \times x$
4: \hspace{1em} $t_2 = 0.25 \times y$
5: \hspace{1em} $x = t_1 + t_2$
6: \hspace{1em} $y = 0.5 \times y$
7: end while

std. widening
- terminates at iteration 5
- large over approximation

(b) Applying standard widening at iteration 5
Example 1: Application to Program 1

(b) Applying standard widening at iteration 5

\textit{std. widening}

(b) Over-approximation of the reachable states, applying $\nabla_{CL,k}$-widening with $k = 4$ at iteration 5

\textbf{coefficient-limiting widening}
Example 2: Non-convex polyhedra (w/o convex hull)

Program 2
1: initially: $x \in [4, 8]$, $y \in [4, 8]$
2: while true do
3: \hspace{10pt} $t1 = 0.125 \times x$
4: \hspace{10pt} $t2 = 0.433 \times y$
5: \hspace{10pt} $t3 = 0.25 \times y$
6: \hspace{10pt} $t4 = 0.2165 \times x$
7: \hspace{20pt} $x = t1 + t2$
8: \hspace{20pt} $y = t3 + t4$

(a) No termination without widening

(b) Termination with $\nabla_{CL,k}$-widening with $k = 4$
Summary

• new widening operator based on bounding coefficients
• error-preserving reductions
  - transition merging
  - variable elimination
• implementation using PHAVer

next steps
• exercise on benchmarks
• integration with standard tools
• incorporating other sources of errors (e.g., inf, NaN)
Focus of the toolchain

The toolchain turns the *functional controller design* into a *software implementation*: a collection of integrated components executed by a *robust component platform* that runs on a *system/hardware platform*. 
Toolchain overview

Functional Design
Simulink/Stateflow

Model-based Design
• Componentization
• Architecture modeling
• Deployment modeling

Model-based Code Generation

Analysis
• Platform effects
• Verification

Schedule Generation

Schedule
SW Platform
Analysis and verification

- Platform effects analysis
  - *TrueTime* model for platform
  - Generator to build Simulink model for functional model + platform model
  - Continuous time, high-precision Simulink simulation allows studying subtle platform effects

- Verification (*Leverage NASA MICTES Project - JPF*)
  - Functional model state reachability verification
  - Functional code verification

*Objective: To give feedback for the designer*
ESMoL: Embedded System Modeling Language

- **Components:** ( Imported from Simulink)
  - Synchronous dataflow network + Statecharts
  - Run at a fixed rate - periodic execution:
    \[ x(k+1) = f(x(k),u(k)) \] – State update
    \[ y(k+1) = g(x(k),u(k)) \] – Output Update
  - Responsible for (periodic) I/O interactions

- **Component architecture:**
  - Components are statically scheduled according to their rates
  - Communication is facilitated by time-triggered messages (also statically scheduled)

- **Hardware platform:**
  - Each node runs a cyclic, static, timed-triggered schedule
  - Nodes communicate via time-triggered messages

- **Deployment model:**
  - Time-triggered tasks hosting components, interfacing via messages, mapped to processor nodes
Execution: Time-triggered Platform

Execution model:
- Timer/clock-interrupt driven periodic execution of tasks according to a static schedule
- Tasks do not preempt, except in case of overruns
- If a task overruns its WCET, it is cancelled and the next task is started
- Tasks are scheduled with sufficient slack to allow for I/O interrupts
- I/O-driven I/O communicates with component/tasks via shared buffers
A Platform for Experimentation

Model-based Toolchain

Real-time Simulation Platform ‘Virtual plant’

Next steps:
- Supporting other Models of Computation
- Integration of heterogeneous MoC-s
- Fault management