Imperial College London

Number Representations for Embedding Optimization Algorithms in Cyber-Physical Systems

Eric Kerrigan, Juan Jerez, Stefano Longo, George Constantinides
Department of Electrical and Electronic Engineering
Department of Aeronautics

EPSRC Engineering and Physical Sciences Research Council
Seventh Framework Programme
EMBOCON
esa
XILINX
National Instruments
The MathWorks

http://cyberphysicalsystems.org

- "Cyber-Physical Systems (CPS) are integrations of **computation**, **networking**, and **physical** processes. **Embedded** computers and networks monitor and control the physical processes, with feedback loops where **physical processes affect computations and vice versa**...The technology builds on the older (but still very young) discipline of embedded systems, computers and software embedded in devices whose principle mission is not computation, such as cars, toys, medical devices, and scientific instruments."

Microsoft® Academic Search Beta

Advanced Search

- Authors »
- Publications »
- Conferences »
- Journals »
- Organizations »
- Keywords »

Academic > Top conferences in Real-Time & Embedded Systems

1 - 27 of 27 results

Computer Science | Real-Time & Embedded Systems | All Years

Conferences	Publications	H-Index
RTSS - IEEE Real-Time Systems Symposium	1160	85
CDC - Conference on Decision and Control	26379	83
SenSys - Conference On Embedded Networked Sensor Systems	635	69
Hybrid Systems	783	54
CHES - Cryptographic Hardware and Embedded Systems	428	53
RTAS - IEEE Real Time Technology and Applications Symposium	363	46
ECRTS - Euromicro Conference on Real-Time Systems	700	39
CASES - Compilers, Architecture, and Synthesis for Embedded Systems	407	34
EMSOFT - International Workshop on Embedded Systems	351	31
ISORC - Object-Oriented Real-Time Distributed Computing	770	29
FTRTFT - Formal Techniques in Real-Time and Fault-Tolerant Systems	218	27
RTCSA - Real-Time Computing Systems and Applications	815	26
LCTES - Languages, Compilers, and Tools for Embedded Systems	203	20

CDC 2012 - Papers on "Networks"

Communication networks	MoC02.3, ThA01.1, ThA01.2, ThA01.3, ThA01.4, ThA01.5, ThA01.6, ThA01.7, ThC01.4, ThC04.6, ThC08.3, ThC08.4, TuA01.6, TuA04.1, TuB01.2, TuB01.6, TuC02.3, TuC03.1, TuC04.2, WeA04.6, WeA17.4, WeB14.4
Network analysis and control	MoA07.1, MoA07.6, MoB01.6, MoB11.1, ThA01.3, ThA01.4, ThA01.5, ThA01.7, ThA03.4, ThA06.2, ThA07.3, ThA13.7, ThB03.3, ThB07.3, ThB14.2, ThB17.1, ThC01.3, ThC04.6, ThC07.1, ThC16.3, TuA01.2, TuA01.3, TuA01.4, TuA03.2, TuA07.3, TuA10.3, TuA15.5, TuB01.2, TuB01.6, TuB02.1, TuB03.2, TuC01.2, TuC02.2, TuC02.5, TuC09.3, WeA01.1, WeA01.3, WeA01.4, WeA01.5, WeA01.6, WeA01.7, WeA07.4, WeA14.4, WeA16.7, WeB01.1, WeB01.4, WeB01.5, WeB01.6, WeB07.3, WeB09.5, WeB11.4, WeC01.4, WeC01.6
Networked control systems	MoA01.1, MoA01.2, MoA01.3, MoA01.4, MoA01.5, MoA01.6, MoA01.7, MoA02.2, MoA02.3, MoA02.4, MoA02.7, MoA03.1, MoA03.6, MoA04.4, MoA07.1, MoA07.3, MoA07.7, MoA11.2, MoB01.1, MoB01.2, MoB01.3, MoB01.4, MoB01.5, MoB01.6, MoB02.5, MoB02.6, MoB03.3, MoB05.3, MoB11.4, MoB17.4, MoC01.1, MoC01.2, MoC01.3, MoC01.4, MoC01.5, MoC01.6, MoC02.6, ThA01.1, ThA01.5, ThA03.6, ThA05.7, ThA06.5, ThA16.1, ThB01.1, ThB01.2, ThB01.3, ThB01.4, ThB01.5, ThB03.2, ThB03.5, ThB03.6, ThB09.2, ThB10.6, ThC01.1, ThC01.2, ThC01.3, ThC01.4, ThC01.5, ThC01.6, ThC07.2, ThC07.3, ThC08.1, ThC10.6, TuA01.1, TuA01.2, TuA01.3, TuA01.4, TuA01.5, TuA01.6, TuA01.7, TuA02.2, TuA03.2, TuA06.4, TuA06.6, TuA10.4, TuA12.6, TuA17.4, TuB01.5, TuB02.3, TuB02.6, TuB17.1, TuC01.1, TuC01.2, TuC01.3, TuC01.4, TuC01.5, TuC02.3, TuC02.4, TuC04.3, TuC11.1, TuC17.5, WeA01.2, WeA03.5, WeA04.6, WeA06.2, WeA07.3, WeB01.2, WeB01.5, WeB03.1, WeB03.3, WeB07.2, WeB07.4, WeB07.6, WeB08.5, WeC03.3, WeC03.4, WeC14.2, WeC16.4
Sensor networks	MoA01.4, MoA02.1, MoA02.3, MoB02.1, MoB02.2, MoB02.3, MoB02.4, MoB02.5, MoB02.6, MoB03.1, MoB03.2, MoB13.2, MoC02.1, MoC02.2, MoC02.3, MoC02.4, MoC02.5, MoC02.6, ThC08.2, ThC08.3, TuA02.1, TuA02.6, TuB02.1, TuB02.2, TuB02.3, TuB02.4, TuB02.5, TuB02.6, TuB04.5, TuC02.1, TuC02.2, TuC02.3, TuC02.4, TuC02.5, TuC02.6, TuC04.2, TuC04.3, TuC06.4, TuC14.1, TuC14.5, WeA03.1, WeA05.3, WeB07.1, WeB14.5, WeC01.2, WeC01.4, WeC07.4, WeC07.5
Transportation networks	MoA07.2, MoA07.5, MoA12.1, MoA12.6, ThB12.6, ThB16.1, ThB16.2, ThC07.1, WeC14.1

CDC 2012 - Papers on "Computation"

Computational methods	MoA10.3, MoA15.1, MoA15.2, MoA15.3, MoA15.4, MoA15.5, MoA15.6, MoA15.7, MoA16.5, MoA17.7, MoB07.3, MoB15.5, MoC05.4, MoC15.2, MoC16.3, ThA05.1, ThA12.3, ThA14.6, ThA17.7, ThB06.4, ThC06.4, ThC13.2, ThC14.1, ThC14.5, TuA02.5, TuA06.5, TuB09.1, TuB09.5, TuC05.3, TuC05.6, TuC06.2, TuC15.3, WeA01.6, WeA04.4, WeA10.7, WeA13.3, WeA17.2, WeB10.3, WeC09.2, WeC10.2, WeC14.4, WeC17.5
Computer networks	MoA10.6, MoA10.7, TuC14.1, WeA17.4, WeB14.2, WeC01.1, WeC01.4, WeC01.5

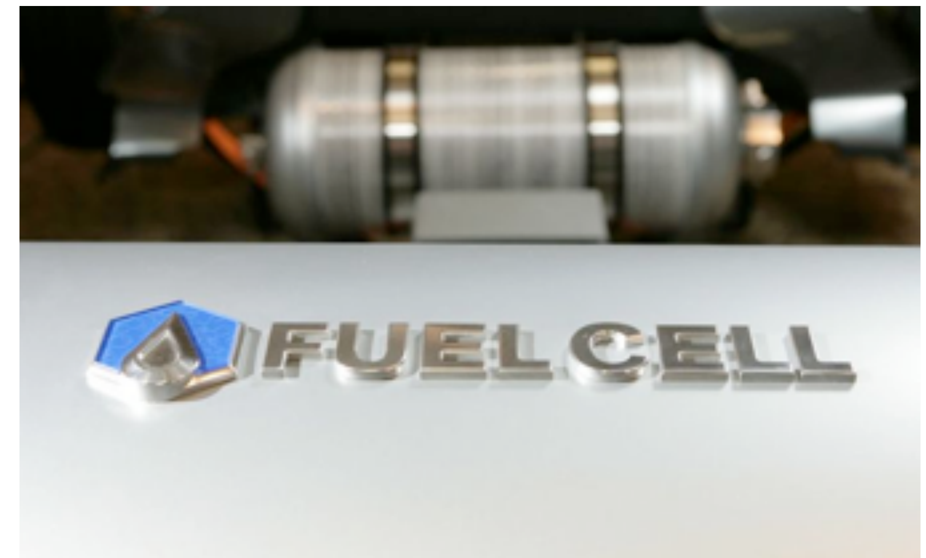
CDC 2012 - Papers on “Embedded Systems”

CDC 2012 - Papers on “Real-time Systems”

Embedded Optimization (Optimal Control / Estimation / DSP) for Cyber-Physical Systems

Applications for Embedded Optimization (Optimal Control / Estimation / DSP)









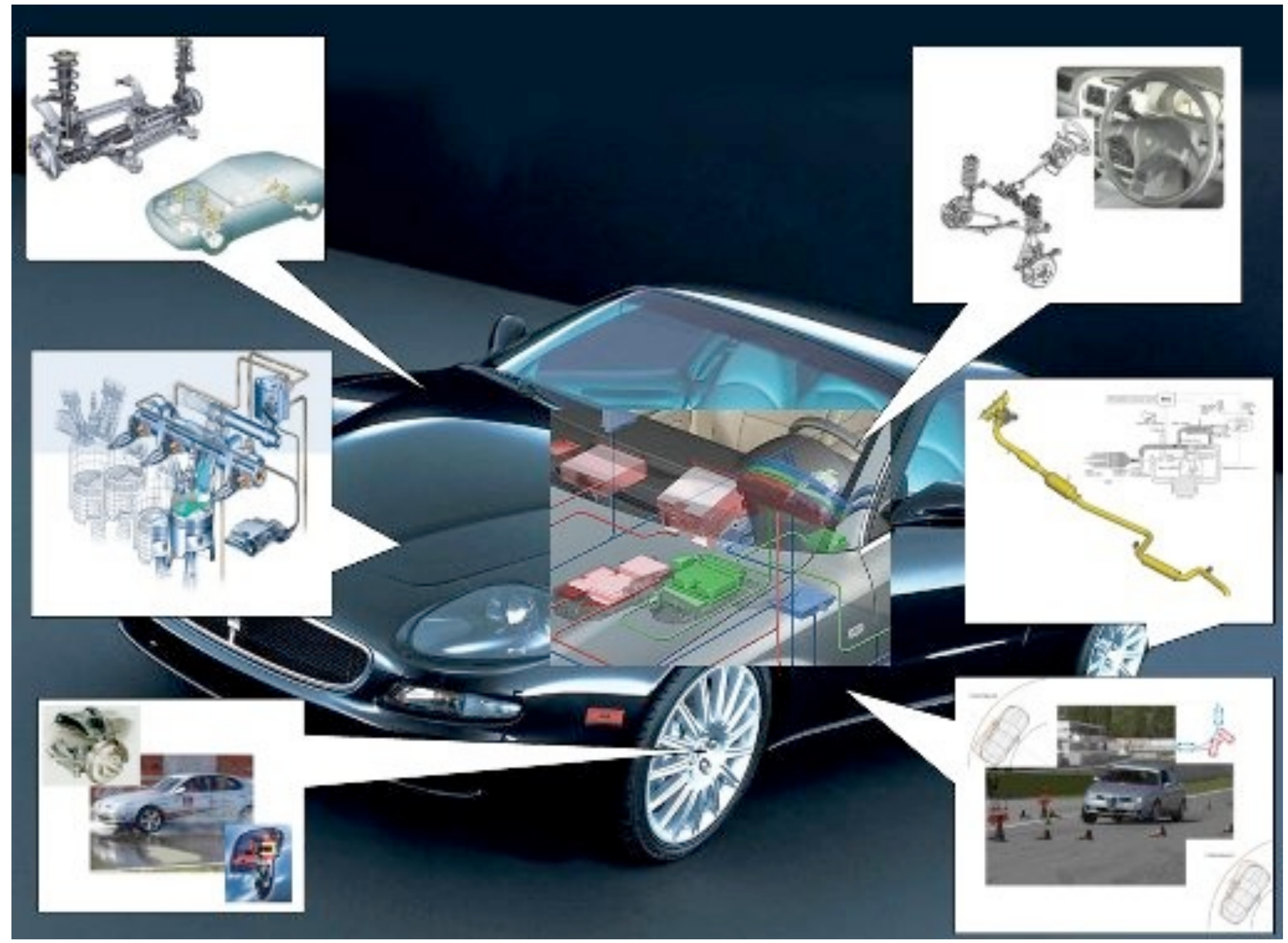
Computing and Cyber-Physical Systems

"All too often, today's students use **laptop** [or **desktop**] computers to perform their computing, which shields them from dealing with any of the physical **constraints** they will face in the **real world**. This approach is akin to **trying to learn skiing while standing comfortably in the après ski lounge**."

Wolf, Cyber-Physical Systems, *Computer*, 2009.

Challenges for Cyber-Physical Systems

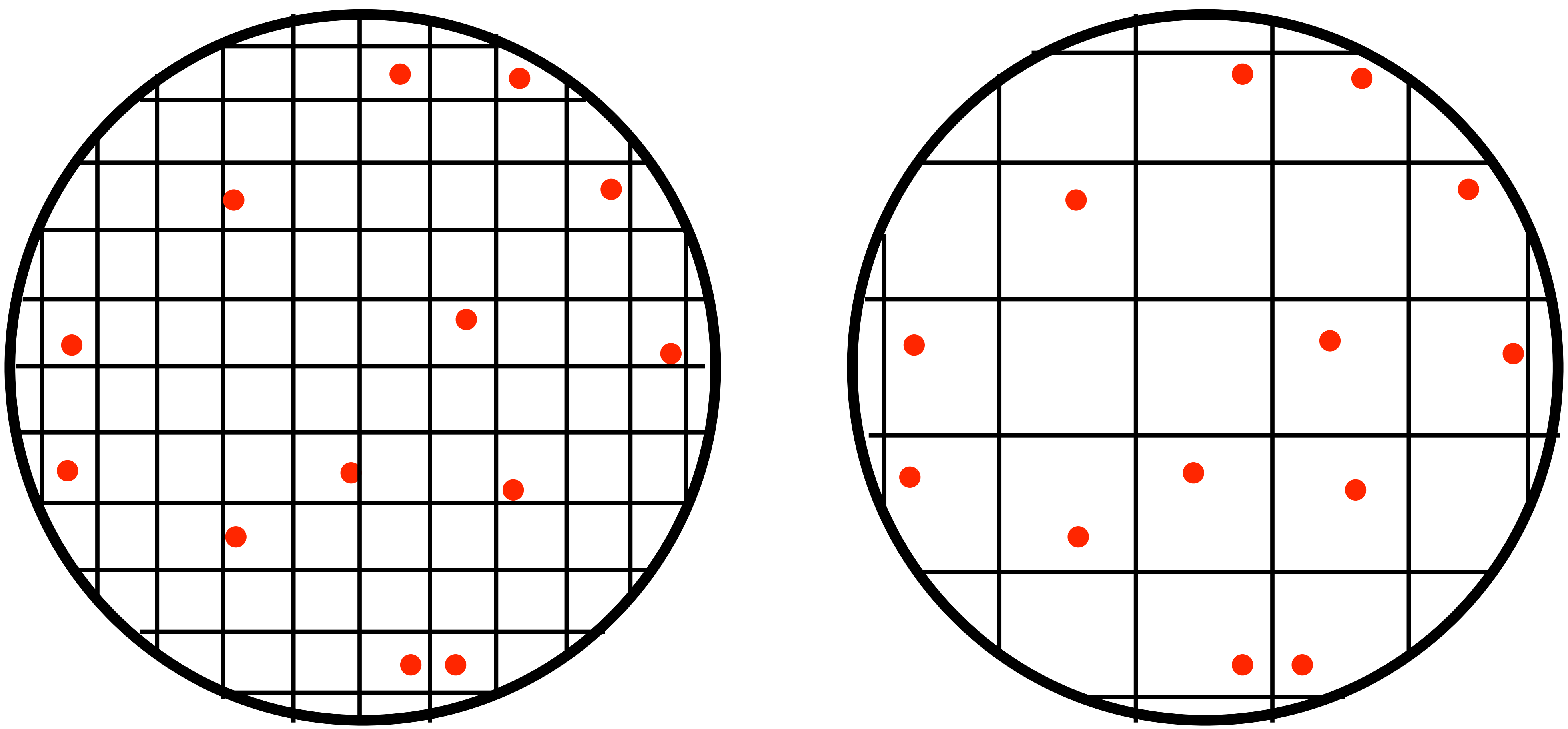
- Cost
- Energy
- Speed
- Reliability



- Predictability / real-time (fast is *not* equal to real-time)

Number representation (e.g. fixed/floating-point, #bits) has a **major** impact on the design

Size is Very Important in Microprocessor Design



Die area = 1
Working = 64

Die area = 4
Working = 4

Cost per die = $f(area^{\boldsymbol{x}})$, $\boldsymbol{x} \in [2,4]$

Computational Resources for an Adder

Xilinx Virtex-7 XT 1140 FPGA:

Number representation	Registers/Flip-Flops (FFs)	Latency/delay (clock cycles)
double floating-point 52-bit mantissa	1046	14
single floating-point 23-bit mantissa	557	11
fixed-point 53 bits	53	1
fixed-point 24 bits	24	1

Cheap and **low power** processors often only have **fixed-point**

Dynamic Optimization

$$\min_{x(\cdot),u(\cdot),p} J(y(\cdot),x(\cdot),u(\cdot),p)$$

$$F(y(t),\dot{x}(t),x(t),u(t),p,t) = 0, \quad \forall t \in [t_0, t_f)$$
$$G(y(t),\dot{x}(t),x(t),u(t),p,t) \leq 0, \quad \forall t \in [t_0, t_f)$$

Discretized and approximated by finite-dimensional NLP:

$$\min_{\theta} V(\theta)$$

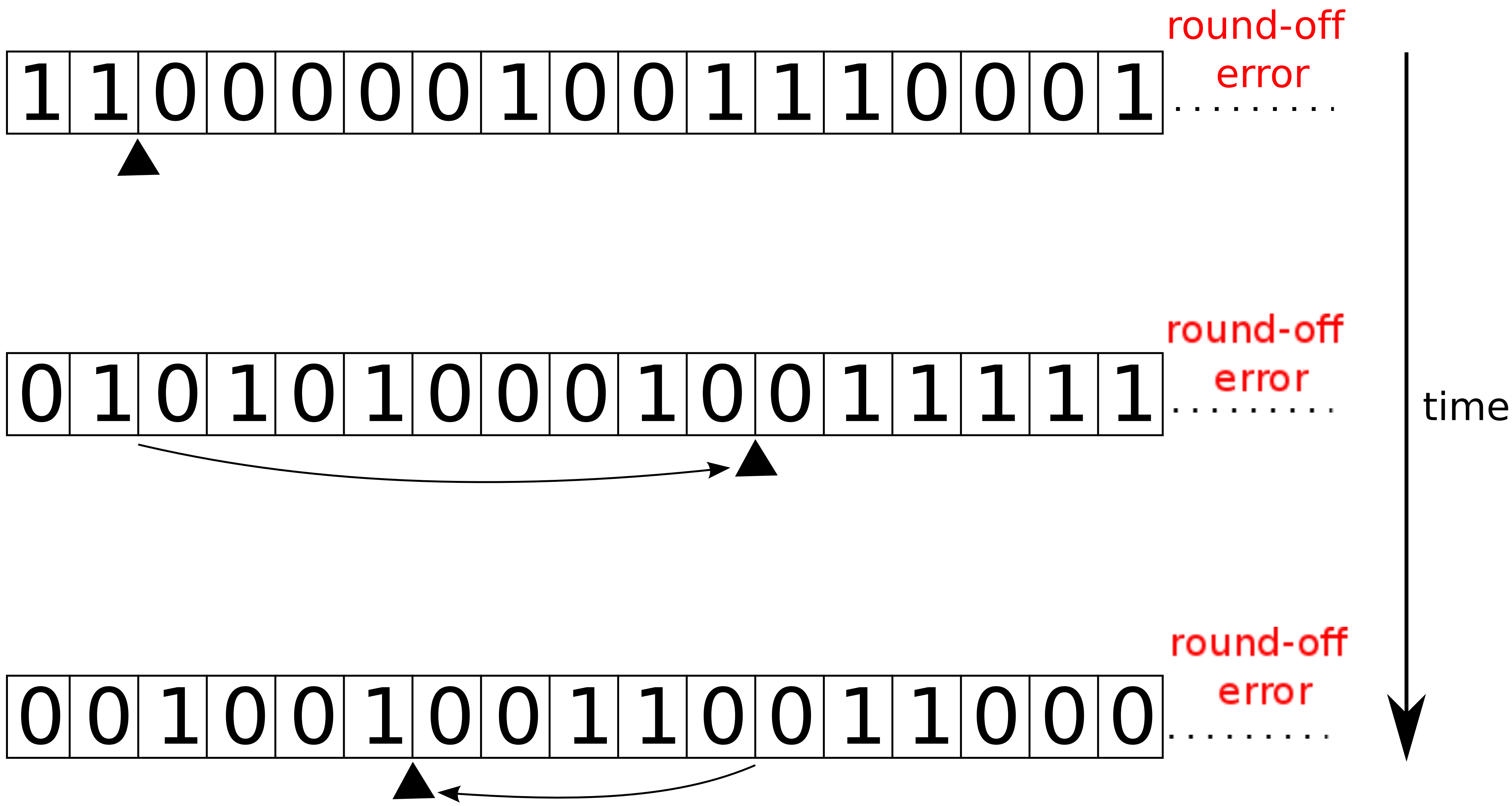
$$f(\theta) = 0$$

$$g(\theta) \leq 0$$

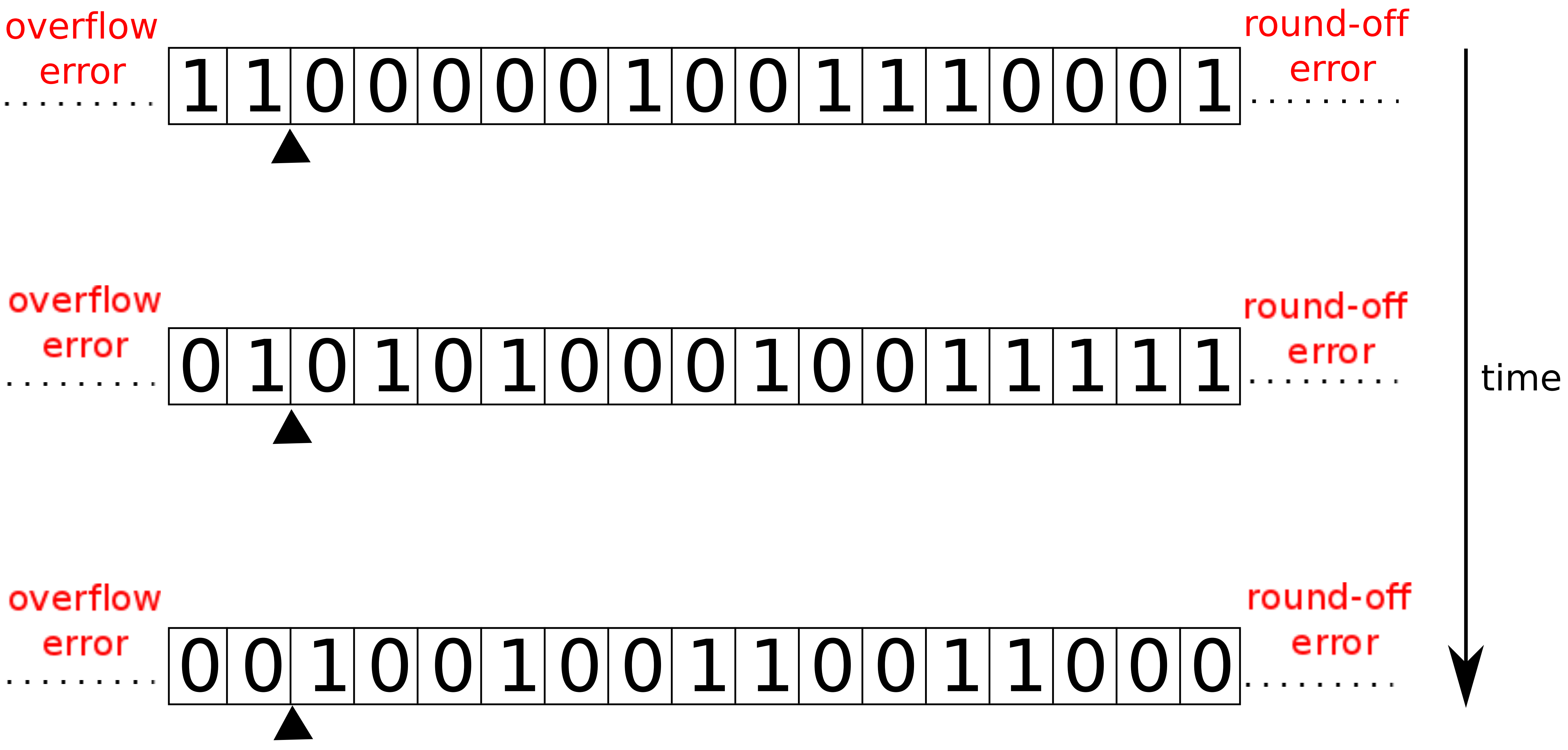
$$\theta \in \mathbb{R}^n,\; f : \mathbb{R}^n \to \mathbb{R}^m,\; g : \mathbb{R}^n \to \mathbb{R}^p$$

Fixed-Point Arithmetic

Floating-Point Arithmetic



Fixed-Point Arithmetic



Challenges for Fixed-Point Arithmetic

- Number of bits for **integer** and fractional part?
    - Determine **worst-case peak** values
    - Optimization algorithms are **nonlinear** and **recursive**
- **Search direction** most computationally critical part:

$$A\xi = b$$

- **Iterative linear solvers** preferred: CG, MINRES, GMRES

Lanczos Algorithm (Kernel of CG/MINRES)

$$\mathsf{Q}_i^T \mathsf{A} \mathsf{Q}_i = \mathsf{T}_i := \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{i-1} \\ 0 & & \beta_{i-1} & \alpha_i \end{bmatrix}$$

Given q_1 such that $\|q_1\|_2 = 1$ and an initial value $\beta_0 := 1$

for $i = 1$ to i_{max} **do**

1. $q_i \leftarrow \frac{q_i}{\beta_{i-1}}$
2. $z_i \leftarrow \mathsf{A} q_i$
3. $\alpha_i \leftarrow q_i^T z_i$
4. $q_{i+1} \leftarrow z_i - \alpha q_i - \beta_{i-1} q_{i-1}$
5. $\beta_i \leftarrow \|q_{i+1}\|_2$

end for

Lanczos Algorithm (Kernel of CG/MINRES)

$$\mathsf{Q}_i^T \mathsf{A} \mathsf{Q}_i = \mathsf{T}_i := \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{i-1} \\ 0 & & \beta_{i-1} & \alpha_i \end{bmatrix}$$

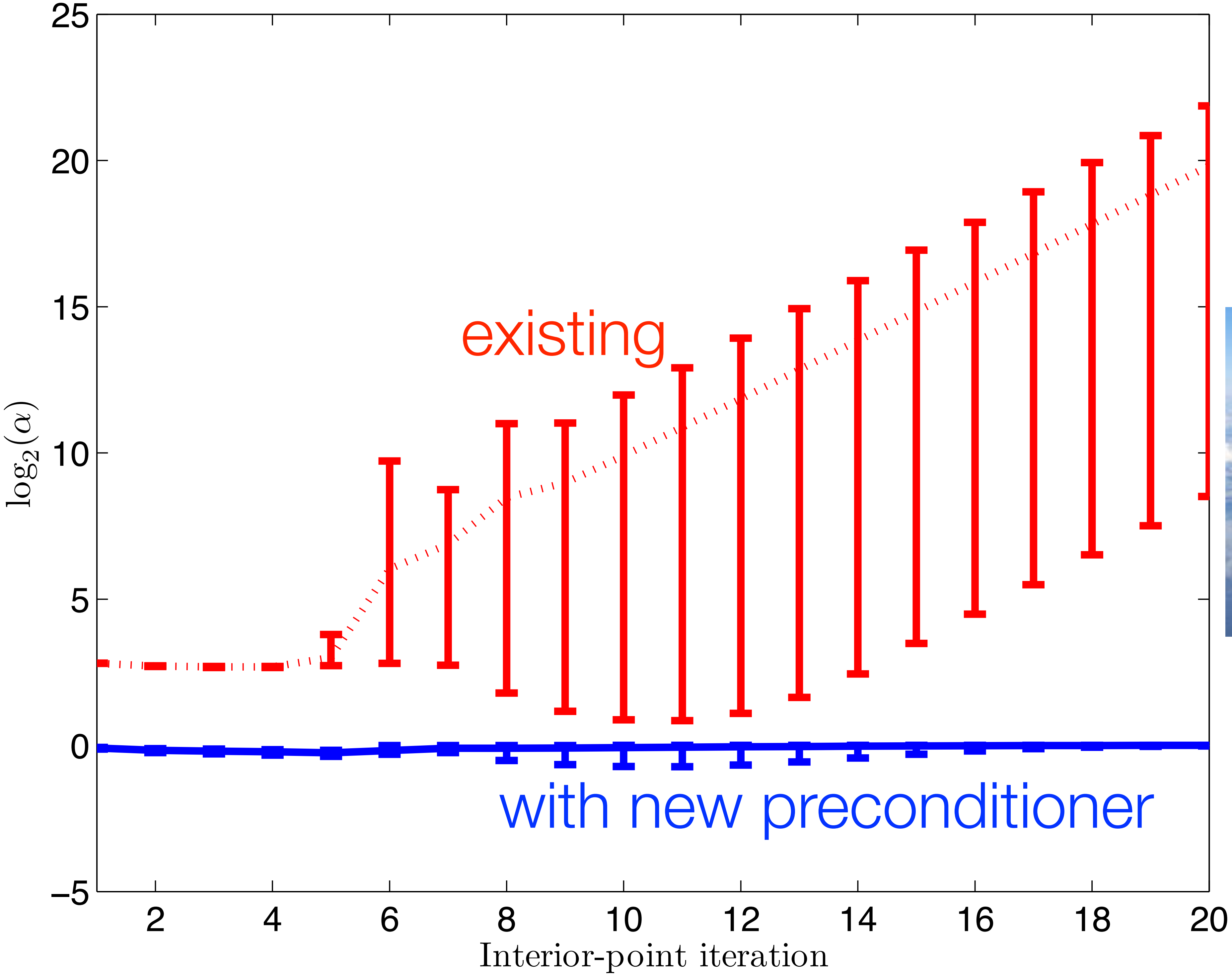
Given q_1 such that $\|q_1\|_2 = 1$ and an initial value $\beta_0 := 1$

for $i = 1$ to i_{max} **do**

1. $q_i \leftarrow \frac{q_i}{\beta_{i-1}}$
2. $z_i \leftarrow \mathsf{A} q_i$
3. $\alpha_i \leftarrow q_i^T z_i$
4. $q_{i+1} \leftarrow z_i - \alpha q_i - \beta_{i-1} q_{i-1}$
5. $\beta_i \leftarrow \|q_{i+1}\|_2$

end for

Evolution of Variables in Primal-dual Interior Point



Optimal control of a Boeing 747

On-line Diagonal Preconditioner / Scaler

$$\mathsf{A}\xi = \mathsf{b}, \quad \mathsf{A} = \mathsf{A}'$$

$$\mathsf{S}_{kk} := \sum_{j=1}^{\mathsf{N}} |\mathsf{A}_{kj}| \quad \text{(1-norm of row } k\text{)}$$

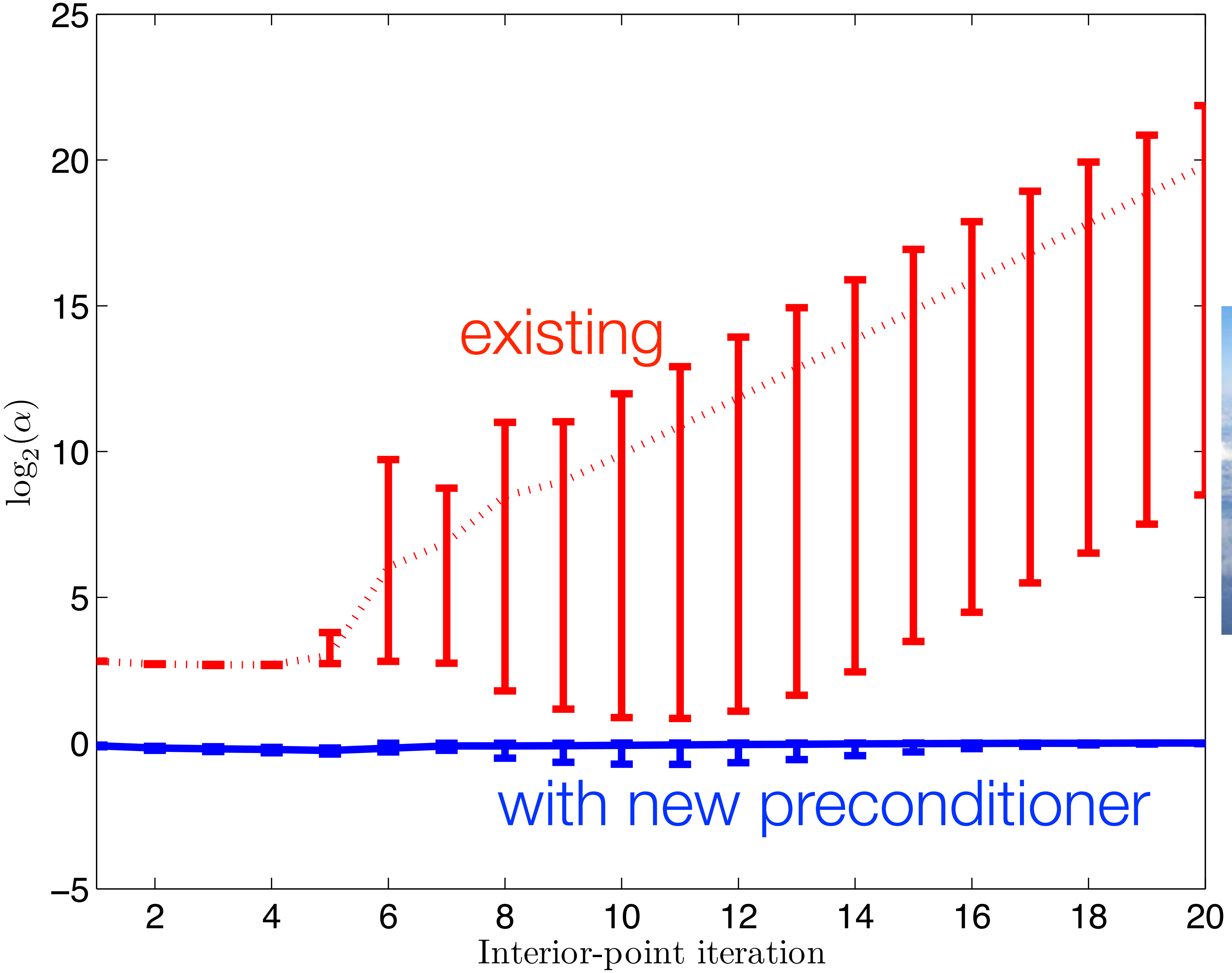
$$\mathsf{S}^{-\frac{1}{2}}\mathsf{A}\mathsf{S}^{-\frac{1}{2}}\psi = \mathsf{S}^{-\frac{1}{2}}\mathsf{b} \Leftrightarrow \widehat{\mathsf{A}}\psi = \widehat{\mathsf{b}} \Rightarrow \rho\left(\widehat{\mathsf{A}}\right) \leq 1$$

$$\xi = \mathsf{S}^{-\frac{1}{2}}\psi$$

Theorem (Avoiding overflow in fixed-point)
All variables in Lanczos algorithm are between -2 and 2

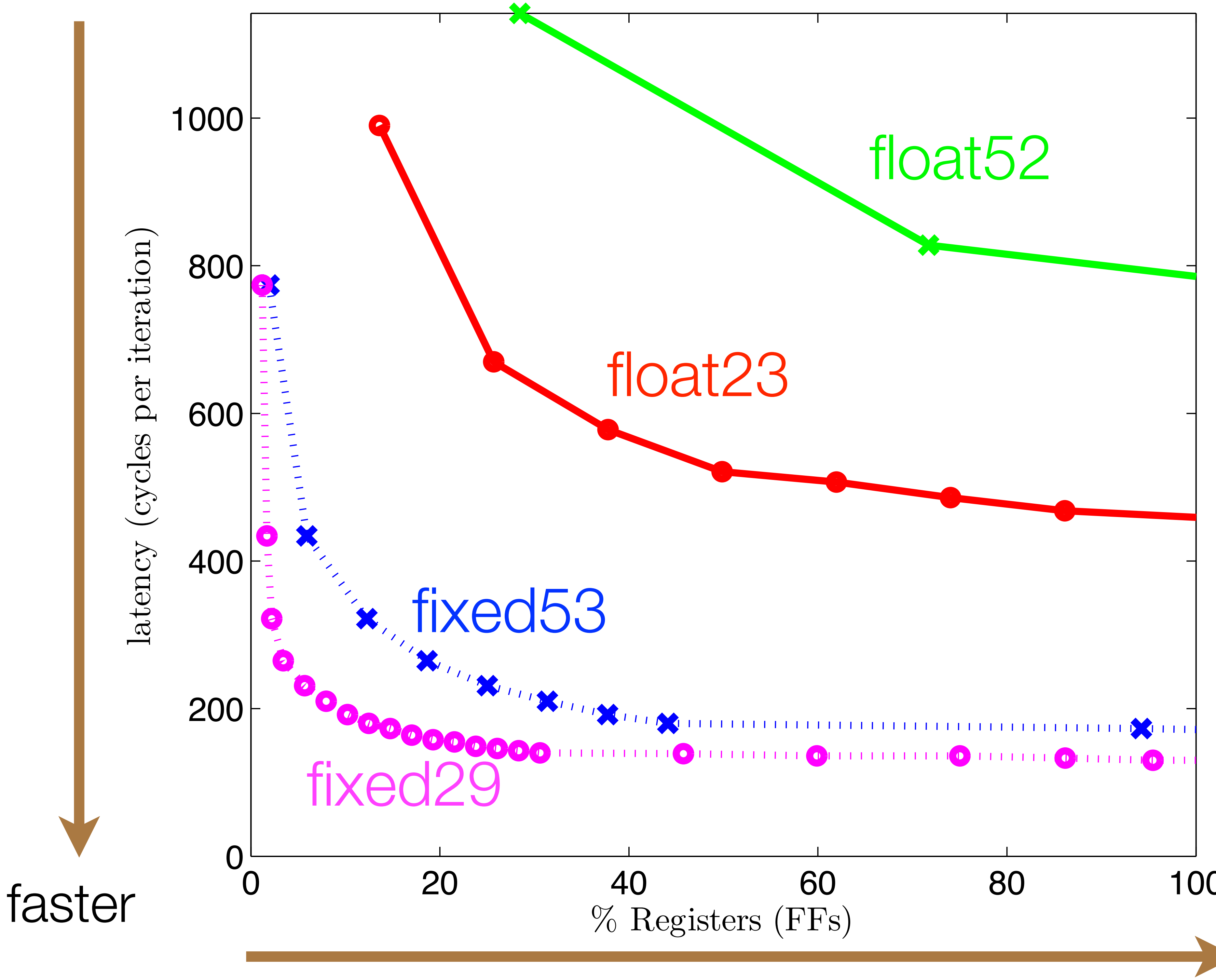
Proof: Proc. IEEE Conference on Decision and Control 2012

Evolution of Variables in Primal-dual Interior Point



Optimal control of a Boeing 747

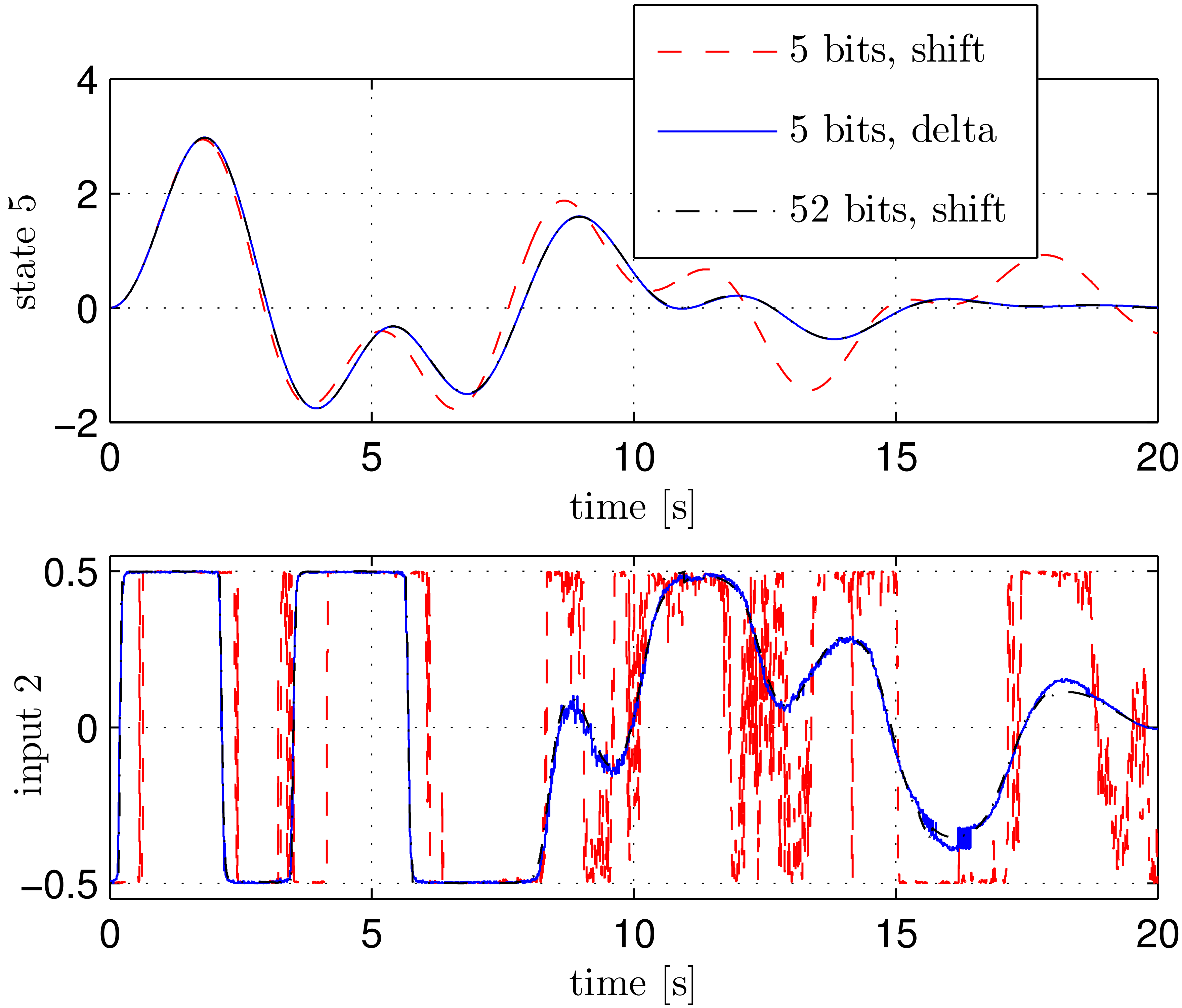
Trade-offs on an FPGA (same accuracy)



Xilinx Virtex-7 XT 1140 with matrices from the optimal control of a Boeing 747

Low-Precision Arithmetic

Optimal Control in Low Precision Arithmetic



Mass-spring system with 3 masses (6 states) and 2 inputs, sample period = 10ms

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh + h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$x_{k+1} = A_s x_k + B_s u_k$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \dots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh + h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$\boxed{x_{k+1} = A_s x_k + B_s u_k}$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \dots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Shift Form

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

Sample period *h* and piecewise constant input (ZOH):

$$u(t) = u(kh) =: u_k, \quad \forall t \in [kh, kh + h)$$

Exact solution/discrete-time model to compute $x_k := x(kh)$

$$\boxed{(x_{k+1} - x_k)/h = (A_s x_k + B_s u_k - x_k)/h}$$

$$A_s := e^{A_c h} = I + A_c h + \frac{(A_c h)^2}{2!} + \frac{(A_c h)^3}{3!} + \dots$$

$$\lim_{\|A_c h\| \to 0} A_s = I, \qquad \lim_{\|A_c h\| \to 0} B_s = 0$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$(x_{k+1} - x_k)/h = (A_s x_k + B_s u_k - x_k)/h$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = \frac{(A_s - I)}{h} x_k + \frac{B_s}{h} u_k$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k$$

Sampled-data Representation in Delta Form

Middleton and Goodwin (IEEE TAC, 1986):

$$\frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k$$

$$A_\delta = A_c + \frac{A_c^2 h}{2!} + \frac{A_c^3 h^2}{3!} + \dots$$

$$\lim_{\|A_c h\| \to 0} A_\delta = A_c, \qquad \lim_{\|A_c h\| \to 0} B_\delta = B_c$$

Equivalent to shift form in **infinite** precision arithmetic
Different from shift form in **finite** precision arithmetic

Optimization Problem Using Shift Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & x_1' & u_1' & x_2' & \cdots & u_{N-1}' & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, \\ x_{k+1} &= A_s x_k + B_s u_k, \quad \forall k \in \{0, 1, \ldots, N-1\} \\ J x_k + E u_k &\leq d, \quad \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

Optimization Problem Using Shift Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & x_1' & u_1' & x_2' & \cdots & u_{N-1}' & x_N' \end{bmatrix}'$$

subject to

$$x_0 = \hat{x},$$

$$x_{k+1} = A_s x_k + B_s u_k, \quad \forall k \in \{0, 1, \ldots, N-1\}$$

$$J x_k + E u_k \leq d, \quad \forall k \in \{0, 1, \ldots, N-1\}$$

$$\delta_k := \frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k$$

Optimization Problem Using Delta Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & \boxed{\delta_0'} & x_1' & u_1' & \boxed{\delta_1'} & x_2' & \cdots & u_{N-1}' & \boxed{\delta_{N-1}'} & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, \\ \delta_k &= A_\delta x_k + B_\delta u_k, && \forall k \in \{0, 1, \ldots, N-1\} \\ x_{k+1} &= x_k + h\delta_k, && \forall k \in \{0, 1, \ldots, N-1\} \\ Jx_k + Eu_k &\le d, && \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

$$\boxed{\delta_k := \frac{x_{k+1} - x_k}{h} = A_\delta x_k + B_\delta u_k}$$

Optimization Problem Using Delta Form

$$\min_{\theta} \sum_{k=0}^{N-1} \ell_k(x_k, u_k)$$

$$\theta := \begin{bmatrix} u_0' & \delta_0' & x_1' & u_1' & \delta_1' & x_2' & \cdots & u_{N-1}' & \delta_{N-1}' & x_N' \end{bmatrix}'$$

subject to

$$\begin{aligned} x_0 &= \hat{x}, \\ \delta_k &= A_\delta x_k + B_\delta u_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ x_{k+1} &= x_k + h\delta_k, & \forall k \in \{0, 1, \ldots, N-1\} \\ J x_k + E u_k &\leq d, & \forall k \in \{0, 1, \ldots, N-1\} \end{aligned}$$

Solving the Optimization Problem

- Solve linearized **KKT system** (Rao, Wright, Rawlings; JOTA, 1998):

$$\mathsf{A}\xi = \mathsf{b}$$

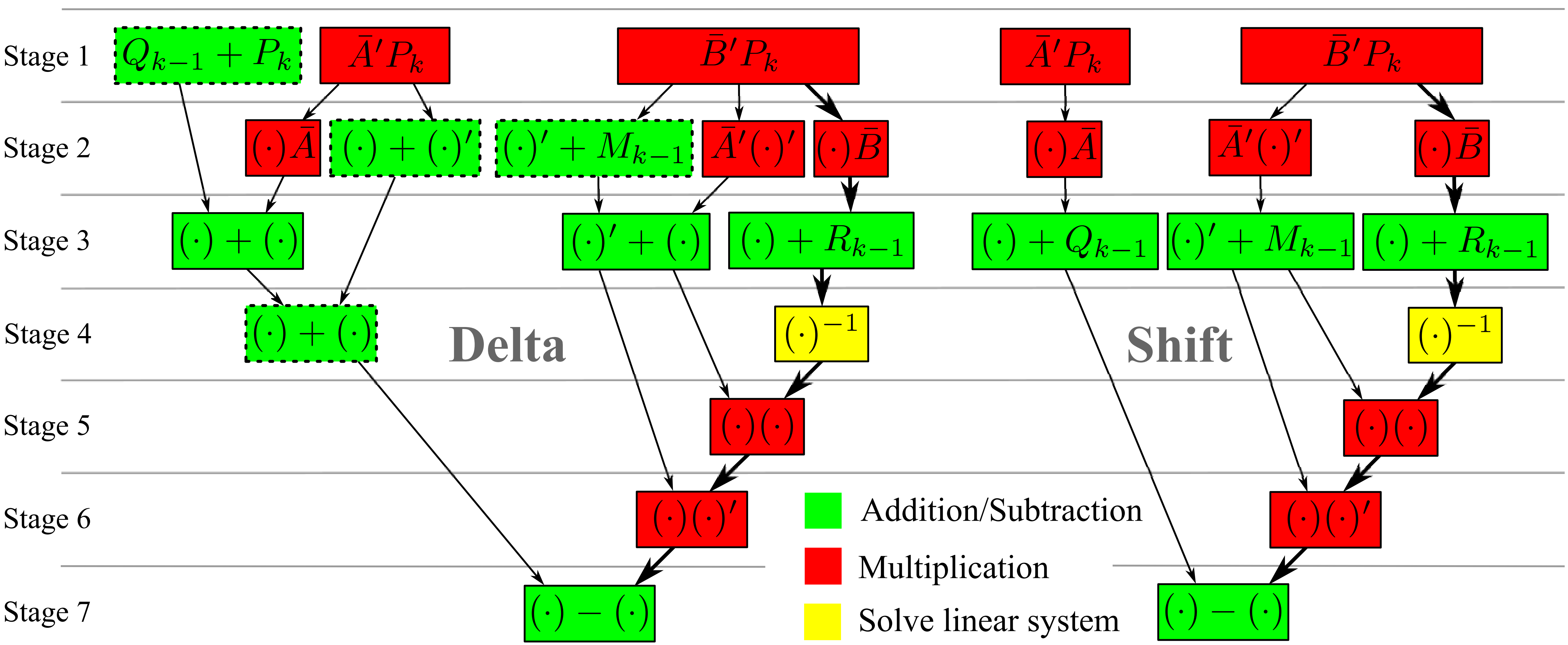
- **Interleave** search direction variables:

$$\xi := \begin{bmatrix} \Delta u_0' & \Delta \gamma_0' & \Delta \delta_0' & \Delta \lambda_1' & \Delta x_1' & \cdots & \Delta x_N' \end{bmatrix}'$$

- **Block elimination** results in Riccati recursions:

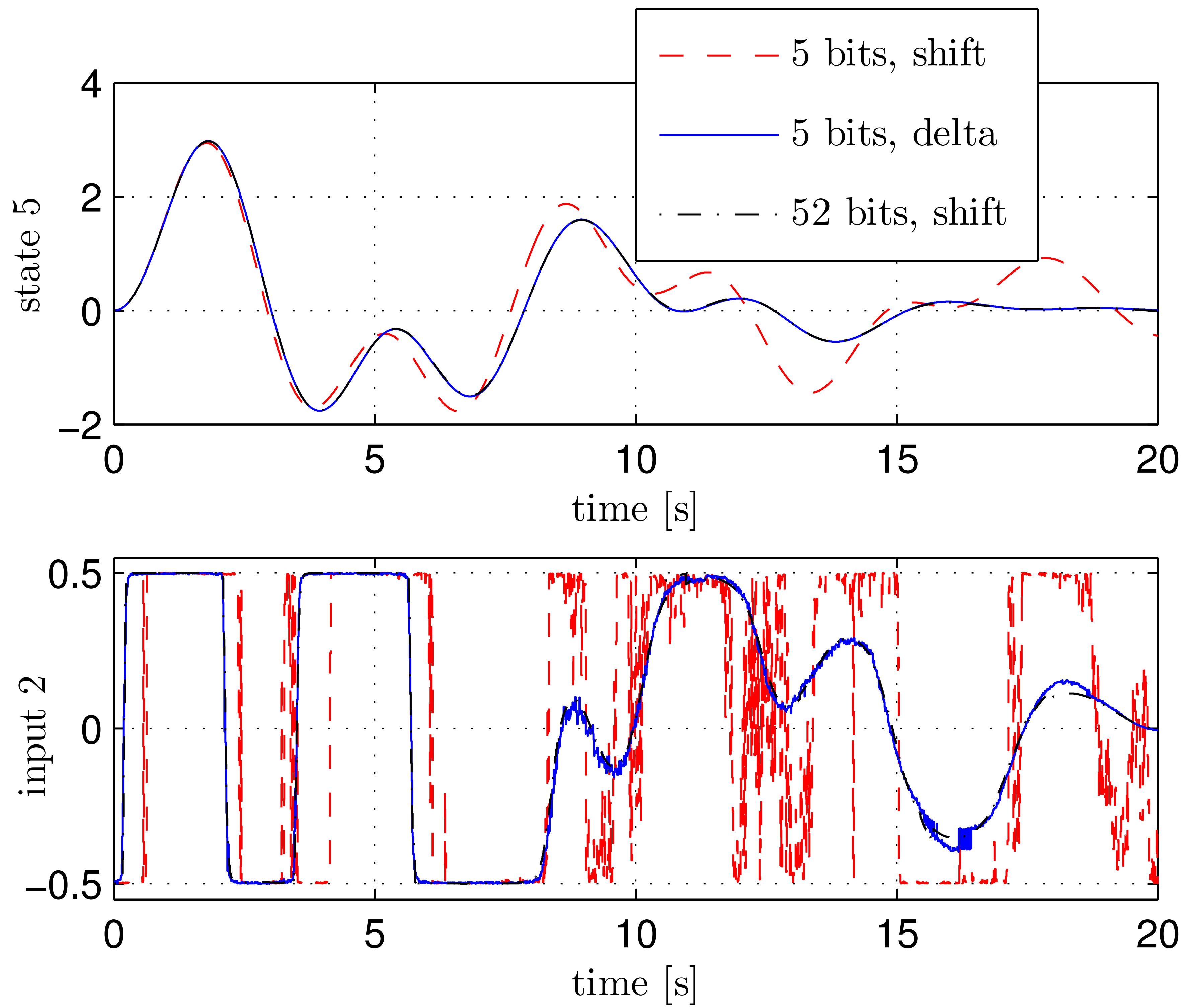
$$\begin{aligned} P_{k-1} := Q_{k-1} + P_k + h^2 A_\delta' P_k A_\delta + h A_\delta' P_k + h P_k A_\delta \\ -(M_{k-1} + h^2 A_\delta' P_k B_\delta + h P_k B_\delta)(R_{k-1} + h^2 B_\delta' P_k B_\delta)^{-1} \\ (M_{k-1}' + h^2 B_\delta' P_k A_\delta + h B_\delta' P_k) \end{aligned}$$

Data Dependencies in Riccati Recursion



Same amount of multipliers, adders and computational delay for a custom circuit, e.g. FPGA

Optimal Control in Low Precision Arithmetic



Mass-spring system with 3 masses (6 states) and 2 inputs, sample period = 10ms

Conclusions

- Number representation major factor that determines cost, energy, computational delay and accuracy
- Fixed-point: **Precondition** to get tight **analytical bounds** on variables in **Lanczos** algorithm to avoid **overflow**
- Low precision: **Sampled-data model** and **optimization method** crucial to successful implementation
- **Co-design** algorithm and hardware to use “just the right amount” of computational resources

Open Research Questions

- Other sampled-data and number representations?
- Nonlinear systems?
- Which algorithms map easily to low precision, fixed-point or other number representations?
- A priori guarantees on accuracy, closed-loop stability, robustness and performance?
- Need control + optimization + numerics + computing