Model Predictive Control Theory and Applications

Francesco Borrelli

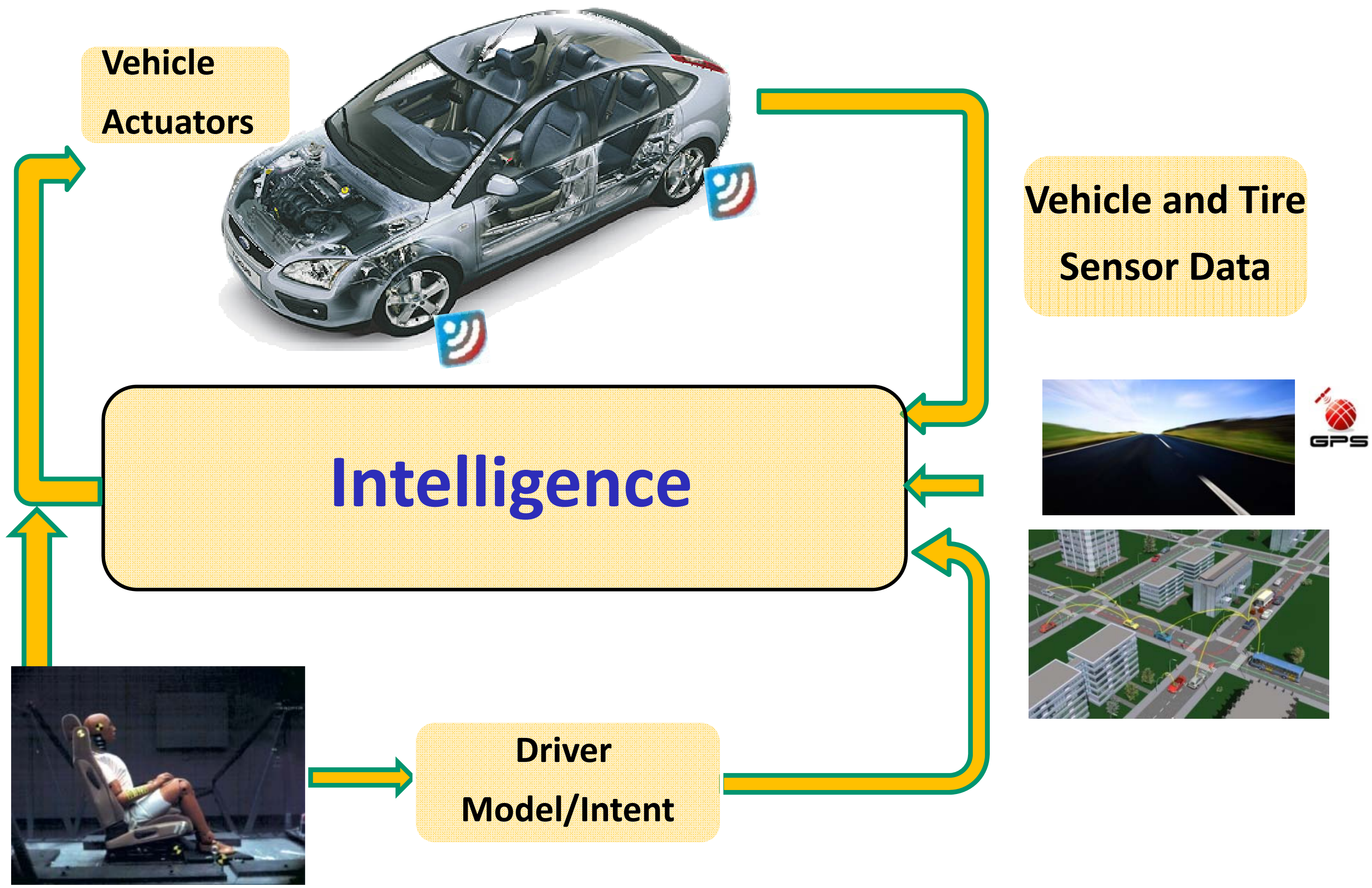
Department of Mechanical Engineering

University of California

Berkeley, USA

www.mpc.berkeley.edu

Automotive Cyber-Physical System

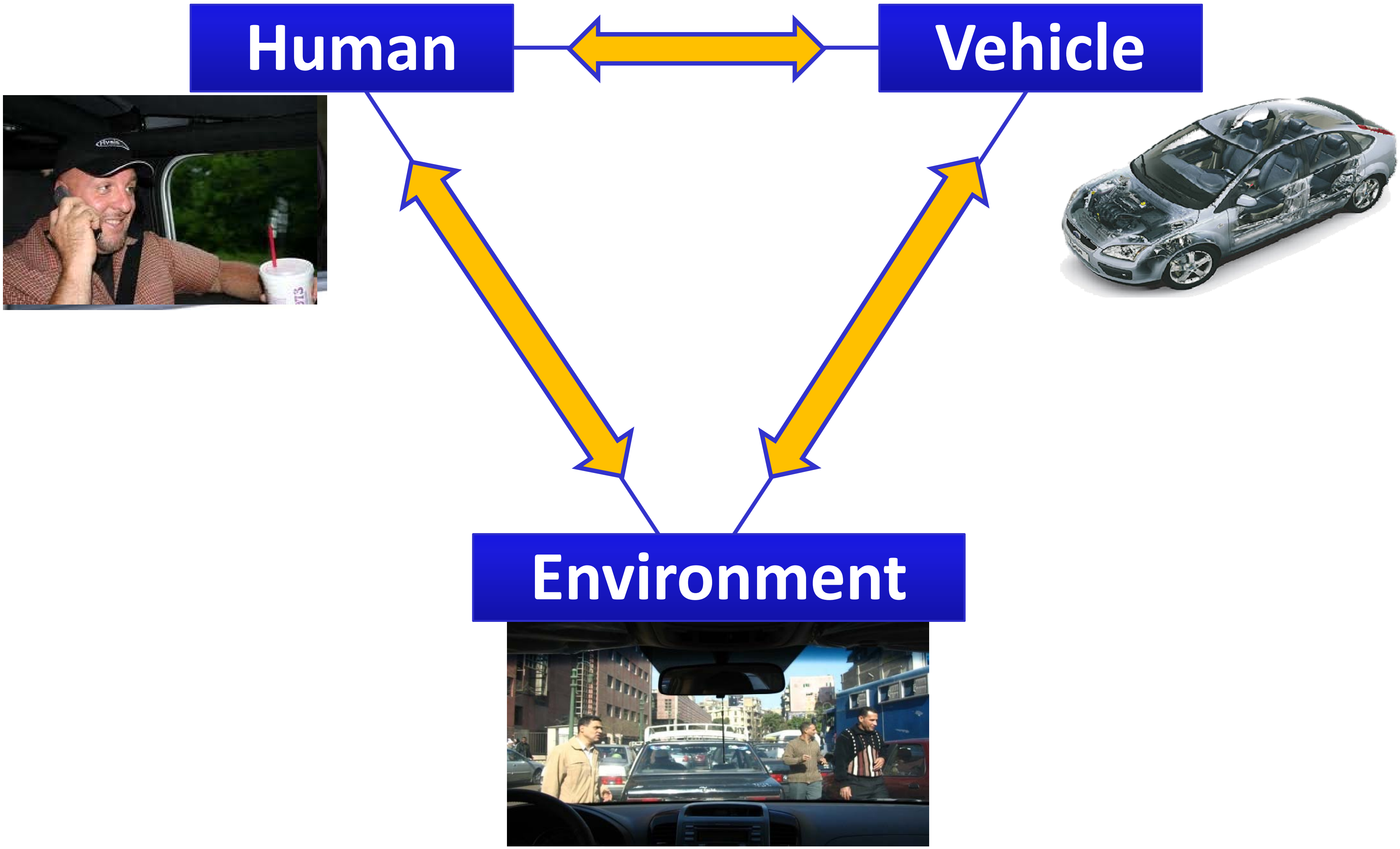


Safety | **Comfort** | **Efficiency**

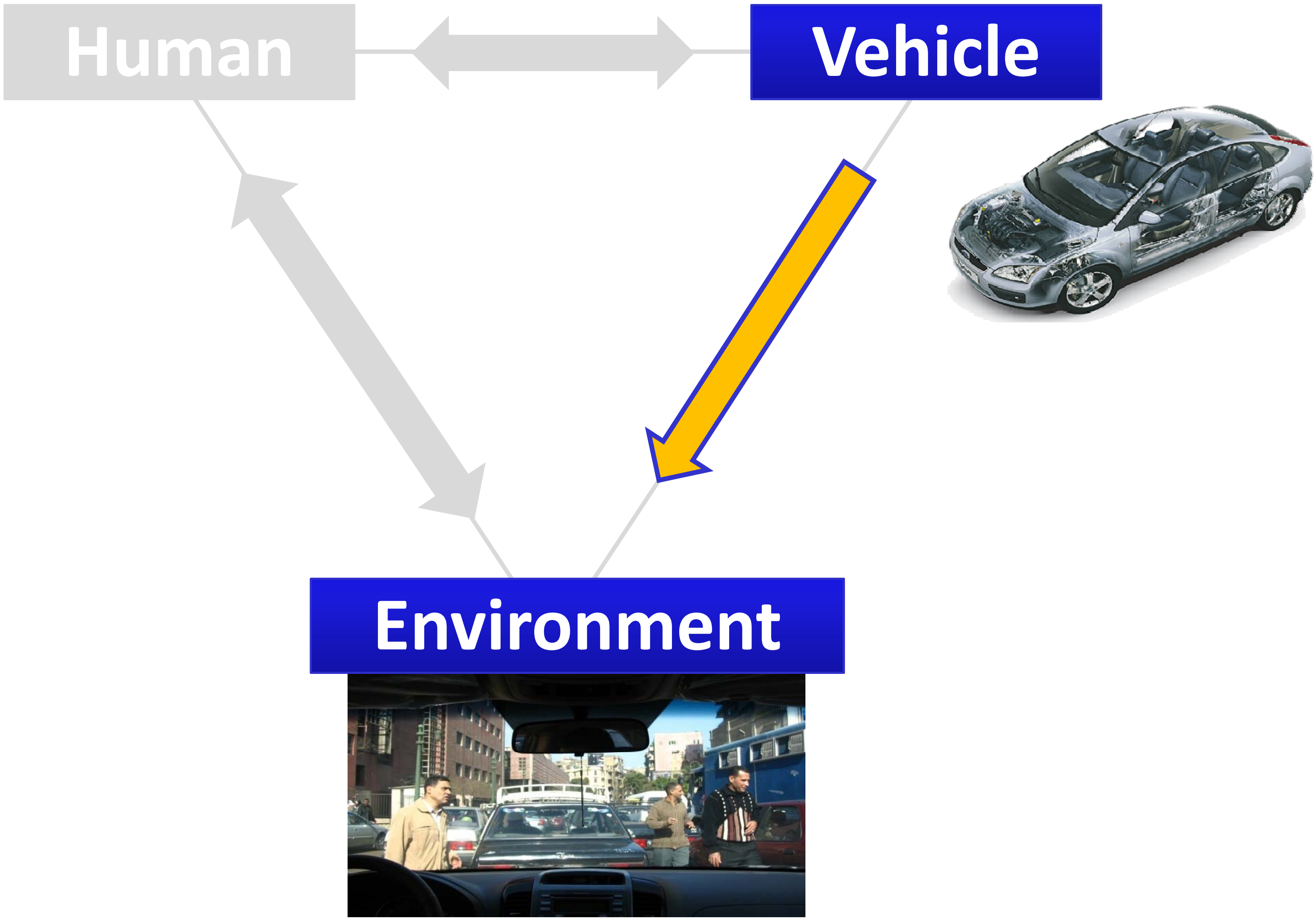
Lane Departure A14 Highway – June 2009

www.mpc.berkeley.edu

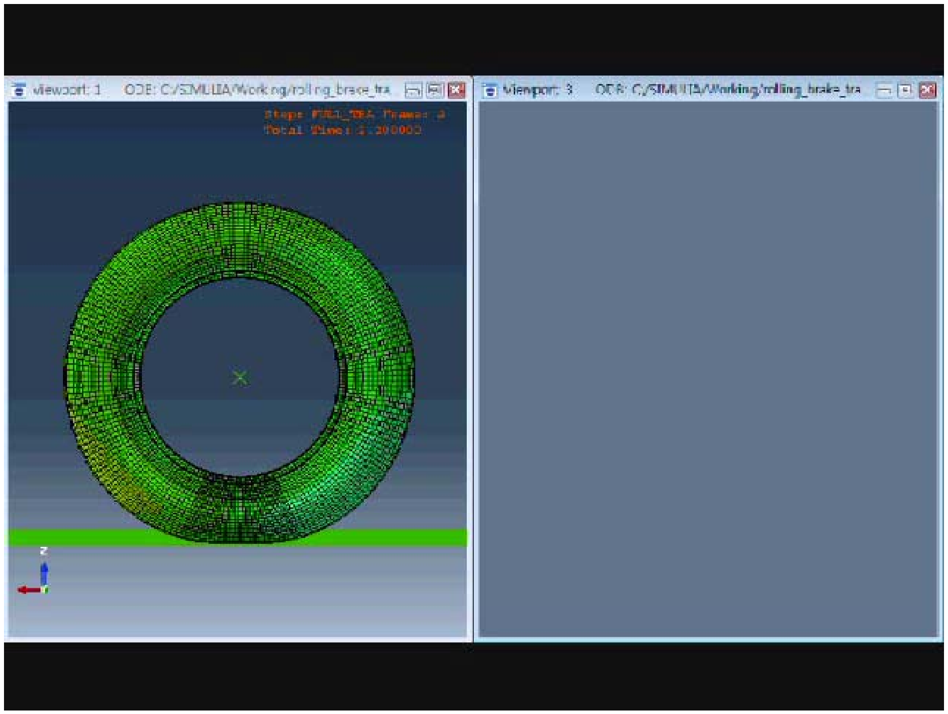
CPS-Synoptic Scheme

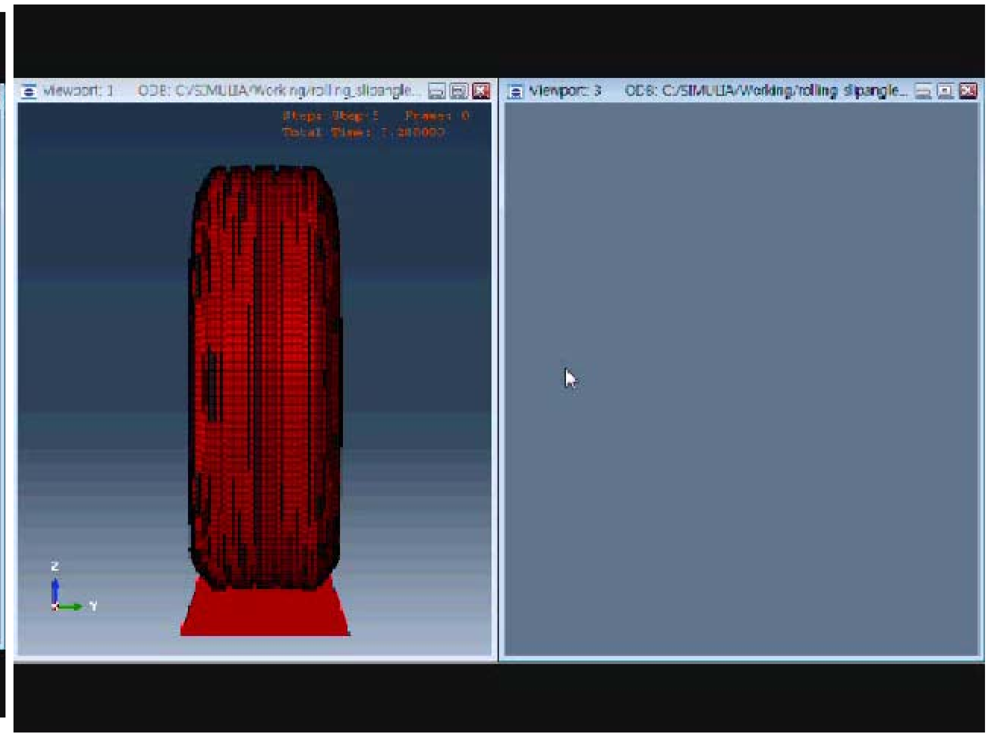


CPS-Synoptic Scheme



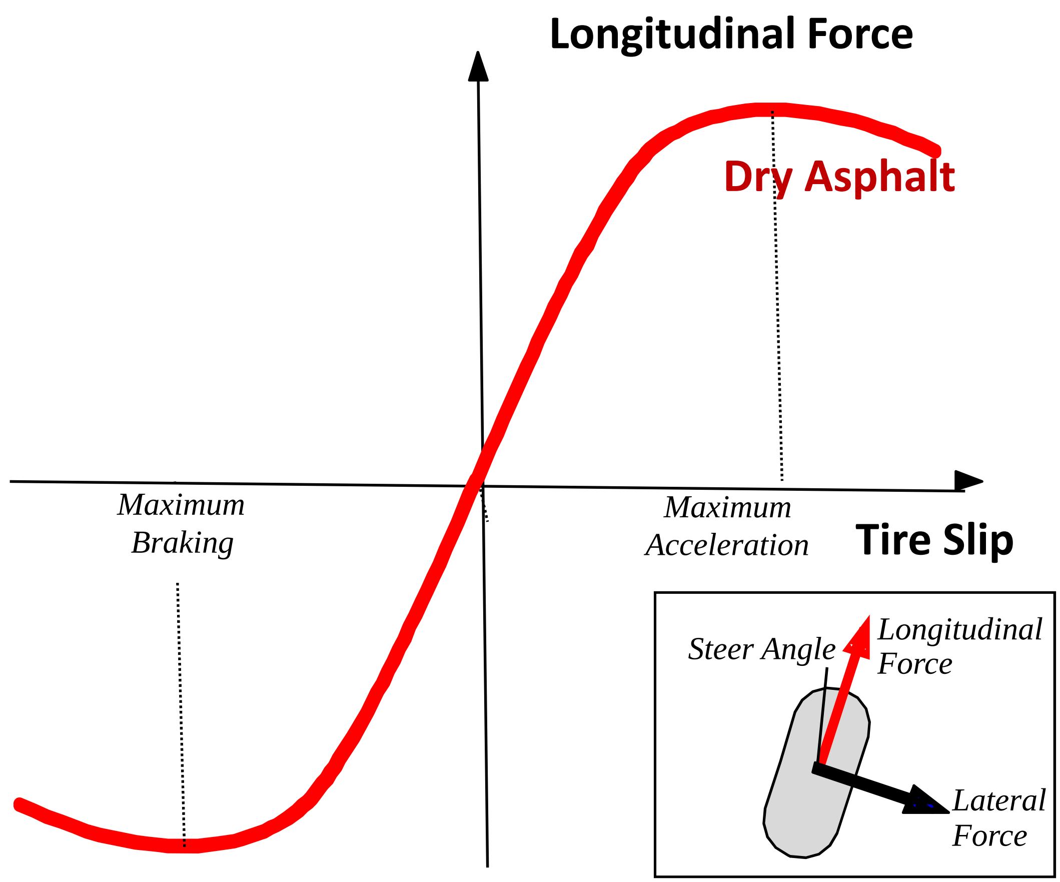
Vehicle-Road Interaction
FEM Simulation



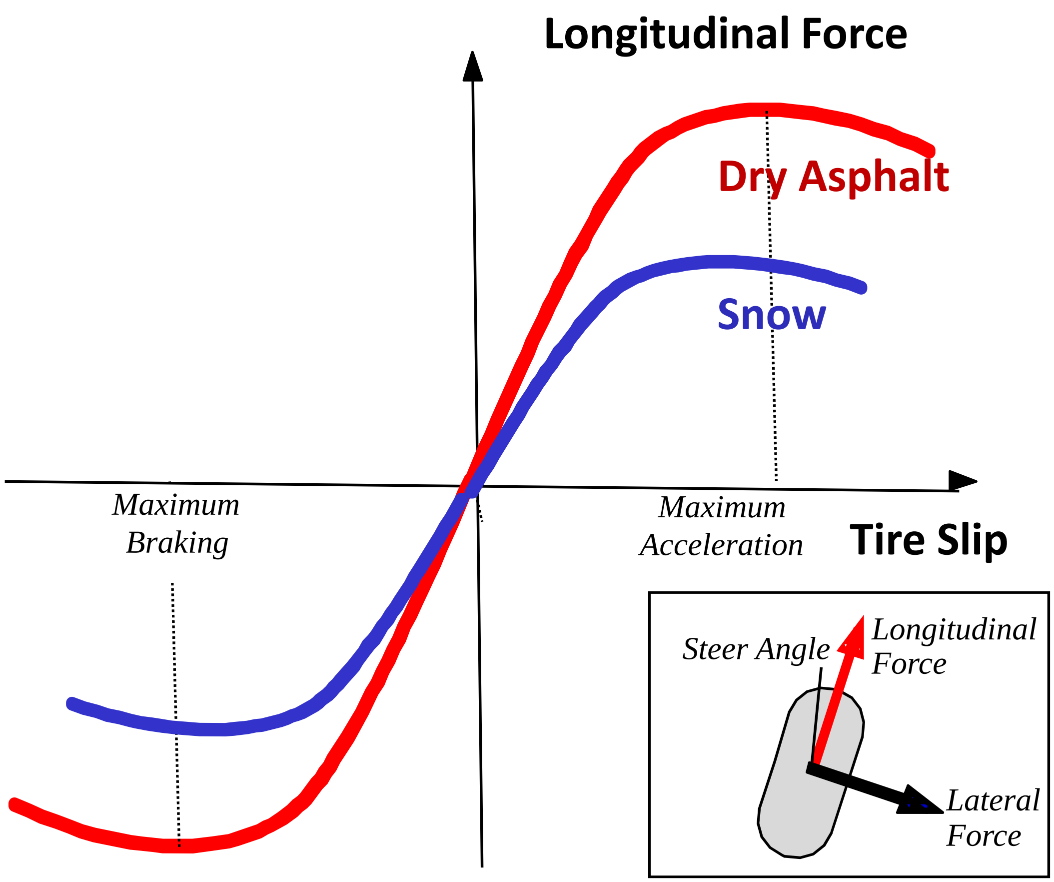


www.mpc.berkeley.edu

Vehicle-Road Interaction
Simplified Models

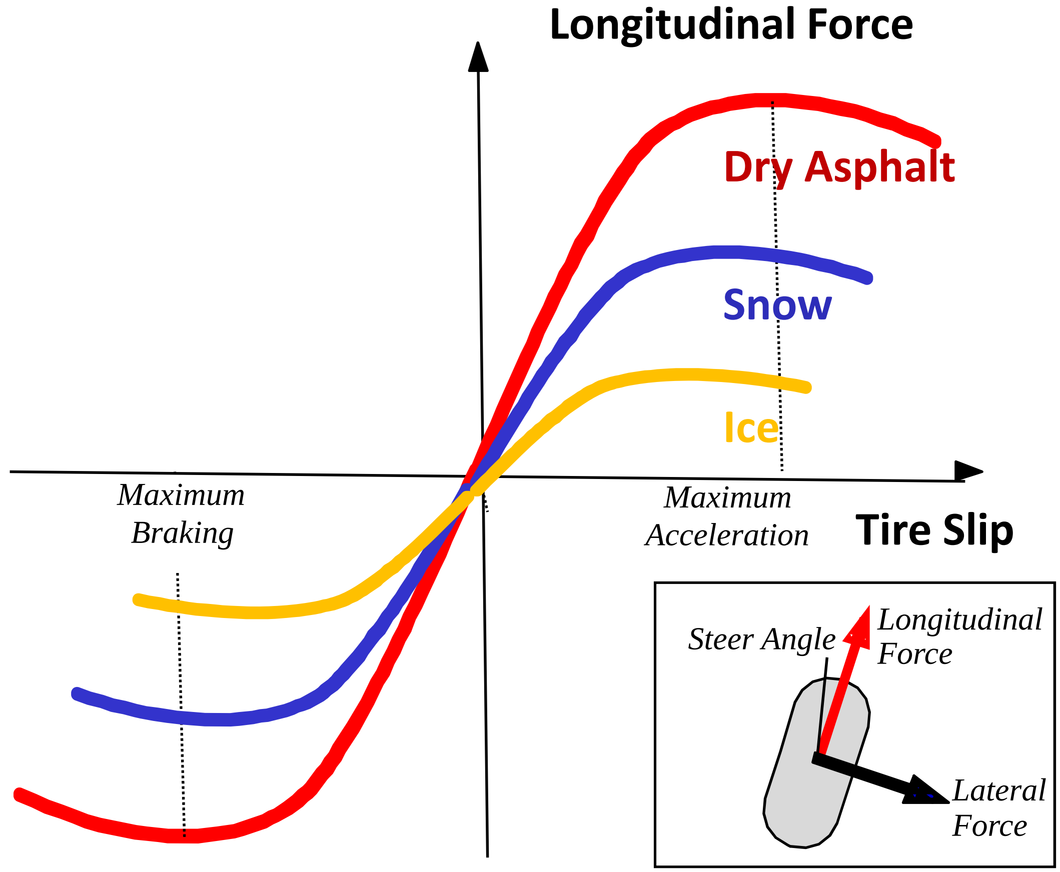


Vehicle-Road Interaction
Simplified Models



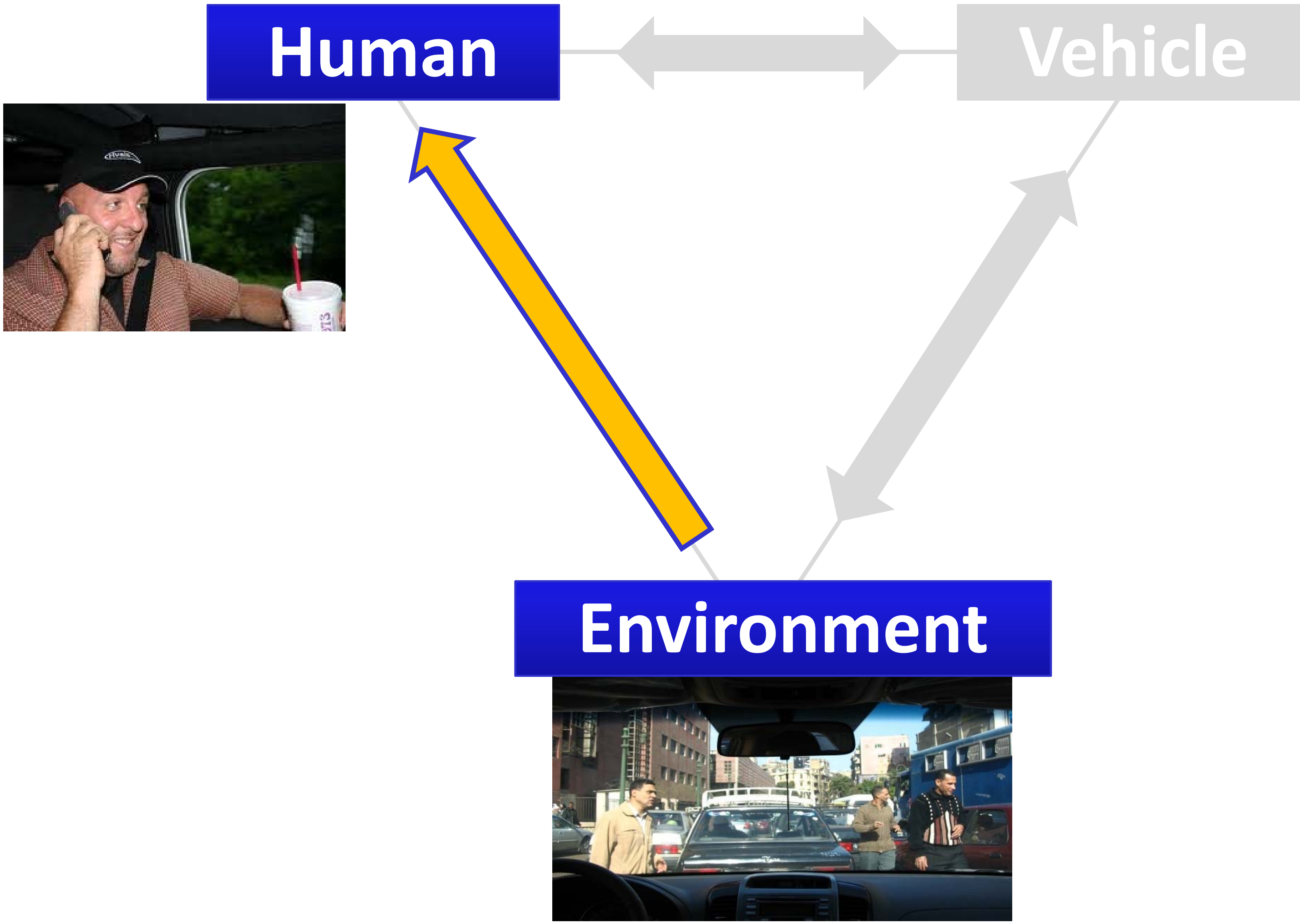
www.mpc.berkeley.edu

Vehicle-Road Interaction
Simplified Models

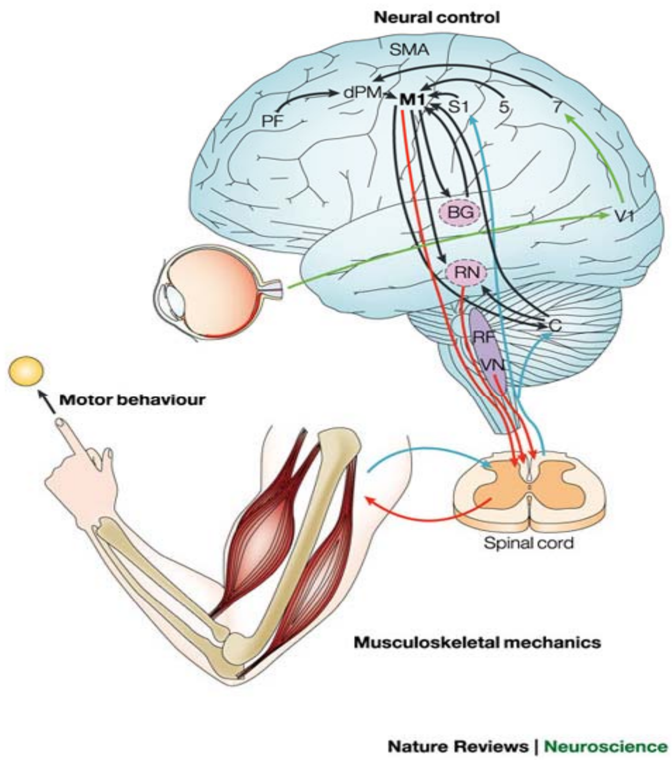


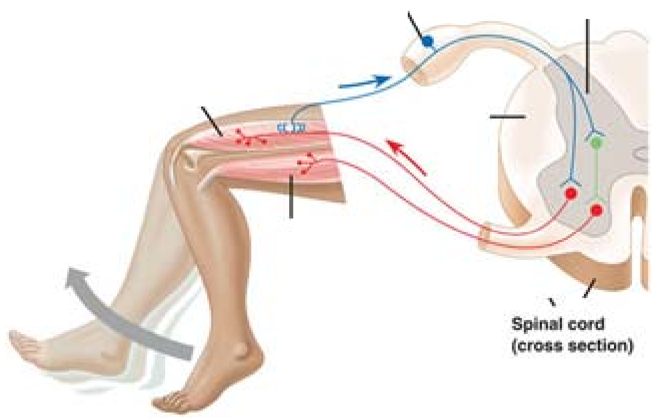
www.mpc.berkeley.edu

CPS-Synoptic Scheme



Environment-Human Interaction





Sensory Inputs

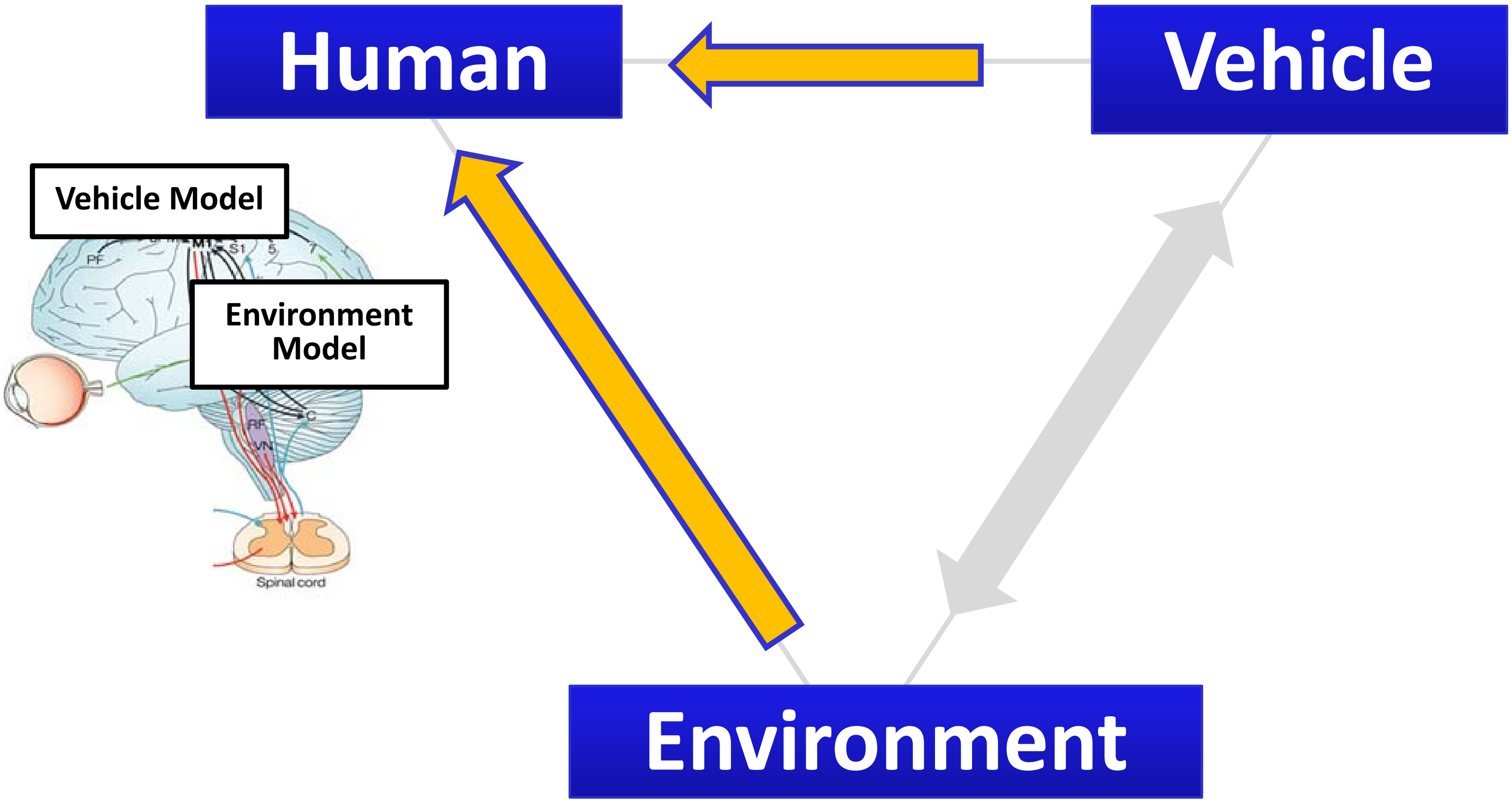
Brain

Spinal Cord

Muscles

http://www.nature.com/nrn/journal/v5/

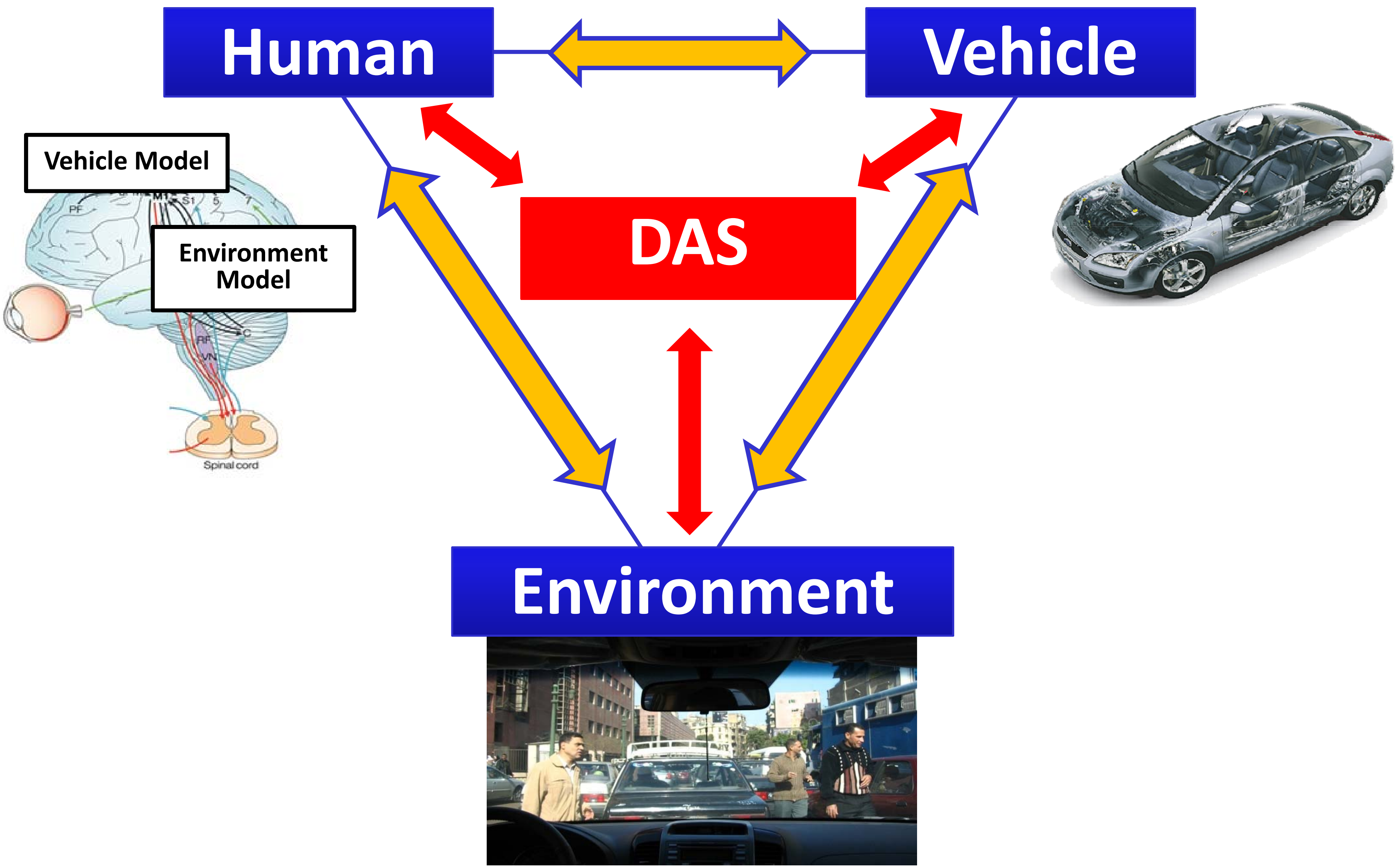
CPS-Synoptic Scheme



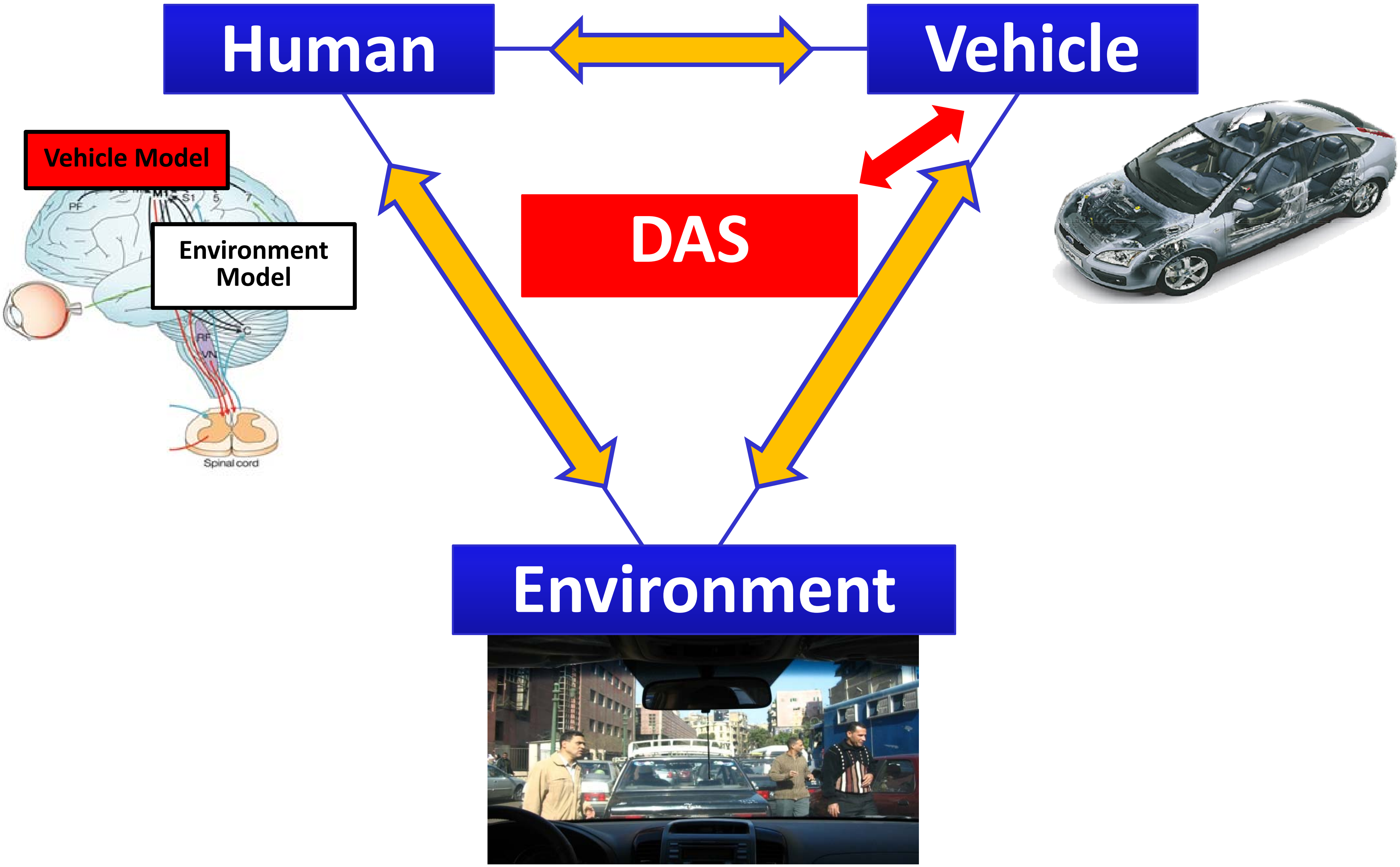
``We know that a lot of the brain has an internal neural simulator”… “to anticipate or predict the future for a given a input”

Eric Kandel (Charlie Rose interview, 2008)

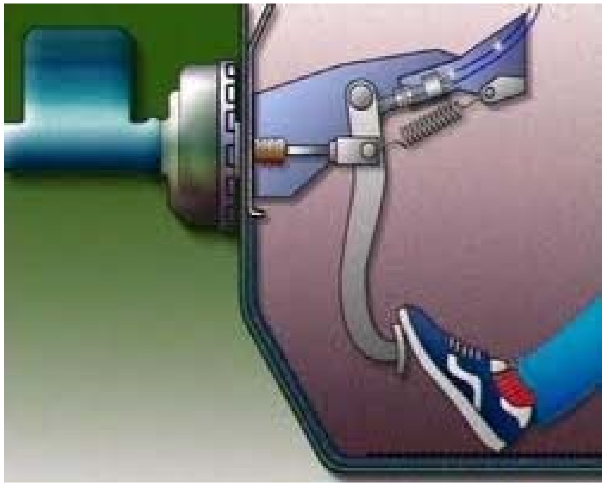
CPS with Driver Assistance System (DAS)

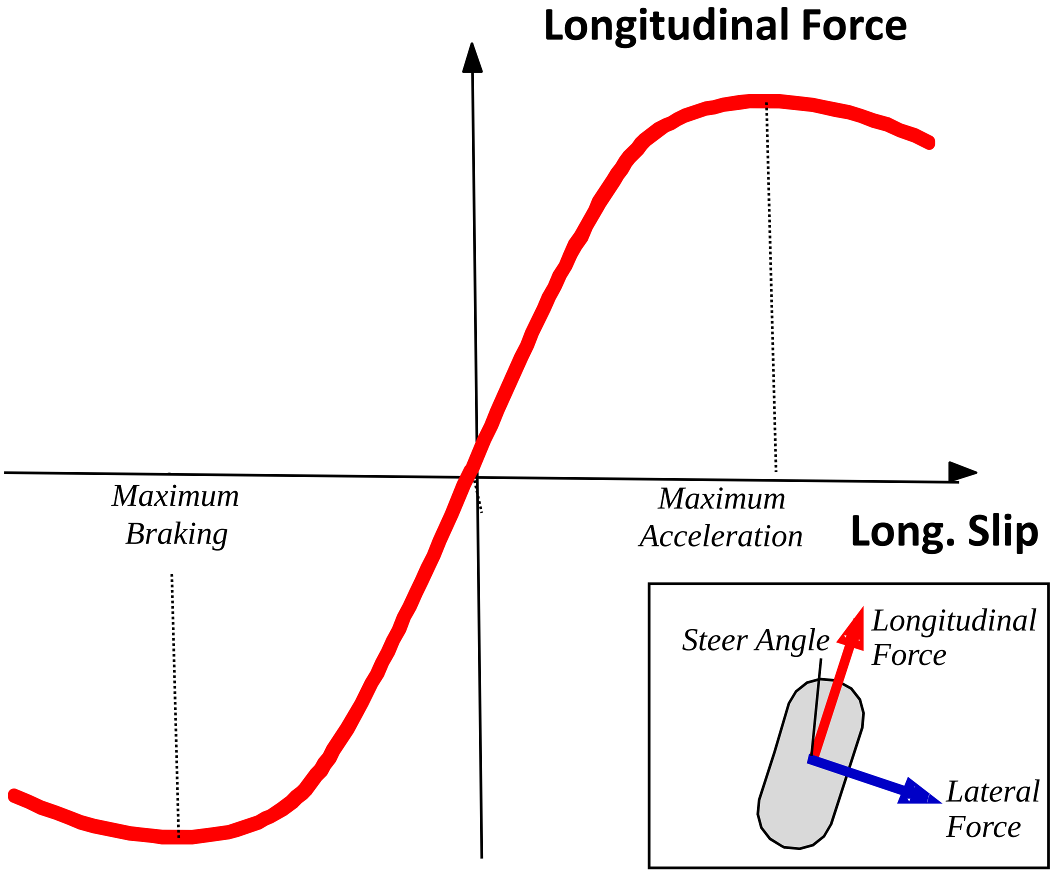


Vast Majority of DAS systems



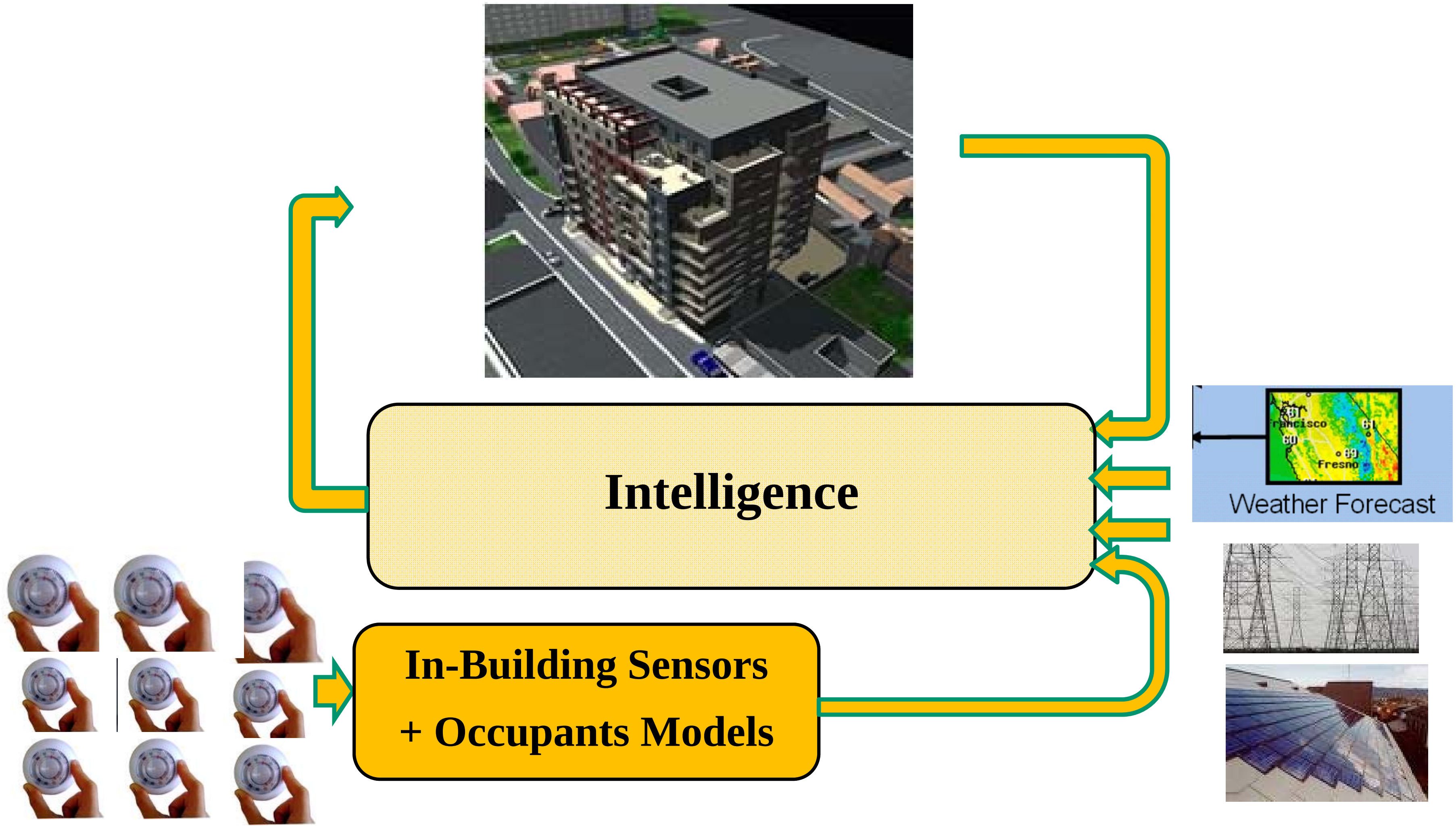
Anti-lock Braking Systems





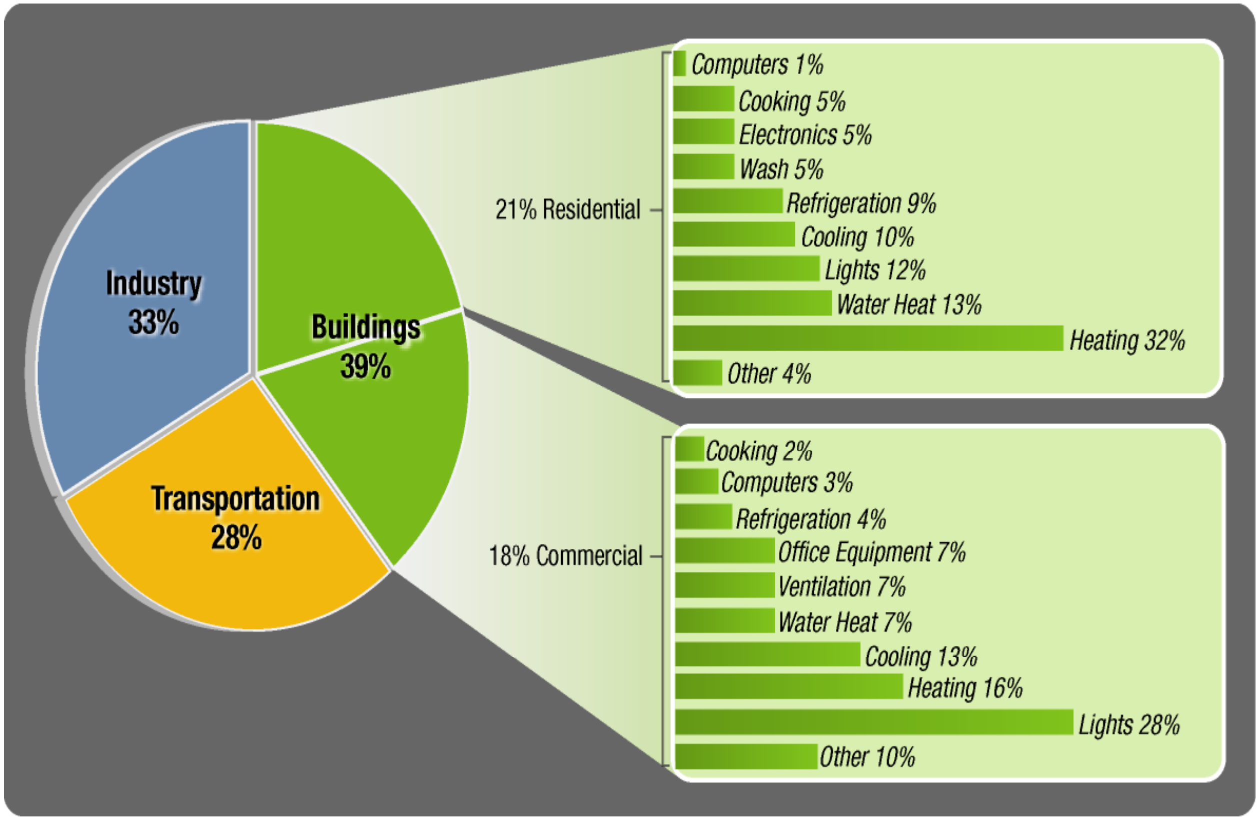
Counter-Steering and Over-Steering

The Building Control Systems of the Future
Predictive and Adaptive Autonomy



Building Energy Demand Challenge

Buildings use 71% of U.S. electricity and 55% of its natural gas

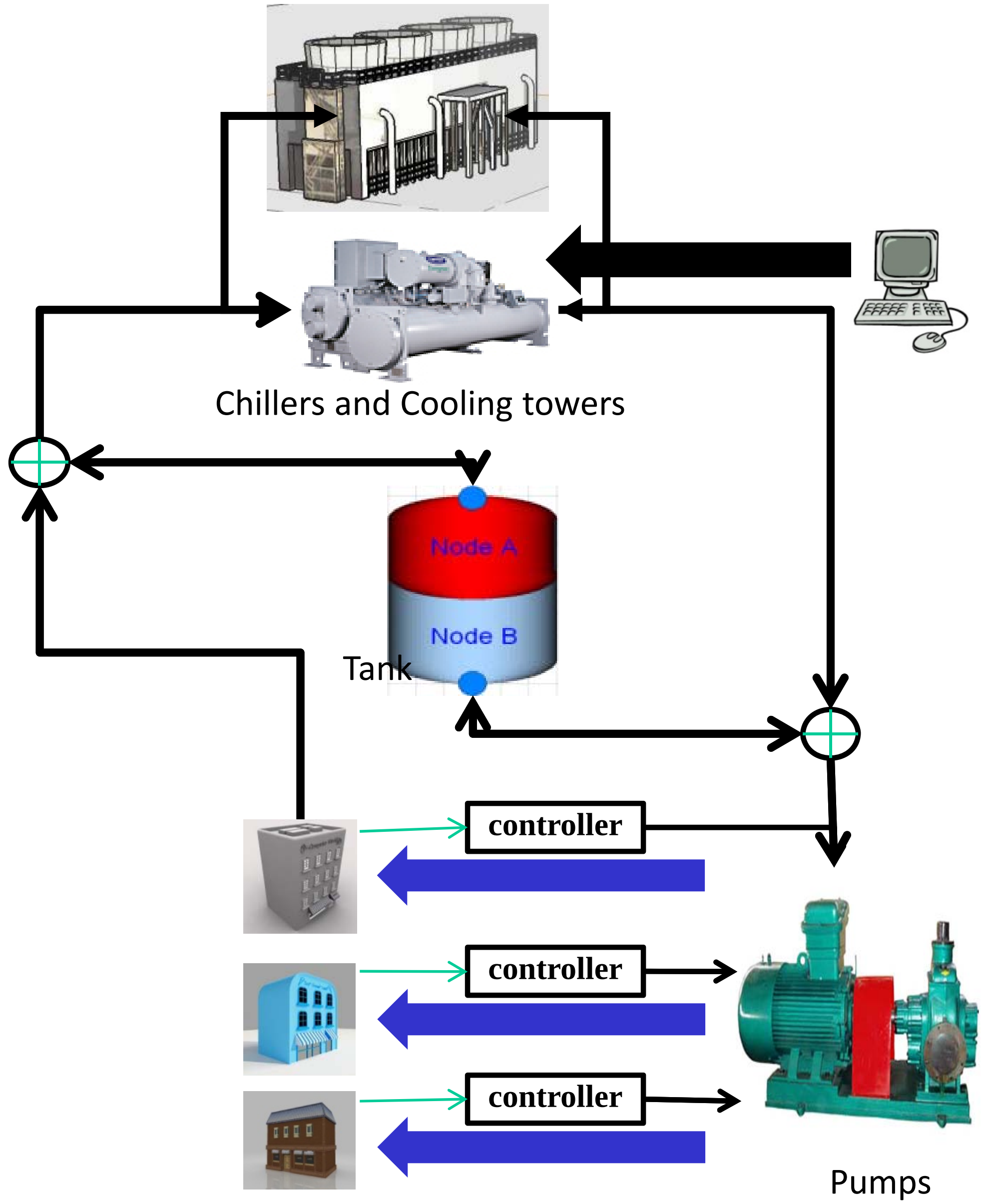


- By 2030
  - 17% growth in electricity demand
  - Additional 200GW of electricity at cost of $500=1000B, or $25-50B/yr

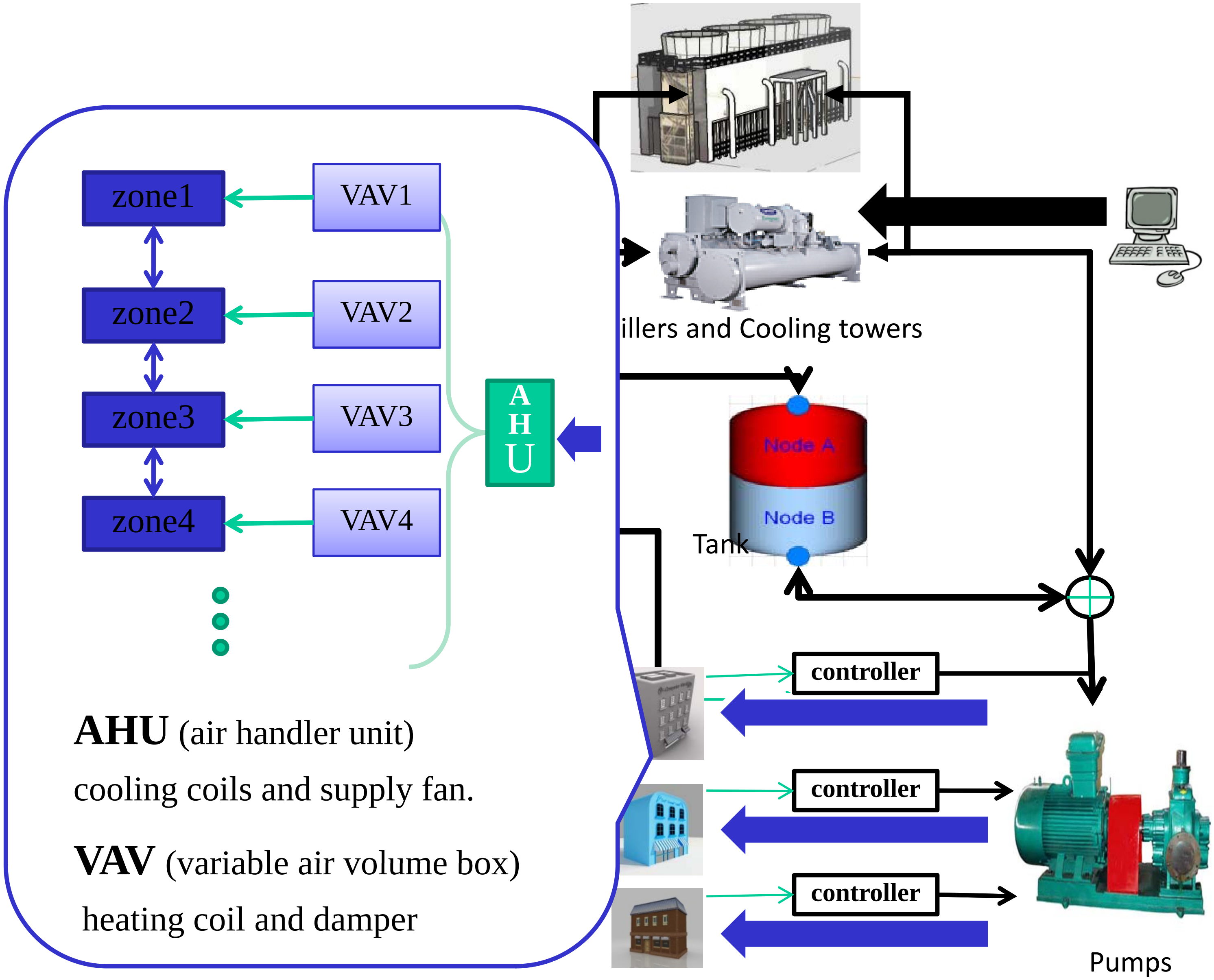
Buildings construction/renovation contributed 9.5% to US GDP and employs approximately 8 million people. Buildings' utility bills totaled $370 Billions in 2005

Source: Buildings Energy Data Book 2007

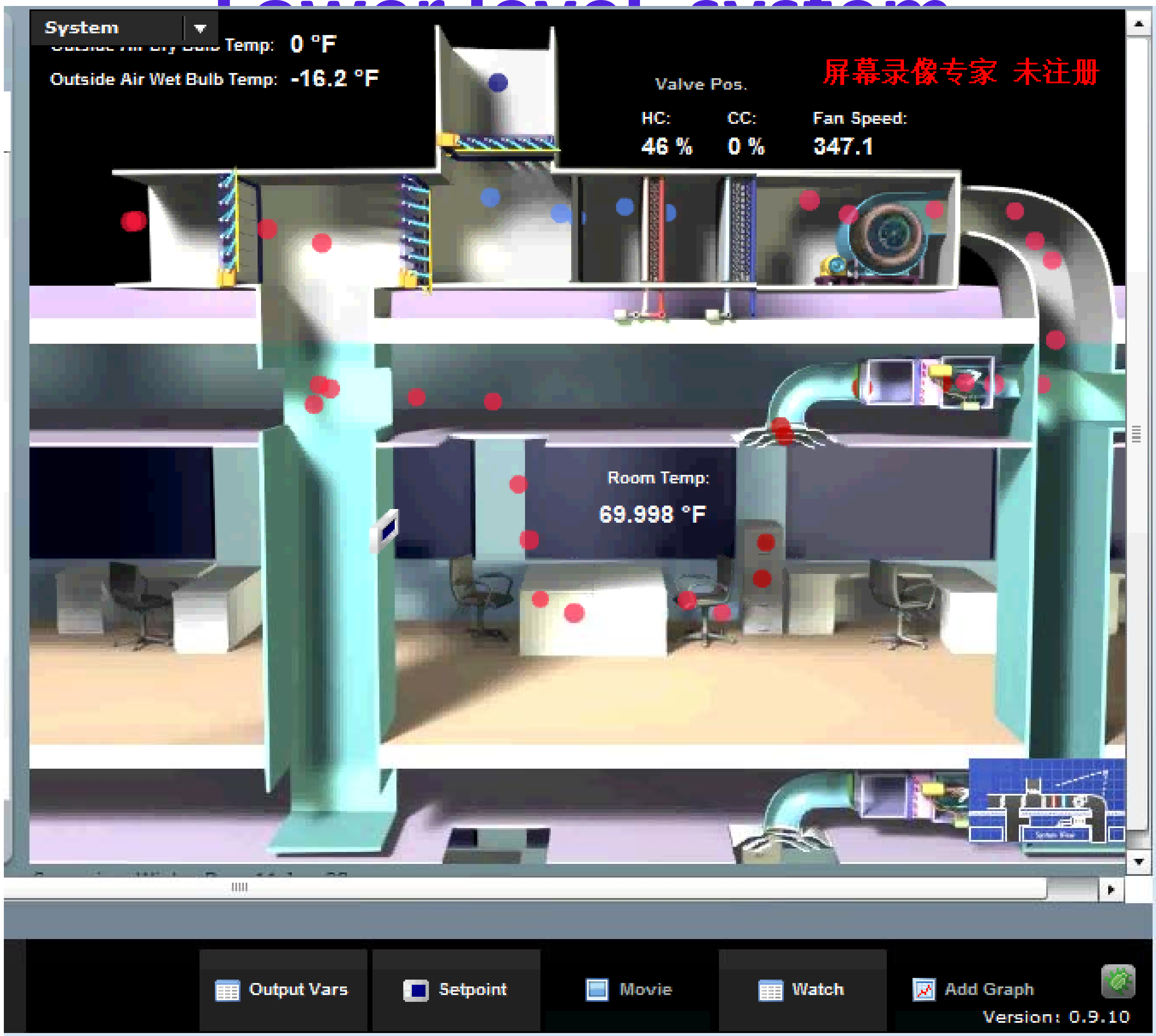
System Description



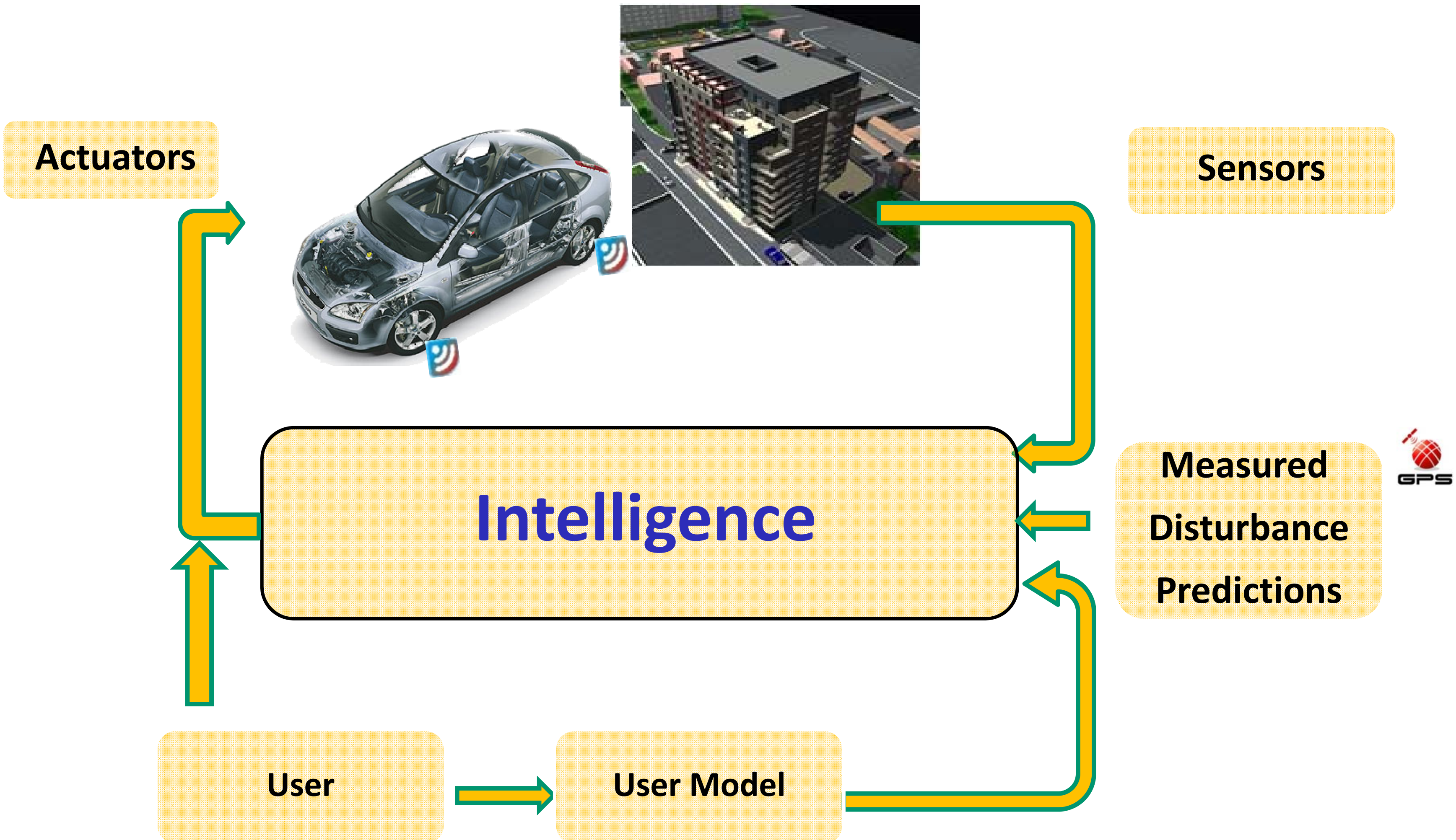
System Description



Lower level system



Autonomous System

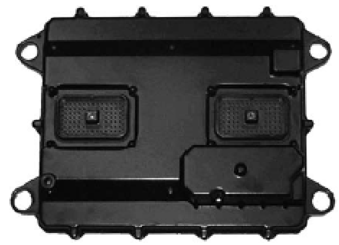


Safety | **Comfort** | **Efficiency**

Control Hardware Platforms for Real-Time Implementation

Hi-End PC

4Ghz, 1 Terabyte



Automotive

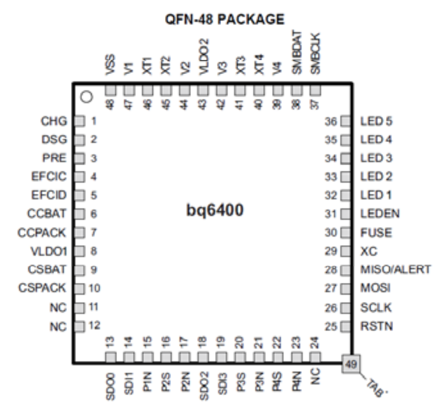
50Mhz, 2 Mbytes

Micro Robotics

16 Mhz, 128Kbytes

Zone Controller

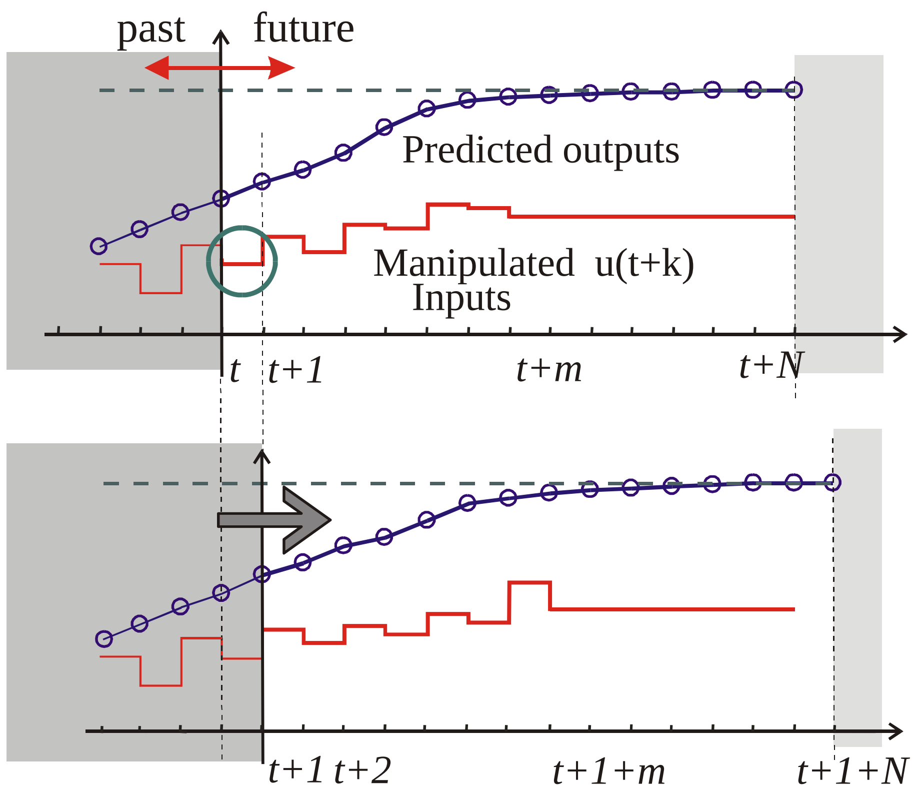
12 Mhz, 512 KByte



Battery Management

8 Mhz, 60 KByte

Model Predictive Control



- Optimize at time *t* (new measurements)
- Only apply the first optimal move *u(t)*
- Repeat the whole optimization at time *t +1*
- Optimization using current measurements œ Feedback

www.mpc.berkeley.edu

MPC Algorithm

$$\min_{U} \int_{t}^{t+N} l(x(t), u(t))$$

$$\text{subj. to} \begin{cases} f(\dot{x}, x, u) = 0 \\ u(\tau) \in \mathcal{U}, \; \forall \, \tau \in [t, t+N] \\ x(\tau) \in \mathcal{X}, \; \forall \, \tau \in [t, t+N] \end{cases}$$

At time t:

- Measure (or estimate) the current state ***x(t)***
- Find the optimal input profile U^* (t)
- Apply only u(τ) , $\tau \in [t, t+\Delta]$

Repeat the same procedure at time $t + \Delta$

www.mpc.berkeley.edu

MPC Algorithm

$$\min_{U} \sum_{k=t}^{t+N-1} l(x_k, u_k)$$

$$\text{subj. to} \begin{cases} x_{k+1} = f(x_k, u_k) \\ u_k \in \mathcal{U} \\ x_k \in \mathcal{X} \\ x_t = x(t) \end{cases}$$

At time t:

- Measure (or estimate) the current state ***x(t)***
- Find the optimal input sequence $U^* = \{\mathbf{u}^*_{\mathbf{t}}, u^*_{t+1}, u^*_{t+2}, \ldots, u^*_{t+N-1}\}$
- Apply only u(t)=$\mathbf{u}^*_{\mathbf{t}}$, and discard $u^*_{t+1}, u^*_{t+2}, \ldots$

Repeat the same procedure at time *t +1*

Predictive, Multivariable, Model Based, Constraints Satisfaction

Important Issues in Model Predictive Control

1. **Feasibility and Stability**
   Optimization problem may be infeasible at some future time step

   Even assuming perfect model, no disturbances:

   predicted open-loop trajectories
   $\neq$
   closed-loop trajectories

2. **Performance**
   What is achieved by repeatedly minimizing $\sum_{k=t}^{t+N-1} l(x_k, u_k)$

3. **Computation**
   Can we guarantee real-time implementation on embedded platforms

Important Issues in Model Predictive Control

1. **Feasibility and Stability**
   Optimization problem may be infeasible at some future time step

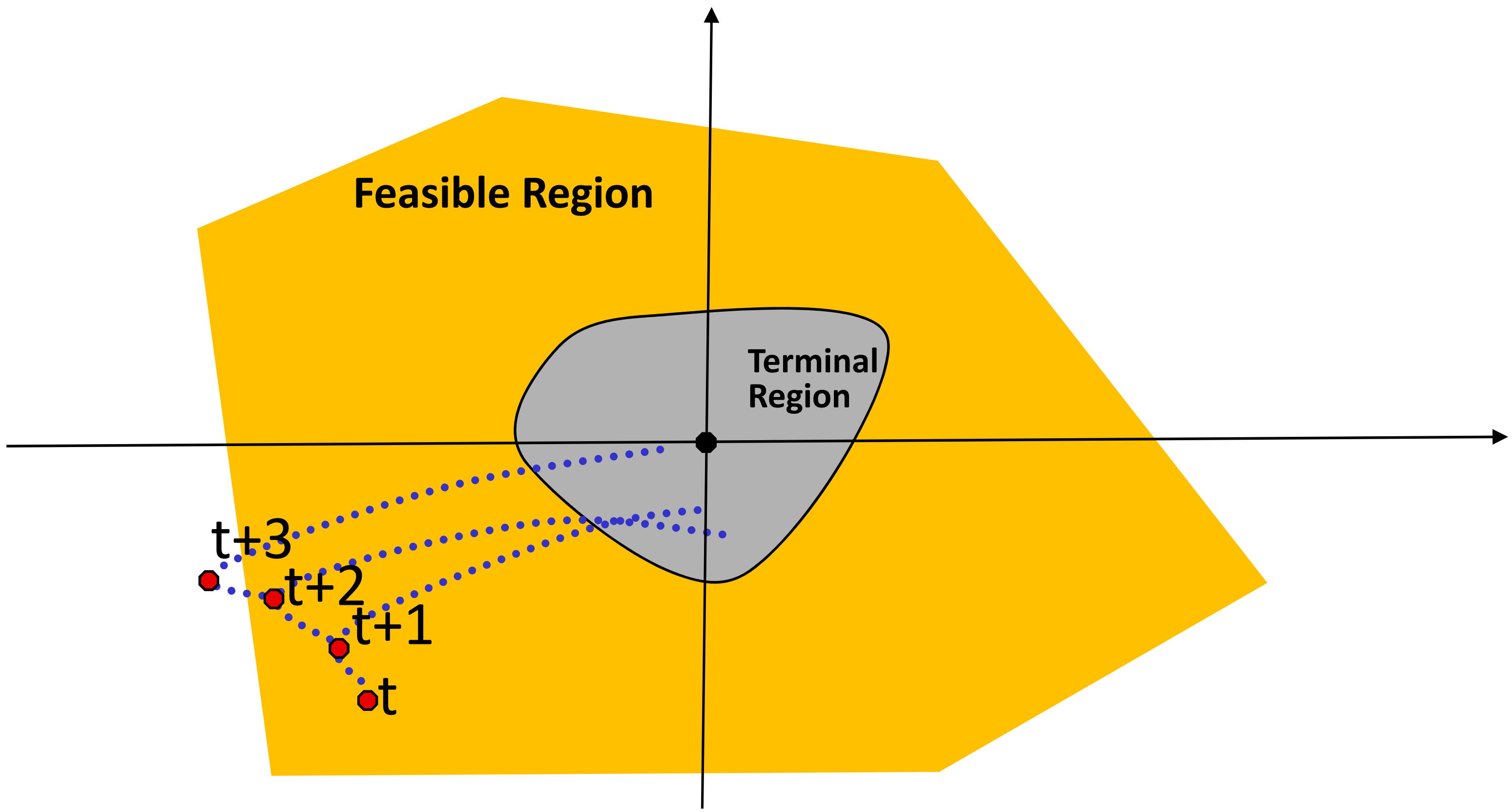
   Even assuming perfect model, no disturbances:

   predicted open-loop trajectories
   $\neq$
   closed-loop trajectories

2. **Performance**
   What is achieved by repeatedly minimizing

3. **Computation**
   Can we guarantee real-time implementation on embedded platforms

Feasiblity and Stability Issue



www.mpc.berkeley.edu

Feasibility and Stability Constraints

$$\min_{U} \sum_{k=t}^{t+N-1} l(x_k, u_k) + p(x_{t+N})$$

$$\text{subj. to} \begin{cases} x_{k+1} = f(x_k, u_k) \\ u_k \in \mathcal{U} \\ x_k \in \mathcal{X} \quad x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{cases}$$

Modified Problem

(Large Body of Literature)

X_f ***(Robust) Invariant Set***

p(x) Control Lyapunov Function

A sketch of the proof....

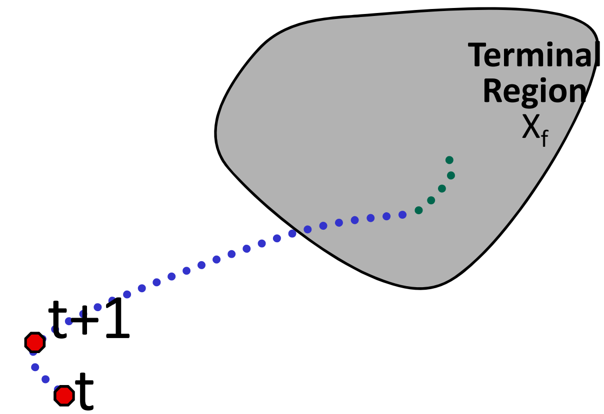
- Define the cost $J(x_t, U) = \sum_{k=t}^{t+N-1} l(x_k, u_k) + p(x_{t+N})$

- Conside the optimal input at time t

$$U^*(t) = \{u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*\}.$$

- And the sequence

$$U_{shift} \triangleq \{u_{t+1}^*, \ldots, u_{t+N-1}^*, v\}.$$



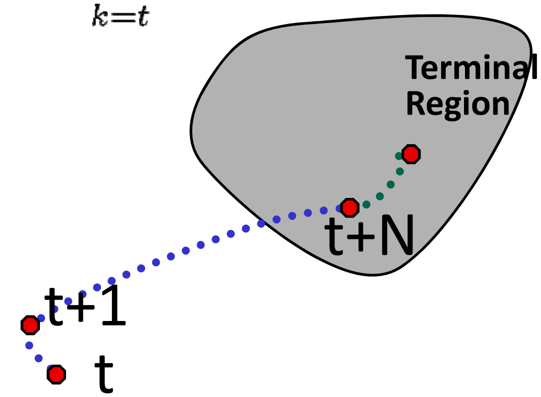
X_f is a controlled invariant $\Rightarrow$ U_{shift} feasible at time t+1

A sketch of the proof….

$$J(x(t), U) = \sum_{k=t}^{t+N-1} l(x_k, u_k) + p(x_{t+N})$$

- Consider the value function

$$V(x_t) = J(x_t, U_t^*)$$



- By simple manipulation

$$V(x_{t+1}) = J(x_{t+1}, U_{t+1}^*) \leq J(x_{t+1}, U_{shift}) =$$
$$V(x_t) - l(x_t, u_t^*) - p(x_{t+N}) + l(x_{t+N}, v) + p(f(x_{t+N}, v))$$

•If

$$\min_{v \in \mathcal{U},\ f(x,v) \in \mathcal{X}_f} (p(f(x,v) - p(x) + l(x,v)) \leq 0$$

•Then

$$V(x_{t+1}) - V(x_t) \leq -l(x_t, u_t^*)$$

If ***p(x)***, ***l(x,u)>0*** , V(t) nonnegative and decreasing along the closed loop trajectories… stability using Lyapunov arguments

Ongoing Research:

Robust Invariant Computation for switched linear systems

Robust Invariance for interconnected linear systems

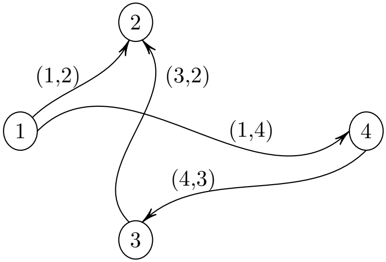
Decentralized Robust Control Invariants in Network Flows

Francesco Borrelli
Miroslav Barić

University of California at Berkeley,
Mechanical Engineering Department

October 23, 2010

Problem Definition



$\mathcal{N} = \{1, 2, 3, 4\}$,
$\mathcal{E} = \{(1, 2), (3, 2), (1, 4), (4, 3)\}$

- directed graph $G = \{\mathcal{N}, \mathcal{E}\}$,
- each node $i \in \mathcal{N}$ features dynamics:

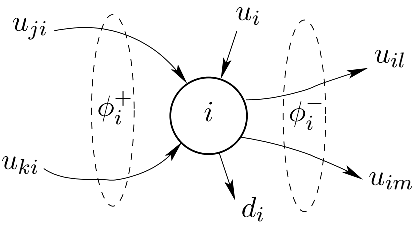
$$x_i^+ = x_i + \phi_i - d_i,$$

where: $\phi_i = u_i + \phi_i^+ - \phi_i^-$

- bounded disturbance (demand) d_i
- dynamics of the whole network:

$$x^+ = x + \phi - d$$

Problem Definition



- directed graph $G = \{\mathcal{N}, \mathcal{E}\}$,
- each node $i \in \mathcal{N}$ features dynamics:

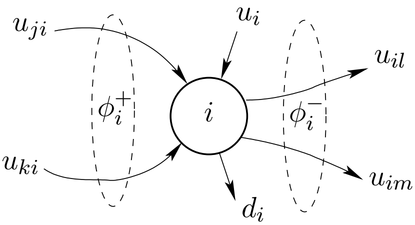
$$x_i^+ = x_i + \phi_i - d_i,$$

  where: $\phi_i = u_i + \phi_i^+ - \phi_i^-$

- bounded disturbance (demand) d_i
- dynamics of the whole network:

$$x^+ = x + \phi - d$$

Problem Definition



- directed graph $G = \{\mathcal{N}, \mathcal{E}\}$,
- each node $i \in \mathcal{N}$ features dynamics:

$$x_i^+ = x_i + \phi_i - d_i,$$

  where: $\phi_i = u_i + \phi_i^+ - \phi_i^-$
- bounded disturbance (demand) d_i
- dynamics of the whole network:

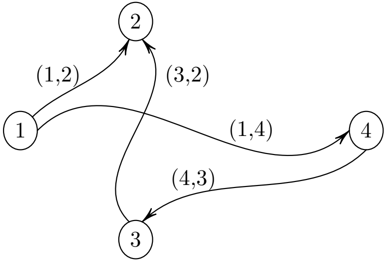
$$x^+ = x + \phi - d$$

Centralized Decision Model

At each time instance controller:

- **knows**: the current state vector x,
- **decides on**: ϕ

Problem Definition



$\mathcal{N} = \{1, 2, 3, 4\}$,
$\mathcal{E} = \{(1, 2), (3, 2), (1, 4), (4, 3)\}$

- directed graph $G = \{\mathcal{N}, \mathcal{E}\}$,
- each node $i \in \mathcal{N}$ features dynamics:

$$x_i^+ = x_i + \phi_i - d_i,$$

where: $\phi_i = u_i + \phi_i^+ - \phi_i^-$

- bounded disturbance (demand) d_i
- dynamics of the whole network:

$$x^+ = x + \phi - d$$

Centralized Decision Model

- **Objective**: Move flow ϕ while satisfying state and input constraints for all admissible disturbances.

Motivation

- Networks of integrators commonly used to model systems for transfer and storage of materials, products, energy ...
  - production–distribution systems,
  - interconnected battery cells,
  - data storage networks etc.
- Problem approached from different perspectives:
  - optimization (since '60): Gale, Ford & Fulkerson, Rockafellar, Wets, Bertsekas, Fujishige ...
  - **dynamical systems:** Blanchini et al.
- **Our contribution**: address complexity issues by considering decentralized constrained robust control

Problem Definition: Constraints

States and disturbances

- given $x \in \mathbb{R}^n$ denote: $x(\mathcal{S}) := \sum_{i \in \mathcal{S}} x_i$, for some $\mathcal{S} \subseteq \{1, \dots, n\}$
- bounding functions: $\underline{x}, \overline{x}, \underline{u}, \overline{u}, \underline{d}, \overline{d} \colon 2^{\mathcal{N}} \to \mathbb{R}$,

$$x \in \mathcal{X} := \{x \colon \underline{x}(\mathcal{S}) \le x(\mathcal{S}) \le \overline{x}(\mathcal{S}),\ \forall \mathcal{S} \subseteq \mathcal{N}\},$$
$$d \in \mathcal{D} := \left\{d \colon \underline{d}(\mathcal{S}) \le d(\mathcal{S}) \le \overline{d}(\mathcal{S}),\ \forall \mathcal{S} \subseteq \mathcal{N}\right\}$$

Example for $\mathcal{N} = \{1, 2\}$:

$$\underline{x}(\{1\}) \le x_1 \le \overline{x}(\{1\}),$$
$$\underline{x}(\{2\}) \le x_2 \le \overline{x}(\{2\}),$$
$$\underline{x}(\{1,2\}) \le x_1 + x_2 \le \overline{x}(\{1,2\}).$$

Problem Definition: Constraints

Flow

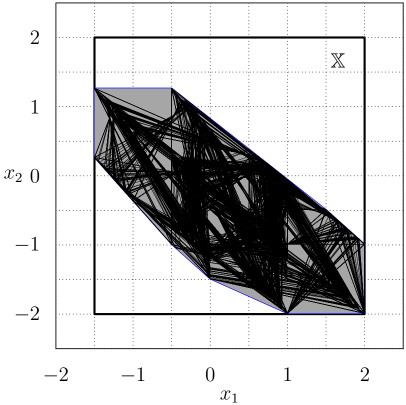
- bounding functions:
  $\underline{u}, \overline{u} \colon 2^{\mathcal{N}} \to \mathbb{R}, \;\; u \in \mathcal{U} := \{u \colon \underline{u}(\mathcal{S}) \leq u(\mathcal{S}) \leq \overline{u}(\mathcal{S}), \; \forall \mathcal{S} \subseteq \mathcal{N}\}$
- capacity function: $\overline{c} \colon (\mathcal{S}, \mathcal{S}') \to \mathbb{R}$ defined for all $\mathcal{S} \subseteq \mathcal{N}$
- $f_{ij} := u_{ij} - u_{ji}$
- $f_{\mathcal{S},\mathcal{S}'} := \displaystyle\sum_{i \in \mathcal{S}, j \in \mathcal{N} \setminus \mathcal{S}} f_{ji}$

$$\phi \in \mathcal{F} := \left\{ \phi \colon \;\; f_{\mathcal{S}\mathcal{S}'} \leq \overline{c}(\mathcal{S}, \mathcal{S}'), \;\; u \in \mathcal{U}, \quad \text{for all } \mathcal{S} \subset \mathcal{N} \;\; \right\}.$$

Preliminaries: Robust Control Invariance

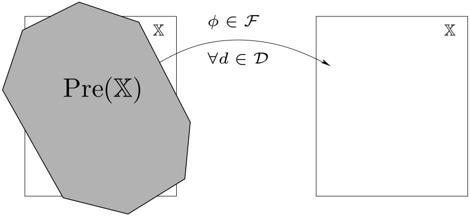
Definition (RCI set)

A set $\mathcal{R} \subseteq \mathcal{X}$ is a robust control invariant (RCI) set if for all $x \in \mathcal{R}$ there exists a flow $\phi \in \mathcal{F}$ such that $x + \phi - d \in \mathcal{R}$ *for all* $d \in \mathcal{D}$.



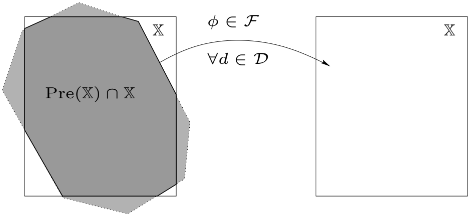
Control Invariant Computation

- Mapping $\mathrm{Pre}(\cdot)$: $\mathrm{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



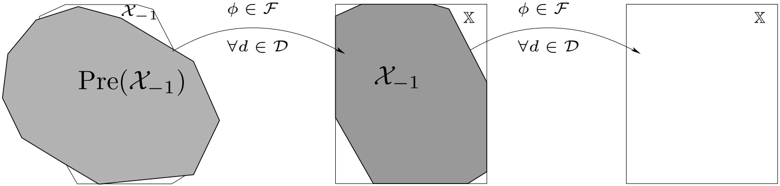
Control Invariant Computation

- Mapping $\mathrm{Pre}(\cdot)$: $\mathrm{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



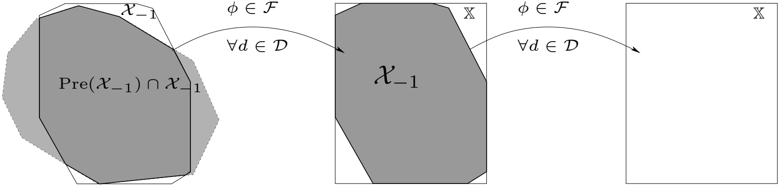
Control Invariant Computation

- Mapping $\mathrm{Pre}(\cdot)$: $\mathrm{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



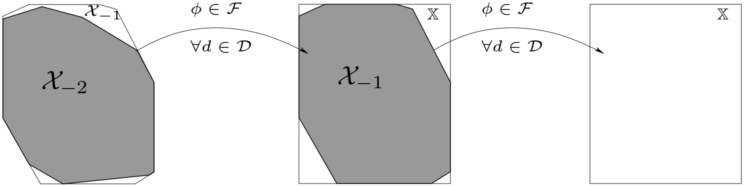
Control Invariant Computation

- Mapping $\mathrm{Pre}(\cdot)$: $\mathrm{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



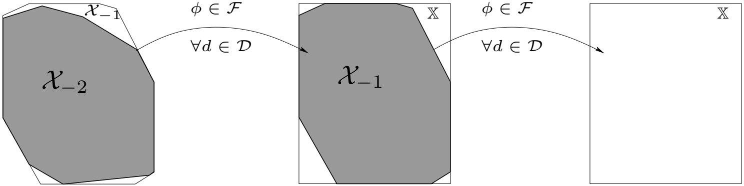
Control Invariant Computation

- Mapping $\text{Pre}(\cdot)$: $\text{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



Control Invariant Computation

- Mapping $\mathrm{Pre}(\cdot)$: $\mathrm{Pre}(\mathcal{X})$ is the set of states robustly controllable into $\mathcal{X}$.



- Repeat this until:

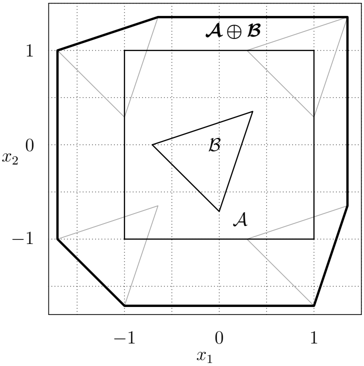
$$\boxed{\mathrm{Pre}(\mathcal{X}_{-i}) = \mathrm{Pre}(\mathcal{X}_{-i+1})}$$

- Fixed point of $\mathrm{Pre}(\cdot)$: **Robust Control Invariant Set** (RCI set).

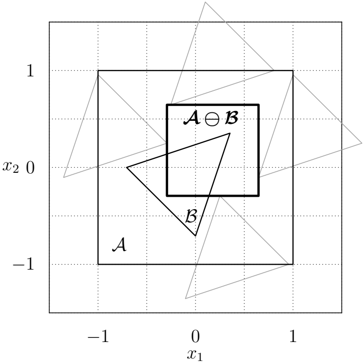
Some Preliminaries... START

Minkowski sum and difference

Minkowski sum $\oplus$



Pontryagin difference $\ominus$



$$\mathcal{A} \oplus \mathcal{B} := \{a + b \colon a \in \mathcal{A}, b \in \mathcal{B}\}$$

$$\mathcal{A} \ominus \mathcal{B} := \{z \colon z + b \in \mathcal{A}, \forall b \in \mathcal{B}\}$$

Controllable Sets for Linear Systems

Consider constrained linear discrete–time system:

$$x^+ = Ax + Bu + Gw, \quad (x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$$

Set of states controllable into the set $\mathcal{Y}$:

$$\mathrm{Pre}(\mathcal{Y}) := \{x \colon \exists u \in \mathcal{U} \text{ such that } Ax + Bu + Gw \in \mathcal{Y},\ \forall w \in \mathcal{W}\}$$

$\Rightarrow$ $Ax + Bu \in \mathcal{Y} \ominus G\mathcal{W}$,

$\Rightarrow$ for each $z \in \mathcal{Y} \ominus G\mathcal{W}$ $\exists x, u \in \mathcal{U}$ such that:

$$Ax = z + (-Bu),$$

$\Rightarrow$ $Ax \in [\mathcal{Y} \ominus G\mathcal{W}] \oplus (-B\mathcal{U})$,

Controllable Sets for Linear Systems

Consider constrained linear discrete–time system:

$$x^+ = Ax + Bu + Gw, \quad (x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$$

Set of states controllable into the set $\mathcal{Y}$:

$$\mathrm{Pre}(\mathcal{Y}) := \{x \colon \exists u \in \mathcal{U} \text{ such that } Ax + Bu + Gw \in \mathcal{Y},\ \forall w \in \mathcal{W}\}$$

$\Rightarrow$ $Ax + Bu \in \mathcal{Y} \ominus G\mathcal{W}$,

$\Rightarrow$ for each $z \in \mathcal{Y} \ominus G\mathcal{W}$ $\exists x, u \in \mathcal{U}$ such that:

$$Ax = z + (-Bu),$$

$\Rightarrow$ $Ax \in [\mathcal{Y} \ominus G\mathcal{W}] \oplus (-B\mathcal{U})$,

Controllable Sets for Linear Systems

Consider constrained linear discrete–time system:

$$x^+ = Ax + Bu + Gw, \quad (x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$$

Set of states controllable into the set $\mathcal{Y}$:

$$\mathrm{Pre}(\mathcal{Y}) := \{x \colon \exists u \in \mathcal{U} \text{ such that } Ax + Bu + Gw \in \mathcal{Y},\ \forall w \in \mathcal{W}\}$$

$\Rightarrow$ $Ax + Bu \in \mathcal{Y} \ominus G\mathcal{W}$,

$\Rightarrow$ for each $z \in \mathcal{Y} \ominus G\mathcal{W}$ $\exists x, u \in \mathcal{U}$ such that:

$$Ax = z + (-Bu),$$

$\Rightarrow$ $Ax \in [\mathcal{Y} \ominus G\mathcal{W}] \oplus (-B\mathcal{U})$,

Controllable Sets for Linear Systems

Consider constrained linear discrete–time system:

$$x^+ = Ax + Bu + Gw, \quad (x, u, w) \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$$

Set of states controllable into the set $\mathcal{Y}$:

$$\boxed{\mathrm{Pre}(\mathcal{Y}) := \{x\colon \exists u \in \mathcal{U} \text{ such that } Ax + Bu + Gw \in \mathcal{Y},\ \forall w \in \mathcal{W}\}}$$

$\Rightarrow$ $Ax + Bu \in \mathcal{Y} \ominus G\mathcal{W}$,

$\Rightarrow$ for each $z \in \mathcal{Y} \ominus G\mathcal{W}$ $\exists x, u \in \mathcal{U}$ such that:

$$Ax = z + (-Bu),$$

$\Rightarrow$ $Ax \in [\mathcal{Y} \ominus G\mathcal{W}] \oplus (-B\mathcal{U})$,

$\Rightarrow$ $\boxed{\mathrm{Pre}(\mathcal{Y}) = \{x\colon Ax \in [\mathcal{Y} \ominus G\mathcal{W}] \oplus (-B\mathcal{U})\}}$

Controllable Sets for Linear Systems

Computations for polyhedral constraints

Given polyhedra $\mathcal{P}_1 = \{\mathbf{x}\colon \mathbf{H}_1\mathbf{x} \leq \mathbf{k}_1\}$ and $\mathcal{P}_2 = \{\mathbf{x}\colon \mathbf{H}_2\mathbf{x} \leq \mathbf{k}_2\}$:

- Pontryagin difference $\mathcal{P}_1 \ominus \mathcal{P}_2$:

$$\begin{aligned}\mathcal{P}_1 \ominus \mathcal{P}_2 &= \{\mathbf{x}\colon \mathbf{H}_1(\mathbf{x}+\mathbf{y}) \leq \mathbf{k}_1,\ \forall \mathbf{y} \in \mathcal{P}_2\} = \\ &= \{\mathbf{x}\colon \mathbf{H}_1\mathbf{x} \leq \tilde{\mathbf{k}}_1\},\end{aligned}$$

  where: $\tilde{k}_{1i} = k_{1i} - \max_{y \in \mathcal{P}_2} \mathbf{h}_{1i}^T y$.

- Minkowski sum $\mathcal{P}_1 \oplus \mathcal{P}_2$:

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \mathrm{Proj}_z \left\{ (\mathbf{z}, \mathbf{y}_1, \mathbf{y}_2)\colon \begin{array}{c} \mathbf{z} = \mathbf{y}_1 + \mathbf{y}_2, \\ \mathbf{H}_1\mathbf{y}_1 \leq \mathbf{k}_1, \\ \mathbf{H}_2\mathbf{y}_2 \leq \mathbf{k}_2. \end{array} \right\}$$

Some Preliminaries... END

Good and Bad News

Theorem Blanchini 1998, Borrelli 2009, Gale–Hoffmann theorem (Gale, 1957)

- maximal RCI set $\mathcal{R}_\infty$ for a network of integrators given as:

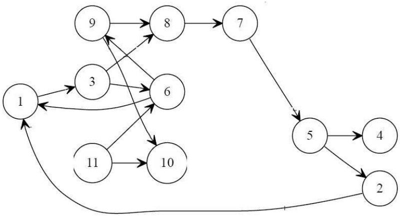
$$\mathcal{R}_\infty = \underbrace{\{[\mathcal{X} \ominus (-\mathcal{D})] \oplus (-\mathcal{F})\}}_{\mathrm{Pre}(\mathcal{X})} \cap \mathcal{X}$$

- $\mathcal{R}_\infty$ non–empty iff:

$$\boxed{\mathcal{D} \subseteq \mathcal{F} \ \text{ and } \ \mathcal{X} \ominus (-\mathcal{D}) \neq \emptyset.}$$

Worst Case Exponentially Complex

Numerical Example



	Node i										
	1	2	3	4	5	6	7	8	9	10	11
$\overline{u}(i)$	1	1	2.5	1.5	3.5	1	4	1	2.5	1	5
$\overline{d}(i)$	3	2.5	2	2	2.5	1.5	3	2	2	2.5	1
$\overline{x}(i)$	5	5	5	5	5	5	5	5	5	5	5

NETWORK AND CONSTRAINTS

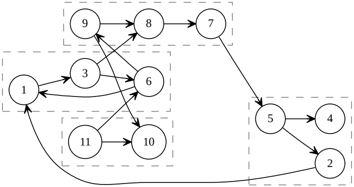
- coupling constraints

$$x_1 + x_3 + x_6 \leq 10, \quad x_2 + x_4 + x_5 \leq 10, \quad x_7 + x_8 + x_9 \geq 2$$

- centralized control: set of admissible flows $\mathcal{F}$ defined by 360 (non–redundant) inequalities

Decentralized Robust Control Invariance

Decision model

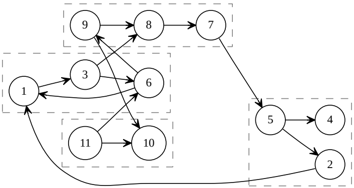


- consider a network partition $\Delta := \{\mathcal{S}_i\}_{i=1}^{q}$, where $\mathcal{N} = \bigcup_{i=1}^{q} \mathcal{S}_i$ and $\mathcal{S}_i$ are mutually disjoint
- dynamics of the ith group of nodes:

$$x_{\mathcal{S}_i}^{+} = x_{\mathcal{S}_i} + u_{\mathcal{S}_i} - \phi_{\mathcal{S}_i}^{-} + \phi_{\mathcal{S}_i}^{+} - d_{\mathcal{S}_i},$$

Decentralized Robust Control Invariance

Decision model



- consider a network partition $\Delta := \{\mathcal{S}_i\}_{i=1}^{q}$, where $\mathcal{N} = \bigcup_{i=1}^{q} \mathcal{S}_i$ and $\mathcal{S}_i$ are mutually disjoint
- dynamics of the ith group of nodes:

$$x_{\mathcal{S}_i}^{+} = x_{\mathcal{S}_i} + u_{\mathcal{S}_i} - \phi_{\mathcal{S}_i}^{-} + \phi_{\mathcal{S}_i}^{+} - d_{\mathcal{S}_i},$$

Decision model

At each time instance the controller for the ith group of nodes:

- **knows**: the current local state vector $x_{\mathcal{S}_i}$,
- **decides on**: $u_{\mathcal{S}_i}$ and $\phi_{\mathcal{S}_i}^{-}$

For the ith controller: $\phi_{\mathcal{S}_i}^{+}$ unknown and part of the disturbance $\tilde{d}_{\mathcal{S}_i} := d_{\mathcal{S}_i} - \phi_{\mathcal{S}_i}^{+}$.

Decentralized Robust Control Invariance

Decentralized RCI Set

An RCI set $\mathcal{R}^{\Delta}$ is a decentralized RCI set w.r.t. the network partition Δ if

(i) $x + \phi(x) - d \in \mathcal{R}^{\Delta}$ for all $d \in \mathcal{D}$, and

(ii) Invariance–inducing robust control law $\phi(\cdot)$ is composed of local control laws $\tilde{\phi}_{\mathcal{S}_i}(\cdot)$ using only local information $x_{\mathcal{S}_i}$.
For each $\mathcal{S}_i \in \Delta$

$$\phi^{-}_{\mathcal{S}_i}(x) + u_{\mathcal{S}_i}(x) = \phi^{-}_{\mathcal{S}_i}(x_{\mathcal{S}_i}) + u_{\mathcal{S}_i}(x_{\mathcal{S}_i}).$$

Decentralized Robust Control Invariance

Decentralized RCI sets, not surprisingly, posses a special structure:

Proposition

If $\mathcal{R}^{\Delta}$ is a decentralized RCI set w.r.t. the network partition $\Delta = \{\mathcal{S}_i\}_{i=1}^{q}$, then:

$$\boxed{\mathcal{R}^{\Delta} = \prod_{i=1}^{q} \mathcal{R}^{\Delta}_{\mathcal{S}_i}}$$

Decentralized Robust Control Invariance

Decentralized RCI sets, not surprisingly, posses a special structure:

Proposition

If $\mathcal{R}^{\Delta}$ is a decentralized RCI set w.r.t. the network partition $\Delta = \{\mathcal{S}_i\}_{i=1}^{q}$, then:

$$\boxed{\mathcal{R}^{\Delta} = \prod_{i=1}^{q} \mathcal{R}^{\Delta}_{\mathcal{S}_i}}$$

Q: What can we say about the existence of decentralized RCI sets for given constraints?

Decentralized RCI set: Parametrization

Main idea

1. Parametrize a family of decentralized RCI sets in bounds that define constraint sets $\mathcal{X}$ and $\mathcal{F}$
2. Solve a centralized problem computing **a** set of parameters for which the global invariant is non-empty
3. Use local parameter values to compute local robust invariants $\mathcal{R}^{\Delta}_{\mathcal{S}_i}$

Decentralized RCI set: Parametrization

Main idea

1. Parametrize a family of decentralized RCI sets in bounds that define constraint sets $\mathcal{X}$ and $\mathcal{F}$
2. Solve a centralized problem computing **a** set of parameters for which the global invariant is non-empty
3. Use local parameter values to compute local robust invariants $\mathcal{R}^{\Delta}_{\mathcal{S}_i}$

Main Results

- Problem 2 is convex!
- $\prod_{i=1}^{q} \mathcal{R}^{\Delta}_{\mathcal{S}_i}$ is a global robust invariant

Decentralized RCI set: Parametrization

- dynamics of the ith subsystem:

$$x_{\mathcal{S}_i}^{+} = x_{\mathcal{S}_i} + \tilde{\phi}_{\mathcal{S}_i} - \tilde{d}_{\mathcal{S}_i} \quad, i = 1, \ldots, q,$$

where $\tilde{\phi}_{\mathcal{S}_i} := u_{\mathcal{S}_i} - \phi_{\mathcal{S}_i}^{-}$ and $\tilde{d}_{\mathcal{S}_i} := d_{\mathcal{S}_i} - \phi_{\mathcal{S}_i}^{+}$.

Decentralized RCI set: Parametrization

- dynamics of the ith subsystem:

$$x^+_{\mathcal{S}_i} = x_{\mathcal{S}_i} + \tilde{\phi}_{\mathcal{S}_i} - \tilde{d}_{\mathcal{S}_i} \quad , i = 1, \ldots, q,$$

  where $\tilde{\phi}_{\mathcal{S}_i} := u_{\mathcal{S}_i} - \phi^-_{\mathcal{S}_i}$ and $\tilde{d}_{\mathcal{S}_i} := d_{\mathcal{S}_i} - \phi^+_{\mathcal{S}_i}$.
- introduce a vector of parameters ξ comprising the bounds:
  - $\underline{\mu}^{\mathcal{S}}$ and $\overline{\mu}^{\mathcal{S}}$, for $u(\mathcal{S})$,
  - $\underline{\psi}^{\mathcal{S}}$ and $\overline{\psi}^{\mathcal{S}}$, for $x(\mathcal{S})$,
  - $\underline{\pi}_{ij}$ and $\overline{\pi}_{ij}$ for u_{ij}, $(i, j) \in \mathcal{E}$,

Decentralized RCI set: Parametrization

- dynamics of the ith subsystem:

$$x^+_{\mathcal{S}_i} = x_{\mathcal{S}_i} + \tilde{\phi}_{\mathcal{S}_i} - \tilde{d}_{\mathcal{S}_i} \quad , i = 1, \dots, q,$$

  where $\tilde{\phi}_{\mathcal{S}_i} := u_{\mathcal{S}_i} - \phi^-_{\mathcal{S}_i}$ and $\tilde{d}_{\mathcal{S}_i} := d_{\mathcal{S}_i} - \phi^+_{\mathcal{S}_i}$.
- introduce a vector of parameters ξ comprising the bounds:
  - $\underline{\mu}^{\mathcal{S}}$ and $\overline{\mu}^{\mathcal{S}}$, for $u(\mathcal{S})$,
  - $\underline{\psi}^{\mathcal{S}}$ and $\overline{\psi}^{\mathcal{S}}$, for $x(\mathcal{S})$,
  - $\underline{\pi}_{ij}$ and $\overline{\pi}_{ij}$ for u_{ij}, $(i,j) \in \mathcal{E}$,
- set of admissible parameters $\xi \in \mathcal{C}$:

$$\mathcal{C} = \left\{ \xi\colon \begin{array}{ll} \underline{x}(\mathcal{S}) \le \underline{\psi}^{\mathcal{S}} \le \overline{\psi}^{\mathcal{S}} \le \overline{x}(\mathcal{S}) & \\ \underline{u}(\mathcal{S}) \le \underline{\mu}^{\mathcal{S}} \le \overline{\mu}^{\mathcal{S}} \le \overline{u}(\mathcal{S}), & \forall \mathcal{S} \subseteq \mathcal{S}_i,\ \forall \mathcal{S}_i \in \Delta \\ \underline{u}_{jk} \le \underline{\pi}_{jk} \le \overline{\pi}_{jk} \le \overline{u}_{jk}, & (j,k) \in \mathcal{E} \end{array} \right\}$$

Decentralized RCI set: Parametrization

Rewrite the feasible flow set as a function of the paramenters

$$\tilde{\mathcal{F}}^{\mathcal{S}_\mathbf{i}}(\xi) = \left\{ \phi \in \mathbb{R}^{|\mathcal{S}_i|} : \ \underline{\sigma}_{\mathcal{S}}(\xi) \leq \phi(\mathcal{S}) \leq \overline{\sigma}_{\mathcal{S}}(\xi) \ , \forall \mathcal{S} \subseteq \mathcal{S}_i \right\}$$

Note that $\underline{\sigma}$ and $\overline{\sigma}$ is linear in the parameters ξ.
Similarly for $\tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi)$ and $\tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi)$.

Decentralized RCI set: Parametrization

Rewrite the feasible flow set as a function of the paramenters

$$\tilde{\mathcal{F}}^{\mathcal{S}_\mathrm{i}}(\xi) = \left\{ \phi \in \mathbb{R}^{|\mathcal{S}_i|} : \ \underline{\sigma}_{\mathcal{S}}(\xi) \leq \phi(\mathcal{S}) \leq \overline{\sigma}_{\mathcal{S}}(\xi) \ , \forall \mathcal{S} \subseteq \mathcal{S}_i \right\}$$

Note that $\underline{\sigma}$ and $\overline{\sigma}$ is linear in the parameters ξ.
Similarly for $\tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi)$ and $\tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi)$.

- parametrized maximal RCI set for the ith subsystem:

$$\mathcal{R}_\xi^{\mathcal{S}_i} := \mathrm{Pre}(\tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi)) \cap \tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi)$$

- parametrized RCI set for the whole network:

$$\mathcal{R}_\xi^{\Delta} := \prod_{i=1}^{q} \mathcal{R}_\xi^{\mathcal{S}_i}$$

Decentralized RCI set: Parametrization

Q: For which parameters $\xi \in \mathcal{C}$ is the set $\mathcal{R}_{\xi}^{\Delta}$ non–empty?

From conditions on non-emptiness of $\mathcal{R}_{\xi}^{\mathcal{S}_i}$ follows that $\mathcal{R}_{\xi}^{\Delta} = \prod_{i=1}^{q} \mathcal{R}_{\xi}^{\mathcal{S}_i}$ is non–empty for all ξ from the set:

$$\Pi := \left\{ \xi \in \mathcal{C} : \begin{array}{l} \tilde{\mathcal{F}}^{\mathcal{S}_i}(\xi) \supseteq \tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi), \text{ and} \\ \tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi) \ominus \left(-\tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi) \right) \neq \emptyset \end{array}, \forall \mathcal{S}_i \in \Delta \right\}$$

Decentralized RCI set: Parametrization

Q: For which parameters $\xi \in \mathcal{C}$ is the set $\mathcal{R}_\xi^\Delta$ non–empty?

From conditions on non-emptiness of $\mathcal{R}_\xi^{\mathcal{S}_i}$ follows that $\mathcal{R}_\xi^\Delta = \prod_{i=1}^q \mathcal{R}_\xi^{\mathcal{S}_i}$ is non–empty for all ξ from the set:

$$\Pi := \left\{ \xi \in \mathcal{C} : \begin{array}{l} \tilde{\mathcal{F}}^{\mathcal{S}_i}(\xi) \supseteq \tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi), \text{ and} \\ \tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi) \ominus \left(-\tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi) \right) \neq \emptyset \end{array}, \ \forall \mathcal{S}_i \in \Delta \right\}$$

Decentralized RCI set: Parametrization

Q: For which parameters $\xi \in \mathcal{C}$ is the set $\mathcal{R}_{\xi}^{\Delta}$ non–empty?

From conditions on non-emptiness of $\mathcal{R}_{\xi}^{\mathcal{S}_i}$ follows that $\mathcal{R}_{\xi}^{\Delta} = \prod_{i=1}^{q} \mathcal{R}_{\xi}^{\mathcal{S}_i}$ is non–empty for all ξ from the set:

$$\Pi := \left\{ \xi \in \mathcal{C} : \begin{array}{l} \tilde{\mathcal{F}}^{\mathcal{S}_i}(\xi) \supseteq \tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi), \text{ and} \\ \tilde{\mathcal{X}}^{\mathcal{S}_i}(\xi) \ominus \left(-\tilde{\mathcal{D}}^{\mathcal{S}_i}(\xi) \right) \neq \emptyset \end{array}, \ \forall \mathcal{S}_i \in \Delta \right\}$$

Theorem

Set Π of parameters ξ for which the $\mathcal{R}_{\xi}^{\Delta} \neq \emptyset$ is a polyhedral set (possibly empty).

Requirement for the proof: Farkas lemma

One formulation

In one of its variants Farkas lemma says:

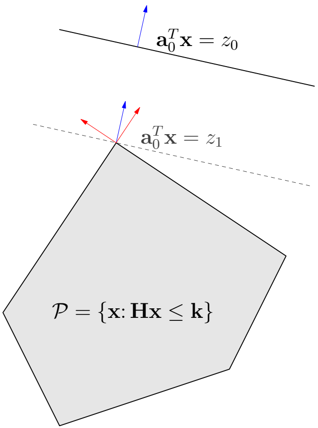
An inequality valid for a polyhedron $\mathcal{P} \neq \emptyset$ can be expressed as a non–negative combination of its inequalities.

More formally:

Farkas lemma

Inequality $\mathbf{a}_0^T\mathbf{x} \leq z_0$ is valid for $\mathbf{x} \in \mathcal{P} \neq \emptyset$ iff there exists $\boldsymbol{\lambda} \geq \mathbb{0}$ such that:

$$\mathbf{H}^T\boldsymbol{\lambda} = \mathbf{a}_0 \text{ and } \boldsymbol{\lambda}^T\mathbf{k} \leq z_0$$

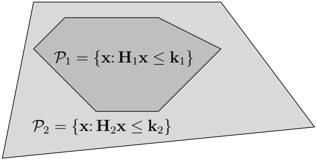


Farkas lemma

Requirement for the proof: Inclusion of polyhedra

- Non–empty polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$:
  $\mathcal{P}_1 \subseteq \mathcal{P}_2$ iff there exists $\Lambda \geq \mathbb{0}$ such that:

$$\Lambda \mathbf{H}_1 = \mathbf{H}_2 \text{ and } \Lambda \mathbf{k}_1 \leq \mathbf{k}_2$$



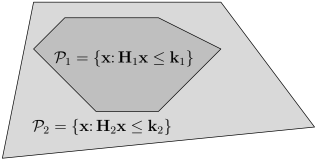
Farkas lemma

Requirement for the proof: Inclusion of polyhedra

- Non–empty polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$:

  $\mathcal{P}_1 \subseteq \mathcal{P}_2$ iff there exists $\Lambda \geq \mathbb{0}$ such that:

  $$\Lambda \mathbf{H}_1 = \mathbf{H}_2 \text{ and } \Lambda \mathbf{k}_1 \leq \mathbf{k}_2$$



- In general linear parametric polyhedral inclusion is nonconvex

$$\Lambda \mathbf{H}_1(\xi) = \mathbf{H}_2(\xi) \text{ and } \Lambda \mathbf{k}_1(\xi) \leq \mathbf{k}_2(\xi)$$

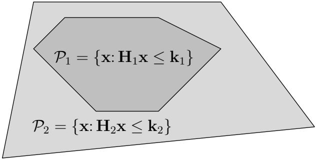
Farkas lemma

Requirement for the proof: Inclusion of polyhedra

- Non–empty polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$:

  $\mathcal{P}_1 \subseteq \mathcal{P}_2$ iff there exists $\Lambda \geq \mathbb{0}$ such that:

$$\Lambda \mathbf{H}_1 = \mathbf{H}_2 \text{ and } \Lambda \mathbf{k}_1 \leq \mathbf{k}_2$$



- In general linear parametric polyhedral inclusion is nonconvex

$$\Lambda \mathbf{H}_1(\xi) = \mathbf{H}_2(\xi) \text{ and } \Lambda \mathbf{k}_1(\xi) \leq \mathbf{k}_2(\xi)$$

- Convex if $\mathbf{H}_1 = \mathbf{H}_2$ or

$$\Lambda \mathbf{H}_1 = \mathbf{H}_2(\xi) \text{ and } \Lambda \mathbf{k}_1 \leq \mathbf{k}_2(\xi)$$

Decentralized RCI set: Final Remarks

- One can select a decentralized RCI set $\mathcal{R}_{\xi}^{\Delta}$ by selecting $\xi \in \Pi$, (one-time or repeated)

Decentralized RCI set: Final Remarks

- One can select a decentralized RCI set $\mathcal{R}_{\xi}^{\Delta}$ by selecting $\xi \in \Pi$, (one-time or repeated)
- Selection of $\xi \in \Pi$: optimization over a polyhedral set

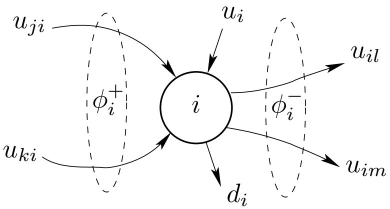
Decentralized RCI set: Final Remarks

- One can select a decentralized RCI set $\mathcal{R}_{\xi}^{\Delta}$ by selecting $\xi \in \Pi$, (one-time or repeated)
- Selection of $\xi \in \Pi$: optimization over a polyhedral set
- For all $\mathcal{S}_i$ much smaller than $|\mathcal{N}|$ decentralized robust invariance might be more manageable than centralized

An Example

Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network

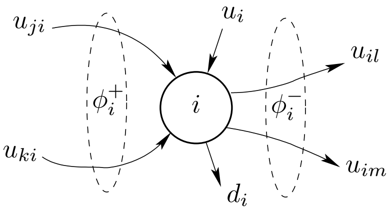


  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)$=0
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

An Example

Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network

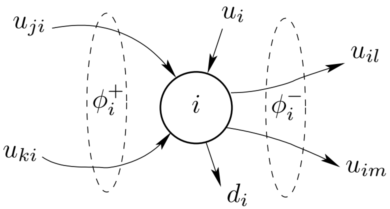


  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)=0$
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

An Example

Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network

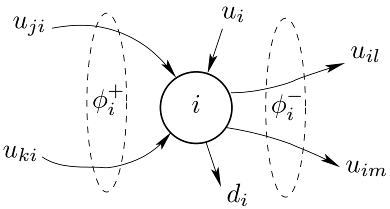


  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)$=0
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

An Example

Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network

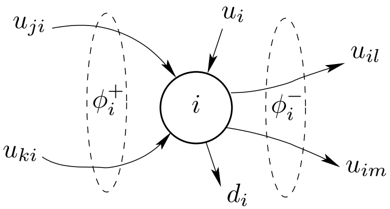


  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)$=0
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

An Example

Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network

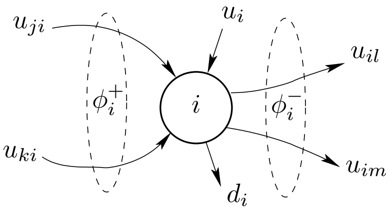


  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)$=0
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

An Example

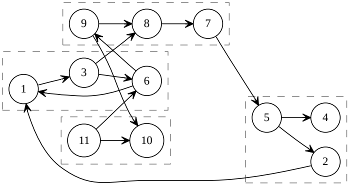
Collaboration with Ford Motor Company

- **Battery blocks in electric vehicles**
  - Multiple battery cells combined into a network



  - Safety critical constraints on local charge levels ($\underline{x}(\{i\}) = 0$, $\overline{x}(\{i\}) = 1$)
  - $\underline{u}(i)$ passive discharging, $\overline{u}(i)$=0
  - Limited charge transfer $\overline{c}(\mathcal{S}, \mathcal{S}')$
  - Unknown but bounded charging and discharging events $\underline{d}(\mathcal{S})$, $\overline{d}(\mathcal{S})$
  - Global constraints: balanced state–of–charge

Numerical Example



	Node i										
	1	2	3	4	5	6	7	8	9	10	11
$\overline{u}(i)$	1	1	2.5	1.5	3.5	1	4	1	2.5	1	5
$\overline{d}(i)$	3	2.5	2	2	2.5	1.5	3	2	2	2.5	1
$\overline{x}(i)$	5	5	5	5	5	5	5	5	5	5	5

ORIGINAL NETWORK AND CONSTRAINTS

- coupling constraints

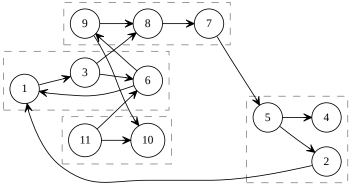
$$x_1 + x_3 + x_6 \leq 10, \quad x_2 + x_4 + x_5 \leq 10, \quad x_7 + x_8 + x_9 \geq 2$$

- only the parameters $\underline{\pi}_{ij}, \overline{\pi}_{ij}$ (bounds on transfers u_{ij}) optimized by minimizing the cost:

$$\sum_{(i,j)\in\mathcal{E}} |\underline{\pi}_{ij}| + |\overline{\pi}_{ij}|,$$

with constraints: $-2 \leq \underline{\pi}_{ij} \leq \overline{\pi}_{ij} \leq 2$

Numerical Example



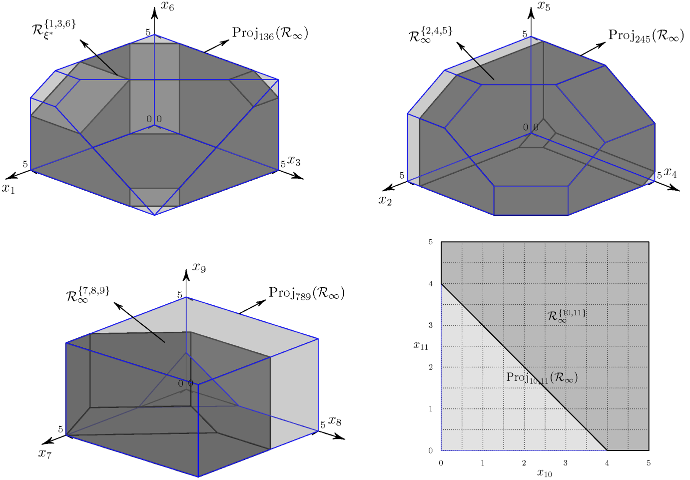
	Edges (i,j)						
	(1,3)	(2,1)	(3,6)	(3,8)	(5,2)	(5,4)	(6,1)
$\underline{u}_{ij}$	0	0	0	0	0	0	0
$\overline{u}_{ij}$	1	1	0.5	1	1.5	0.5	2
	Edges (i,j)						
	(6,9)	(7,5)	(8,7)	(9,8)	(9,10)	(11,6)	(11,10)
$\underline{u}_{ij}$	0	1	0	0	-0.5	2	-2
$\overline{u}_{ij}$	2	1	1	1	2	2	2

OPTIMIZED FLOW CAPACITIES FOR WHICH $\mathcal{R}^{\Delta} \neq \emptyset$

- centralized control: set of admissible flows $\mathcal{F}$ defined by 360 (non–redundant) inequalities
- decentralized control: sets $\tilde{\mathcal{F}}^{\mathcal{S}_1}(\xi^*)$, $\tilde{\mathcal{F}}^{\mathcal{S}_2}(\xi^*)$, $\tilde{\mathcal{F}}^{\mathcal{S}_3}(\xi^*)$ and $\tilde{\mathcal{F}}^{\mathcal{S}_4}(\xi^*)$ for the selected parameter vector $\xi^* \in \Pi$ defined with 14,12,12 and 6 inequalities respectively

Numerical Example

Resulting sets



Examples

- **Networked of UV**
  - Local: Input actuators, bank angles, flight envelop
  - Network: collision avoidance, max communication range
- **Lithium-Ion Batteries. Node: Cells**

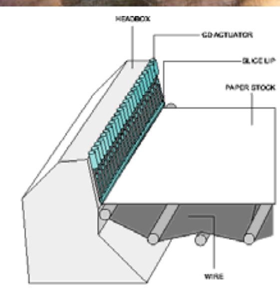
  Local: min and max charge level

  Network: charging and discharging capacity

- **Paper Machine. Node: Actuators**

  Local: min and max position

  Network: max relative distance between neighbors



- **Building Cooling and Heating Systems**

  Local: Min and Max temperature of thermal zones

  Network: Max cooling and heating capacity

Important Issues in Model Predictive Control

1. **Feasibility and Stability**
   Optimization problem may be infeasible at some future time step

   Even assuming perfect model, no disturbances:

   predicted open-loop trajectories
   $\neq$
   closed-loop trajectories

2. **Performance**
   What is achieved by repeatedly minimizing $\sum_{k=t}^{t+N-1} l(x_k, u_k)$

3. **Computation**
   Can we guarantee real-time implementation on embedded platforms

Important Issues in Model Predictive Control

1. **Feasibility and Stability**
   Optimization problem may be infeasible at some future time step

   Even assuming perfect model, no disturbances:

   predicted open-loop trajectories
   $\neq$
   closed-loop trajectories

2. **Performance**
   What is achieved by repeatedly minimizing

3. **Computation**
   Can we guarantee real-time implementation on embedded platforms

Computation

Control Hardware Platforms for Real-Time Implementation

Hi-End PC

4Ghz, 1 Terabyte

Automotive

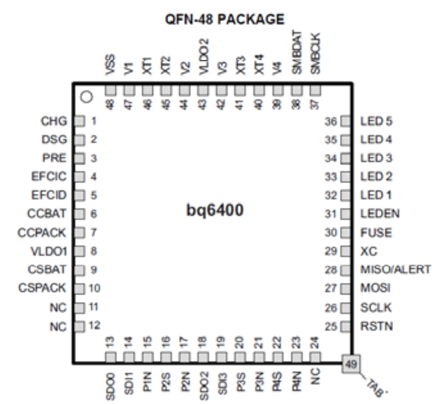
50Mhz, 2 Mbytes

Micro Robotics

16 Mhz, 128Kbytes

Zone Controller

12 Mhz, 512 KByte



Battery Management

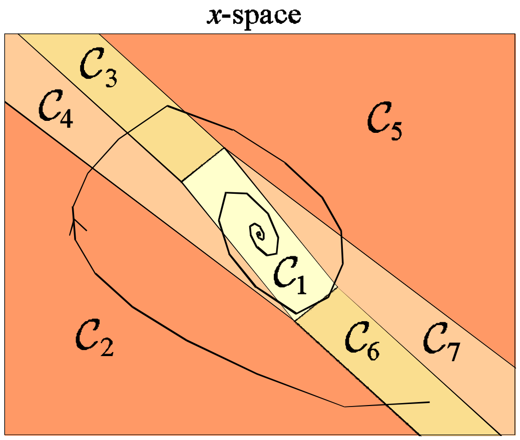
8 Mhz, 60 KByte

PieceWise Affine (Hybrid) MPC

Borrelli from 1999 to today

$$\min_{U} \sum_{k=0}^{N-1} \|x_k\|_p^Q + \|u_k\|_p^R$$

$$\text{subj. to} \begin{cases} x_{k+1} = A^i x_k + B^i u_k + c^i \\ \quad \text{if } [x_k, u_k] \in \mathcal{X}^i \\ x_k \in \mathcal{X}, \ u_k \in \mathcal{U} \\ \mathcal{X}^i, \ \mathcal{X}, \ \mathcal{U} \text{ polyhedra} \end{cases}$$



$$x_k \in \mathbb{R}^n \times \{0,1\}^{n_b}, \ u_k \in \mathbb{R}^m \times \{0,1\}^{m_b}, \ U = \{u_0, u_1, \ldots, u_{N-1}\}, \ p = 1, 2, \infty$$

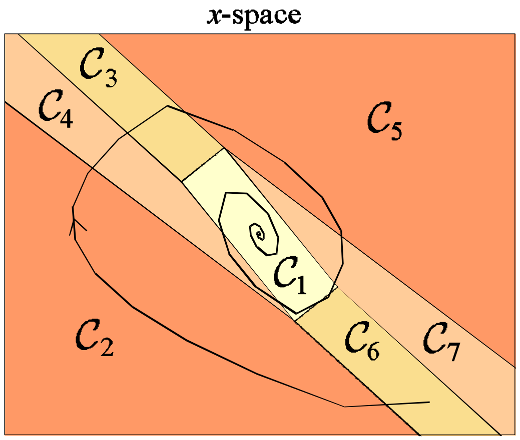
- Understanding solution structures and properties
- Solution computational methods and tools

PieceWise Affine (Hybrid) MPC

Borrelli from 1999 to today

$$\min_U \sum_{k=0}^{N-1} \|x_k\|_p^Q + \|u_k\|_p^R$$

$$\text{subj. to} \quad \begin{cases} x_{k+1} = A^i x_k + B^i u_k + c^i \\ \qquad \text{if } [x_k, u_k] \in \mathcal{X}^i \\ x_k \in \mathcal{X},\ u_k \in \mathcal{U} \\ \mathcal{X}^i,\ \mathcal{X},\ \mathcal{U} \text{ polyhedra} \end{cases}$$



$$\begin{array}{ll} \min_\epsilon & \epsilon^T H_\epsilon \epsilon + \epsilon^T H_x x_0 + f_\epsilon^T \epsilon + f_x^T x_0 \\ \text{subj. to} & G\epsilon \leq w + F x_0 \end{array}$$

On line solution of a

Mixed-Integer Linear/Quadratic Program

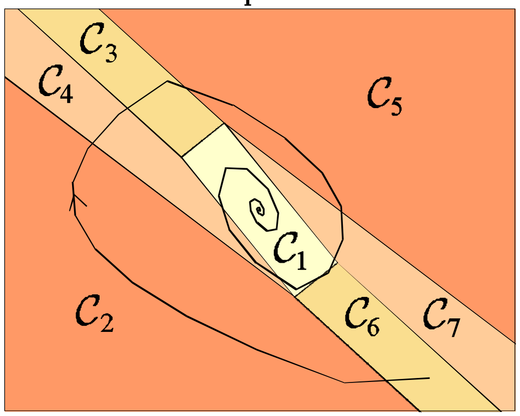
PieceWise Affine (Hybrid) MPC

Borrelli from 1999 to today

$$\min_{U} \sum_{k=0}^{N-1} \|x_k\|_p^Q + \|u_k\|_p^R$$

subj. to

$$\begin{cases} x_{k+1} = A^i x_k + B^i u_k + c^i \\ \quad \text{if } [x_k, u_k] \in \mathcal{X}^i \\ x_k \in \mathcal{X},\ u_k \in \mathcal{U} \\ \mathcal{X}^i,\ \mathcal{X},\ \mathcal{U} \text{ polyhedra} \end{cases}$$



$$x_k \in \mathbb{R}^n \times \{0,1\}^{n_b},\ u_k \in \mathbb{R}^m \times \{0,1\}^{m_b},\ U = \{u_0, u_1, \ldots, u_{N-1}\},\ p = 1, 2, \infty$$

- **Understanding solution structures and properties**
- Solution computational methods and tools

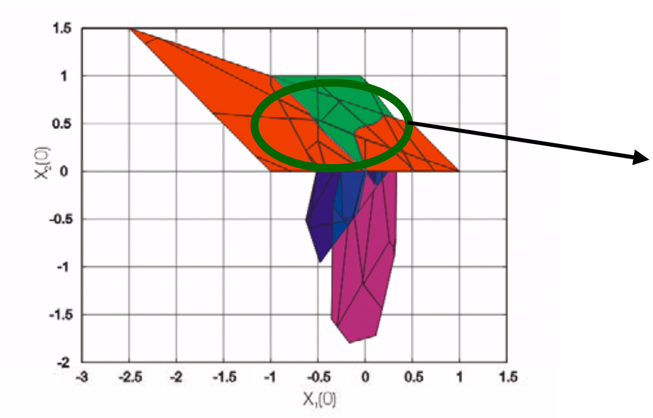
Characterization of the Solution (p=1,2,∞)

Borrelli et al, ACC, 2000
Borrelli et al, TAC, 2003
Borrelli et al, AUTOMATICA, 2005

The solution to the optimal control problem is a time varying PWA state feedback control law of the form

$$u_k^*(x_k) = \left\{ \begin{array}{lll} F_k^1 x_k + G_k^1 & \text{if} & x_k \in CR_k^1 \\ \vdots & \vdots & \vdots \\ F_k^R x_k + G_k^R & \text{if} & x_k \in CR_k^R \end{array} \right\}$$

$\{CR_k^i\}_{i=1}^R$ is a partition of the set of feasible states $\mathcal{X}_k$.



- **p=1, p=∞:**

$$CR_k^i = \{x \ : \ M_k^i x \leq K_k^i\}$$

- **p=2:**

$$CR_k^i = \{x \ : \ x' L_k^i(j) x + M_k^i(j) x \leq K_k^i(j)\}$$

Very useful result for MPC implementation

Characterization of the Solution (p=1,2,∞)

via Multiparametric Programming

Consider the mixed-integer mathematical program

$$\begin{array}{rl} \min_{\epsilon} & \epsilon^T H_\epsilon \epsilon + \epsilon^T H_x x_0 + f_\epsilon^T \epsilon + f_x^T x_0 \\ \text{subj. to} & G\epsilon \le w + F x_0 \end{array}$$

ε is the optimization variable, x_0 a parameter that affects the solution ε^*

Compute $\varepsilon^*(x_0)$ *for* $x_0 \in \mathcal{X}$

Non-Linear Parametric Optimization, Properties of point-to-set mappings

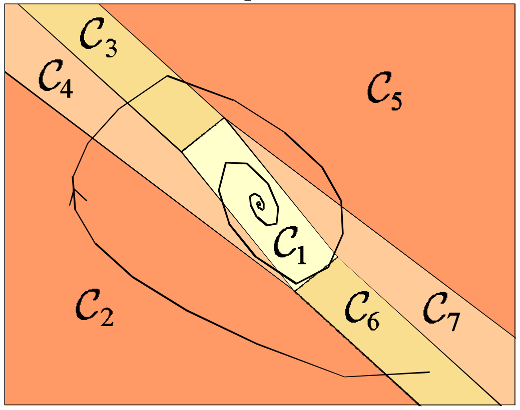
Borrelli, B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, Klatte, Sontag, Dua, Pistikopoluous, Bemporad, Morari, …..

PieceWise Affine (Hybrid) MPC

Borrelli from 1999 to today

$$\min_{U} \sum_{k=0}^{N-1} \|x_k\|_p^Q + \|u_k\|_p^R$$

$$\text{subj. to} \begin{cases} x_{k+1} = A^i x_k + B^i u_k + c^i \\ \quad \text{if } [x_k, u_k] \in \mathcal{X}^i \\ x_k \in \mathcal{X},\ u_k \in \mathcal{U} \\ \mathcal{X}^i,\ \mathcal{X},\ \mathcal{U} \text{ polyhedra} \end{cases}$$

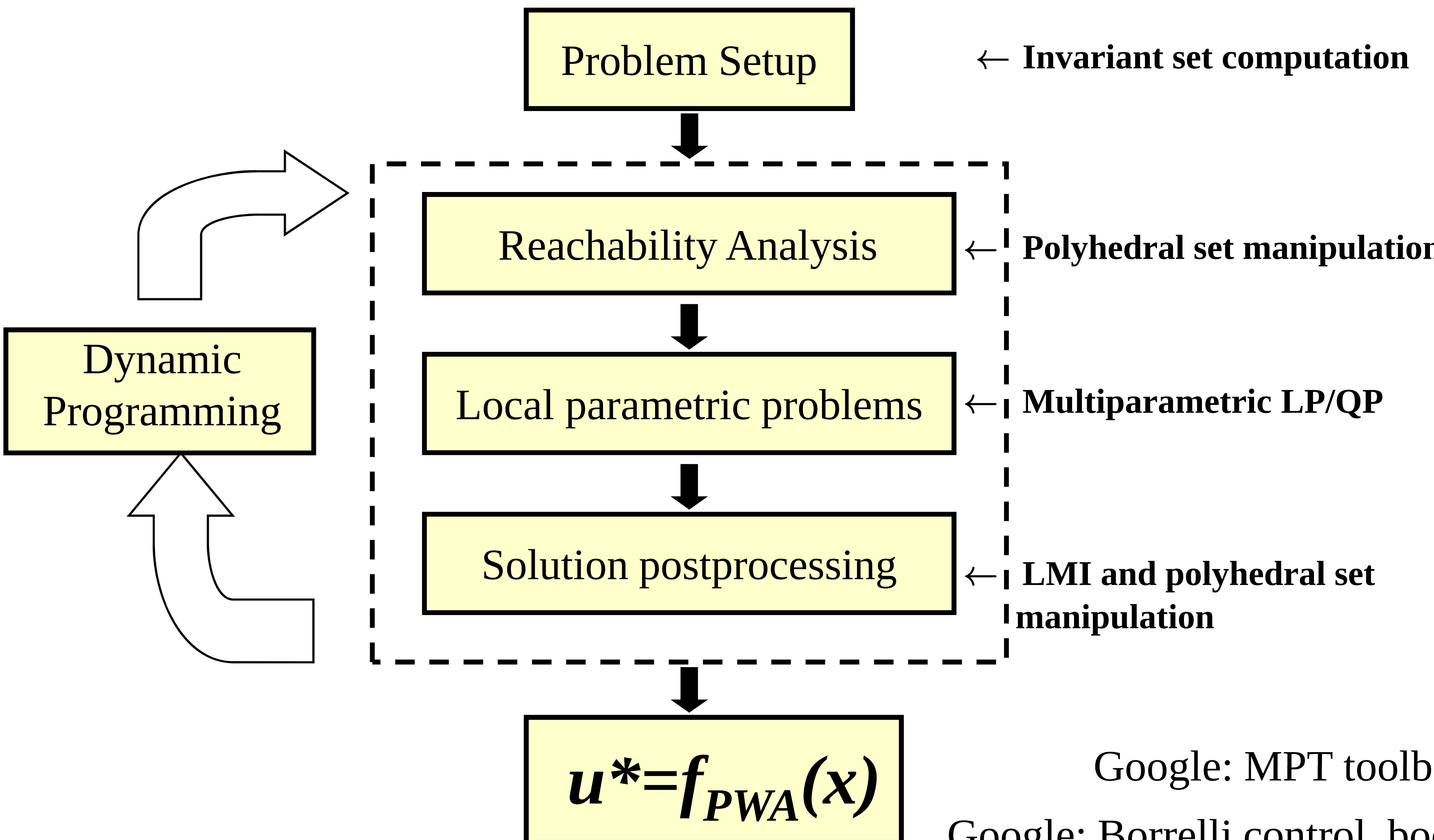


$$x_k \in \mathbb{R}^n \times \{0,1\}^{n_b},\ u_k \in \mathbb{R}^m \times \{0,1\}^{m_b},\ U = \{u_0, u_1, \ldots, u_{N-1}\},\ p = 1, 2, \infty$$

- Understanding solution structures and properties
- **Solution computational methods and tools**

Computational Flow

Borrelli et al, JOTA, 2003
Borrelli et al, AUTOMATICA, 2006
Baotic, Borrelli et al, SICON, 2007



Google: MPT toolbox

Google: Borrelli control book

Summary: Explicit vs Online

x →

$$u(x) = \begin{cases} F_1 x + G_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \vdots \\ F_N x + G_N & \text{if } H_N x \leq K_N \end{cases}$$

OR

Online Solution of a (mixed-integer) Linear/Quadratic Program

→ $u^*(x)$

- Explicit solution is ALWAYS faster than an active set solver
- Algorithms HAVE to be compared in Memory-CPU plane
- Alternative Algorithms exists with different trade-off

F, **Borrelli, J. Pekar, M. Baotic, G. Stewart On The Computation Of Model Predictive Control Laws. Automatica 2010**

Online vs Explicit

$$\begin{array}{ll} \min\limits_{U} & \frac{1}{2}U'HU + g(x)'U \\ \text{subj. to} & GU \leq b(x) \end{array}$$

$$G \in \mathbb{R}^{m\times n},\ b(x) \in \mathbb{R}^{m},\ g(x) \in \mathbb{R}^{n},\ b(x) = b^r + B^x x \text{ and } g(x) = F'x$$

$$\begin{array}{ll} \min\limits_{U} & \frac{1}{2}U'HU + g(x)'U \\ \text{subj. to} & G_{\mathcal{A}}U = b_{\mathcal{A}}(x) \end{array}$$

$$\begin{array}{l} HU + g(x) + G'_{\mathcal{A}}\lambda = 0 \\ G_{\mathcal{A}}U = b_{\mathcal{A}}(x). \end{array} \Rightarrow \begin{bmatrix} H & G'_{\mathcal{A}} \\ G_{\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} -g(x) \\ b_{\mathcal{A}}(x) \end{bmatrix} \Rightarrow \begin{array}{l} U^* = -Lg(x) + Tb_{\mathcal{A}}(x) \\ \lambda^* = -T'g(x) + Sb_{\mathcal{A}}(x) \end{array}$$

$$\begin{array}{l} L = H^{-1} - H^{-1}G'_{\mathcal{A}}(G_{\mathcal{A}}H^{-1}G'_{\mathcal{A}})^{-1}G_{\mathcal{A}}H^{-1} \\ T = H^{-1}G'_{\mathcal{A}}(G_{\mathcal{A}}H^{-1}G'_{\mathcal{A}})^{-1} \\ S = -(G_{\mathcal{A}}H^{-1}G'_{\mathcal{A}})^{-1} \end{array}$$

Online vs Explicit

$$U^* = -Lg(x) + Tb_{\mathcal{A}}(x)$$
$$\lambda^* = -T'g(x) + Sb_{\mathcal{A}}(x)$$

Dual Feasibility

$d^* = \min_{i \in \{1,\ldots,|\mathcal{A}|\}} \lambda_i^*$
If $d^* < 0$ then $\mathcal{A}^+ = \mathcal{A} \setminus \mathcal{A}(p)$
where p is the argmin

Primal Feasibility

$f^* = \min_{i \in \mathcal{I} \setminus \mathcal{A}} b_i(x) - G_i U^*$
If $f^* < 0$ then $\mathcal{A}^+ = \mathcal{A} \cup p$
where p is the argmin

$$U^* = (TB_{\mathcal{A}}^x - LF')x + Tb_{\mathcal{A}}^r = F_{\mathcal{A}}x + c_{\mathcal{A}}$$
$$\lambda^* = (SB_{\mathcal{A}}^x - T'F')x + Sb_{\mathcal{A}}^r = F_{\mathcal{A}}^d x + c_{\mathcal{A}}^d$$

$$\mathcal{P}_p = \{x : \quad G_{\mathcal{I}\setminus\mathcal{A}} U^*(x) < b_{\mathcal{I}\setminus\mathcal{A}}(x)\} \quad H_p x \le K_p\}$$

$$\mathcal{P}_d = \{x : \quad \lambda^*(x) \ge 0\} \quad H_d x \le K_d\}$$

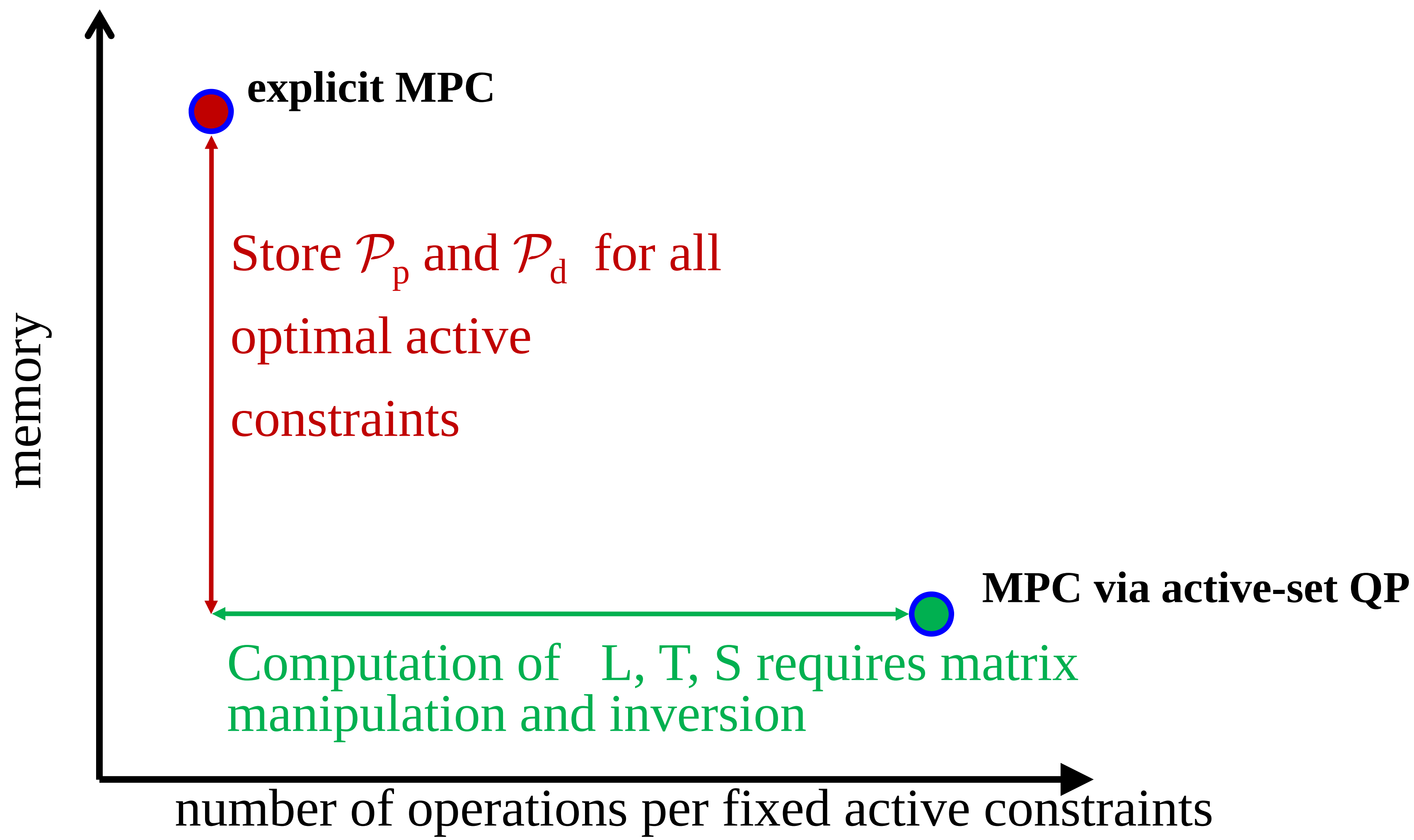
Dual Feasibility

$d^* = \min_i -(H_d)_i x + (K_d)_{(i)}$
If $d^* < 0$ goto neighbouring region p
where p is the argmin

Primal Feasibility

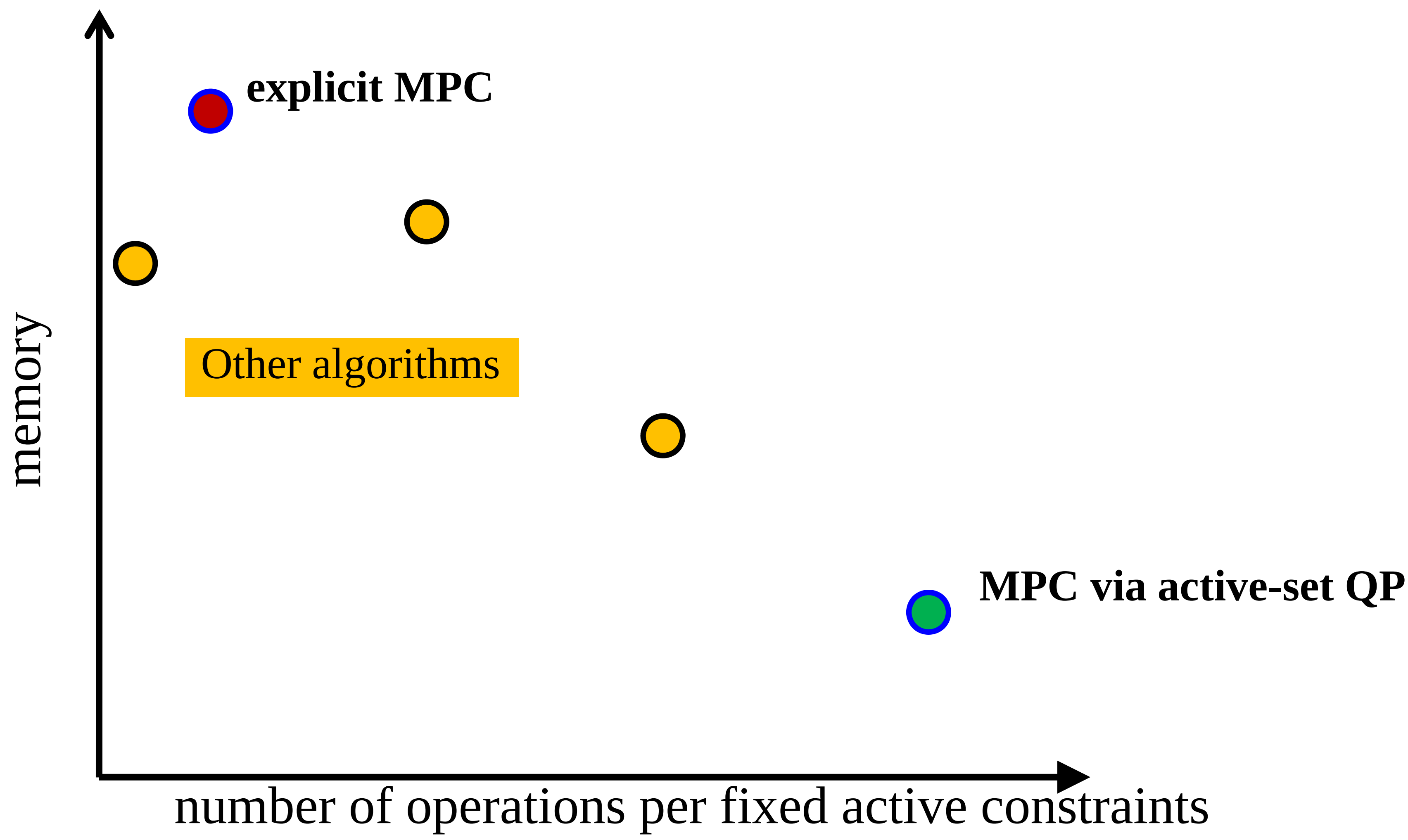
$f^* = \min_i -(H_p)_{(i)} x + (K_p)_{(i)}$
If $f^* < 0$ goto neighbouring region p
where p is the argmin

Trading off CPU Time and Memory in MPC



Trading off CPU Time and Memory in MPC

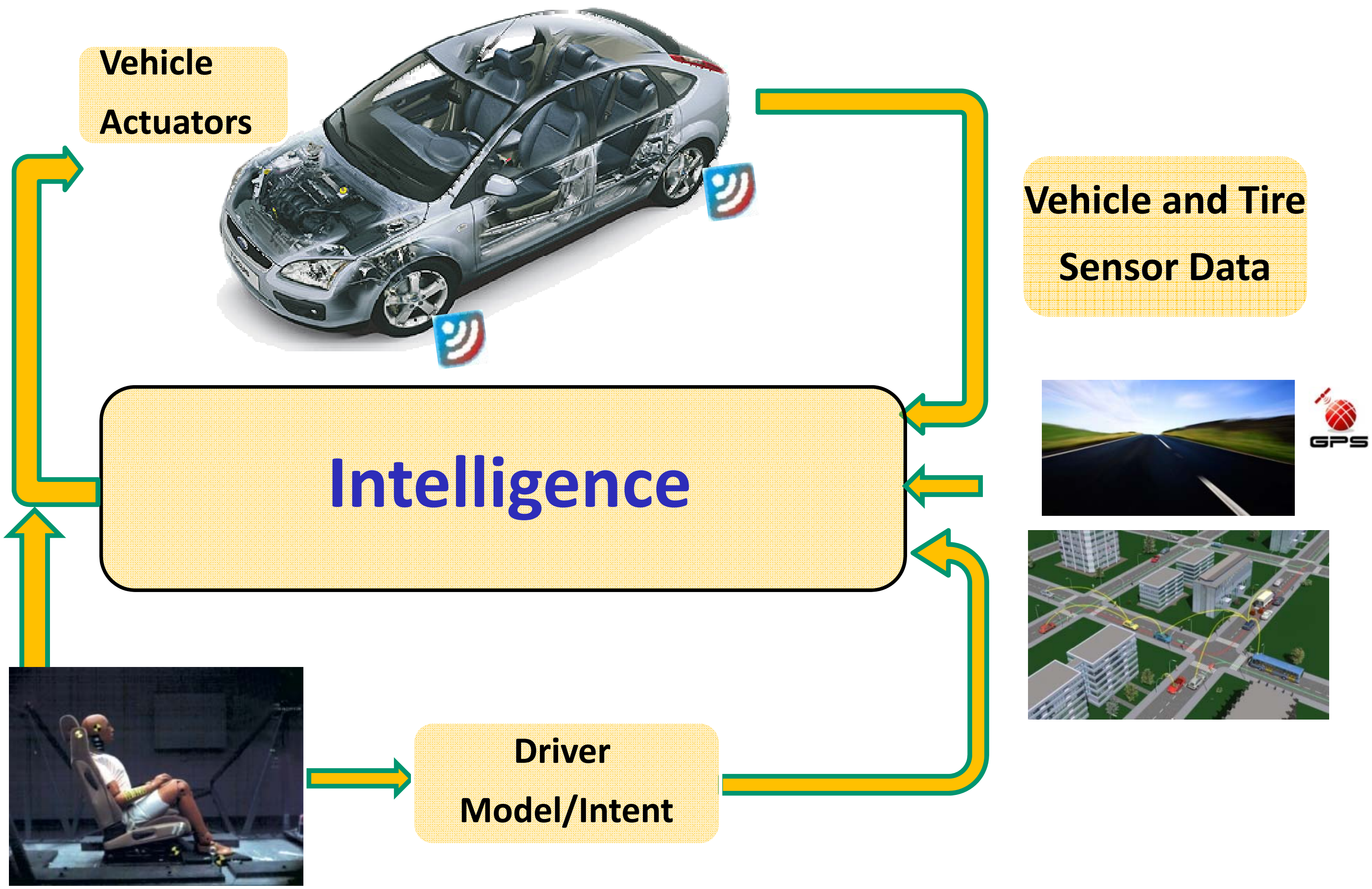
[Borrelli, Baotic, Pekar, Stewart, ECC09]



Spice this up below with Videos and reach computation of NSF\(Put equations)

who driver is the bounded disturbance....

Automotive Cyber-Physical System



Safety | Comfort | Efficiency

Control Hardware Platforms for Real-Time Implementation

Hi-End PC

4Ghz, 1 Terabyte

Automotive

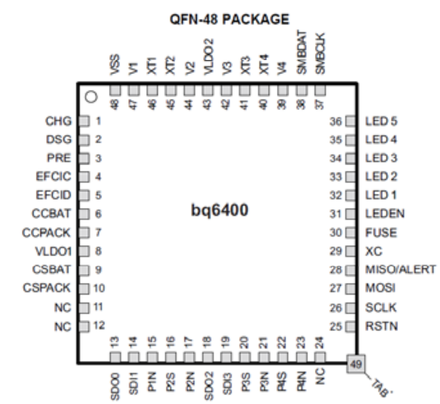
50Mhz, 2 Mbytes

Micro Robotics

16 Mhz, 128Kbytes

Zone Controller

12 Mhz, 512 KByte

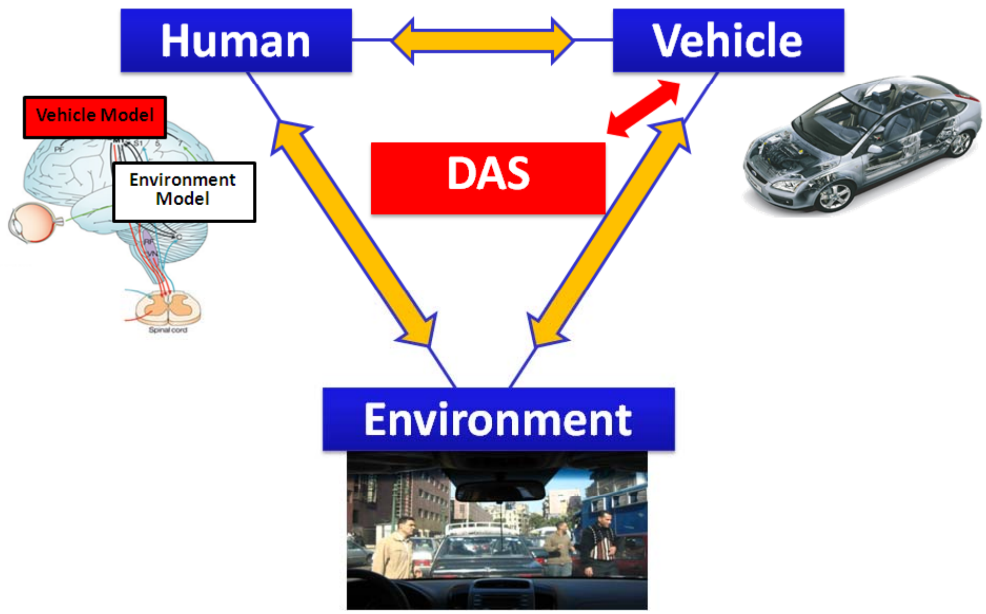


Battery Management

8 Mhz, 60 KByte

CPS Main Issues

- Complexity/Compositionality
- No Guarantees/Heuristic Tunings
- Human Motion/perception/cognition ignored



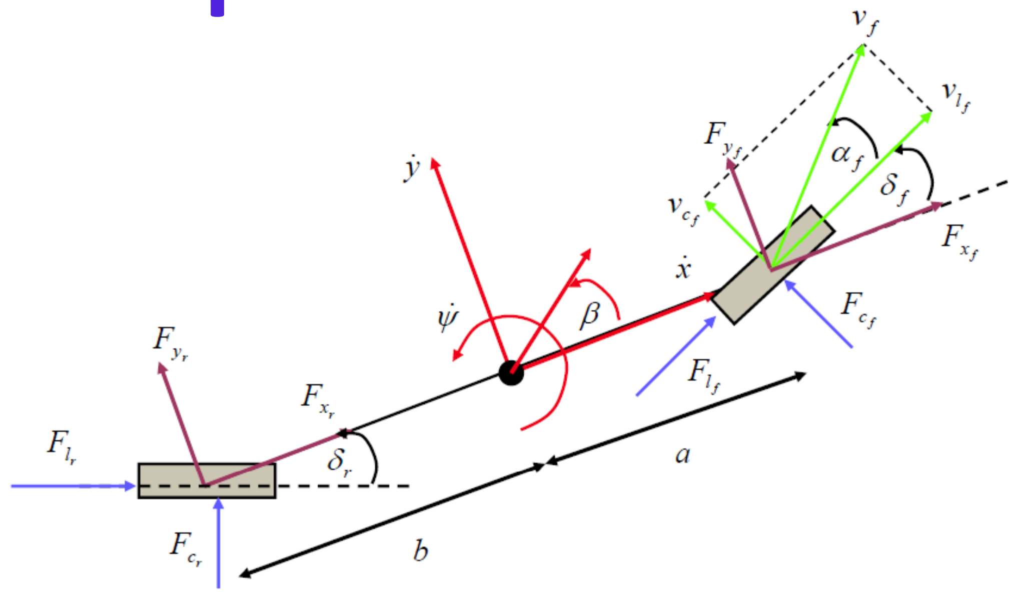
2D Example

States: $\dot{y}, \dot{\psi}$

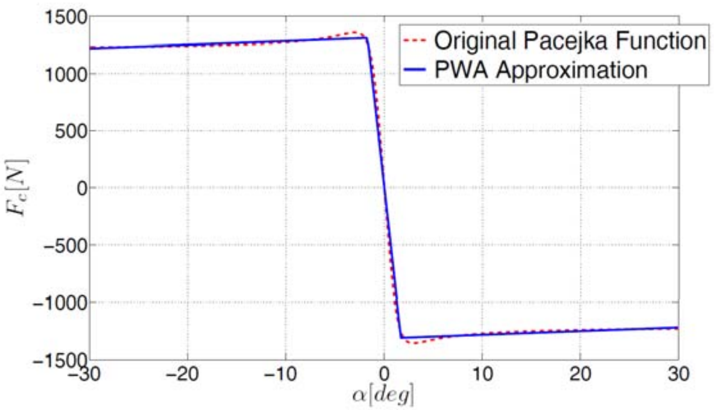
Inputs: M

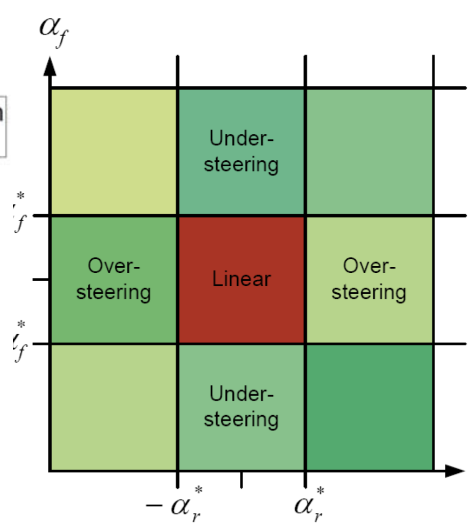
Disturbance: δ_f

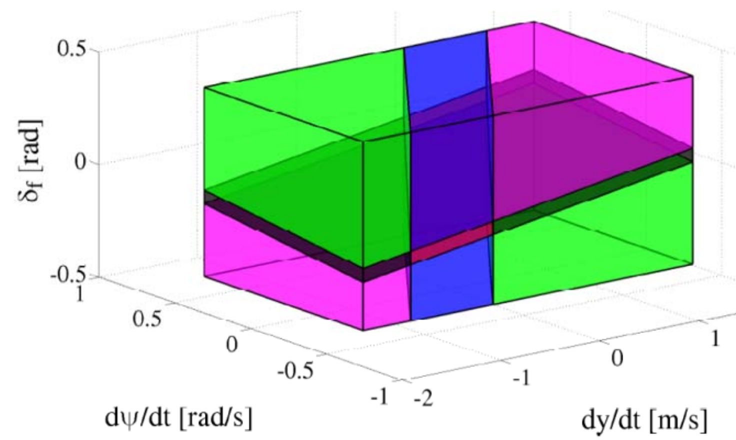
Assume constant: μ, V_x



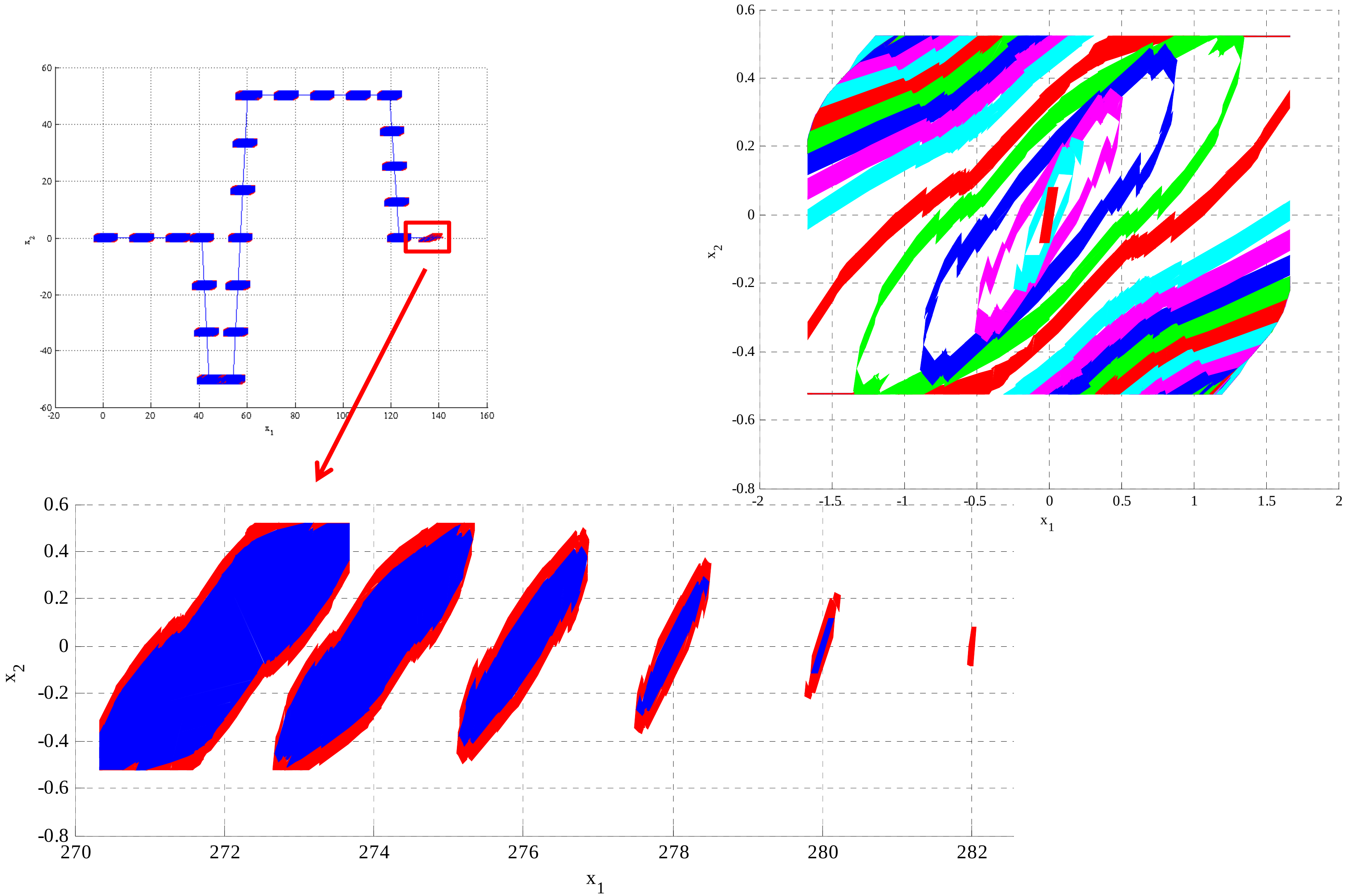
PWA approximation of Pacejka tire model



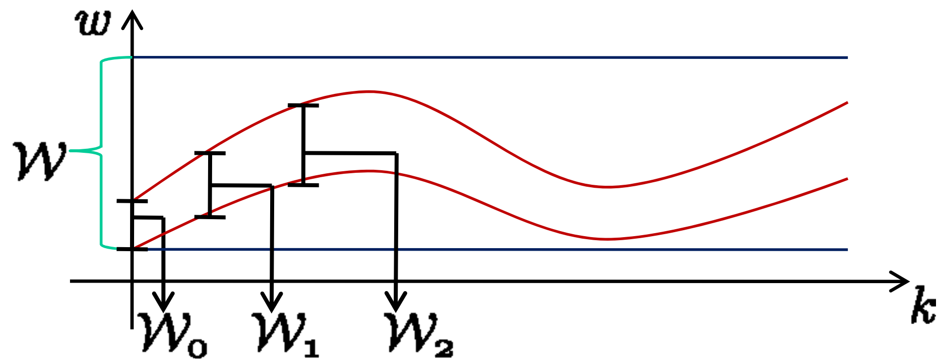




2D Example – Robust Set Computation



Simple Autonomy Concept



At time j

- Driver intent w_j
- Among all possible actuations $\overline{u}_j$ choose the one that solves

$$min_{u_j} \; \|u_j - w_j\| \; subj.to. \; x_{j+1} \in \mathcal{X}_{j+1}$$

Adaptive and Predictive Autonomy

From 2-D to 12-D Example
Experimental results @ 72 Kph on Ice

LAPACK

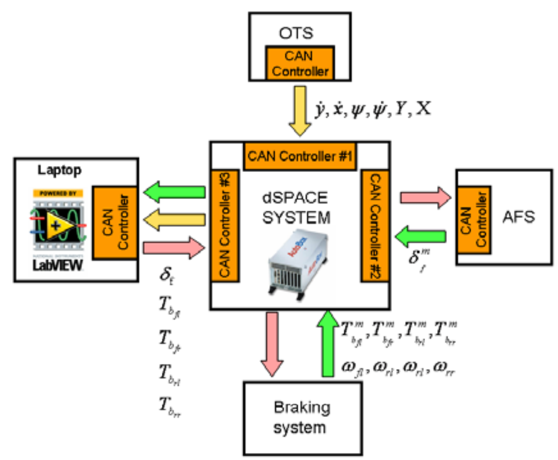
dSPACE

NAG

COIN-OR: Computational Infrastructure for Operations Research

LabVIEW

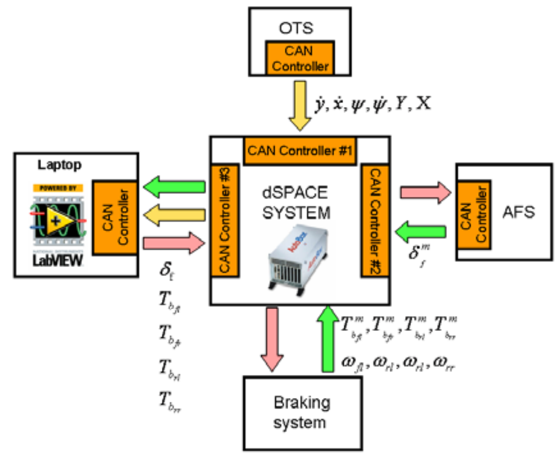
Stanford Business Software Inc.
DISTRIBUTION OF SOL/UCSD OPTIMIZATION SOFTWARE



FORD RESEARCH LABORATORY

Ford

Experimental results @ 72 Kph on Ice



Nonlinear MPC not real-time feasible in standard prototyping hardware, experiments only up to 25kph

Explicit Solution exceeds Hardware Memory.

Enabler for Real-Time Implementation: Modified LTV Predictive Control

Conclusions/ Research Activities

1. Model Predictive Control
2. Invariant Computation for Switched Linear systems and Linear Large Scale systems
3. Real-time Computation
4. Enhancing the safety of Autonomous Systems

www.mpc.berkeley.edu

Hybrid Predictive Control
Real Time Implementation

(MI) Linear/Quadratic Program

Online solution of an optimization problem

- Difficulty to verify
- Source code might be required

Explicit Solution

Online evaluation of a look-up table

- Easy to Implement/verify
- Number of regions can explode

Value of Predictions

Explicit MPC Enabler for

Real-Time Implementation on “Small&Fast” Processes

Hierarchical and Distributed MPC