EVENT-DRIVEN CONTROL AND OPTIMIZATION:
WHERE LESS IS OFTEN MORE...

C. G. Cassandras
Division of Systems Engineering
and Dept. of Electrical and Computer Engineering
and Center for Information and Systems Engineering
Boston University
OUTLINE

- Reasons for EVENT-DRIVEN Control and Optimization
- EVENT-DRIVEN Control in Distributed Systems
- EVENT-DRIVEN Control in Managing Uncertainty
- EVENT-DRIVEN Sensitivity Analysis
EVENT-DRIVEN CONTROL: Act only when needed (or on TIMEOUT) - not based on a clock
Many systems are naturally **Discrete Event Systems (DES)** (e.g., Internet) → *all* state transitions are event-driven

Most of the rest are **Hybrid Systems (HS)** → *some* state transitions are event-driven

Many systems are **distributed** → components interact asynchronously (through events)

Time-driven sampling inherently inefficient ("open loop" sampling)
Many systems are stochastic → actions needed in response to random events

Event-driven methods provide significant advantages in computation and estimation quality

System performance is often more sensitive to event-driven components than to time-driven components

Many systems are wirelessly networked → energy constrained → time-driven communication consumes significant energy UNNECESSARILY!

Christos G. Cassandras
CODES Lab. - Boston University
TIME-DRIVEN v EVENT-DRIVEN SYSTEMS

**TIME-DRIVEN SYSTEM**

**STATES**

\[ x(t) \]

**STATE SPACE:**
\[ X = \mathbb{R} \]

**DYNAMICS:**
\[ \dot{x} = f(x, t) \]

**EVENT-DRIVEN SYSTEM**

**STATES**

\[ s_1, s_2, s_3, s_4 \]

**STATE SPACE:**
\[ X = \{ s_1, s_2, s_3, s_4 \} \]

**DYNAMICS:**
\[ x' = f(x, e) \]
SYNCHRONOUS v ASYNCHRONOUS BEHAVIOR

- Indistinguishable events
- Wasted clock ticks
- More wasted clock ticks
- Even more wasted clock ticks

Christos G. Cassandras
CODES Lab. - Boston University
SYNCHRONOUS v ASYNCHRONOUS COMPUTATION

- Time-driven (synchronous) implementation:
  - Sum repeatedly evaluated unnecessarily
  - When evaluation is actually needed, it is done at the wrong times!

SYNCHRONOUS v ASYNCHRONOUS COMPUTATION

Christos G. Cassandras
CODES Lab. - Boston University
EVENT-DRIVEN CONTROL IN DISTRIBUTED SYSTEMS
MOTIVATIONAL PROBLEM: **COVERAGE CONTROL**

Deploy sensors to maximize “event” detection probability

- unknown event locations
- event sources may be mobile
- sensors may be mobile

Perceived event density (data sources) over given region (mission space)

*Meguerdichian et al*, IEEE INFOCOM, 2001

*Cortes et al*, IEEE Trans. on Robotics and Automation, 2004

*Cassandras and Li*, Eur. J. of Control, 2005

*Ganguli et al*, American Control Conf., 2006

*Hussein and Stipanovic*, American Control Conf., 2007

*Hokayem et al*, American Control Conf., 2007
OPTIMAL COVERAGE IN A MAZE

http://codescolor.bu.edu/coverage

Zhong and Cassandras, 2008

Christos G. Cassandras

CODES Lab. - Boston University
COVERAGE: PROBLEM FORMULATION

- $N$ mobile sensors, each located at $s_i \in \mathbb{R}^2$
- Data source at $x$ emits signal with energy $E$
- Signal observed by sensor node $i$ (at $s_i$)

**SENSING MODEL:**

$$p_i(x, s_i) \equiv P[\text{Detected by } i \mid A(x), s_i]$$

($A(x) =$ data source emits at $x$)

- Sensing attenuation:

$$p_i(x, s_i) \text{ monotonically decreasing in } d_i(x) \equiv \|x - s_i\|$$
Joint detection prob. assuming sensor independence
(s = [s₁,...,sₙ] : node locations)

\[ P(x,s) = 1 - \prod_{i=1}^{N} [1 - p_i(x,s_i)] \]

OBJECTIVE: Determine locations s = [s₁,...,sₙ] to maximize total Detection Probability:

\[ \max_s \int_{\Omega} R(x) P(x,s) dx \]
DISTRIBUTED COOPERATIVE SCHEME

- Set

\[ H(s_1, \ldots, s_N) = \int_{\Omega} R(x) \prod_{i=1}^{N} \left( 1 - p_i(x) \right) dx \]

- Maximize \( H(s_1, \ldots, s_N) \) by forcing nodes to move using gradient information:

\[ \frac{\partial H}{\partial s_k} = \int_{\Omega} R(x) \prod_{i=1, i \neq k}^{N} \left( 1 - p_i(x) \right) \frac{\partial p_k(x)}{\partial d_k(x)} \frac{s_k - x}{d_k(x)} dx \]

\[ s_i^{k+1} = s_i^k + \beta_k \frac{\partial H}{\partial s_i^k} \]

Desired displacement = \( V \cdot \Delta t \)

Cassandras and Li, 2005
Zhong and Cassandras, 2011
… has to be autonomously evaluated by each node so as to determine how to move to next position:

\[
\frac{\partial H}{\partial s_k} = \int_{\Omega} R(x) \prod_{i=1, i \neq k}^{N} [1 - p_i(x)] \frac{\partial p_k(x)}{\partial d_k(x)} \frac{s_k - x}{d_k(x)} dx
\]

- Use truncated \( p_i(x) \) \( \Rightarrow \Omega \) replaced by node neighborhood
- Discretize \( p_i(x) \) using a local grid
$N$ system components
(processors, agents, vehicles, nodes),
one common objective:

\[ \min_{s_1, \ldots, s_N} H(s_1, \ldots, s_N) \]
\[ \text{s.t. constraints on each } s_i \]

\[ \min_{s_1} H(s_1, \ldots, s_N) \]
\[ \text{s.t. constraints on } s_1 \]

\[ \vdots \]

\[ \min_{s_N} H(s_1, \ldots, s_N) \]
\[ \text{s.t. constraints on } s_N \]
Controllable state \( s_i, i = 1, \ldots, n_i \)

\[
    s_i(k + 1) = s_i(k) + \alpha_i d_i(s(k))
\]

- **Step Size**
- **Update Direction**, usually
  \[
    d_i(s(k)) = -\nabla_i H(s(k))
  \]

\( i \) requires knowledge of all \( s_1, \ldots, s_N \)

Inter-node communication

\[
    \min_{s_i} H(s_1, \ldots, s_N) \\
    s.t. \text{ constraints on } s_i
\]
SYNCHRONIZED (TIME-DRIVEN) COOPERATION

Drawbacks:
- Excessive communication (critical in wireless settings!)
- Faster nodes have to wait for slower ones
- Clock synchronization infeasible
- Bandwidth limitations
- Security risks
Nodes not synchronized, delayed information used

Update frequency for each node is bounded + technical conditions

\[ s_i(k+1) = s_i(k) + \alpha_i d_i(s(k)) \]

converges

Bertsekas and Tsitsiklis, 1997
**UPDATE at** $i$ : locally determined, arbitrary (possibly periodic)

**COMMUNICATE from** $i$ : only when absolutely necessary
Node state at any time $t$: $x_i(t)$

Node state at $t_k$: $s_i(k)$

$\Rightarrow s_i(k) = x_i(t_k)$

AT UPDATE TIME $t_k$:

$s_j^i(k)$: node $j$ state estimated by node $i$

Estimate examples:

$\rightarrow s_j^i(k) = x_j(\tau^j(k))$  
Most recent value

$\rightarrow s_j^i(k) = x_j(\tau^j(k)) + \frac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_i \cdot d_j\left(x_j(\tau^j(k))\right)$  
Linear prediction
WHEN SHOULD A NODE COMMUNICATE?

AT ANY TIME $t$:

- $x_i^j(t)$: node $i$ state estimated by node $j$

- If node $i$ knows how $j$ estimates its state, then it can evaluate $x_i^j(t)$

- Node $i$ uses
  - its own true state, $x_i(t)$
  - the estimate that $j$ uses, $x_i^j(t)$

... and evaluates an ERROR FUNCTION $g(x_i(t), x_i^j(t))$

**Error Function examples:**

$$\|x_i(t) - x_i^j(t)\|_1, \quad \|x_i(t) - x_i^j(t)\|_2$$
WHEN SHOULD A NODE COMMUNICATE?

Compare ERROR FUNCTION \( g(x_i(t), x_j^i(t)) \) to THRESHOLD \( \delta_i \)

Node \( i \) communicates its state to node \( j \) only when it detects that its true state \( x_i(t) \) deviates from \( j\)'s estimate of it \( x_j^i(t) \) so that \( g(x_i(t), x_j^i(t)) \geq \delta_i \)

\( \Rightarrow \) \textbf{Event-Driven} Control
Asynchronous distributed state update process at each $i$:

\[ s_i(k + 1) = s_i(k) + \alpha \cdot d_i(s^i(k)) \]

Estimates of other nodes, evaluated by node $i$

\[ \delta_i(k) = \begin{cases} K_\delta \| d_i(s^i(k)) \| & \text{if } k \text{ sends update} \\ \delta_i(k - 1) & \text{otherwise} \end{cases} \]

**THEOREM:** Under certain conditions, there exist positive constants $\alpha$ and $K_\delta$ such that

\[ \lim_{k \to \infty} \nabla H(s(k)) = 0 \]

**INTERPRETATION:**

*Event-driven cooperation achievable with minimal communication requirements $\Rightarrow$ energy savings*

Zhong and Cassandras, IEEE TAC, 2010
Error function trajectory with NO DELAY

Red curve: $g(x_i, x_i')$
Black curve: $g(x_i, \tilde{x}_i')$

DELAY

COONVERGENCE WHEN DELAYS ARE PRESENT
Add a boundedness assumption:

**ASSUMPTION:** There exists a non-negative integer $D$ such that if a message is sent before $t_{k-D}$ from node $i$ to node $j$, it will be received before $t_k$.

**INTERPRETATION:** at most $D$ state update events can occur between a node sending a message and all destination nodes receiving this message.

**THEOREM:** Under certain conditions, there exist positive constants $\alpha$ and $K_\delta$ such that

$$
\lim_{{k \to \infty}} \nabla H(s(k)) = 0
$$

**NOTE:** The requirements on $\alpha$ and $K_\delta$ depend on $D$ and they are tighter.

---

Zhong and Cassandras, IEEE TAC, 2010
SYNCHRONOUS v ASYNCHRONOUS
OPTIMAL COVERAGE PERFORMANCE

SYNCHRONOUS v ASYNCHRONOUS:
No. of communication events for a deployment problem with obstacles

SYNCHRONOUS v ASYNCHRONOUS:
Achieving optimality in a problem with obstacles

Energy savings + Extended lifetime

Christos G. Cassandras
CODES Lab. - Boston University
DEMO: OPTIMAL DISTRIBUTED DEPLOYMENT WITH OBSTACLES – SIMULATED AND REAL

Christos G. Cassandras

CODES Lab. - Boston University
Important to note:
There is no external control causing this behavior. Algorithm includes tracking functionality automatically.
EVENT-DRIVEN CONTROL IN MANAGING UNCERTAINTY
UNCERTAINTY: CONTRAST TWO APPROACHES

**ESTIMATE-AND-PLAN**
- Decisions planned ahead
- Need accurate stochastic models
- Curse of dimensionality

- Dynamic Programming (DP)
- Markov Decision Processes (MDP)

**HEDGE-AND-REACT**
- Delay decisions until last possible instant
- No (detailed) stochastic model
- Simpler opt. problems

- Receding Horizon Control (RHC)
- Model Predictive Control (MPC)

Christos G. Cassandras
CODES Lab. - Boston University
UNCERTAINTY: CONTRAST TWO APPROACHES

ESTIMATE-AND-PLAN VS HEDGE-AND-REACT

Christos G. Cassandras

CODES Lab. - Boston University
TIME-DRIVEN: \[ \Delta \text{ must be small,}
\text{Computationally intensive} \]

EVENT-DRIVEN: \[ \text{Computational intensity}
\text{depends on event frequency} \]
COOPERATIVE RECEDING HORIZON (CRH) CONTROL: MAIN IDEA

- Do not attempt to assign nodes to targets
- Cooperatively steer nodes towards “high expected reward” regions
- Repeat process when event occurs
- Worry about final node-target assignment at the last possible instant

Turns out nodes converge to targets on their own!

Solve optimization problem by selecting all $u_i$ to maximize total expected rewards over $H$
MAIN IDEA IN CRH APPROACH:

- Replace complex *Discrete Stochastic Optimization* problem by a sequence of simpler *Continuous Optimization* problems
- Solve each new problem whenever a PREDEFINED EVENT occurs (e.g., some node gets to some target)
- ... or when a RANDOM EVENT (or a TIMEOUT) occurs

But how do we guarantee that nodes ultimately head for the desired DISCRETE TARGET POINTS?
**STABILITY ANALYSIS**

**TARGETS:** $y_i$  
**NODES:** $x_j$  
**DISTANCE:** $d_{ij}$

**DEFINITION:** Node trajectory $x(t) = [x_1(t), \ldots, x_M(t)]$ generated by a controller is **stationary**, if there exists some $t_v < \infty$, such that $\|x_j(t_v) - y_i\| \leq s_i$ for some $i = 1, \ldots, N$, $j = 1, \ldots, M$.

**QUESTION:**  
*Under what conditions is a CRH-generated trajectory stationary?*
Local minima of objective function \( J(x) : x^l = (x^l_1, \ldots, x^l_M) \in \mathbb{R}^{2M}, \ l = 1, \ldots, L \)

Vector of node positions at \( k \)th iteration of CRH controller: \( x_k \)

**Theorem:** Suppose \( H_k = \min d_{ij}(t_k) \).
If, for all \( l = 1, \ldots, L \), \( x^l_j = y_i \) for some \( i = 1, \ldots, N, j = 1, \ldots, M \), then \( J(x_k) - J(x_{k+1}) > b \) \( (b > 0 \text{ is a constant}) \).

*If all local minima coincide with targets, the CRH-generated trajectory is stationary*
MAIN STABILITY RESULT

QUESTION:

When do all local minima coincide with target points?

1 Node, \( N \) targets

2 Nodes, 1 target

2 Nodes, 2 targets

If there exists a \( y_i \) s.t.

\[
R_i - \left| \sum_{j=1, j \neq i}^{N} R_j \frac{y_i - y_j}{\| y_i - y_j \|} \right| > 0
\]

Li and Cassandras, IEEE TAC, 2006
BOSTON UNIVERSITY TEST BEDS

Christos G. Cassandras

CISE - CODES Lab. - Boston University
II. 2 Robots, 4 Targets Case
EVENT-DRIVEN SENSITIVITY ANALYSIS
REAL-TIME STOCHASTIC OPTIMIZATION

CONTROL/DECISION
(Parameterized by $\theta$)

SYSTEM

PERFORMANCE
$E[L(\theta)]$

GOAL:
$max_{\theta \in \Theta} E[L(\theta)]$

$\theta_{n+1} = \theta_n + \eta_n \nabla L(\theta_n)$

GRADIENT ESTIMATOR

$\nabla L(\theta)$

$x(t)$

$\nabla L(\theta)$

DIFFICULTIES:
- $E[L(\theta)]$ NOT available in closed form
- $\nabla L(\theta)$ not easy to evaluate
- $\nabla L(\theta)$ may not be a good estimate of $\nabla E[L(\theta)]$

Christos G. Cassandras

CISE - CODES Lab. - Boston University
REAL-TIME STOCHASTIC OPTIMIZATION FOR DES: INFINITESIMAL PERTURBATION ANALYSIS (IPA)

CONTROL/DECISION (Parameterized by $\theta$) → Discrete Event System (DES) → PERFORMANCE $E[L(\theta)]$

$\theta_{n+1} = \theta_n + \eta_n \nabla L(\theta_n)$

IPA

$\nabla L(\theta)$

For many (but NOT all) DES:
- Unbiased estimators
- General distributions
- Simple on-line implementation

A general framework for an IPA theory in Hybrid Systems?

\[ \theta_{n+1} = \theta_n + \eta_n \nabla L(\theta_n) \]
Performance metric (objective function):

\[ J(\theta; x(\theta,0), T) = E[L(\theta; x(\theta,0), T)] \]

\[ L(\theta) = \sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t) dt \]

IPA goal:

- Obtain unbiased estimates of \( \frac{dJ(\theta; x(\theta,0), T)}{d\theta} \), normally \( \frac{dL(\theta)}{d\theta} \)

- Then: \( \theta_{n+1} = \theta_n + \eta_n \frac{dL(\theta_n)}{d\theta} \)

NOTATION:

\[ x'(t) = \frac{\partial x(\theta,t)}{\partial \theta}, \quad \tau_k' = \frac{\partial \tau_k(\theta)}{\partial \theta} \]
HYBRID AUTOMATA

$$G_h = (Q, X, E, U, f, \phi, Inv, guard, \rho, q_0, x_0)$$

- **Q**: set of discrete states (modes)
- **X**: set of continuous states (normally $\mathbb{R}^n$)
- **E**: set of events
- **U**: set of admissible controls
- **f**: vector field, $f : Q \times X \times U \to X$
- **$\phi$**: discrete state transition function, $\phi : Q \times X \times E \to Q$
- **Inv**: set defining an invariant condition (domain), $Inv \subseteq Q \times X$
- **guard**: set defining a guard condition, $guard \subseteq Q \times Q \times X$
- **$\rho$**: reset function, $\rho : Q \times Q \times X \times E \to X$
- **$q_0$**: initial discrete state
- **$x_0$**: initial continuous state
HYBRID AUTOMATA

Unreliable machine with timeouts

\[ x(t) : \text{physical state of part in machine} \]
\[ \tau(t) : \text{clock} \]

\[ \alpha : \text{START}, \beta : \text{STOP}, \gamma : \text{REPAIR} \]
THE IPA CALCULUS
IPA: THREE FUNDAMENTAL EQUATIONS

System dynamics over \( (\tau_k(\theta), \tau_{k+1}(\theta)) \): 
\[
\dot{x} = f_k(x, \theta, t)
\]

NOTATION: 
\[
x'(t) = \frac{\partial x(\theta, t)}{\partial \theta}, \quad \tau'_k = \frac{\partial \tau_k(\theta)}{\partial \theta}
\]

1. Continuity at events: 
\[
x(\tau_k^+) = x(\tau_k^-)
\]

Take \( d/d\theta \): 
\[
x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)]\tau'_k
\]

If no continuity, use reset condition \( \Rightarrow \) 
\[
x'(\tau_k^+) = \frac{d\rho(q, q', x, \nu, \delta)}{d\theta}
\]
IPA: THREE FUNDAMENTAL EQUATIONS

2. Take \( d/d\theta \) of system dynamics \( \dot{x} = f_k(x, \theta, t) \) over \((\tau_k(\theta), \tau_{k+1}(\theta))\):

\[
\frac{dx'(t)}{dt} = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta}
\]

Solve

\[
\frac{dx'(t)}{dt} = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta}
\]

over \((\tau_k(\theta), \tau_{k+1}(\theta))\):\

\[
x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[ \int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial x} du} dv + x'(\tau_k^+) \right]
\]

NOTE: If there are no events (pure time-driven system), IPA reduces to this equation

initial condition from 1 above
3. Get $\tau'_k$ depending on the event type:

- **Exogenous event**: By definition, $\tau'_k = 0$

- **Endogenous event**: occurs when $g_k(x(\theta, \tau_k), \theta) = 0$

$$\tau'_k = -\left[\frac{\partial g}{\partial x} f_k(\tau^-_k)\right]^{-1}\left(\frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau^-_k)\right)$$

- **Induced events**:

$$\tau'_k = -\left[\frac{\partial y_k(\tau_k)}{\partial t}\right]^{-1} y'_k(\tau^+_k)$$
IPA: THREE FUNDAMENTAL EQUATIONS

Ignoring resets and induced events:

1. \( x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)] \cdot \tau_k' \)

2. \( x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} \, du} \left[ \int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial x} \, du} \, dv + x'(\tau_k^+) \right] \)

3. \( \tau_k' = 0 \quad \text{or} \quad \tau_k' = -\left[ \frac{\partial g}{\partial x} f_k(\tau_k^-) \right]^{-1} \left( \frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau_k^-) \right) \)

Recall:

\[
\begin{align*}
    x'(t) &= \frac{\partial x(\theta, t)}{\partial \theta} \\
    \tau_k' &= \frac{\partial \tau_k(\theta)}{\partial \theta}
\end{align*}
\]
IPA PROPERTIES

Back to performance metric: \[ L(\theta) = \sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t) \, dt \]

NOTATION: \[ L'_k(x, \theta, t) = \frac{\partial L_k(x, \theta, t)}{\partial \theta} \]

Then: \[
\frac{dL(\theta)}{d\theta} = \sum_{k=0}^{N} \left[ \tau'_{k+1} \cdot L_k(\tau_{k+1}) - \tau'_k \cdot L_k(\tau_k) + \int_{\tau_k}^{\tau_{k+1}} L'_k(x, \theta, t) \, dt \right]
\]

What happens at event times

What happens between event times
IPA PROPERTIES

**THEOREM 1**: If either 1,2 holds, then \(dL(\theta)/d\theta\) depends only on information available at event times \(\tau_k\):

1. \(L(x, \theta, t)\) is independent of \(t\) over \([\tau_k(\theta), \tau_{k+1}(\theta)]\) for all \(k\)

2. \(L(x, \theta, t)\) is only a function of \(x\) and for all \(t\) over \([\tau_k(\theta), \tau_{k+1}(\theta)]\):

\[
\frac{d}{dt} \frac{\partial L_k}{\partial x} = \frac{d}{dt} \frac{\partial f_k}{\partial x} = \frac{d}{dt} \frac{\partial f_k}{\partial \theta} = 0
\]

[**IMPLICATION**: - Performance sensitivities can be obtained from information limited to event times, which is easily observed

- **No need to track system in between events**!]

Christos G. Cassandras

CISE - CODES Lab. - Boston University
Evaluating $x(t; \theta)$ requires full knowledge of $w$ and $f$ values (obvious)

However, $\frac{dx(t; \theta)}{d\theta}$ may be independent of $w$ and $f$ values (NOT obvious)

It often depends only on:
- event times $\tau_k$
- possibly $f(\tau_{k+1}^-)$

Christos G. Cassandras

CISE - CODES Lab. - Boston University
IPA PROPERTIES

In many cases:

- **No need for a detailed model** (captured by $f_k$) to describe state behavior in between events.

- This explains why **simple abstractions of a complex stochastic system** can be adequate to perform sensitivity analysis and optimization, as long as event times are accurately observed and local system behavior at these event times can also be measured.

- This is true in **abstractions of DES as HS** since:
  
  Common performance metrics (e.g., workload) satisfy THEOREM 1.
SOLVING PROBLEMS WITH LINEAR TIME-DRIVEN DYNAMICS

\[ \min_{u_1, \ldots, u_K} \sum_{k=0}^{K} \left[ \int_{\tau_k}^{\tau_{k+1}} L_k(x(t), u(t)) dt + \psi_i(\tau_k) \right] \]

s.t.

\[ \dot{x} = a_k x + b_k u_k, \quad t \in (\tau_k, \tau_{k+1}] \]

Common to parameterize controls using basis functions \( \beta_l(t) \):

\[ u_k(t) = \sum_{l=0}^{L} \theta_{k,l} \beta_l(t) \]
Recall IPA equations:

1. \( x'(\tau_k^+)=x'(\tau_k^-)+[f_{k-1}(\tau_k^-)-f_k(\tau_k^+)] \cdot \tau_k' \)

2. \( x'(t)=e^{\alpha_k} \left[ \int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial \theta} du} dv + x'(\tau_k^+) \right] \)

3. \( \tau_k'=0 \) or \( \tau_k' = \left[ \frac{\partial g}{\partial x} f_k(\tau_k^-) \right]^{-1} \left( \frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau_k^-) \right) \)

When endogenous event \([g(x(\theta, \tau_k), \theta) = 0]\) occurs at \(\tau_k\):

1. \( \frac{\partial x(\tau_k^+)}{\partial \theta_{k,l}} = \frac{\partial x(\tau_k^-)}{\partial \theta_{k,l}} + \left( a_{k-1} - a_k \right) x(\tau_k^-) + b_{k-1} \sum_{l=0}^{L} \theta_{k-1,l} \beta_l(\tau_k^-) - b_k \sum_{l=0}^{L} \theta_{k,l} \beta_l(\tau_k^-) \frac{\partial \tau_k}{\partial \theta_{k,l}} \)

2. \( \frac{\partial x(t)}{\partial \theta_{k,l}} = e^{a_k(t-\tau_k)} \left[ \frac{b_k \beta_l}{a_k} \left[ 1-e^{-a_k(t-\tau_k)} \right] + \frac{\partial x(\tau_k^+)}{\partial \theta_{k,l}} \right] \)
Data collection: relatively easy...

Control: a challenge...