Controller Aware dL

Josh Clune (jclune@andrew.cmu.edu)
A Typical Story

The World
A Typical Story

The World

Cyber-Physical System
A Typical Story

The World

Goal

Cyber-Physical System

...
A Typical Story

Cyber-Physical Systems consider their environment

The World → Goal → Cyber-Physical System

The World: Busy traffic scene.
Goal: A car with a red circle around it, indicating no collision.
Cyber-Physical System: A self-driving car on the road.
A Typical Story

Cyber-Physical Systems consider their environment

The World  →  Goal  →  Cyber-Physical System

Cyber-Physical Systems affect the world
A Typical Story

Cyber-Physical Systems consider their environment

The World

Goal

Cyber-Physical System

Cyber-Physical Systems affect the world

Opportunities for Error:

- Assumptions about the world
- Implementing the cyber-physical system
- Defining and proving the goal
A Typical Story

Cyber-Physical Systems consider their environment

The World

Goal

Measurement

Cyber-Physical System

Control

Cyber-Physical Systems affect the world

Opportunities for Error:

- Assumptions about the world
- Implementing the cyber-physical system
- Defining and proving the goal
A Typical Story

Cyber-Physical Systems consider their environment

The World

Goal

Cyber-Physical System

Measurement

Control

Cyber-Physical Systems affect the world

- Opportunities for Error:
  - Assumptions about the world
  - Implementing the cyber-physical system
  - Defining and proving the goal

Measurement and control phenomena are also opportunities for error
Measurement and control phenomena exist in practically all cyber-physical systems.

All hardware is imperfect.

Modeling these imperfections is a fundamental problem.

Controller Aware dL is meant to make modeling hardware imperfections easier.
Goals for Controller Aware dL

- Assumptions about hardware imperfections should be explicit and easily extractable

- Reasoning about hardware imperfections should be separate from reasoning about the rest of the model

- Controller Aware dL should only reduce conceptual overhead
Measurable/Controllable Variables

**Measurable**
Variables the CPS uses hardware to estimate

**Controllable**
Variables that the CPS uses hardware to approximately control
Syntax of Controller Aware dL

term \( e \) ::= \( x \) | \( c \) | \( e_1 + e_2 \) | \( e_1 * e_2 \) | \( \neg x \)

formula \( P, Q \) ::= \( e_1 = e_2 \) | \( e_1 \leq e_2 \) | \( \neg P \) | \( P \land Q \) | \( P \lor Q \) | \( P \rightarrow Q \) | \( \forall x P \) | \( \exists x P \) | \( \langle \alpha \rangle P \) | \( [\alpha] P \)

hybrid program \( \alpha, \beta \) ::= \( x := e \) | \( x := \ast \) | \( ? P \) | \( x' = e \land P \) | \( \alpha \cup \beta \) | \( \alpha; \beta \) | \( \alpha^* \) | measure \( x \) | set \( x \) \( e \)
Approximating Measurable Variables

Length $x$
Approximating Measurable Variables

Length $x$

$x$ is between 5 and 6
Approximating Measurable Variables

Length $x$

$x$ is between 5.3 and 5.4
Approximating Measurable Variables

- $\sim x$ has some value in a range $[a, b]$

- The exact value of $\sim x$ is nondeterministically chosen
Approximating Measurable Variables

- ~x has some value in a range \([a, b]\)
- The exact value of ~x is nondeterministically chosen
- (~x)^2 should always be positive
  - (~x)^2 should also always equal ~x*~x
Approximating Measurable Variables

- \( \sim x \) has some value in a range \([a, b]\)

- The exact value of \( \sim x \) is nondeterministically chosen

- \((\sim x)^2\) should always be positive
  - \((\sim x)^2\) should also always equal \(\sim x^*\sim x\)

- The exact value of \(\sim x\) should be chosen when the programmer wants. This is what “measure x” is for
Setting Controllable Variables

- No wind
- Consistent wind
- Weak, gusty wind
- Strong, gusty wind
Setting Controllable Variables

- $x := e$ assigns $x$ the value of $e$ exactly
- “set $x e$” assigns $x$ the approximate value of $e$
- After “set $x e$”, $x$ has a value in a range $[a, b]$
- The exact value of $x$ is nondeterministically chosen
Restrictions on Ranges

- How should we define these ranges of uncertainty?
Restrictions on Ranges

- How should we define these ranges of uncertainty?
- For each variable \( x \), pick real values \( \varepsilon_{x1} \) and \( \varepsilon_{x2} \) and use the range \([x - \varepsilon_{x1}, x + \varepsilon_{x2}]\)
Restrictions on Ranges

- How should we define these ranges of uncertainty?

- For each variable $x$, pick real values $\varepsilon_{x1}$ and $\varepsilon_{x2}$ and use the range $[x - \varepsilon_{x1}, x + \varepsilon_{x2}]$
  - We may want $\varepsilon_{x1}$ and $\varepsilon_{x2}$ to depend on $x$
Restrictions on Ranges

- How should we define these ranges of uncertainty?

- For each variable $x$, pick real values $\varepsilon_{x1}$ and $\varepsilon_{x2}$ and use the range $[x - \varepsilon_{x1}, x + \varepsilon_{x2}]$
  - We may want $\varepsilon_{x1}$ and $\varepsilon_{x2}$ to depend on $x$

- For each variable $x$, pick expression $e_1$ and $e_2$ and use the range $[e_1, e_2]$
Restrictions on Ranges

- How should we define these ranges of uncertainty?

- For each variable $x$, pick real values $\varepsilon_{x1}$ and $\varepsilon_{x2}$ and use the range $[x - \varepsilon_{x1}, x + \varepsilon_{x2}]$
  - We may want $\varepsilon_{x1}$ and $\varepsilon_{x2}$ to depend on $x$

- For each variable $x$, pick expression $e_1$ and $e_2$ and use the range $[e_1, e_2]$
  - We may want the range to depend on the environment in a way that isn’t expressible at the term level
Restrictions on Ranges

- For each variable $x$, define the range $[a, b]$ in terms of dL hybrid programs $\alpha$ and $\beta$.

- $[e_1, e_2]$ represented as:
  - $\alpha = (~x := e_1)$, $\beta = (~x := e_2)$

- $[e_1, e_2]$ if it’s raining hard, $[e_3, e_4]$ otherwise
  - $\alpha = (\text{is\_raining}; (~x := e_1)) \cup (\text{is\_raining}; (~x := e_3))$
  - $\beta = (\text{is\_raining}; (~x := e_2)) \cup (\text{is\_raining}; (~x := e_4))$
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \]

\[ ((x + vT < l \land v \leq 0) \lor (x + vT > r \land v \geq 0)) \text{ then } v := -v; \]

\[ t := 0; \{ x' = v, t' = 1 \land 0 \leq t \leq T \}^* \]

\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \]

\[ (\text{if } x + vT < l \land v \leq 0 \lor x + vT > r \land v \geq 0) \text{ then } v := -v; \]

\[ t := 0; \{x' = v, t' = 1 \land 0 \leq t \leq T\}^* \]

\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \]

\[(\text{if } x + vT < l \land v \leq 0 \lor x + vT > r \land v \geq 0)\]

then \( (v_{old} := v; v := *; ?(-v_{old} - \varepsilon_1 \leq v \land v \leq -v_{old} + \varepsilon_2)); \)

\[ t := 0; \{x' = v, t' = 1 \& 0 \leq t \leq T\})^* \]

\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \]
\[ [(\text{if} (x + vT < l \land v \leq 0 \lor x + vT > r \land v \geq 0) \]
\text{then} \ (v_{\text{old}} := v; v := *; ?(-v_{\text{old}} - \varepsilon_1 \leq v \land v \leq -v_{\text{old}} + \varepsilon_2)); \]
\[ t := 0; \{x' = v, t' = 1 \& 0 \leq t \leq T}\}^* \]
\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow (v_{\text{approx}} := *; ?(v - \varepsilon_3 \leq v_{\text{approx}} \land v + \varepsilon_4); \]

\textbf{if} \ (x + v_{\text{approx}}T < l \land v_{\text{approx}} \leq 0 \lor x + v_{\text{approx}}T > r \land v_{\text{approx}} \geq 0) \ \\
\textbf{then} \ (v_{\text{old}} := v_{\text{approx}}; v := *; ?(-v_{\text{old}} - \varepsilon_1 \leq v \land v \leq -v_{\text{old}} + \varepsilon_2)); \ \\
\quad t := 0; \{x' = v, t' = 1 \& 0 \leq t \leq T\})^* \]

\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[ l \leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \]

\[ [(v_{approx} := *; \ ?(v - \varepsilon_3 \leq v_{approx} \land v + \varepsilon_4)); \]

\[ \text{if}(x + v_{approx}T < l \land v_{approx} \leq 0 \lor x + v_{approx}T > r \land v_{approx} \geq 0) \]

\[ \text{then} \ (v_{old} := v_{approx}; v := *; ?(-v_{old} - \varepsilon_1 \leq v \land v \leq -v_{old} + \varepsilon_2)); \]

\[ t := 0; \ \{x' = v, t' = 1 \ \& \ 0 \leq t \leq T\}]^* \]

\[ l \leq x \land x \leq r \]
A Simple Example: Ping Pong in 1D

\[
\begin{align*}
    l &\leq x \land x \leq r \land v \geq 0 \land T > 0 \land l + 2vT \leq r \rightarrow \\
    [(v_{\text{approx}} := *; ?(v - \varepsilon_3 \leq v_{\text{approx}} \land v + \varepsilon_4)); \\
    x_{\text{approx}} := *; ?(x - \varepsilon_5 \leq x_{\text{approx}} \land x + \varepsilon_6)); \\
    \text{if}(x_{\text{approx}} + v_{\text{approx}}T < l \land v_{\text{approx}} \leq 0 \lor x_{\text{approx}} + v_{\text{approx}}T > r \land v_{\text{approx}} \geq 0) \\
    \text{then} (v_{\text{old}} := v_{\text{approx}}; v := *; ?(-v_{\text{old}} - \varepsilon_1 \leq v \land v \leq -v_{\text{old}} + \varepsilon_2)); \\
    t := 0; \{x' = v, t' = 1 \land 0 \leq t \leq T\}^*]
\end{align*}
\]

\[l \leq x \land x \leq r\]
A Simple Example: Ping Pong in 1D

\[
l \leq x \wedge x \leq r \wedge v \geq 0 \wedge T > 0 \wedge l + 2vT \leq r \rightarrow \]

\[
[(\text{measure } x; \text{measure } v; \]
\text{if}(\sim x + \sim vT < l \wedge \sim v \leq 0 \lor \sim x + \sim vT > r \wedge \sim v \geq 0) \text{ then set } v (\sim v); \]
\text{t := 0; } \{x' = v, t' = 1 \& 0 \leq t \leq T\}]^{*}
\]

\[
l \leq x \wedge x \leq r
\]
I’ve given the high-level vision and motivation for Controller Aware dL

There are a lot of technical details that go into actually making Controller Aware dL feasible

- Giving a formal semantics
- Formalizing the restrictions on ranges
- Defining a translation to dL
- Proving said translation sound