Recall: Prerequisites

To design a (gas burner) controller that meets its requirements, we need:

- A formal model of behaviour in (quantitative) time,
- A language to concisely, conveniently specify requirements on behaviour,
- A language to specify behaviour of controllers,
- A notion of "meet" and a methodology to verify meeting.

Real-Time Behaviour, More Formally...

State Variables (or Observables)

We assume that the real-time systems we consider is characterised by a finite set of state variables (or observables) $\text{obs}_1, \ldots, \text{obs}_n$ each equipped with a domain $D(\text{obs}_i)$, $1 \leq i \leq n$.

Example: gas burner

- $G$: Time $\rightarrow \{0, 1\}$
- $F$: Time $\rightarrow \{0, 1\}$
- $I$: Time $\rightarrow \{0, 1\}$
- $H$: Time $\rightarrow \{0, 1\}$

System Evolution over Time

One possible evolution (or behaviour) of the considered system over time is represented as a function $\pi: \text{Time} \rightarrow D(\text{obs}_1) \times \cdots \times D(\text{obs}_n)$.

If (and only if) observable $\text{obs}_i$ has value $d_i \in D(\text{obs}_i)$ at time $t \in \text{Time}$, $1 \leq i \leq n$, we set $\pi(t) = (d_1, \ldots, d_n)$.

For convenience, we use $\text{obs}_i: \text{Time} \rightarrow D(\text{obs}_i)$ to denote the projection of $\pi$ onto the $i$-th component.
What's the time?

There are two main choices for the time domain: Time:

- Discrete time: Time = N₀, the set of natural numbers.
- Continuous or dense time: Time = R⁺₀, the set of non-negative real numbers.

Throughout the lecture, we shall use the continuous time model and consider discrete time as a special case.

Because plant models usually live in continuous time, we avoid too early introduction of hardware considerations.

An interesting view: continuous-time is a well-suited abstraction from the discrete-time realm induced by clock cycles etc.

Example: Gas Burner

Gas valve
Flame sensor
Ignition

One possible evolution of the considered system over time is represented as a function π: Time → D(obs₁) × · · · × D(obsₙ).

If (and only if) observable obsᵢ has value dᵢ ∈ D(obsᵢ) at time t ∈ Time, set:

π(t) = (d₁, . . . , dₙ).

For convenience, use obsᵢ: Time → D(obsᵢ).

Level of Details

Note: Depending on the choice of observables, we can describe a real-time system at various levels of detail.

For instance, if the gas valve has different positions, use G: Time → {0, 1, 2, 3} (But: D(G) is never continuous in the lecture, otherwise we had a hybrid system.)

If the thermostat and the controller are connected via a bus and exchange messages, use B: Time → Msg∗ to model the receive buffer as a finite sequence of messages from Msg.

etc.

System Properties

Predicate Logic

ϕ ::= obs(t) = d | ¬ϕ | ϕ₁ ∨ ϕ₂ | ϕ₁ ∧ ϕ₂ | ϕ₁ ⇒ ϕ₂ | ϕ₁ ⇔ ϕ₂ |

∀t ∈ Time • ϕ |

∀t ∈ [t₁ + c₁, t₂ + c₂] • ϕ

where obs an observable,

d ∈ D(obs),

t ∈ Var logical variable,

c₁, c₂ ∈ R⁺₀ constants.
Req's validity.

If 'Req' and 'Des' are described by formulae of first-order predicate logic,

\[ \pi, \beta \vdash \phi \implies \forall \tau, \pi(\phi) = \tau \in \phi \in \phi \\text{\texttt{obs}} \]

We can also use first-order predicate logic to formally specify requirements.

Let 'Req' be a set of system behaviours with the pragmatics of the correct implementation of the correct design shall lie within this set.

For example:

\[
\begin{align*}
\text{Time} & \in \mathbb{N} \\
\text{obs} & : \mathbb{N} \times \mathbb{N} \rightarrow D
\end{align*}
\]

Recall: Predicate Logic, Standard Semantics
Safety Properties

- A safety property states that something bad must never happen [Lamport].
- Example: train inside level crossing with gates open.
- More general, assume observable $C: \text{Time} \rightarrow \{0, 1\}$ where $C(t) = 1$ represents a critical system state at time $t$.
- Then $\forall t \in \text{Time} \cdot \neg C(t)$ is a safety property.
- In general, a safety property is characterised as a property that can be falsified in bounded time.
- But safety is not everything...

Liveness Properties

- The simplest form of a liveness property states that something good eventually does happen.
- Example: gates open for road traffic.
- More general, assume observable $G: \text{Time} \rightarrow \{0, 1\}$ where $G(t) = 1$ represents a good system state at time $t$.
- Then $\exists t \in \text{Time} \cdot G(t)$ is a liveness property.
- Note: not falsified in infinite time.
- With real-time, liveness is too weak...

Bounded Response Properties

- A bounded response property states that the desired reaction on an input occurs within the time interval $[b,e]$.
- Example: from request to secure level crossing to gates closed.
- More general, reconsider good thing $G: \text{Time} \rightarrow \{0, 1\}$ and request $R: \text{Time} \rightarrow \{0, 1\}$.
- Then $\forall t_1 \in \text{Time} \cdot (R(t_1) = \Rightarrow \exists t_2 \in [t_1+10, t_1+15] \cdot G(t_2))$ is a bounded liveness property.
- This property can again be falsified in infinite time.
- With gas burners, this is still not everything...

Duration Properties

- A duration property states that for observation interval $[b,e]$ characterised by a condition $A(b,e)$ the accumulated time in which the system is in a certain critical state has an upper bound $u(b,e)$.
- Example: leakage in gas burner.
- More general, reconsider critical thing $C: \text{Time} \rightarrow \{0, 1\}$.
- Then $\forall b, e \in \text{Time} \cdot (A(b,e) = \Rightarrow \int_{e}^{b} C(t) dt \leq u(b,e))$ is a duration property.
- This property can again be falsified in infinite time.

References

