

# DAIMLERCHRYSLER

## **Logic of Continuous Control**

Edward R. Griffor

DaimlerChrysler AG – November 4, 2006

## Presentation Outline

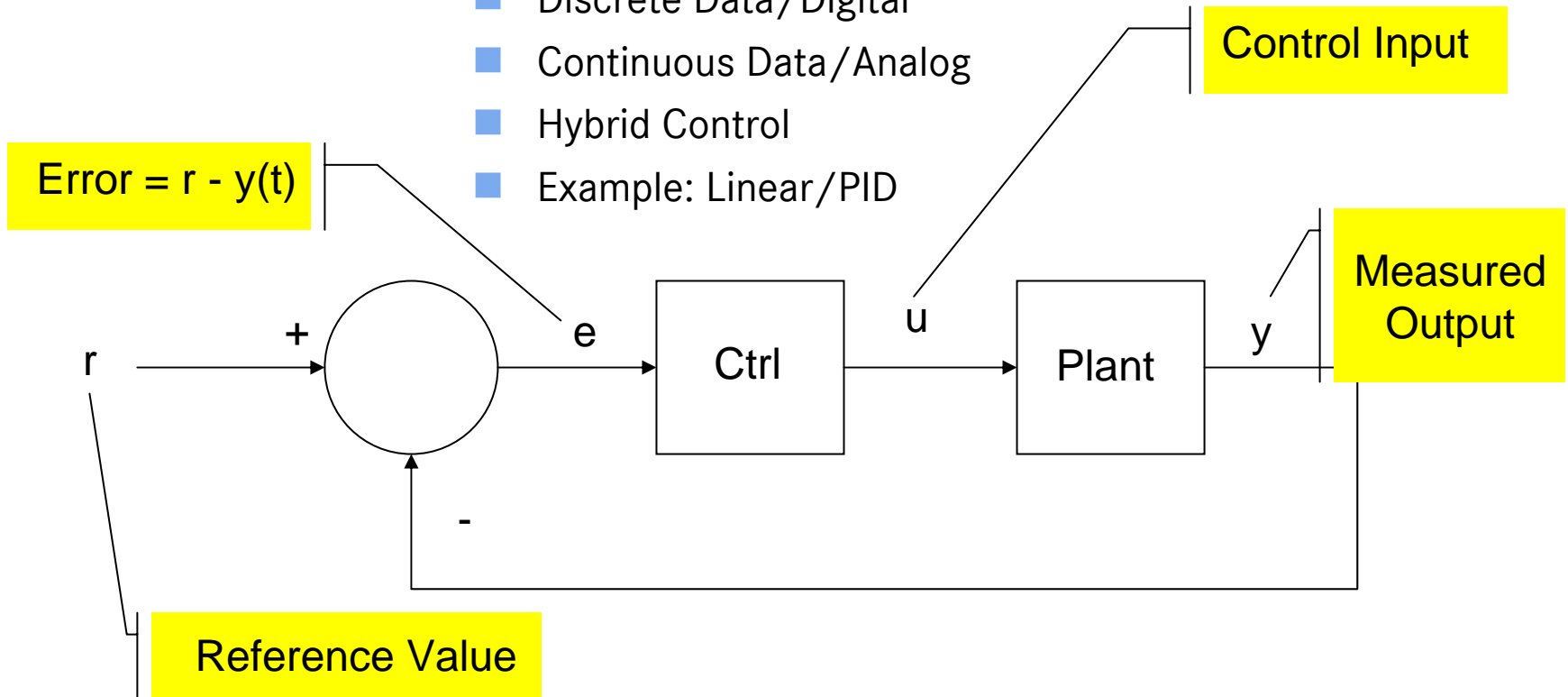
- Analog Control and Real Computation
  - Control Systems
  - Control System Requirements Management
  - System Requirements and Test – The Logic of Ctrl
  - Complex Control Structures
  - System Accidents – Requirement Failure
  - Safety and Failure
  - System Safety/Reliability
  - STAMP - System Failure Models for Defining Semantics
    - STAMP Model (MIT-Leveson)
    - STAMP-Based Hazard Analysis
    - Sample System and Analysis
  - Conclusions
-

# Analog Control and Real Computation

- Continuous/Analog Control (Linear and Non-Linear)
  - Sensor Values as Inputs
    - Distance
    - Temperature
    - Average Chg in Velocity
  - Computations
    - Closing Velocity (Acceptable Rate of Change in Distance)
    - Material Failure (Incremental or Critical Temperature Rate of Change)
    - Duty Cycle Profile (Electronics and Mechanical Systems)
  - Ctrl Structure + Constructive Semantics = Dynamic Specification
  - Laplace Transform as Model-Theoretic Embedding
-

## Control Systems

- Discrete Data/Digital
- Continuous Data/Analog
- Hybrid Control
- Example: Linear/PID



# Control System Requirements Management

- Requirements Management
    - System Behavioral Requirements (e.g. all voltages remain below  $V$ )
    - Levels of Implementation (Multilevel Semantics)
    - Systems Engineering - Links for Dependencies (Cascade and Test)
  - Test Cases “On input  $a_1, \dots, a_k$ , the output  $b$  satisfies  $v(a_1, \dots, a_k) < V$ ”; outputs on constrained inputs satisfy a property; Test Case fails if false, otherwise it succeeds for  $R$ .
  - Sets of Test Cases define ‘Requirements Satisfaction’; the requirement  $R$  is *satisfied* relative to  $F$  (finite set of test cases) if all test cases in  $F$  succeed for  $R$ .
-

## System Requirements and Test – The Logic of Ctrl

- System Behavior as Propositions in the FOL of Control Structures (LCS)
    - Global Variables
    - Variables for I/O
    - Finite Data Types D and Bdd Quantifiers
  - Hybrid Truth Definition for Formulae of LCS – Bounded Quantification for Data Types and Domain Restricted Unrestricted Quantification
    - Test cases are the atoms  $t_1, t_2 \dots t_n$
    - ‘Test’ constructors for Constructive Types
  - $\Phi$  true – Induction on  $\Phi$ 
    - $\Phi$  atomic as for
    - Cf constructive type theory for connectives and bounded quantification
    - feedback as range-defined global quantification
  - Logic to Specify/Design and Construct Ctrl Structure
-

## Complex Control Structures

- Simplified PID Control  
(Proportionality, Integration and Differentiation)
    - Proportionality:  $C(r-y)$
    - Integration:  $\sum C(t)[r - y(t)] \Delta t$
    - Differentiation:  $d/dt[\sum C(t)[r - y(t)] \Delta t]$
  - Hierarchical Control
  - Parameterizing Control Structures  
(Configuration or Parameterization of Features)
-

## ‘System Accidents’ – Requirements Failure

- Accident = Domain Mismatch
  - Component Failure Accidents
    - Single or multiple component failures
    - Usually assume random failure
  - System Accidents
    - Arise from interactions among components
    - Related to interactive complexity and tight coupling in systems
    - Exacerbated by introduction of computers and software
  - Safety = Operation on the Model
-

## Safety and Failure(1)

- Traditionally assumed component failure accidents
    - Redundancy
    - Safety functions and Protection systems
    - Increase component “integrity”
      - Safety margins for physical components
      - Error-free or fault-tolerance for logical and human components
    - FTA, PRA, FMECA, Event Trees, etc.
    - FMEA, DFMEA, etc.
  - This approach works well for traditional electro-mechanical automotive systems
-

## Safety (Operation On the Model) and Failure (2)

- Modern systems rely extensively on software and computers
  - Most software-related accidents are system accidents
    - Software does not “fail” as a mechanical system
    - “Software Failure” box in a fault tree is misleading
  - Increase of automation has large impact on operator error and control
    - Inadequate communication and confusion in distributed control is increasingly a factor in accidents
  - Dysfunctional interactions between the hardware, software and human require more than a component-based reliability approach
-

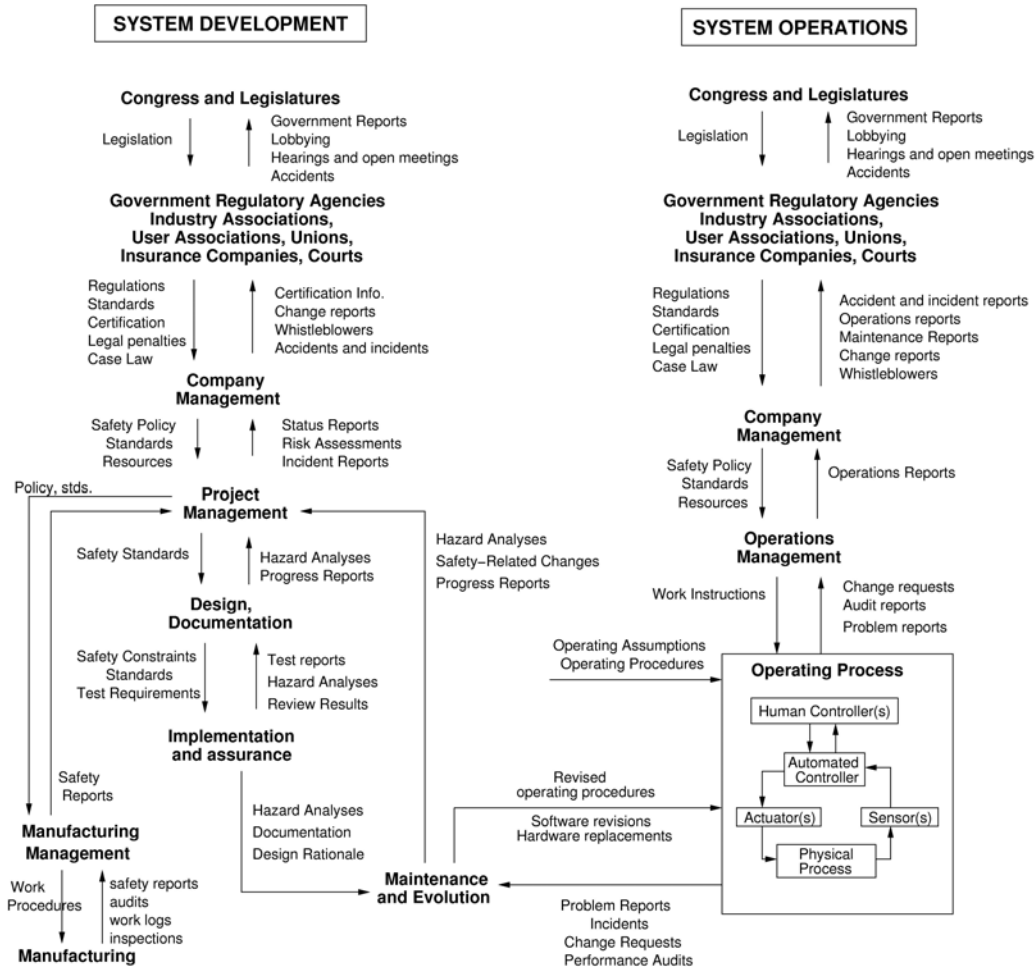
## STAMP (1 - Leveson Group - MIT)

- Systems-Theoretic Accident Model and Processes
    - Accidents are not simply an event or chain of events but involve a complex, dynamic process
    - Based on systems and control theory
  - Accidents arise from interactions among humans, machines, and the environment
    - Not simply linear causality, but more complex types of causal connections
-

## STAMP (2)

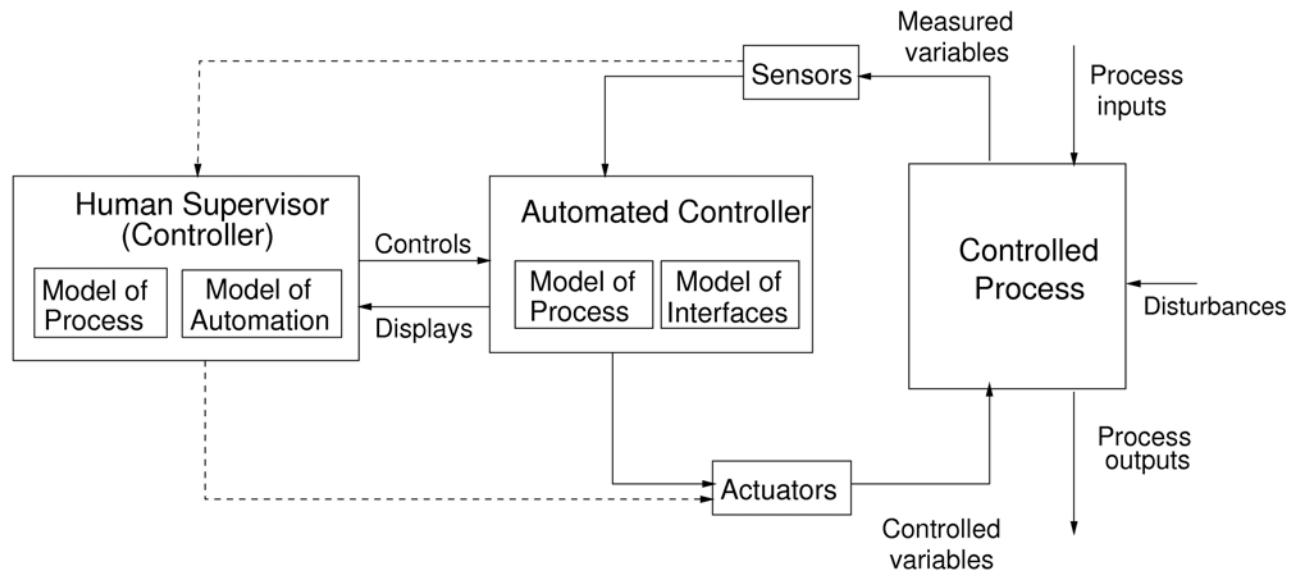
- Safety is an emergent property that arises when system components interact with each other within a larger environment
    - A set of safety constraints related to behavior of system components enforces that property
    - Accidents occur when interactions violate those constraints (a lack of appropriate constraints on the interactions)
    - “Controllers” embody or enforce those constraints
    - Goal of system safety engineering is to identify the safety constraints and enforce them in the system design and/or operation
-

# Conceptual Model



# Process Model and Computation

## Process Models



Process models must contain:

- Required relationship among process variables
- Current state (values of process variables)
- The ways the process can change state

## Uses for STAMP

- Basis for new, more powerful hazard analysis techniques (STPA)
  - Safety-driven design
  - More comprehensive accident/incident investigation and root cause analysis
  - Organizational and cultural risk analysis
    - Defining safety metrics and performance audits
    - Designing and evaluating potential policy and structural improvements
    - Identifying leading indicators of increasing risk (“canary in the coal mine”)
  - New risk management tools
    - Policy analysis and evaluation
    - Risk analysis and control
  - New holistic approaches to security
-

## STAMP-Based Hazard Analysis (STPA)

- Supports a safety-driven design process where:
    - Hazard analysis influences and shapes early design decisions
    - Hazard analysis iterated and refined as design evolves
  - Goals (same as any hazard analysis)
    - Identification of system hazards and related safety constraints necessary to ensure acceptable risk
    - Accumulation of information about how hazards can be violated, which is used to eliminate, reduce and control hazards in system design, development, manufacturing, and operations
-

## STPA - Safety-Driven Design Process

- Start with identifying system requirements and design constraints necessary to maintain safety (class of failures)
  - STPA assists in top-down refinement into requirements and safety constraints on individual components
  - STPA used to identify scenarios in which safety constraints can be violated. Use results to eliminate or control them in design, operations, etc.
-

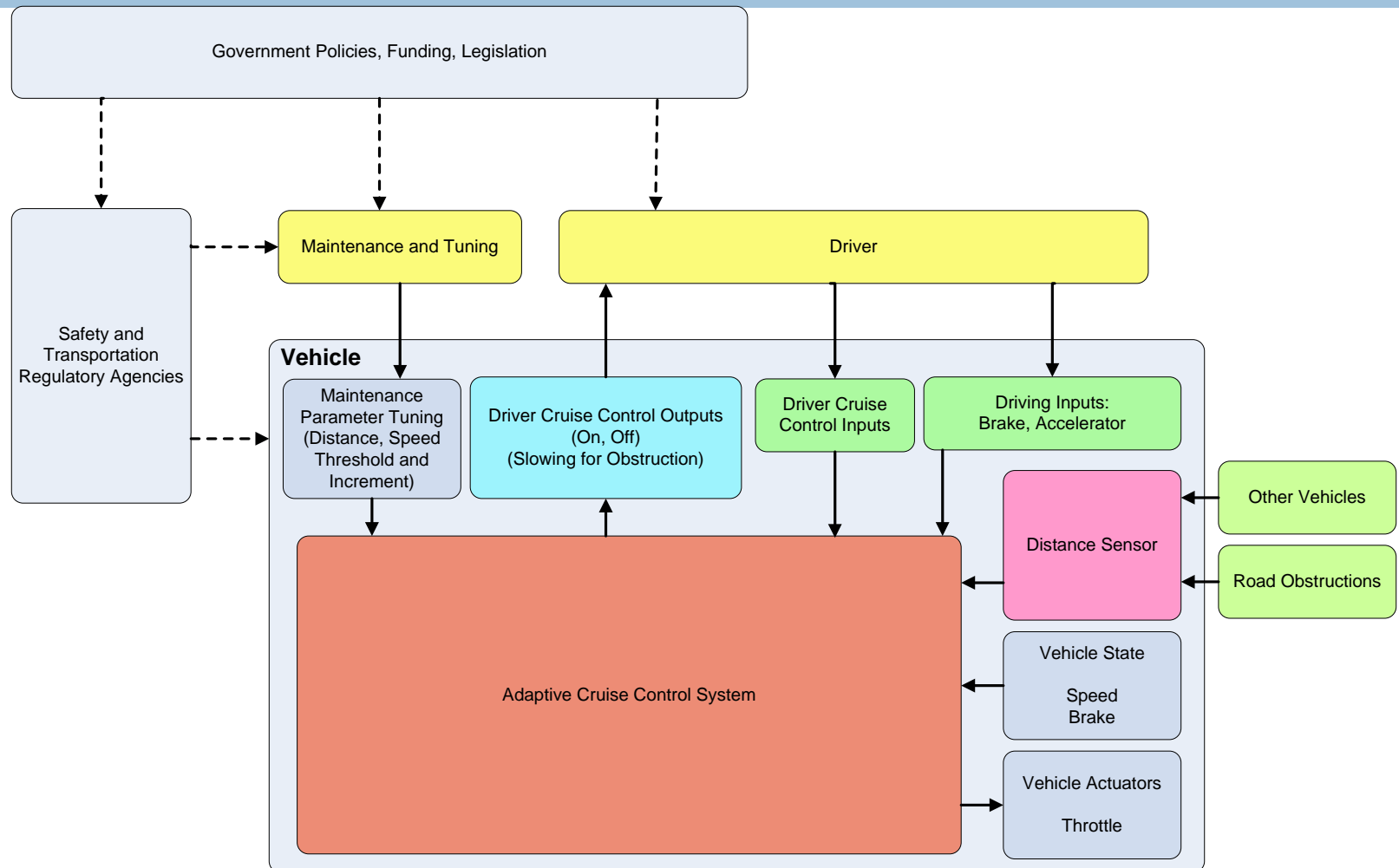
## Example of Adaptive Control Hazards

1. Flight control does not maintain correct speed
  2. Flight control causes inadvertent acceleration
  3. Flight control commands vehicle beyond safe operating capability
  4. Flight control leads to driver complacency
  5. Flight control interferes with driver control of the system
- Turn hazards into design and safety constraints:
    - SC.2: The vehicle must not accelerate inadvertently
-

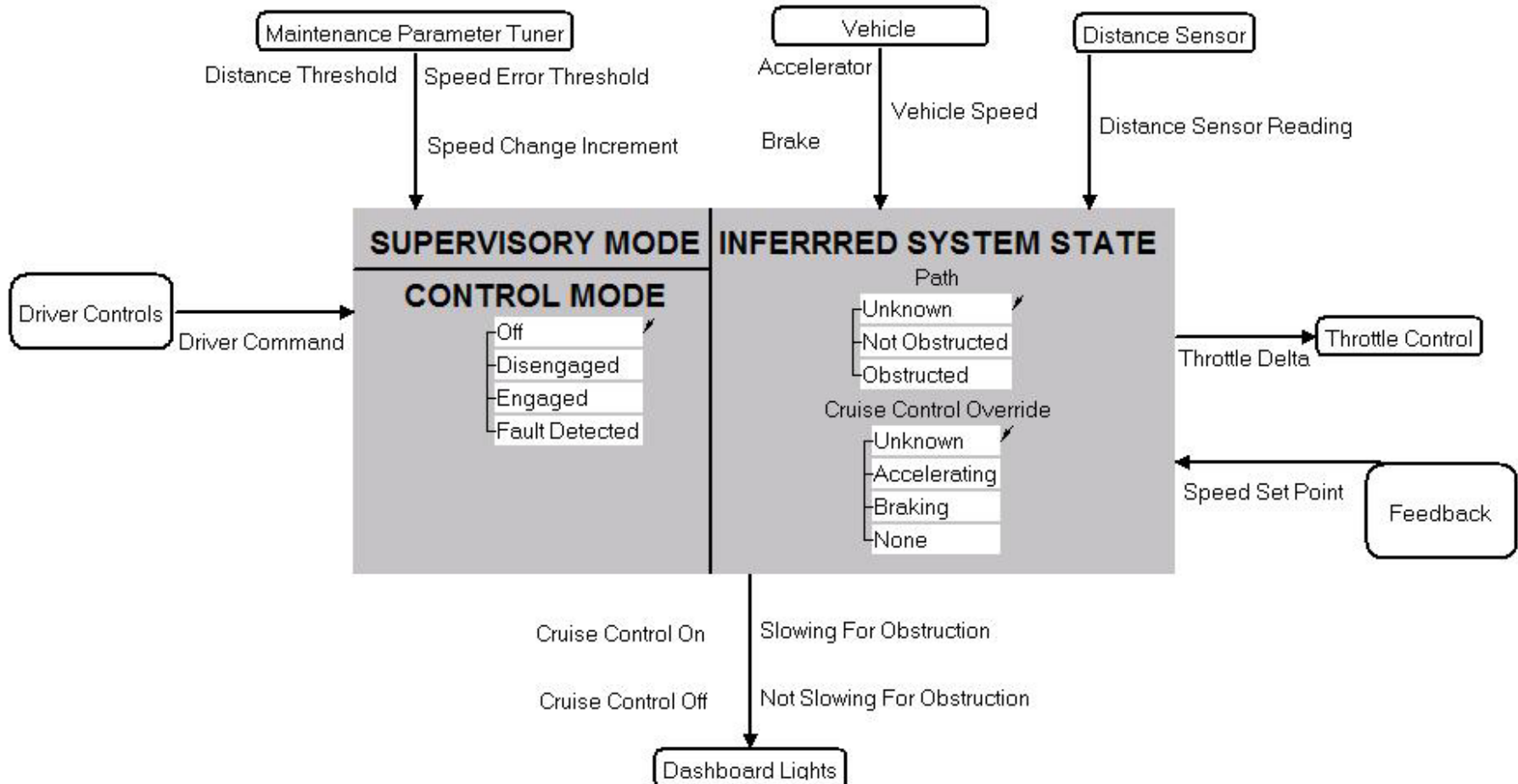
# Taxonomy of Control Flaws

- **Inadequate control actions (enforcement of constraints)**
    - Unidentified hazards
    - Inappropriate, ineffective, or missing control actions for identified hazards
      - Design of control algorithm (process) does not enforce constraints
      - Process models inconsistent, incomplete, or incorrect (lack of linkup)
        - Flaw(s) in creation process
        - Flaws(s) in updating process (asynchronous evolution)
        - Time lags and measurement inaccuracies not accounted for
      - Inadequate coordination among controllers and decision-makers (boundary and overlap areas)
  - **Inadequate Execution of Control Action**
    - Communication flaw
    - Inadequate actuator operation
    - Time lag
  - **Inadequate or missing feedback**
    - Not provided in system design
    - Communication flaw
    - Time lag
    - Inadequate sensor operation (incorrect or no information provided)
-

# Another Control Example – Adaptive Cruise Control (ACC)



## Sample ACC System Controller



# Specifying the Control Logic

## Cruise Control

**Description:** Cruise Control is the main control mode for the cruise control system. The most complicated logic in the system is determining when the cruise control should engage and disengage in order to maintain the safety of the system.

**Comment:**

**References:** ([3.Driver Command](#), [3.Cruise Control Override](#), [3.Vehicle Speed](#))

**Appears In:** ([2.1](#), [2.2](#), [2.5](#), [2.6](#)) ([3.Cruise Control On](#), [3.Cruise Control Off](#), [3.Slowing For Obstruction](#), [3.Not Slowing For Obstruction](#), [3.Throttle Delta](#), [3.Set Next Speed Set Point](#), [3.Decrease Next Speed Set Point](#), [3.Increase Next Speed Set Point](#))

### DEFINITION

= Off

System Start	T	*
Driver Command is Off	*	T

= Disengaged

System Start	F	F	F
Off	T	*	*
Disengaged	*	T	*
Engaged	*	*	T
Driver Command is Off	*	F	*
Driver Command is On	T	*	*
Driver Command is Set Speed	*	F	*
Cruise Control Override in state None	*	*	F

= Engaged

System Start	F	F
Disengaged	T	*
Engaged	*	T
Driver Command is Off	*	F
Driver Command is Set Speed	T	*
Cruise Control Override in state None	*	T

= Fault Detected

System Start	F	F	F	F
Fault Detected	T	*	*	*
Time Since Cruise Control Last Entered Off > 5 seconds	*	T	*	T
Driver Command is Off	F	F	F	F
Vehicle Speed was Never Received	*	T	F	*
Vehicle Speed is Obsolete	*	*	T	*
Cruise Control Override in state Unknown	*	*	*	T

= Fault Detected

System Start
Fault Detected
Time Since Cruise Control Last Entered Off > 5 seconds
Driver Command is Off
Vehicle Speed was Never Received
Vehicle Speed is Obsolete
Cruise Control Override in state Unknown

F	F	F	F
T	*	*	*
*	T	*	T
F	F	F	F
*	T	F	*
*	*	T	*
*	*	*	T

## ‘Constructing’ the Constructive Logic of the Controller

- Engineering the semantics for the logics needed for today’s complex software-intensive systems (STAMP provides one approach)
  - Current applications to aerospace/automotive systems
    - CBR (Condition Based Response)
    - NASA ITA and Exploration System
    - Crew Exploration Vehicle
    - Next Generation Air Transportation System
    - VII (Vehicle Information Infrastructure)
  - Can provide an effective approach to design for safety in modern automotive systems
-

## Questions

---