Analyzing Feature Interactions in Automobiles

John Thomas, Ph.D.
Seth Placke
3.25.14

Outline

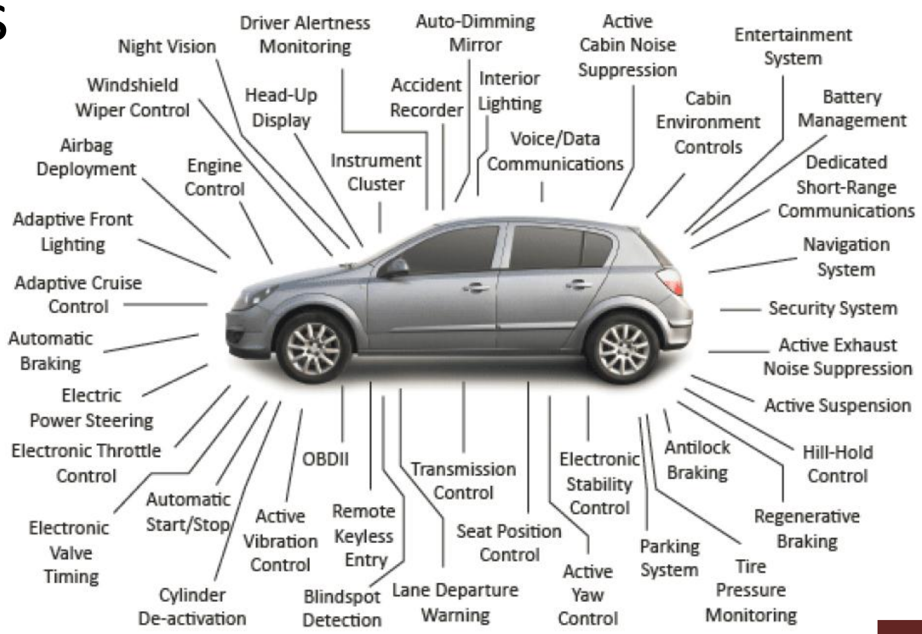
- Project Introduction & Background
- STPA Case Study
- New Strategy for Analyzing Interactions
- Contributions

Project Introduction

Goal: Integrate multiple propulsion and braking control systems into one vehicle.

Problem: These control systems (features) may interact in unsafe and dysfunctional ways

- Large numbers of systems
- Emergent behavior:
  - Difficult to predict
  - Can lead to an accident



(Hommes, 2012)

Project Introduction

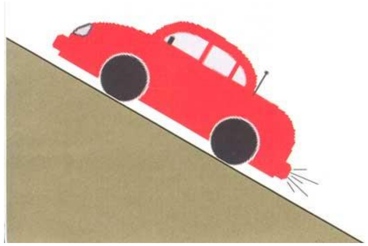
- Ideal System Engineering:
  - Top-down design from the start
- Common Challenges:
  - Upgrades to old systems
  - Adding features, etc…

Project:

- Use STPA to analyze interactions from new controllers
  - STPA to three example features
  - Identify hazards and dysfunctional interactions that arise during feature integration
  - Generalize analysis process for future use during concept development

Project Scope

Auto-Hold: Automatic braking at stops

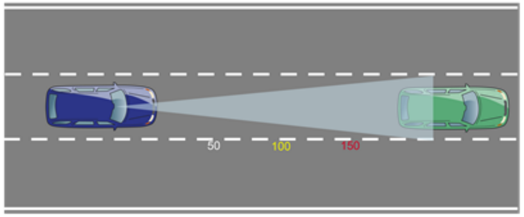


- Take (or release) control of the brakes
- Increase the brake pressure
- Apply the Parking Brake

Engine Stop-Start: Reduce idling at traffic stops

- Shutoff the Engine
- Restart the Engine
- Apply the Parking Brake

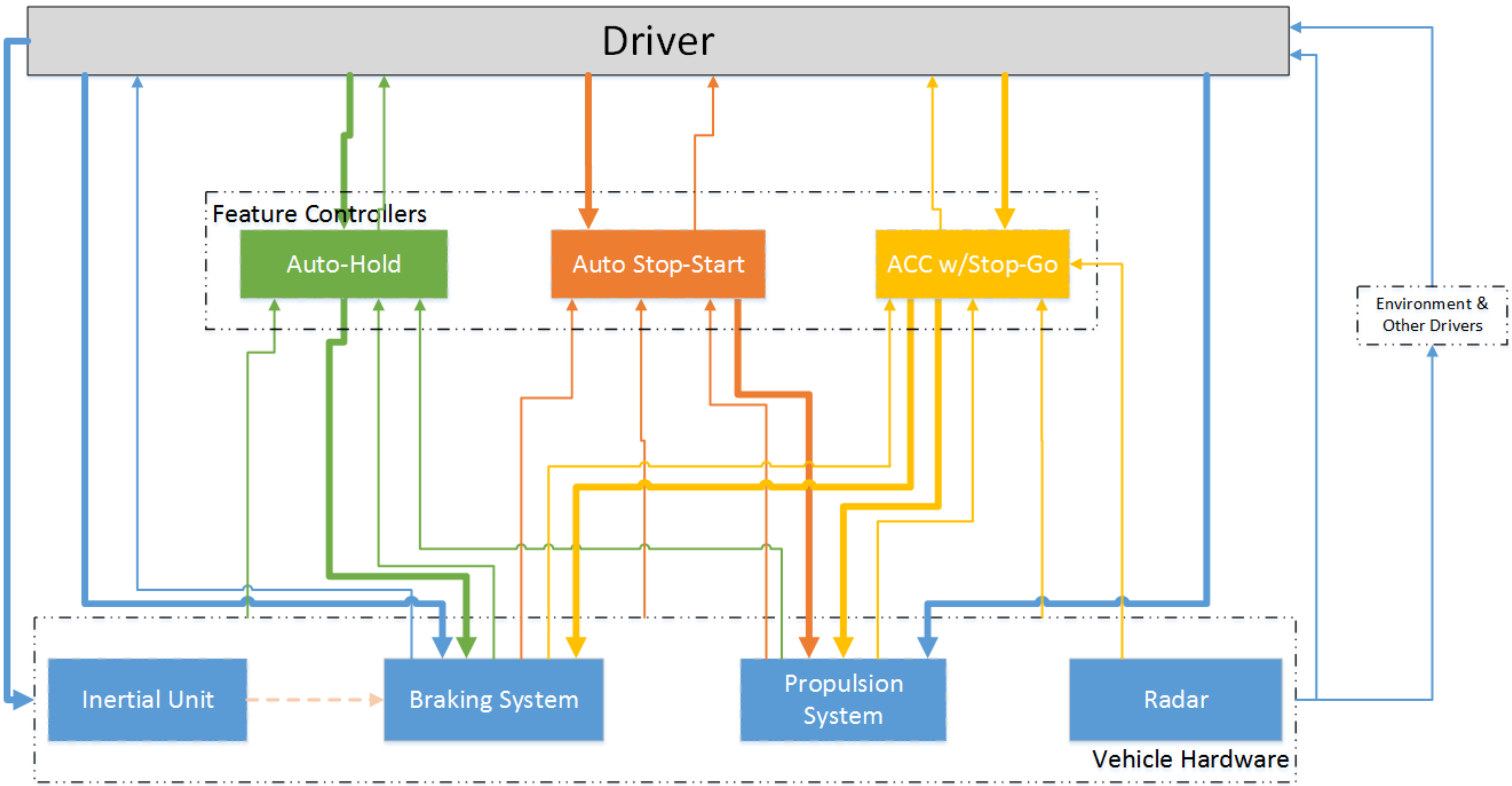
ACC w/Stop-Go: Adaptive Cruise Control at all speeds



- Accelerate
- Brake

All features Safety Critical!

System Control Structure



Control Common Processes & Receive Common Feedback

Individual Analysis Results

- Step 1 UCAs
  - ACC accelerates when too close to leading vehicle
  - ESS shuts-off engine when vehicle is rolling
  - AH holds brakes when vehicle is moving
  - etc…

- Step 2 Causal Factors
  - Brake valve fails
  - Shared bus error
  - Delayed range feedback
  - etc…

	ACC Enabled	ACC Engaged	Driver Present	Etc…	Providing Causes Hazard	Etc…
Accelerate Command	No	*	*	•••	x	•••
	Yes	No	*	•••	x	•••
	Yes	Yes	No	•••	x	•••

Executable Requirements

ACC w/SG

		Accelerate		Brake		
ACC Enabled =	No					
	Yes	T	T	T	T	T
ACC Engaged =	No					
	Yes	T	T	T	T	T
Driver Present =	No					
	Yes	T	T	T	T	T
PRNDL =	!=D					
	D	T	T	T	T	T
Driver (Either) Pedal Input =	Yes					
	No	T	T	T	T	T
Target Locked =	No					T
	Yes		T	T	T	
Distance Above Threshold =	No	T	T	T		
	Yes				T	
Speed Above Threshold =	Yes				T	T
	No	T				

Individual Analysis Summary

- Analyzed the design of each controller, implemented individually
  - Systems were designed independently
  - In isolation each works relatively well
  - Design assumptions may be violated upon integration
- Need to thoroughly analyze the *interactions* between controllers:
  - How does Stop-Start stopping the engine affect Auto-Hold?
  - How do ACC w/SG and Auto-Hold manage the brakes simultaneously?
  - Do the features respond in concert during off nominal situations?
- Can the features issue conflicting commands?

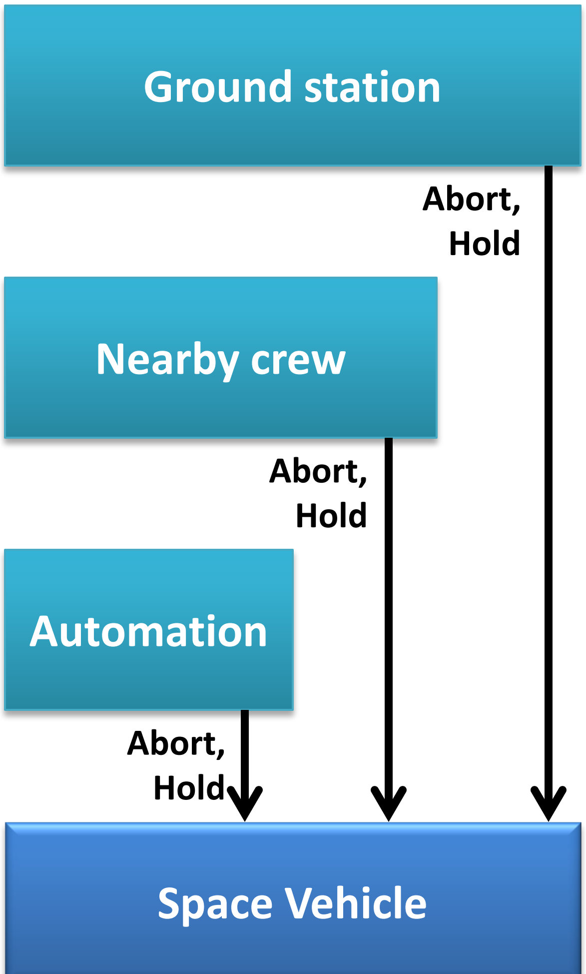
Dangerous Interactions ?

- STPA already performed on individual designs

- System upgrades
  - New controllers, new functionality
  - May interact in hazardous new ways

- Need to start over from blank page?
- Can we leverage existing STPA results?

Multiple Controller Problem

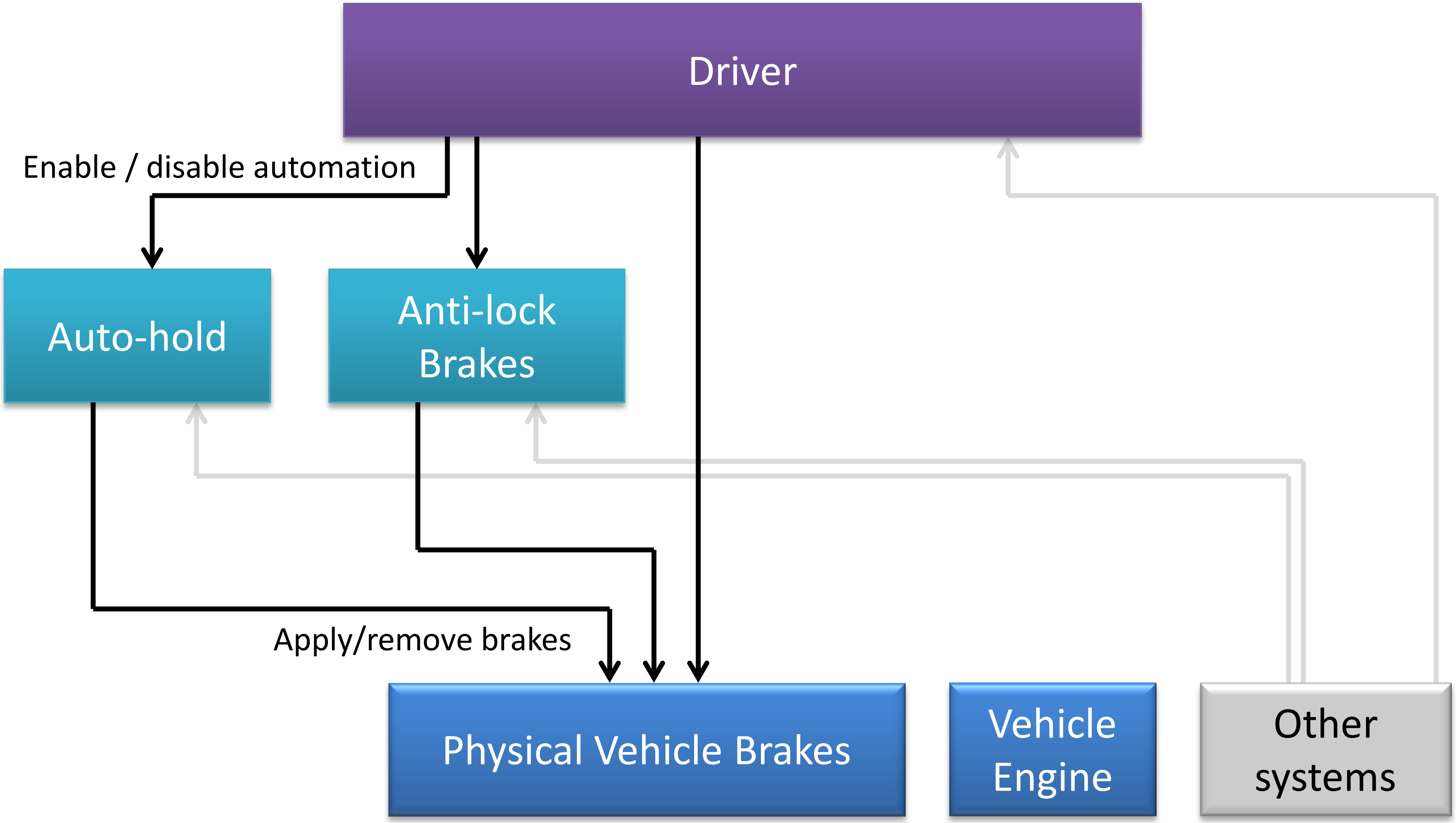


In an emergency situation:

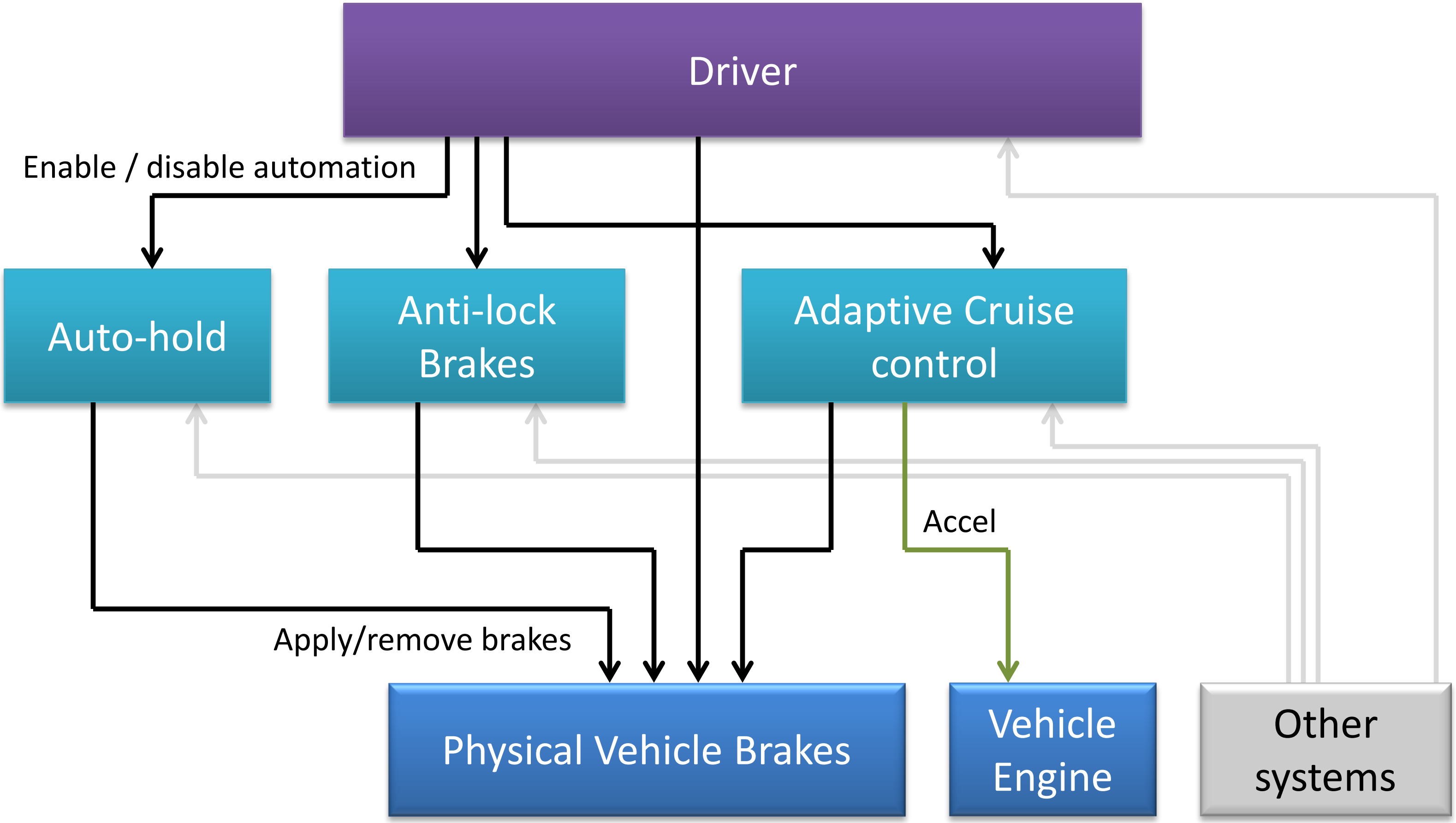
		Automation		Nearby Crew	
		Abort	Hold	Abort	Hold
Automation	Abort	Redundant		Ok	Unsafe
	Hold		Redundant	Undesirable	
Nearby Crew	Abort	Ok	Undesirable	Redundant	
	Hold	Unsafe			Redundant
Ground Station	Abort	Ok	Undesirable	Ok	Undesirable
	Hold	Unsafe		Unsafe	

(See 2012 STAMP Workshop, Ishimatsu)

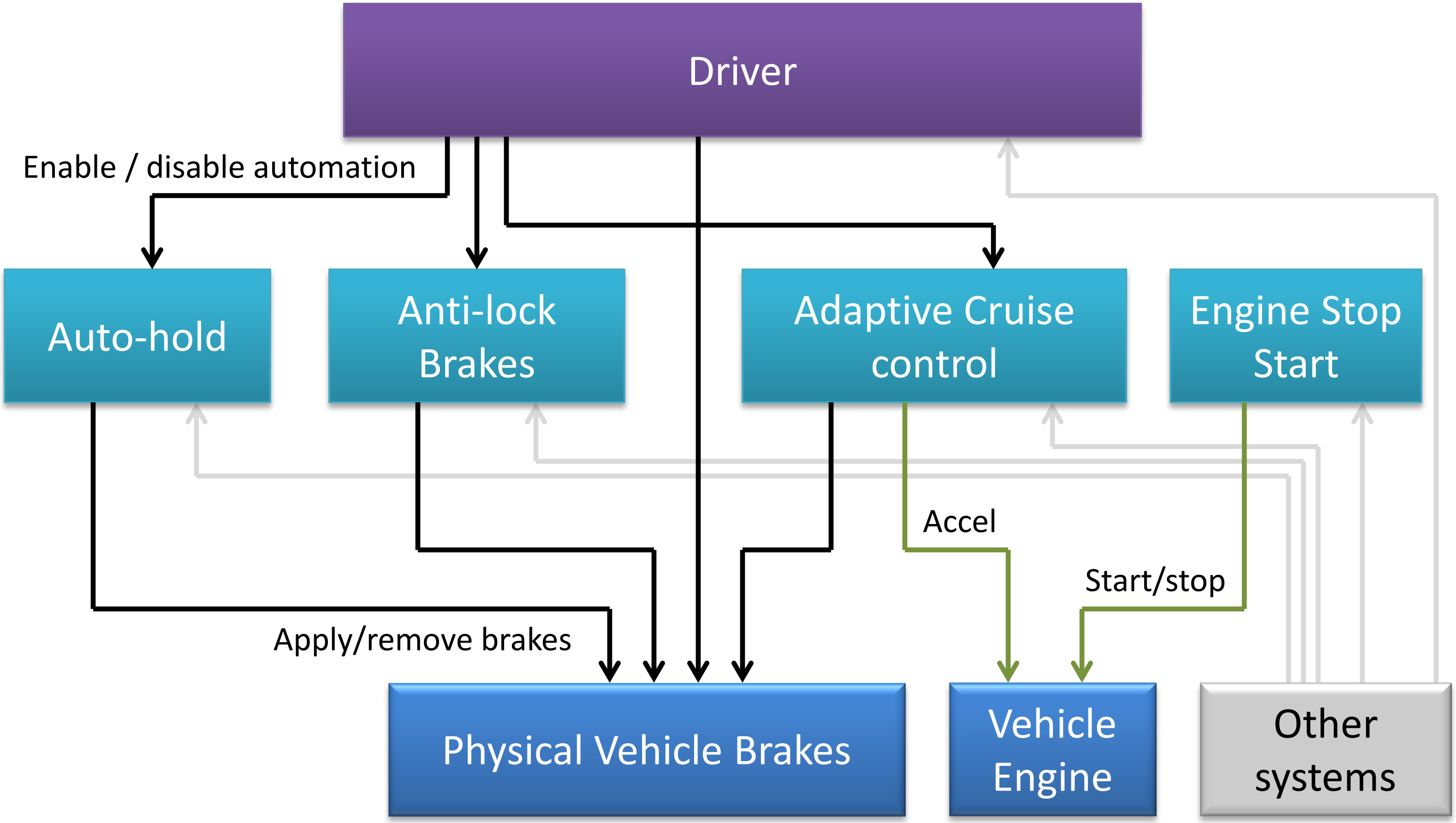
Can it work for many controllers?



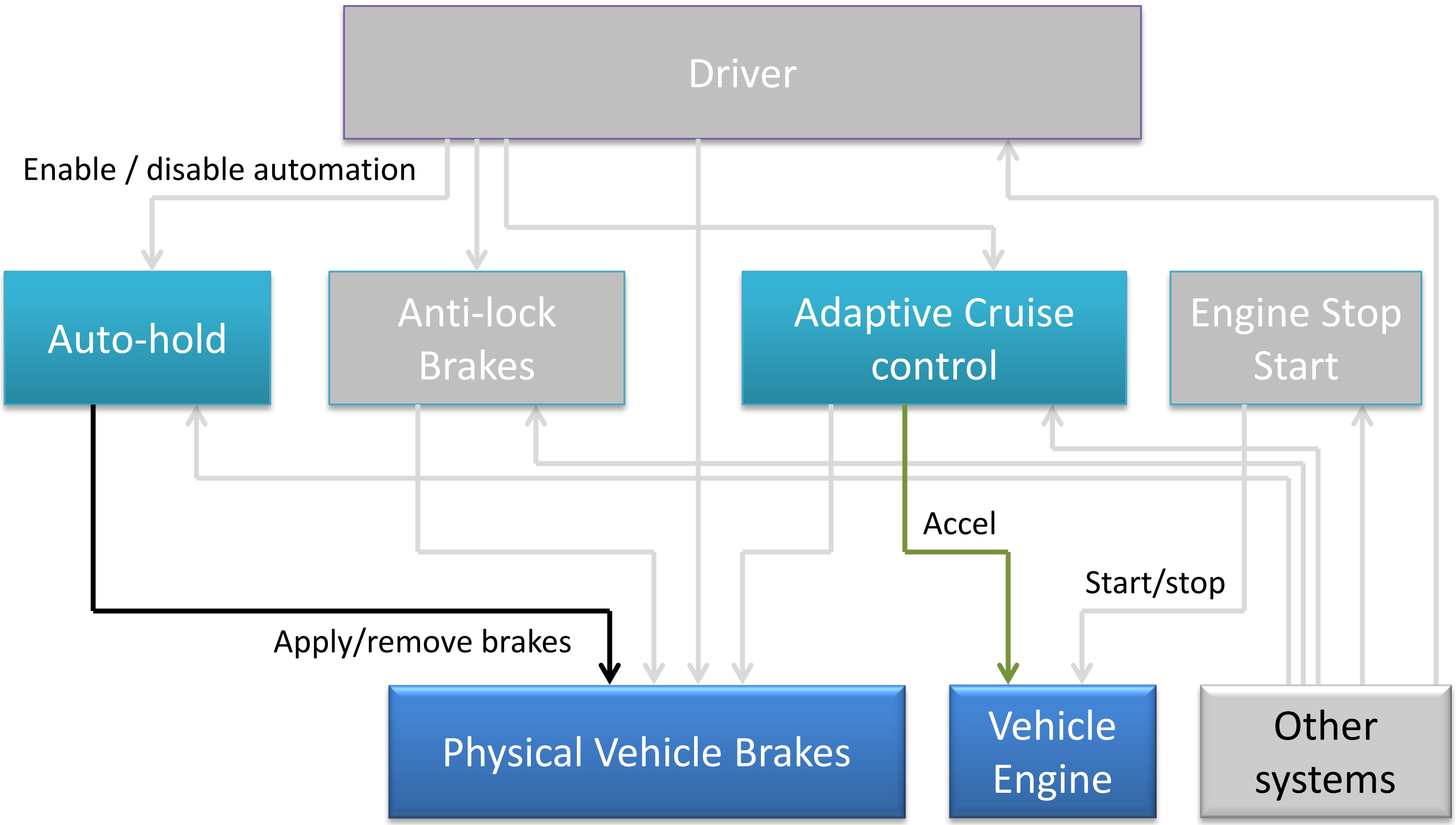
Can it work for many controllers?



Can it work for many controllers?



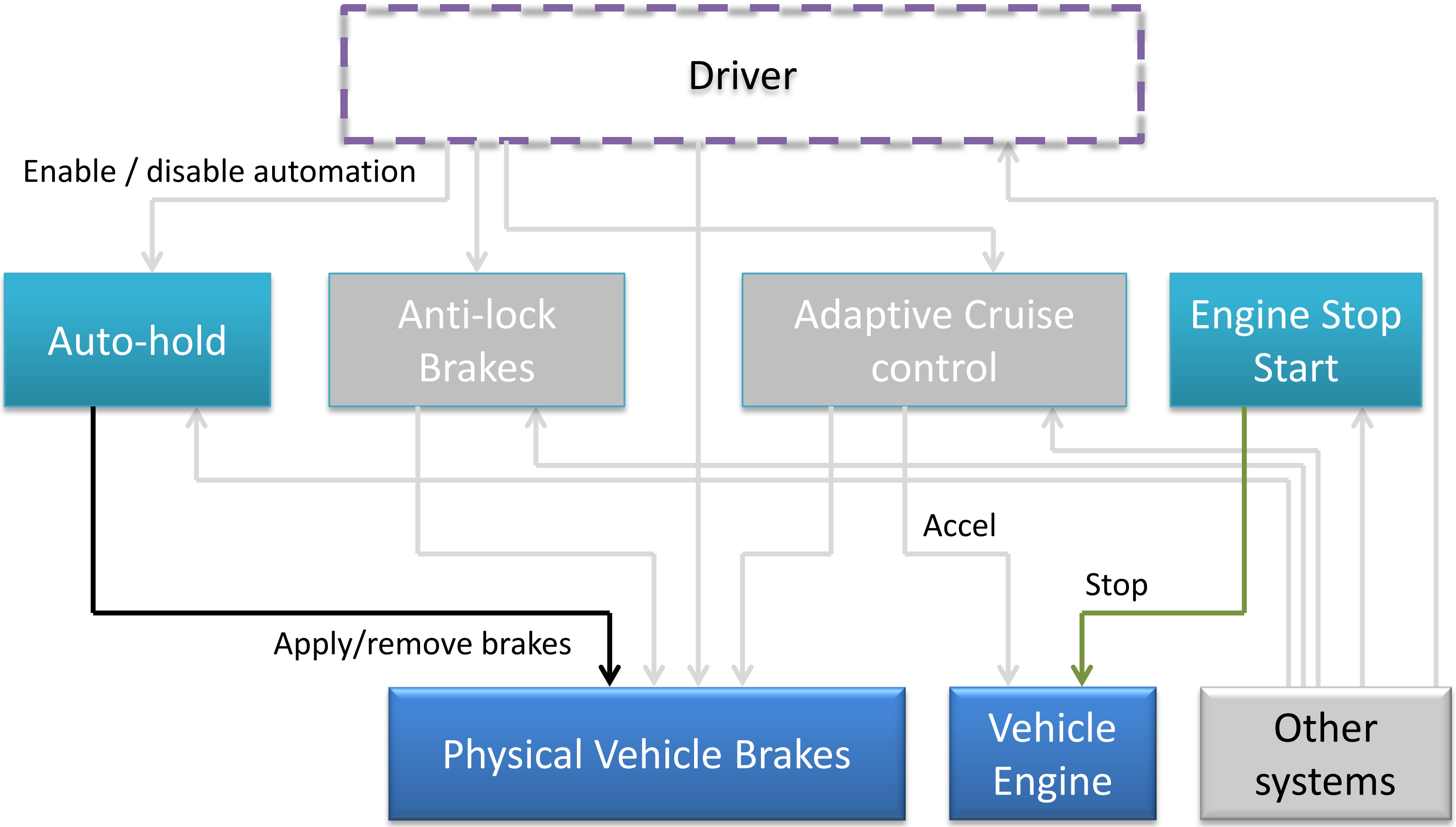
Can it work for many controllers?



Example interaction:

Auto-hold applies brakes
ACC tries to accelerate

Can it work for many controllers?



Example interaction:

Auto-hold applies brakes
Engine-Stop-Start turns engine off
Driver exits vehicle
Driver may be going to look under hood (so be careful starting engine)

Brute force approach (incomplete)

			Auto-Hold			Stop-Start		Driver			
			A	B	C	D	E	D	E	F	G
			Hold	Release	AP	Engine start	Engine stop	Leave	Shift	Gas	Brake
Auto-Hold	1	Hold									
	2	Release									
	3	AP									
Stop-Start	4	Engine start									
	5	Engine stop									
Driver	6	Leave									
	7	Shift									
	8	Gas									
	9	Brake									

Not a good approach for this problem

Brute Force Limitations

- Doesn't scale well
  - Big-O Notation: (characterizes growth rates)
    - $O(n^2)$ analysis points for 2 control actions (2-D matrix)
    - $O(n^3)$ analysis points for 3 control actions (3-D matrix)
    - $O(n^x)$ analysis points for X control actions
- Matrix includes all possible combinations
  - No way to merge rows/columns
  - No way to do abstraction

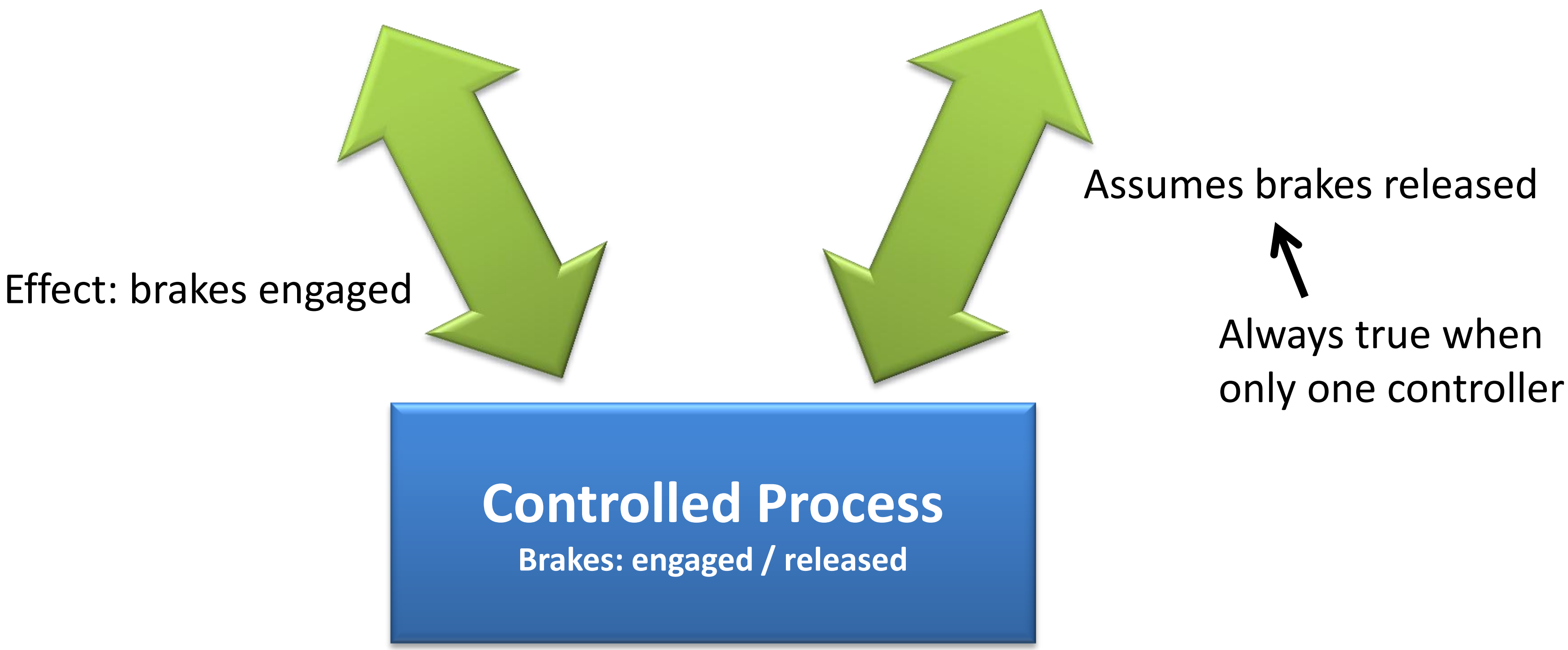
Understanding the Problem

Auto-hold:

- Applies brakes

Adaptive Cruise Control:

- Applies engine throttle



Control Actions and Conditions

	Auto-hold	Adaptive Cruise Control
Conditions assumed/required	Wheels not rotating	Brakes released
Control Action	**Apply Brakes**	**Apply Engine Throttle**
Conditions affected	Brakes engaged	Increased engine speed

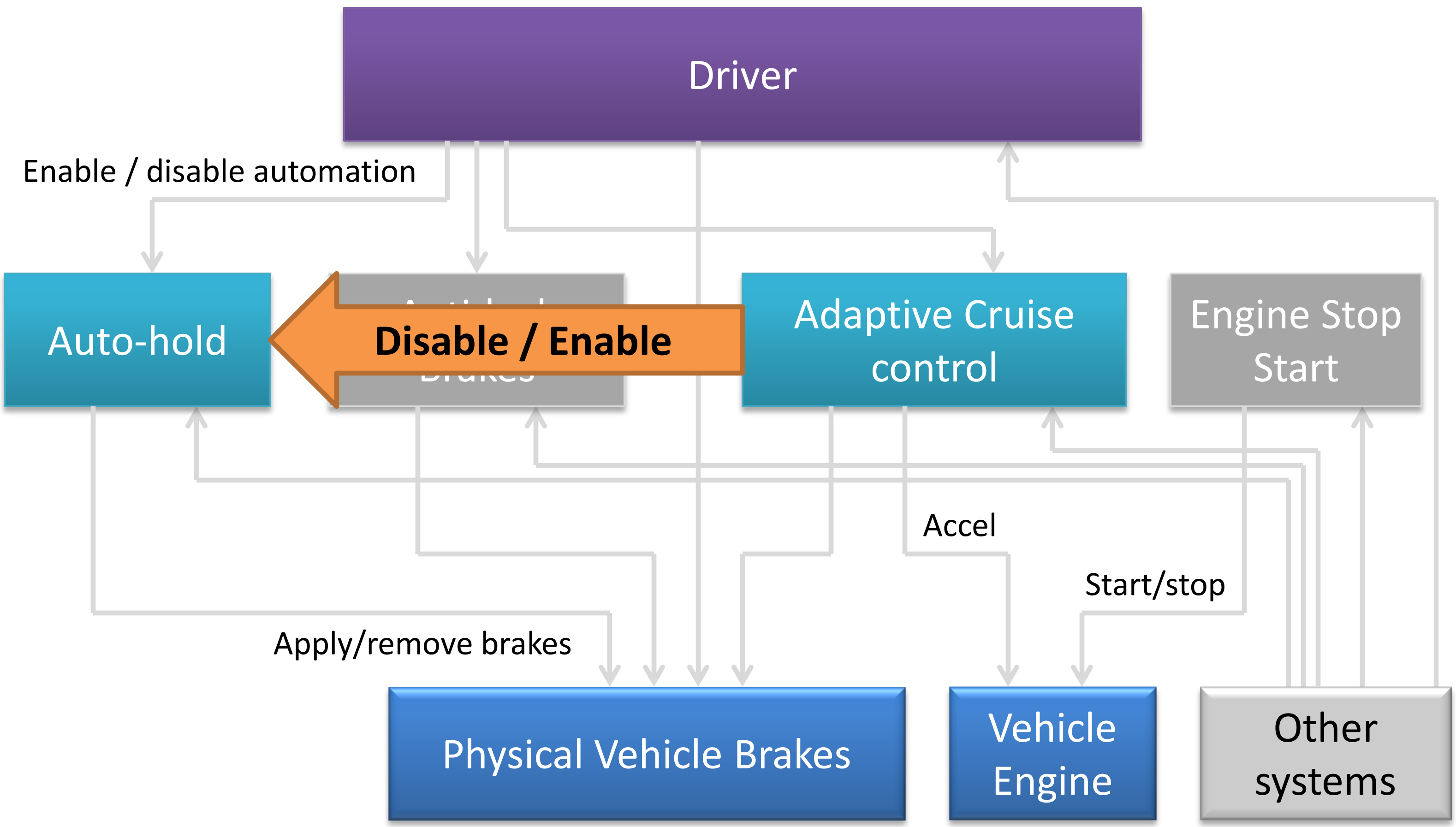
Control Actions and Conditions

	Auto-hold	Adaptive Cruise Control
Conditions assumed/required	Wheels not rotating	Brakes released
Control Action	**Apply Brakes**	**Apply Engine Throttle**
Conditions affected	Brakes engaged	Increased engine speed

New constraint that didn't exist for any individual system!

How could this combination ha ?
- ACC stops on a hill following l , leading car accelerates and ACC applies throttle to follow. AH c rake force is insufficient, and automatically increases brake
- Possible solution: Update design so AH is disabled when ACC active

Updated Control Structure



Possible Solution

	Controller 1		Controller 2	Controller 3	
	Command A	Command B	Command C	Command D	Command E
Design Assumptions & Required Conditions					
Effect on the System					

- A "conditions table" can record all the information needed to identify multiple-controller conflicts.
  - Grows with O(2n) – scalable!

New Approach

Stop-Start	Engine restart	Engine stop	Etc.
Conditions Assumed / Required	**Vehicle Held:** Yes (i.e. **Brakes:** On **|** **Range:** Park **|** **EPB:** Yes) **Wheels Rotating:** No **Restart Possible:** Yes (i.e. **Battery Charge:** High) **Driver Present:** Yes **Range**:!=P,R,N	**SS Enabled:** Yes **AUTO-STOPPED:** Yes **Vehicle Held:** Yes (i.e. **Brake:** On **|** **EPB:** Yes) **Restart Possible:** Yes (i.e. **Battery Charge:** High) **Driver Present:** Yes **Gas Pedal:** No **Auxiliary Power Needs:** Low **Range:** !=P,R,N	
Conditions Affected	**Propulsion:** On **Idle Torque:** Yes **Electrical Power:** On - power reduced ~2s **AUTO-STOPPED:** No	**Propulsion:** Off **Idle Torque:** No **Electrical Power**: Off **AUTO-STOPPED:** Yes	

New Approach

	AH				ESS			ACC w/SG		Driver			
	Hold	Release	AP	Apply EPB	Engine start	Engine stop	Apply EPB	Accel.	Decel.	Leave	Shift	Gas	Brake
Design assumptions / Required Conditions													
System states / conditions changed													

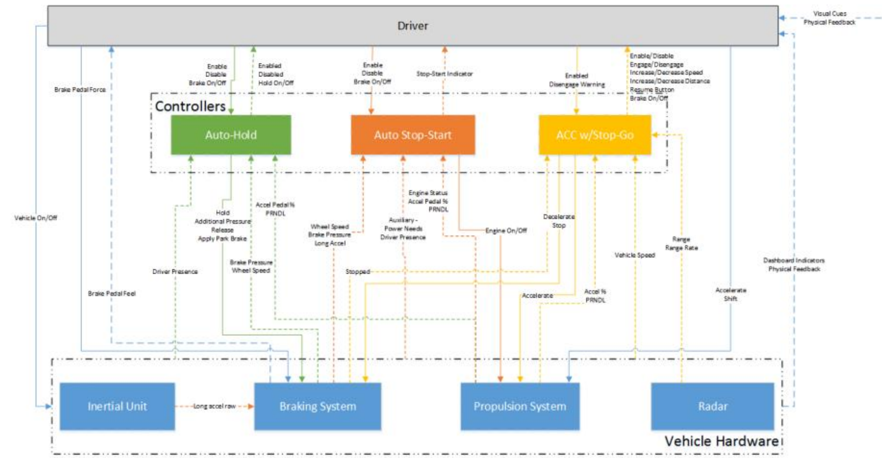
O(2n) – This is scalable!

New Approach

	AH			ESS		ACC w/SG		Driver	
	Hold	Release	AP	Engine start	Engine stop	Accel	Decel	Accel	Brake
Design assumptions / Conditions required to be effective	Car stopped; Battery power available; Little or no propulsion torque; Ability assume brake control	Driver present (to prevent rollback)	Battery power available; Little or no propulsion torque; AH controls brakes (AH in hold mode)	Battery power available; Engine off	Vehicle stopped	Propulsion ready (engine running, in gear); Brakes not applied	Battery power available; Ability to assume brake control; Little or no propulsion torque	Propulsion ready (engine running, in gear) Brakes not applied	Power available (power brakes); Little or no propulsion torque; Brake pedal connected
System states / conditions changed	AH controls brakes; Brakes applied; Brake pedal disconnected	AH releases brake control (brake pedal connected)	AH braking force increased	Propulsion ready after 2s (engine running, idle propulsion torque), electric power significantly reduced for 2s, power available after 2s (battery charging, power brakes, etc)	Propulsion not ready (engine off, no propulsion torque); Limited battery power available	Increased propulsion torque	ACC controls brakes; Brakes applied; Brake pedal disconnected	Increased propulsion torque	Driver controls brakes; Brakes applied

O(2n) – This is scalable!

Results 1



Control actions:

- Auto-Hold applies the parking brake
- ACC attempts to accelerate

Problems/Conflicts:

- ACC does not have the authority to dis-engage the EPB
- Auto-Hold attempting to secure the vehicle while it’s held by ACC

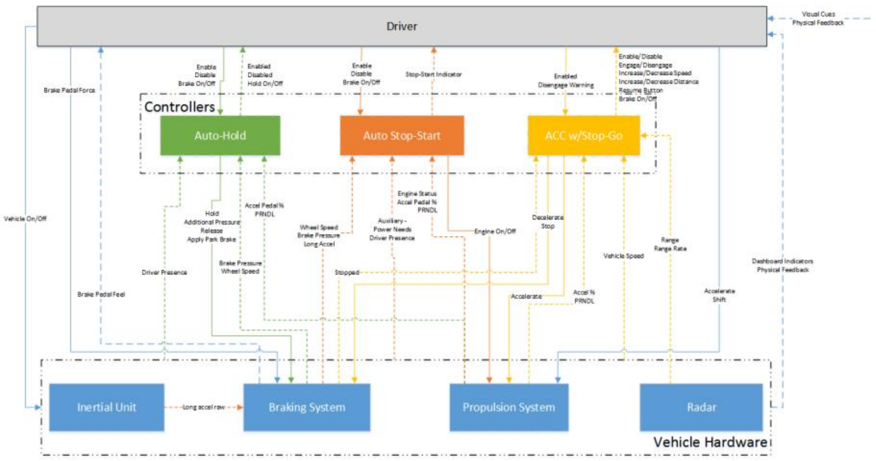
Potential Solutions :

- R-1: ACC may disengage EPB
- R-2: ACC may monitor the state of the EPB
- R-3: EPB may monitor the state of ACC
- R-4: Issuing the EPB turns the features ‘off’
- R-5: Auto-Hold could be disabled when ACC is active (ACC can hold car at stop)

Results 2

Context:

- AH is holding brakes
- Battery charge is low (but sufficient for restart)
- ESS turns engine off to save fuel
- Reduced torque causes vehicle to move i.e. downhill



Controller Response:

- AH attempts to increase brake pressure
- Stop-Start attempts to start vehicle

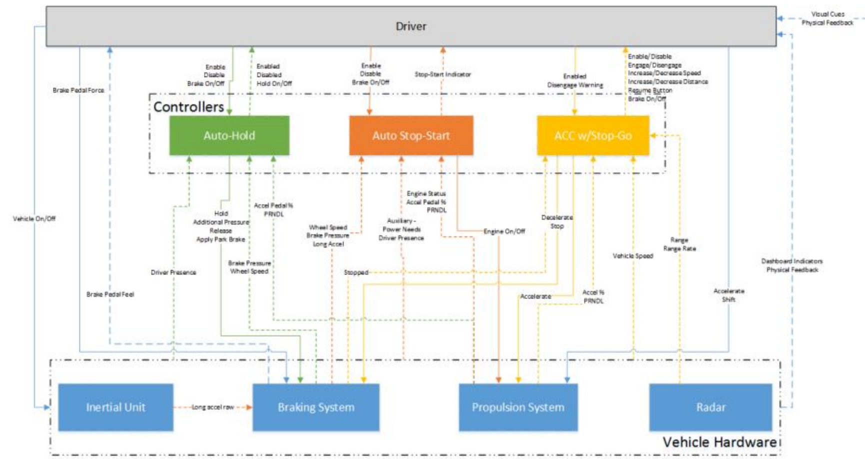
Problem:

- Battery voltage drops, vehicle starts but cannot increase brake pressure for 2s

Potential Solutions / Requirements:

- R-1: AH pump must operate at a low battery voltage
- R-2: ESS must warn AH so pressure can be increased before engine turns off
- R-3: Battery threshold must be sufficient to guarantee simultaneous restart and brake pump

Results 3



Context:

- Auto-Hold is holding vehicle
- ESS stops engine to save fuel
- Driver shifts to reverse
- Driver steps on gas to back up

Problem:

- ESS cannot *Start* the engine (prevented by FMVSS 102)
- AH cannot *Release* (insufficient engine torque)

Potential Solutions / Requirements?

Summary

- Provides a way to analyze interactive effects
  - Can be automated
- Scalable to very complex systems, more than 2 control actions
- Can identify missing feedback / control in the design
- Leverages existing STPA analysis, requirements for independent systems
- Provides a way to identify new Process Model Variables