



TRAFFIC MANAGEMENT FOR URBAN AIR MOBILITY

Suda Bharadwaj, Steven Carr, Natasha Neogi, Hasan Poonawala,
Alejandro Barberia Chueca, and Ufuk Topcu



Outline

- Motivation
- Shields for UAM



Motivation

- Urban Air Mobility (UAM) ecosystem
 - Competing requirements and priorities
 - Complex network of vehicles, large range of control actions



NASA Illustration



Air Traffic Management

Fleet

Mission

Congestion

Local Fleet
Properties

Line-of-sight
Collision
Energy

Individual
vehicle

- Centralized planning grows exponentially with the number of vehicles
- Each agent is unaware of the whole fleet
- Split the problem
 - Individual onboard route
 - Shielding airspace regions to enforce safety constraints



Hierarchical approach is needed!

- Vehicles cannot have full knowledge of the entire fleet!
- Collision avoidance – can be done on board.
- Congestion requirements? Need higher level of planning!



Requirements

- Enforcing FAA Regulations
 - Visual line-of-sight (VLOS) for operators

§107.3 Visual line of sight aircraft operation

With vision that is unaided by any device other than corrective lenses, the remote pilot in command, the visual observer (if one is used), and the person manipulating the flight control of the small unmanned aircraft system must be able to see the unmanned aircraft throughout the entire flight

FAA Commercial Drone Rules Part 107



Shield Synthesis

Bettina Könighofer
Roderick Bloem
Stefan Pranger



Ufuk Topcu
Scott Niekum
Mohammed Alshiekh



TEXAS
The University of Texas at Austin

Rüdiger Ehlers



Universität
Bremen

Thomas Henzinger
Guy Avni
Krishnendu Chatterjee



Institute of Science and Technology

Nils Jansen



Radboud
Universiteit
Nijmegen

Sebastian Junges



Rayna Dimitrova

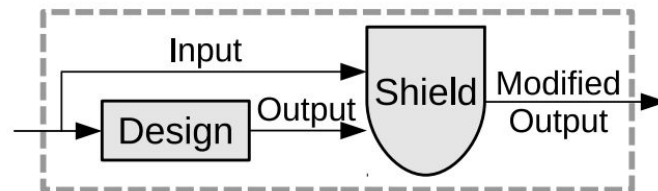


UNIVERSITY OF
LEICESTER



Shields!

- What is a shield *not*?
 - A planner
- What do we want from a shield?
 - Agnostic of underlying goals
 - A run-time enforcer of safety!
 - Guarantee progress (?)

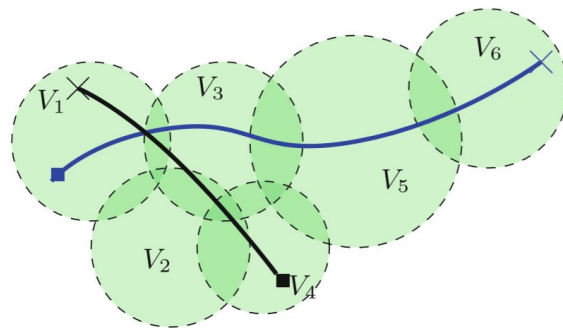




Shields for UAM

Setting

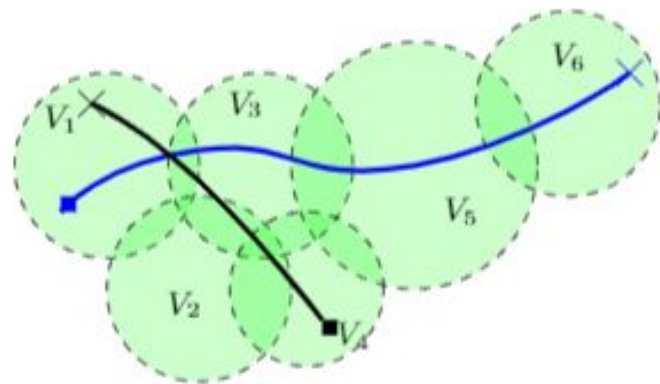
- Vehicles complete trips between origin-destination pairs by moving through *sectors*.
- Sectors cannot accept vehicles if *maximum operational density* will be exceeded.
- Sectors can send idle vehicles out to neighbouring sectors -> *can lead to conflicts!*
- During runtime – avoid conflicts and satisfy safety requirements!





Problem Setting

- Network of k UAM sectors $V_1, \dots, V_k$
- Set of UAM vehicles
 - Tasks (origin-destination)
 - Two action types: loiter or hand-off
- Each agent is modeled by an MDP





Previous approach

Safety Shields for Multi-Agent Systems

ACC-19

Task: Enforce global safety property

1. Quantitative interference costs c_{INT} :
 - Counting cost function
 - Different costs for interferences with different agents
2. Fair Shielding
 - Do not always interfere with the same agent repeatedly



Previous approach

- Shield synthesis is centralized!
- Key assumptions:
 1. **Global knowledge** – shield knows where all agents are at all times
 2. **Global control** – shield can overwrite any agent at any time

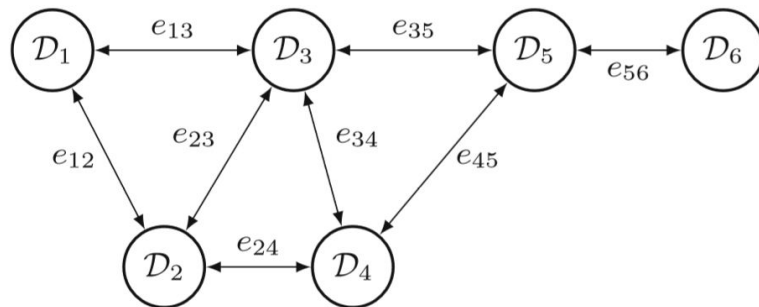
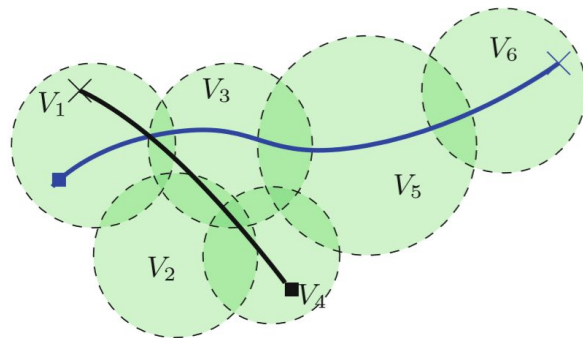
Can we synthesize shields that
satisfies safety properties defined
across a joint output but cannot
control the whole output?



Localized Shield Synthesis

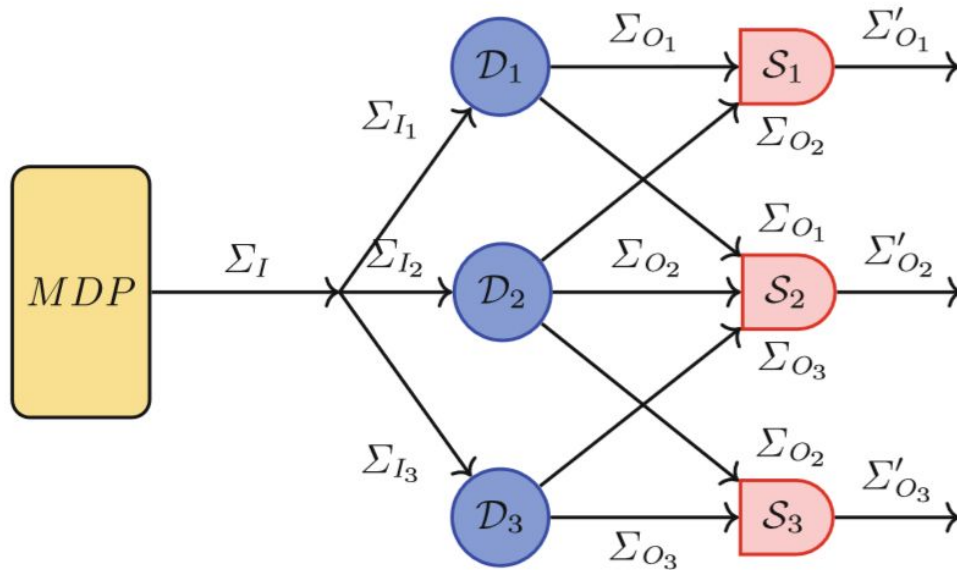
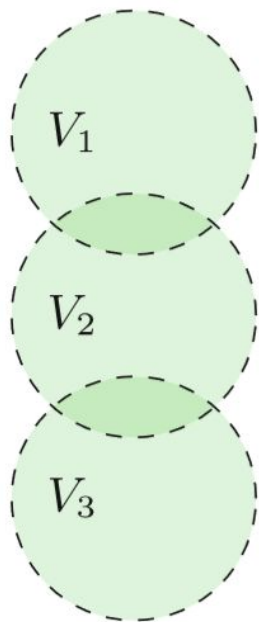
Key assumption:

- The outputs of each sector only affects *connected sectors*
- Connections are defined using a *directed graph*





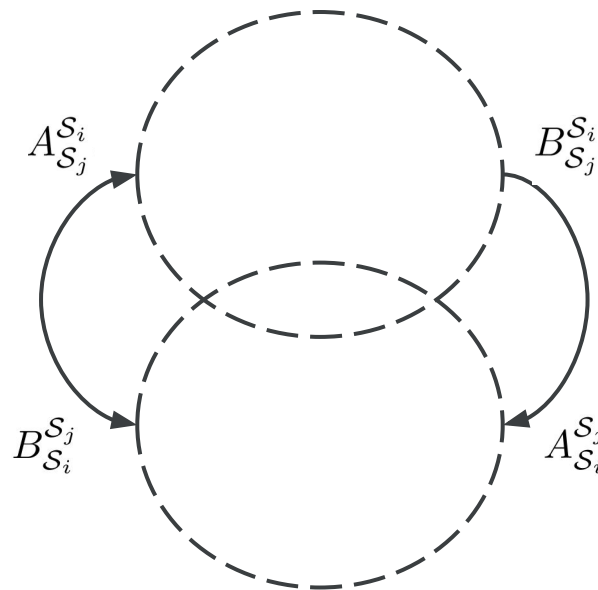
Architecture





Contracts

- How do we guarantee a sector not prevent its neighbours from satisfying safety?
 - Assume-guarantee contracts!
- We define a contract as: $\mathcal{C}_{S_j}^{S_i} = (A_{S_j}^{S_i}, B_{S_j}^{S_i})$
 - *Assumptions* shield i makes on outputs of shield j .
 - *Guarantees* shield i provides to shield j .





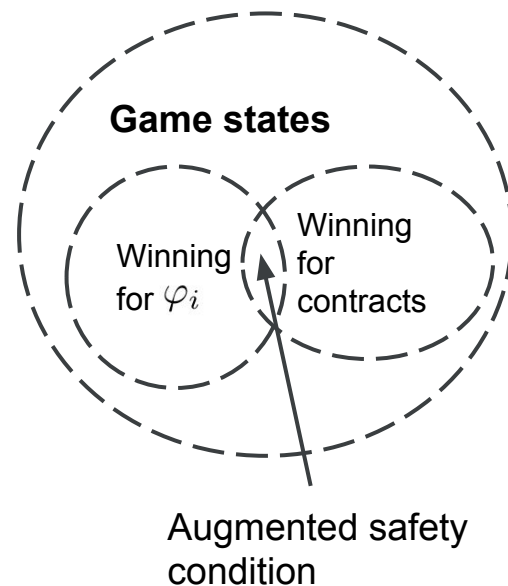
Synthesis overview

1. For each shield construct a game from the given safety specification and augment with contracts.
2. Solve for a permissive strategy.
3. Compute locally optimal deterministic strategy.



Game construction

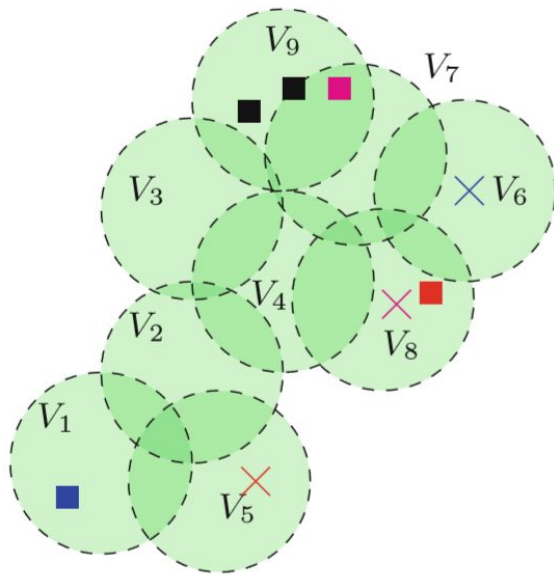
- Game with acceptance condition defined by original safety requirement - e.g., congestion.
- Augment with contract requirements - still a safety game!
- We solve the game using Small bUt Complete GROne Synthesizer (Slugs)





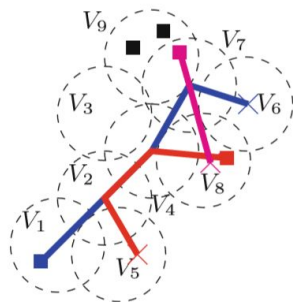
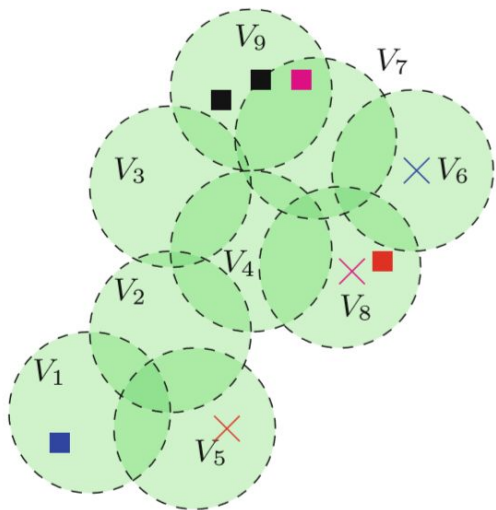
Simulation setting

- Agents start at squares and move to same coloured X.
- Safety:
 - Must not violate congestion requirements
 - Must not pass through same sector at same time
- Shields can make agents wait
- Progress guarantee:
 - Agents must not wait more than given time
- Contracts:
 - Shield can pass max no. agents to neighbours.
 - Shield is allowed to make agents wait.

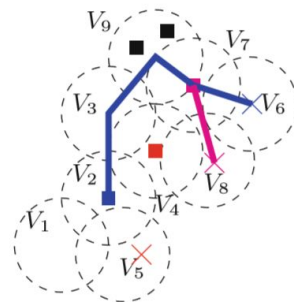




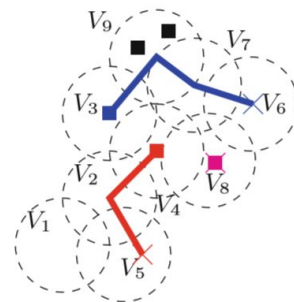
Simulation results



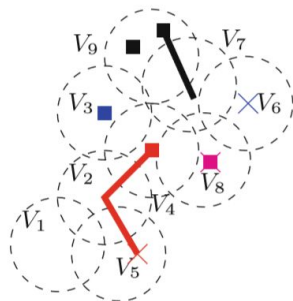
(a) Unshielded: Time = 0



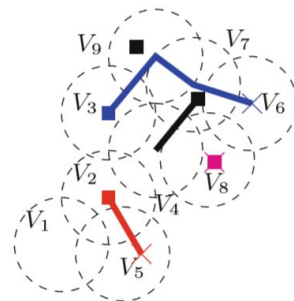
(b) Unshielded: Time = 1



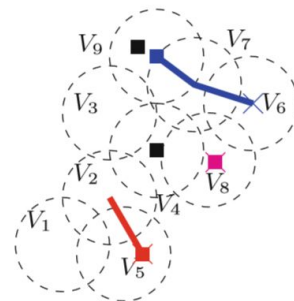
(c) Unshielded: Time = 2



(d) Shielded: Time = 2



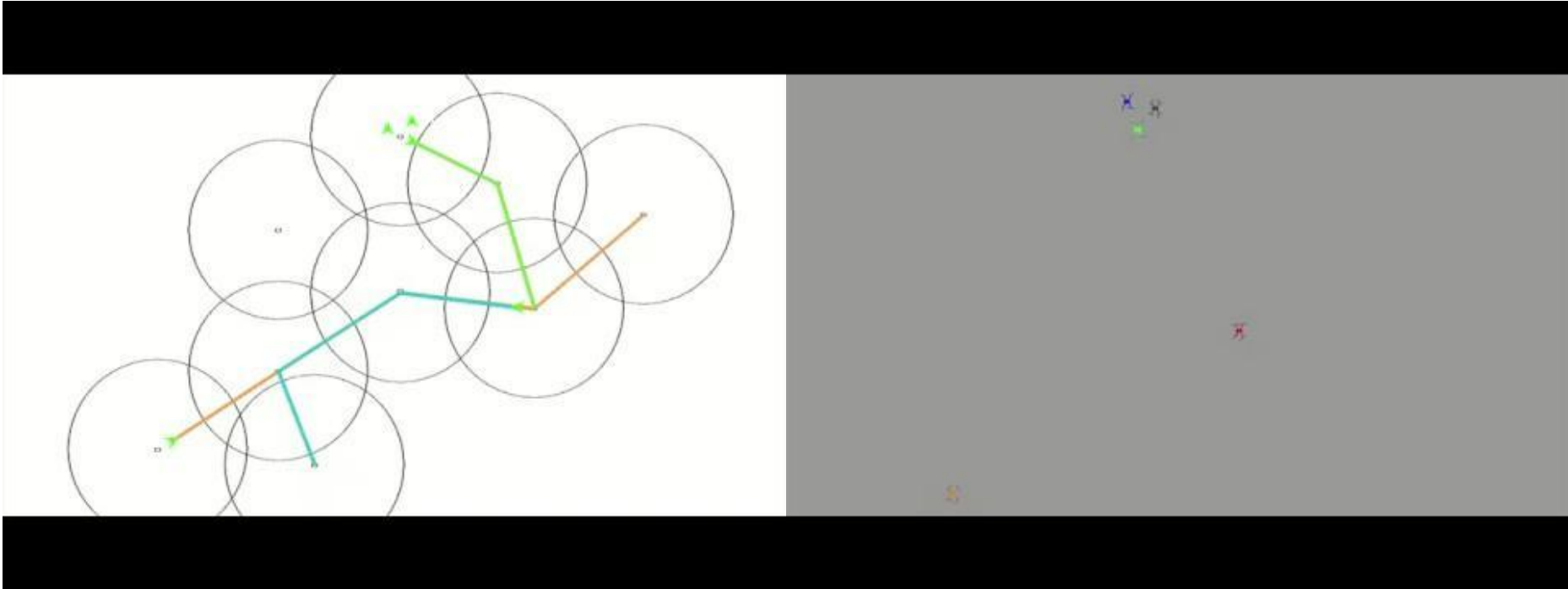
(e) Shielded: Time = 3



(f) Shielded: Time = 4

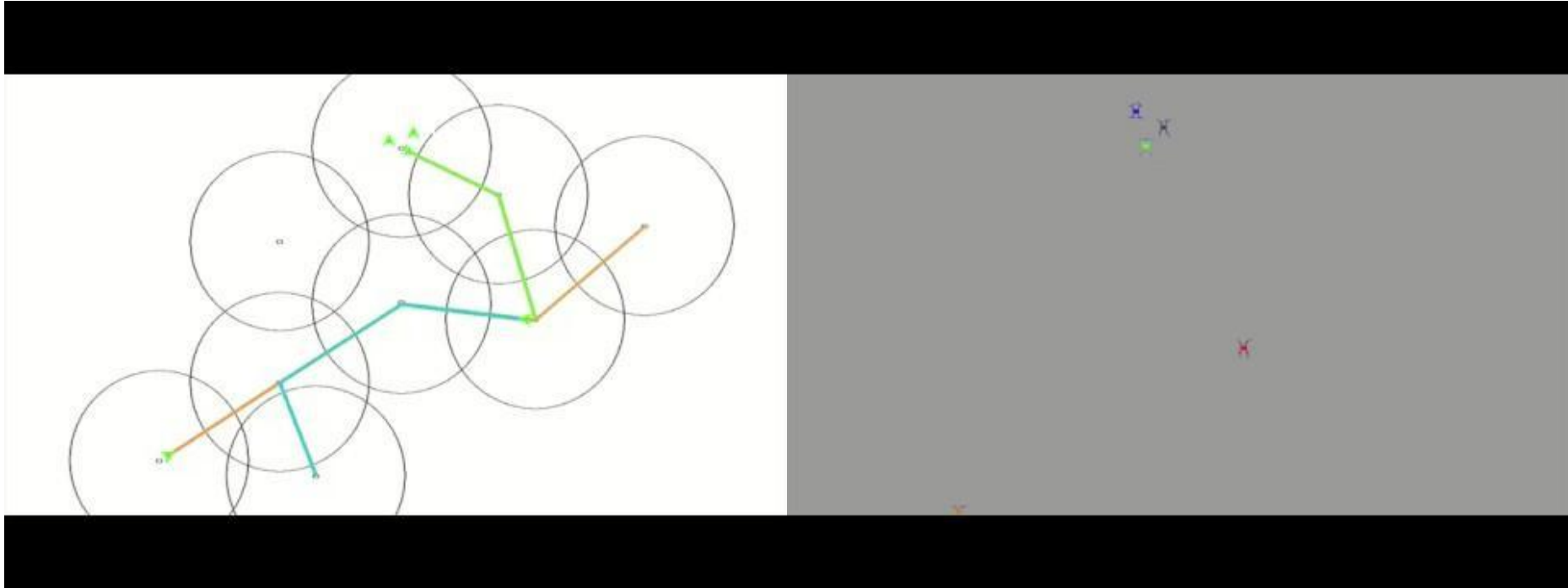


Simulation - unshielded





Simulation - shielded





Ongoing/Future work

- Characterize when safety is always guaranteed - counterexamples!
- Onboard shields for individual safety properties – collision avoidance, energy consumption, LOS, etc
- Challenges:
 - Decentralization needed
 - Consensus required amongst shields
 - Deadlock avoidance



Decentralized shields

- Currently:
 - Fully decentralized shielding procedure
 - Consensus sets agent *priority*
 - Priority is tied to agent *progress* – guarantees deadlock avoidance
 - Also allows for limited communication



Next steps

- Connect sector shields with local decentralized shields.
- Guarantee deadlock avoidance with multiple sources of potential interference!





The University of Texas at Austin

Cockrell School of Engineering