Robustness

Robustness = operate efficiently despite perturbations

Non-nominal conditions and component failures

Changes in user behavior

Malicious attacks

Things can go terribly bad if (design) $\not\Rightarrow$ (robustness)

Robustness

Robustness = operate efficiently despite perturbations

Non-nominal conditions and component failures

Changes in user behavior

Malicious attacks

Things can go terribly bad if (design) $\not\Rightarrow$ (robustness)

Robustness

Robustness = operate efficiently despite perturbations

Non-nominal conditions and component failures

Changes in user behavior

Malicious attacks

Things can go terribly bad if (design) $\not\Rightarrow$ (robustness)

Robustness

Robustness = operate efficiently despite perturbations

Non-nominal conditions and component failures

Changes in user behavior

Malicious attacks

The Register®
Biting the hand that feeds IT

DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE EMERGENT TECH BOOTNOTES LECTURES

US spy drone hijacked with GPS spoof hack, report says

Electronic warfare comes of age – in Iran

By Dan Goodin 15 Dec 2011 at 23:27 158 SHARE ▼

The US stealth drone broadcast last week on Iranian state television was captured by spoofing its GPS coordinates, a hack that tricked the bird into landing in Iranian territory instead of where it was programmed to touch down, *The Christian Science Monitor* reported.

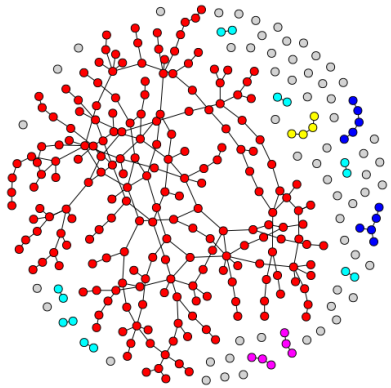
The 1700-word article cited an unnamed Iranian engineer who said he's orkings of the American bat-wing RQ-170 Sentinel issing over Iranian airspace. He said the spoofing craft "land on its own where we wanted it to, without emote-control signals and communications" from the

Things can go terribly bad if (design) $\not\Rightarrow$ (robustness)

Research Focus and Challenges

Transportation

Robotics

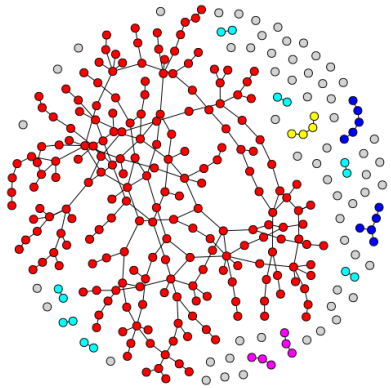


Complex Networks

Research Focus and Challenges

Transportation

Robotics



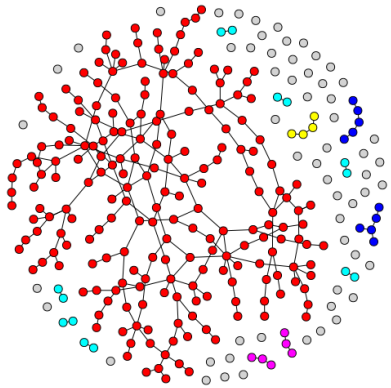
Complex Networks

Physical components + Network

Research Focus and Challenges

Transportation

Robotics



Complex Networks

Physical components + Network

☹ Challenges: increasing complexity, human decisions, operate at capacity

☺ Opportunities: (technology/network) + (control/optimization)

Organization of Dissertation and Talk Outline

Application to traffic and transportation

Robustness of networks with app routing

Application to robotics

Trajectory planning for secure navigation

Organization of Dissertation and Talk Outline

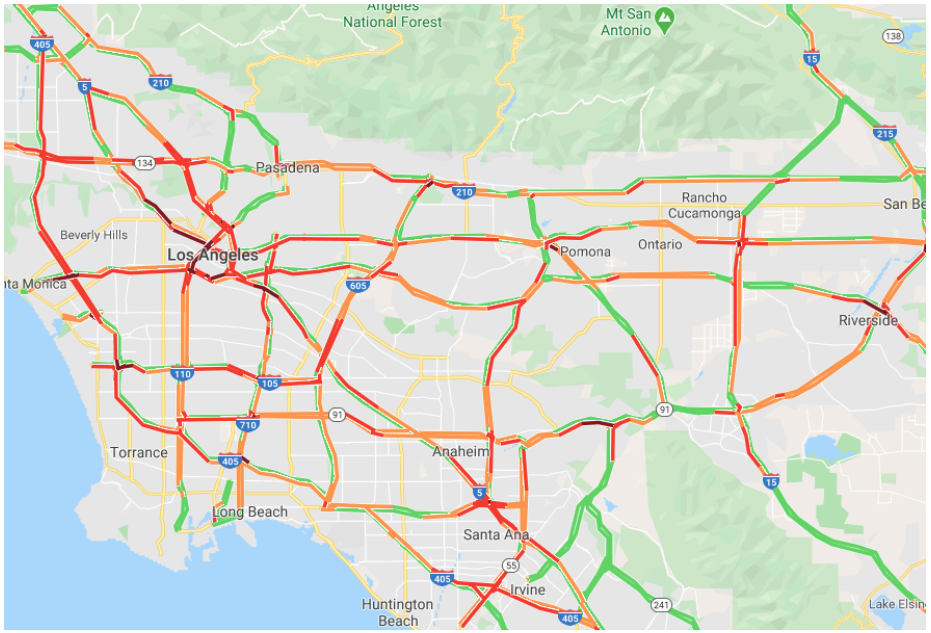
Application to traffic and transportation

Robustness of networks with app routing Part I

Application to robotics

Trajectory planning for secure navigation

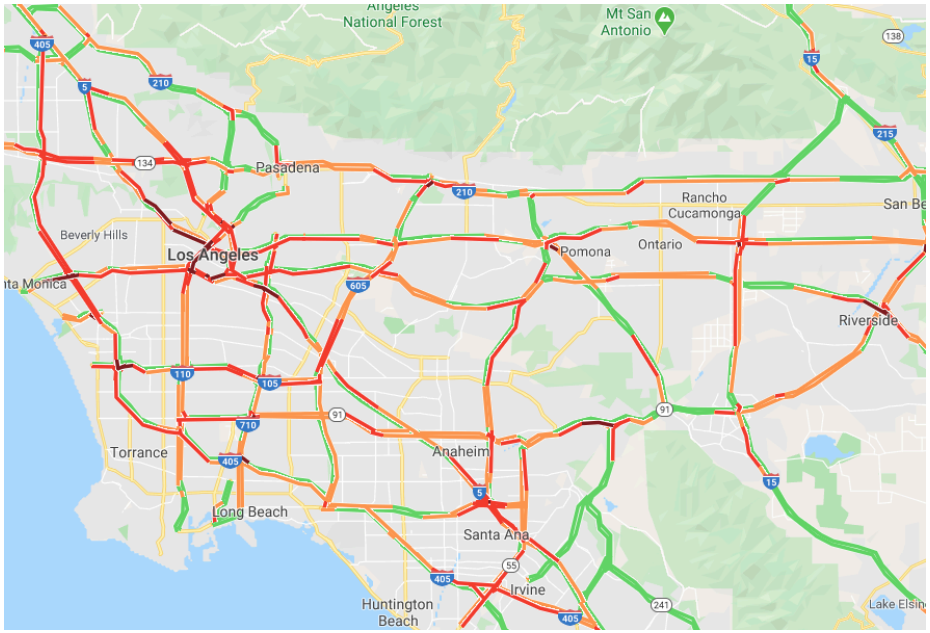
Transportation and Needs for Robustness



(source: Google)

- Transportation: 9% US GDP
- Congestion: wastes 3B Gallons of fuel every year
- Large-scale, complex, rich nonlinear dynamics

Transportation and Needs for Robustness



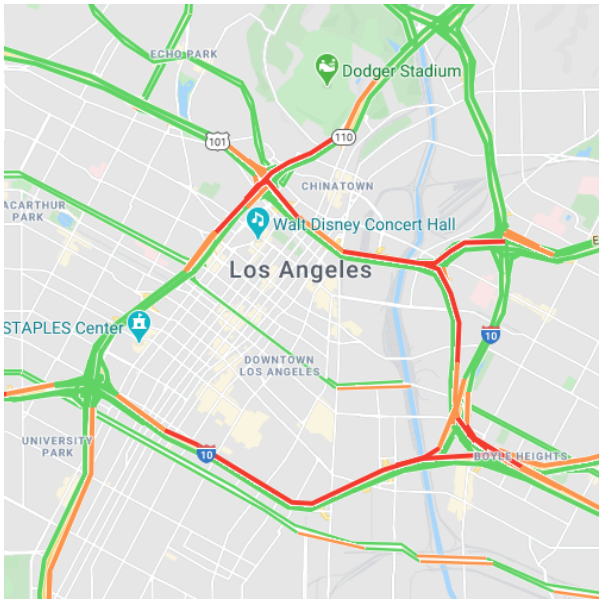
(source: Google)

- Transportation: 9% US GDP
- Congestion: wastes 3B Gallons of fuel every year
- Large-scale, complex, rich nonlinear dynamics

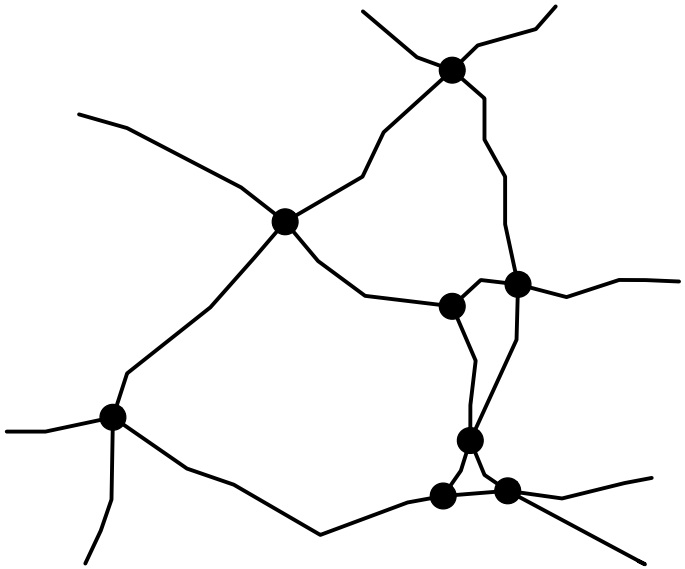
Robustness is extremely relevant problem

- 100 years old and operating at capacity limits
- Things can go really bad (Atlanta 2014, Beijing 2010, Houston 2005, NY 2001)

Modeling Traffic



(source: Google)



Traffic network topology:

(1) Highways  each transfers traffic flows

(2) Junctions ● exchange traffic flows between highways

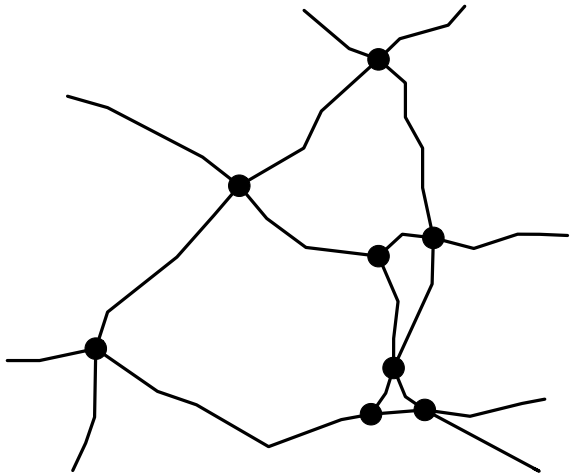
Dynamics in Traffic Networks

(1) **Highways**

Modeled as vehicle accumulators

$$\dot{x}_\ell = f_\ell^{\text{in}}(x) - f_\ell^{\text{out}}(x_\ell)$$

Classical models: each highway has a single flow variable



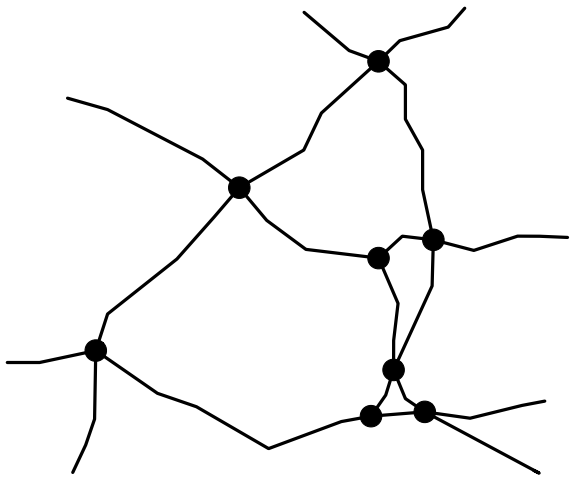
Dynamics in Traffic Networks

(1) **Highways**

Modeled as vehicle accumulators

$$\dot{x}_\ell = f_\ell^{\text{in}}(x) - f_\ell^{\text{out}}(x_\ell)$$

Classical models: each highway has a single flow variable

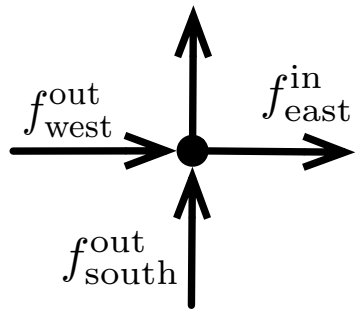


(2) **Junctions**

Transfer flows between highways

$$f_{\text{east}}^{\text{in}} = r_{\text{west}\to\text{east}} f_{\text{west}}^{\text{out}} + r_{\text{south}\to\text{east}} f_{\text{south}}^{\text{out}}$$

Routing is the result of human preferences



The Open Problem of Real-Time Information

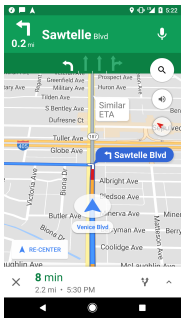
$$\dot{x}_\ell = f_\ell^{\text{in}}(x) - f_\ell^{\text{out}}(x_\ell)$$

$$f_{\text{east}}^{\text{in}} = r_{\text{west}\to\text{east}} f_{\text{west}}^{\text{out}} + r_{\text{south}\to\text{east}} f_{\text{south}}^{\text{out}}$$

The Open Problem of Real-Time Information

$$\dot{x}_\ell = f_\ell^{\text{in}}(x) - f_\ell^{\text{out}}(x_\ell)$$

$$f_{\text{east}}^{\text{in}} = r_{\text{west}\to\text{east}} f_{\text{west}}^{\text{out}} + r_{\text{south}\to\text{east}} f_{\text{south}}^{\text{out}}$$

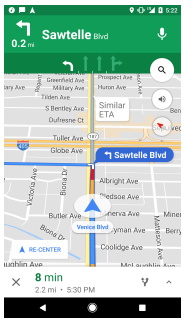


- Effective optimal-route algorithms
- Real-time congestion information

The Open Problem of Real-Time Information

$$\dot{x}_\ell = f_\ell^{\text{in}}(x) - f_\ell^{\text{out}}(x_\ell)$$

$$f_{\text{east}}^{\text{in}} = r_{\text{west}\to\text{east}} f_{\text{west}}^{\text{out}} + r_{\text{south}\to\text{east}} f_{\text{south}}^{\text{out}}$$



- Effective optimal-route algorithms
- Real-time congestion information

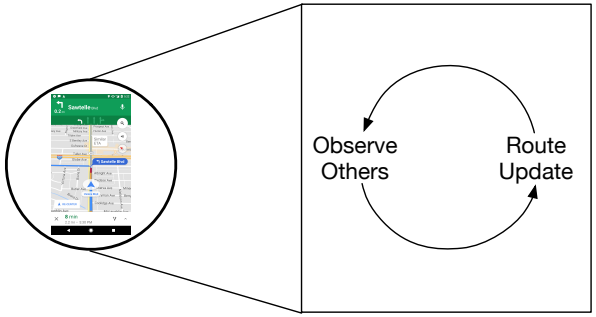
Open problem

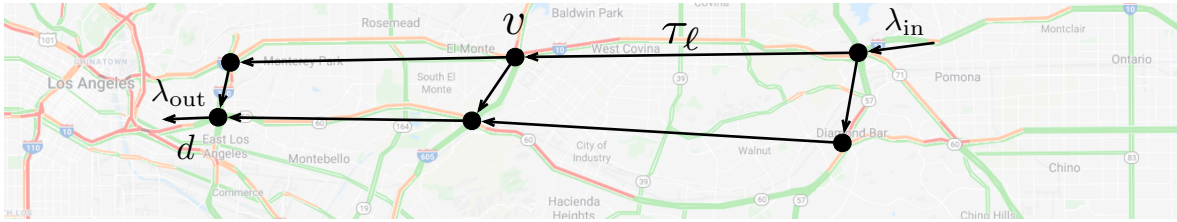
Real-time congestion information $\leftrightarrow$ Robustness of transportation system

Robustness:

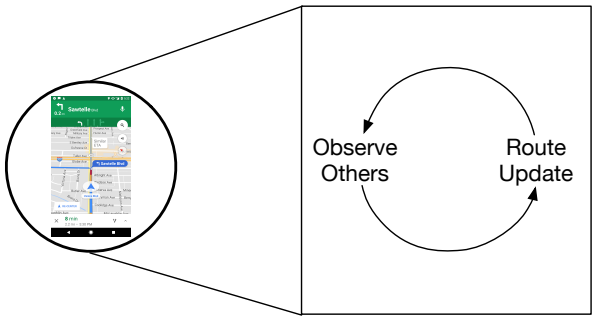
- Transfer largest traffic flows
- Despite noncooperative human behaviors

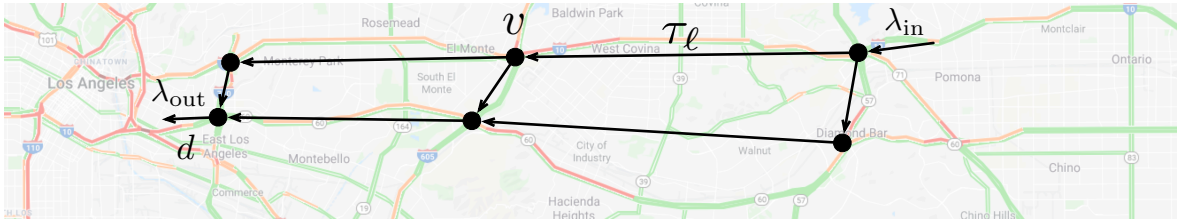
Modeling Navigation Apps





Modeling Navigation Apps

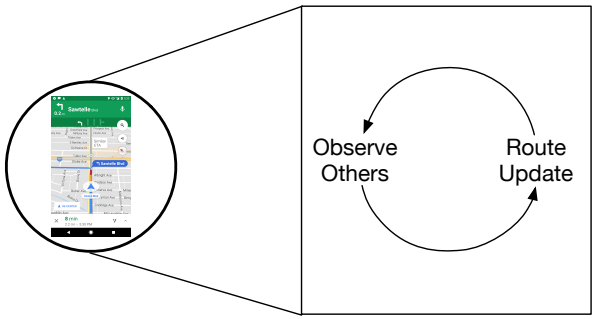


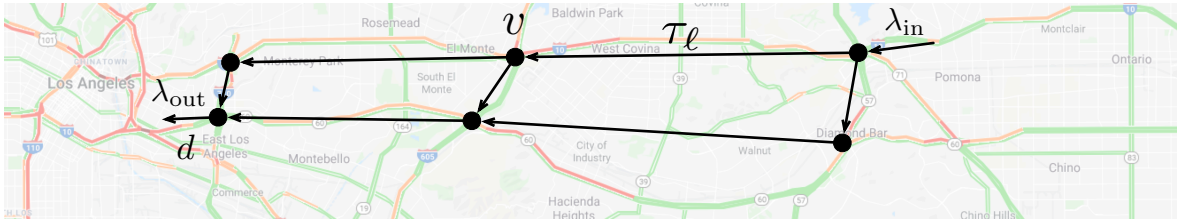


Microscopic: at **every node** drivers minimize travel time to destination

minimize τ_ℓ + (time from v to dest.)

Modeling Navigation Apps



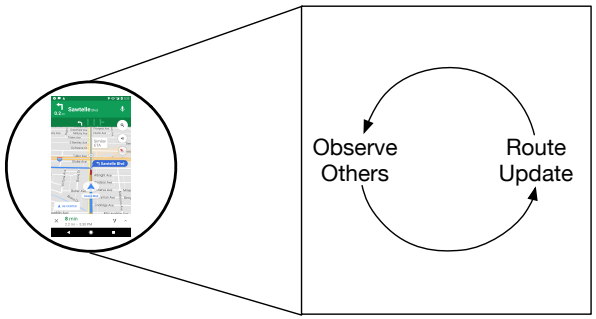


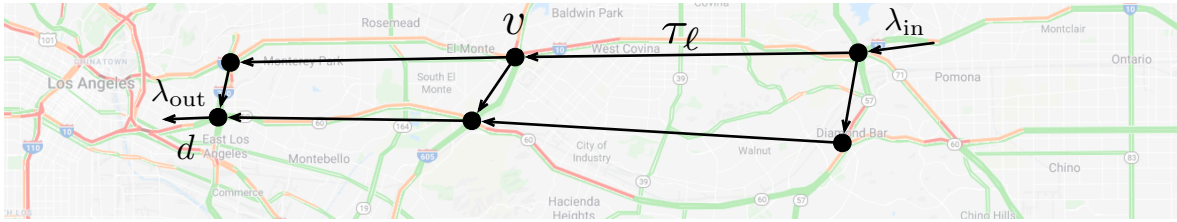
Microscopic: at **every node** drivers minimize travel time to destination

minimize τ_ℓ + (time from v to dest.)

π_ℓ **:= perceived cost** $\rightarrow$ economic cost that drivers associate to each highway

Modeling Navigation Apps





Microscopic: at **every node** drivers minimize travel time to destination

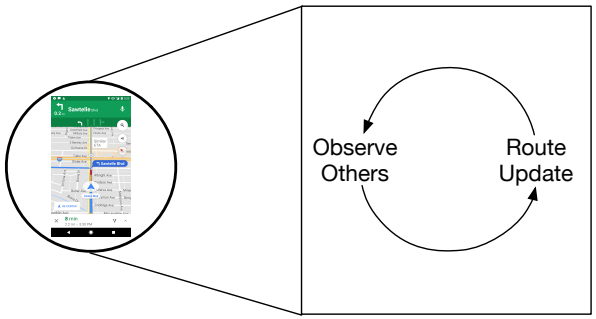
minimize τ_ℓ + (time from v to dest.)

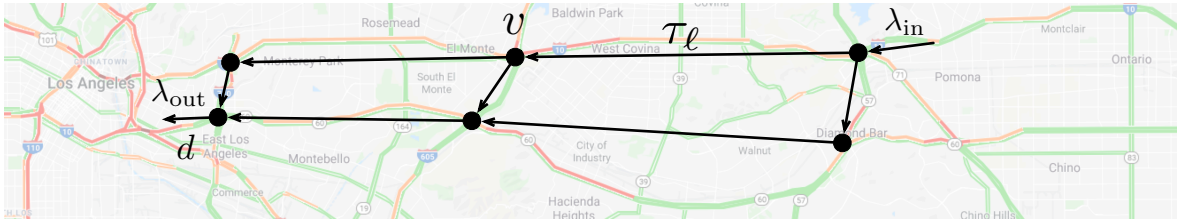
π_ℓ **:= perceived cost** $\rightarrow$ economic cost that drivers associate to each highway

Macroscopic: all drivers minimize their perceived costs

$$\dot{r}_{\ell m} = \text{(aggregate selection)}$$

Modeling Navigation Apps





Microscopic: at **every node** drivers minimize travel time to destination

minimize τ_ℓ + (time from v to dest.)

π_ℓ **:= perceived cost** $\rightarrow$ economic cost that drivers associate to each highway

Macroscopic: all drivers minimize their perceived costs

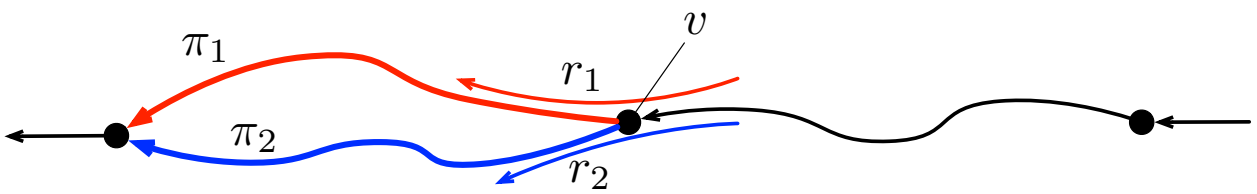
$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$

"Replicator dynamics"

Evolutionary Model of Routing Apps

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$

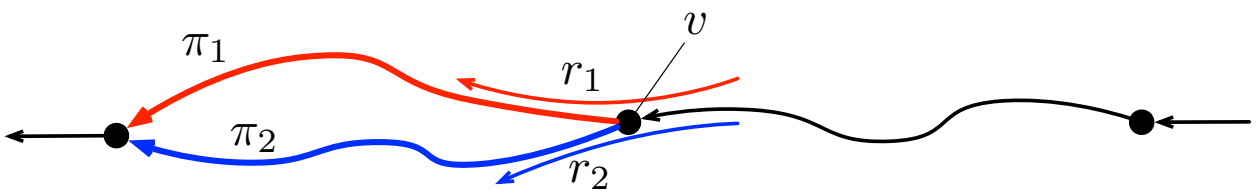


- $r_1 \to$ % of drivers chosing path 1
- $r_1\pi_1 + r_2\pi_2 \to$ Average cost from v to dest.

Evolutionary Model of Routing Apps

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$

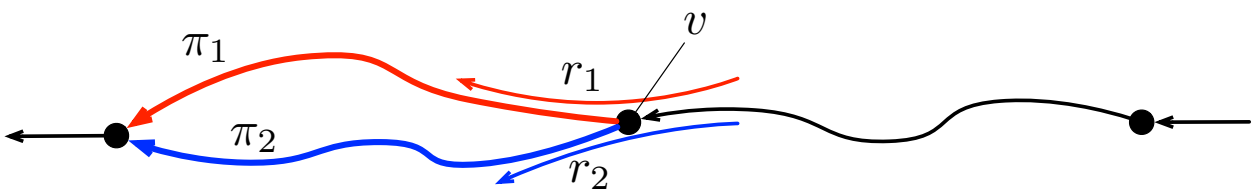


- $r_1 \to$ % of drivers chosing path 1
- $r_1\pi_1 + r_2\pi_2 \longrightarrow$ Average cost from v to dest.

Evolutionary Model of Routing Apps

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$



- $r_1 \to$ % of drivers chosing path 1
- $r_1\pi_1 + r_2\pi_2 \to$ Average cost from v to dest.

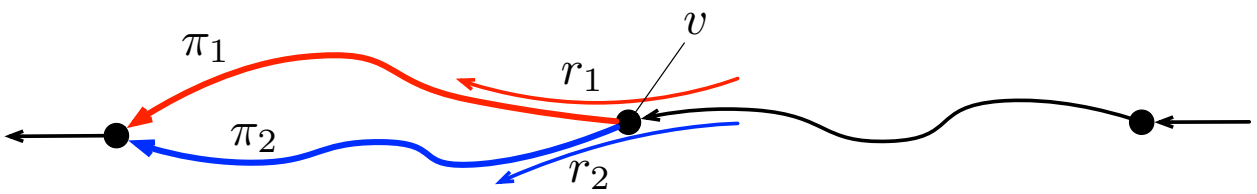
$$\triangle$$

  - if $\pi_1 > \pi_2 \quad \Rightarrow \quad \dot{r}_1 = r_1(\triangle - \pi_1) < 0$

Evolutionary Model of Routing Apps

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$



- $r_1 \to$ % of drivers chosing path 1
- $r_1\pi_1 + r_2\pi_2 \to$ Average cost from v to dest.

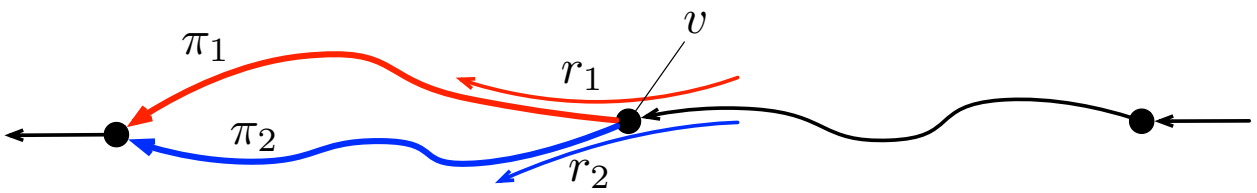
$\triangle$

  - if $\pi_1 > \pi_2 \quad \Rightarrow \quad \dot{r}_1 = r_1(\triangle - \pi_1) < 0$
  - if $\pi_1 < \pi_2 \quad \Rightarrow \quad \dot{r}_1 = r_1(\triangle - \pi_1) > 0$

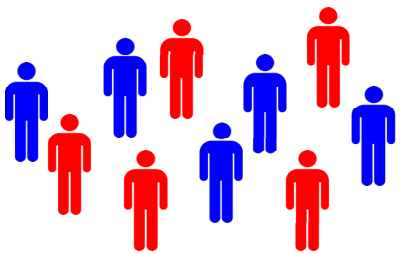
Evolutionary Model of Routing Apps (2)

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$



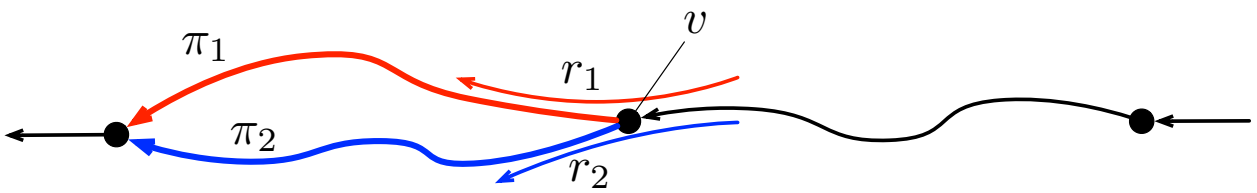
- Red is more convenient than blue ($\pi_1 > \pi_2$)



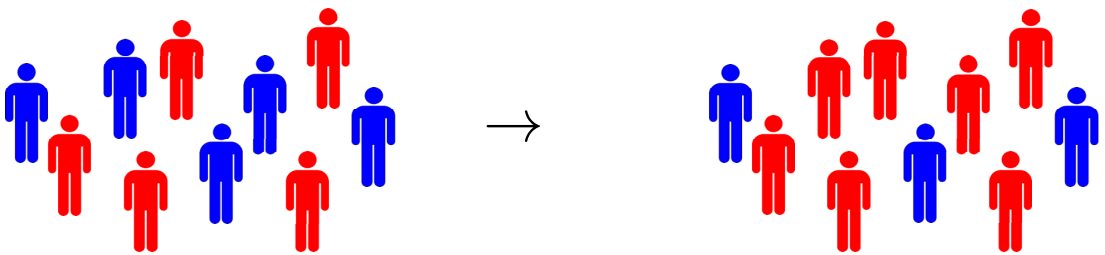
Evolutionary Model of Routing Apps (2)

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$



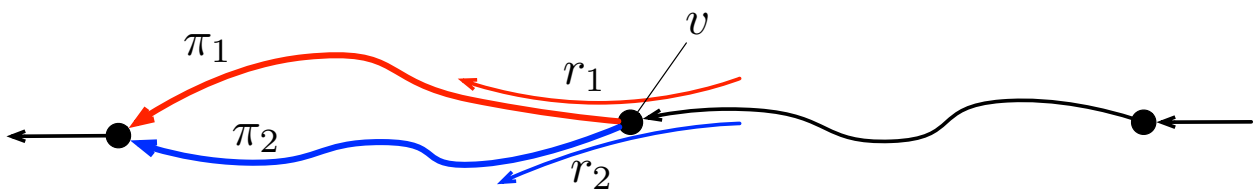
- Red is more convenient than blue ($\pi_1 > \pi_2$)



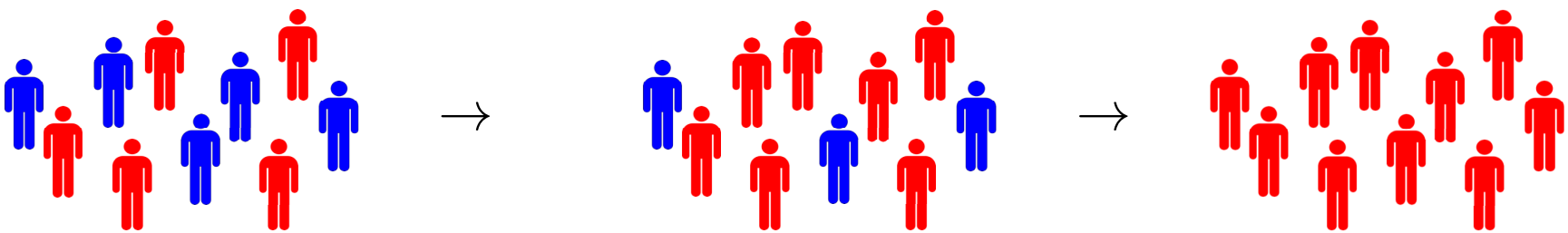
Evolutionary Model of Routing Apps (2)

"Replicator dynamics"

$$\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$



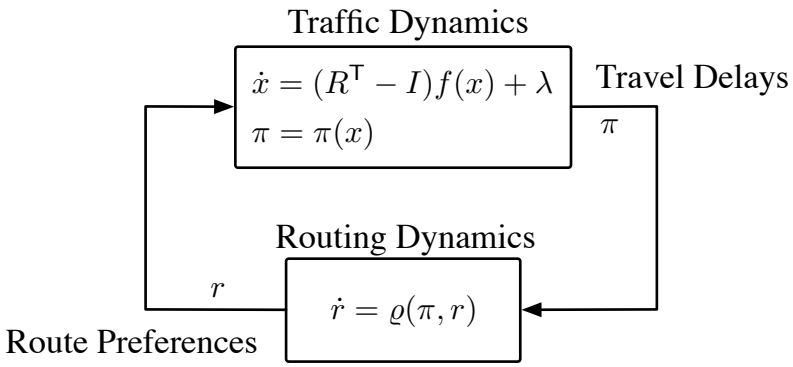
- Red is more convenient than blue ($\pi_1 > \pi_2$)



But changes in the user behavior will change congestion

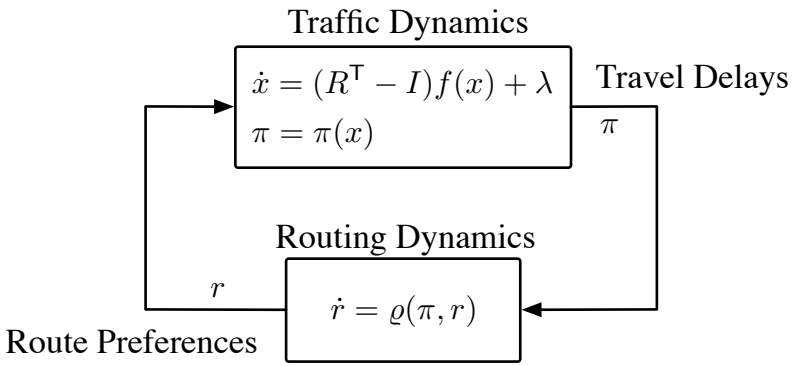
Coupled Traffic and Routing Dynamics

- Congestion affects route choices
- Routing affects congestion



Coupled Traffic and Routing Dynamics

- Congestion affects route choices
- Routing affects congestion



- Nonlinear $\rightarrow$ trajectories difficult to characterize
- We study equilibria, dynamical behavior

Does the system admit equilibrium points?

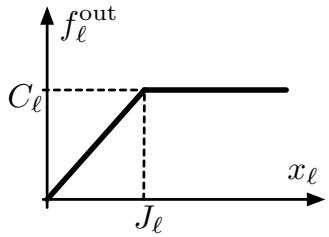
Are the equilibrium points stable?

Existence of Equilibria

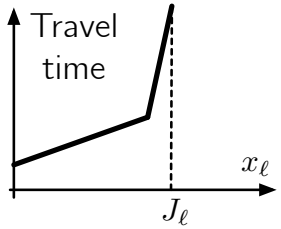
Equilibria (x^*, r^*): if system starts at these points will remain at all times

Technical assumptions:

Roads have flow capacities



Drivers avoid jammed roads

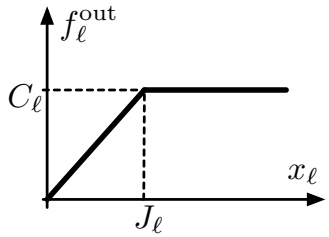


Existence of Equilibria

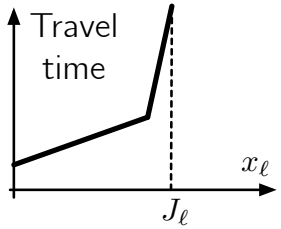
Equilibria (x^*, r^*): if system starts at these points will remain at all times

Technical assumptions:

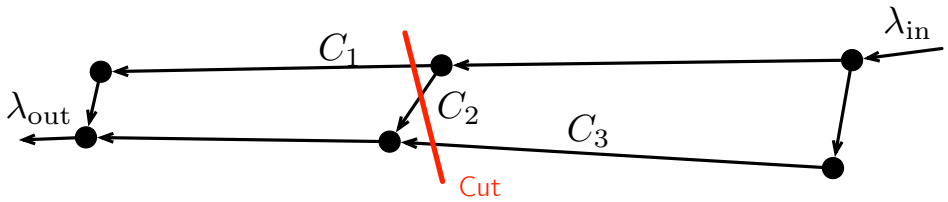
Roads have flow capacities



Drivers avoid jammed roads



Min-cut capacity: capacity of smallest cut that disconnects λ_{in} from λ_{out}

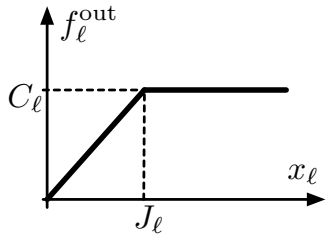


Existence of Equilibria

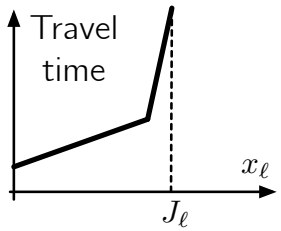
Equilibria (x^*, r^*): if system starts at these points will remain at all times

Technical assumptions:

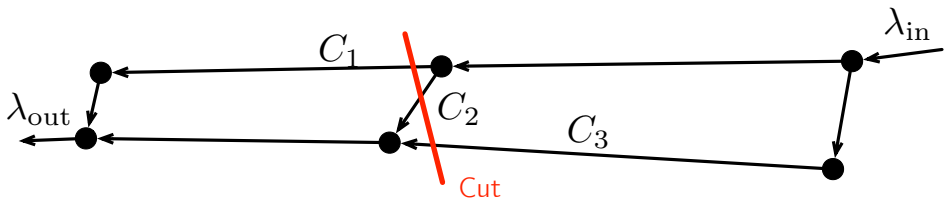
Roads have flow capacities



Drivers avoid jammed roads



Min-cut capacity: capacity of smallest cut that disconnects λ_{in} from λ_{out}



(Bianchin, Pasqualetti, TAC 2020)

Networks with app-routing admit equilibrium $\Leftrightarrow$ $\lambda_{\text{in}} <$ min-cut capacity

Existence of Equilibria: Implications

(Bianchin, Pasqualetti, TAC 2020)

Networks with app-routing admit equilibrium $\Leftrightarrow$ $\lambda_{\text{in}} <$ min-cut capacity

Existence of Equilibria: Implications

(Bianchin, Pasqualetti, TAC 2020)

Networks with app-routing admit equilibrium $\Leftrightarrow$ $\lambda_{\text{in}} <$ min-cut capacity

Implications:

(1) Routing apps $\rightarrow$ maximum network throughput

(2) $\lambda_{\text{in}} \gg 1 \rightarrow$ no equilibria (congestion grows unbounded)

Existence of Equilibria: Implications

(Bianchin, Pasqualetti, TAC 2020)

Networks with app-routing admit equilibrium $\Leftrightarrow$ $\lambda_{\text{in}} <$ min-cut capacity

Implications:

(1) Routing apps $\rightarrow$ maximum network throughput

(2) $\lambda_{\text{in}} \gg 1 \rightarrow$ no equilibria (congestion grows unbounded)

(1) If routing is "free"

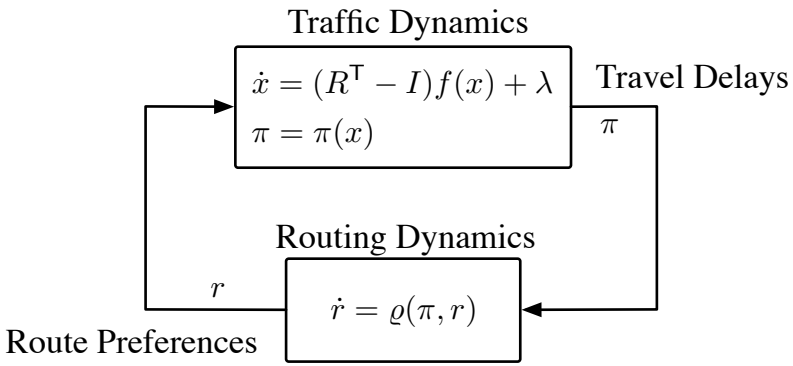
Max-flow theorem $\Rightarrow$ exists maximum flow with finite travel times

(2) If travel times are finite and "fixed"

$\Rightarrow$ Replicator equation admits equilibrium

Coupled Traffic and Routing Dynamics

- Congestion affects route choices
- Routing affects congestion



- Nonlinear $\rightarrow$ trajectories difficult to characterize
- We study equilibria, dynamical behavior

Does the system admit equilibrium points?

Yes, if $\lambda_{\text{in}} <$ min-cut capacity

Are the equilibrium points stable?

Stability of Equilibria

Stability: if system starts near equilibrium
will remain near that operating point

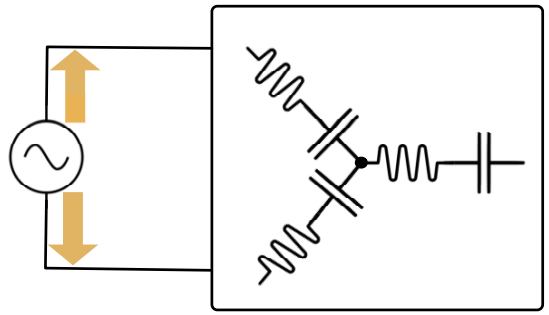
Stability $\Rightarrow$ measure of robustness of the system

Detour: Passivity in Nonlinear Dynamical Systems

Passivity: the system does not generate energy
but instead dissipates, stores, and releases it

Theory inspired from electrical circuits:

When energy is injected
$\Rightarrow$ system stores



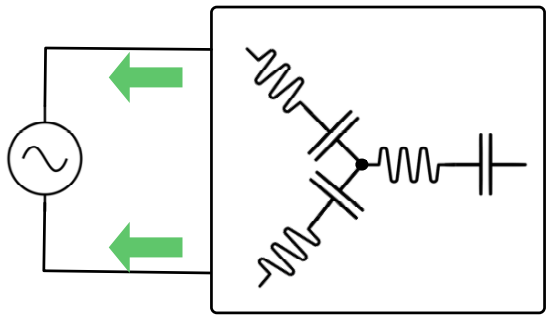
Passive Circuit

Detour: Passivity in Nonlinear Dynamical Systems

Passivity: the system does not generate energy
but instead dissipates, stores, and releases it

Theory inspired from electrical circuits:

When energy is not supplied
$\Rightarrow$ system releases

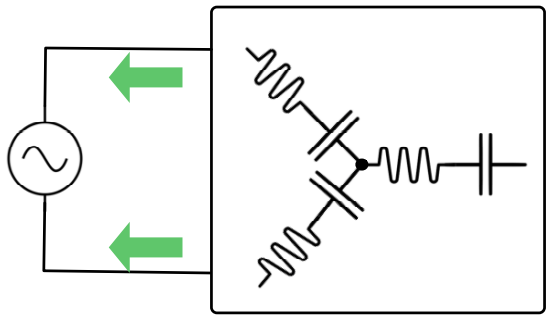

Passive Circuit

Detour: Passivity in Nonlinear Dynamical Systems

Passivity: the system does not generate energy
but instead dissipates, stores, and releases it

Theory inspired from electrical circuits:

When energy is not supplied
$\Rightarrow$ system releases



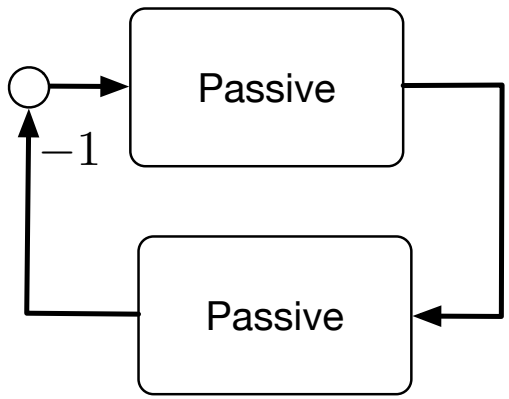
In control systems, a system is passive if

There exists storage function $V \geq 0$
such that $\dot{V} \leq u^{\mathsf{T}} y$

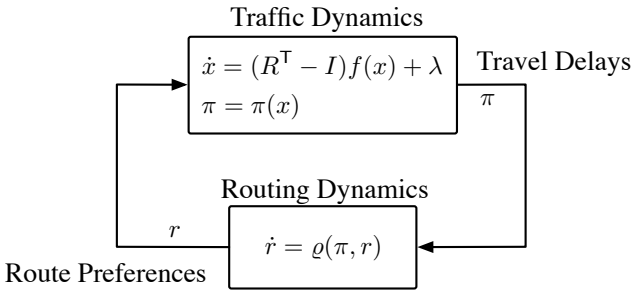
Detour: Passivity in Nonlinear Dynamical Systems

Passivity: the system does not generate energy
but instead dissipates, stores, and releases it

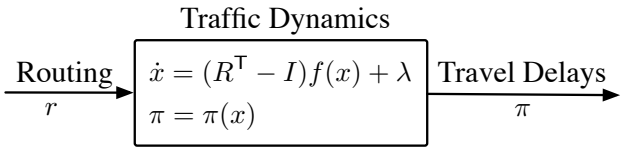
The negative feedback interconnection between
two passive nonlinear systems is passive



Detour: Passivity in Nonlinear Dynamical Systems (2)



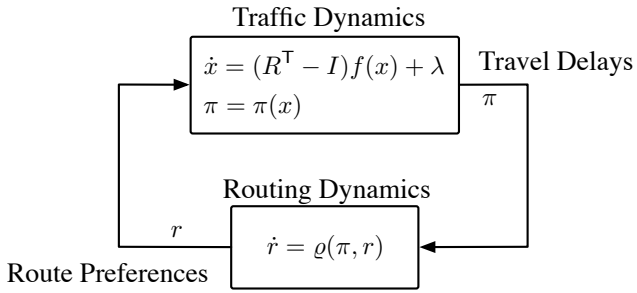
The **open-loop systems are passive**:



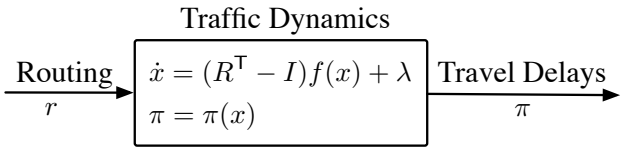
The traffic dynamics are passive

- suboptimal routing $\Rightarrow$ network stores congestion
- optimal routing $\Rightarrow$ congestion is released

Detour: Passivity in Nonlinear Dynamical Systems (2)

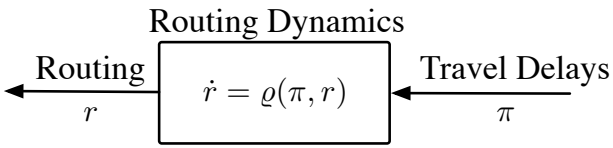


The **open-loop systems are passive**:



The traffic dynamics are passive

- suboptimal routing $\Rightarrow$ network stores congestion
- optimal routing $\Rightarrow$ congestion is released



The routing dynamics are passive

Stability of Equilibria

Stability: if system starts near equilibrium
will remain near that operating point

Stability $\Rightarrow$ measure of robustness of the system

Stability of Equilibria

Stability: if system starts near equilibrium
will remain near that operating point

Stability $\Rightarrow$ measure of robustness of the system

The answer is positive, but only partially:

(Bianchin, Pasqualetti, TAC '20)

Equilibria with app-informed drivers are stable,
but not necessarily asymptotically stable

Stability of Equilibria

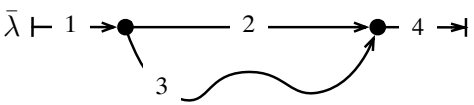
Stability: if system starts near equilibrium
will remain near that operating point

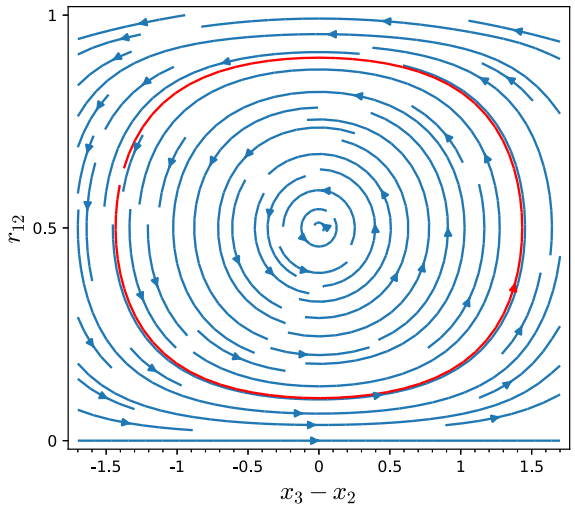
Stability $\Rightarrow$ measure of robustness of the system

The answer is positive, but only partially:

(Bianchin, Pasqualetti, TAC '20)

Equilibria with app-informed drivers are stable,
but not necessarily asymptotically stable





Stability of Equilibria

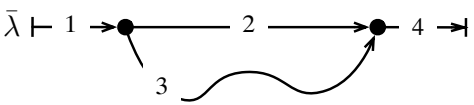
Stability: if system starts near equilibrium
will remain near that operating point

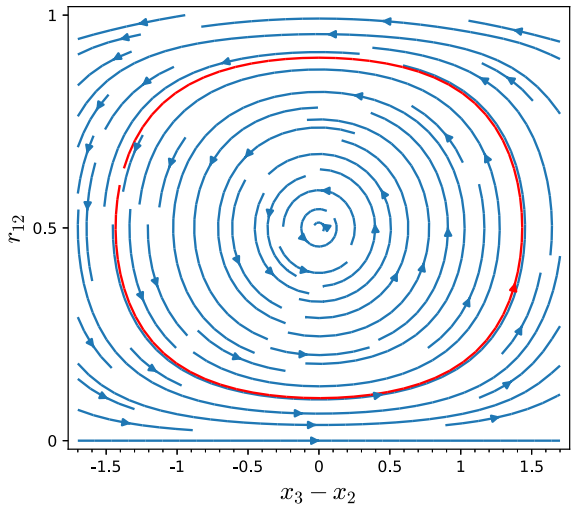
Stability $\Rightarrow$ measure of robustness of the system

The answer is positive, but only partially:

(Bianchin, Pasqualetti, TAC '20)

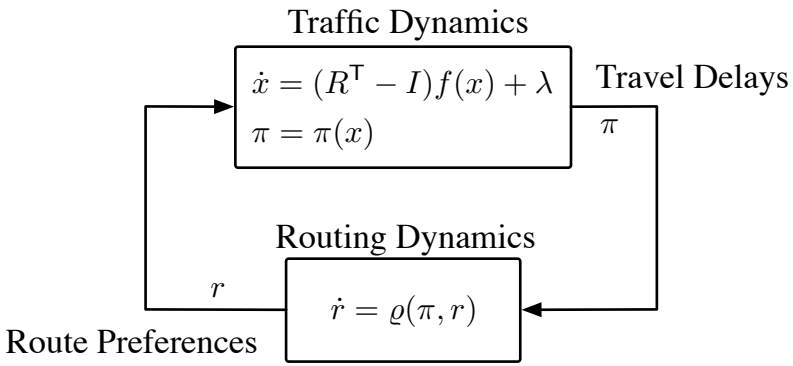
Equilibria with app-informed drivers are stable,
but not necessarily asymptotically stable





Coupled Traffic and Routing Dynamics

- Congestion affects route choices
- Routing affects congestion



- Nonlinear $\rightarrow$ trajectories difficult to characterize
- We study equilibria, dynamical behavior

Does the system admit equilibrium points?

Yes, if $\lambda_{\text{in}} <$ min-cut capacity

Are the equilibrium points stable?

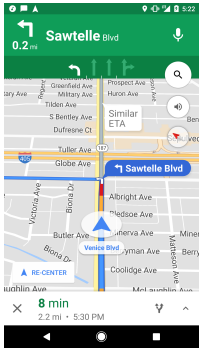
Not necessarily asymptotically stable

Directions

(1) Design navigation apps for better robustness

$$\delta_{\ell m}^{-1}\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$

$\delta_{\ell m} \to$ congestion-dependent reaction

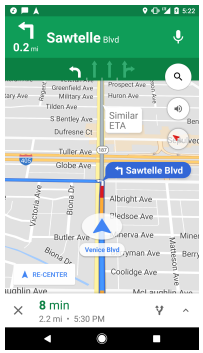


Directions

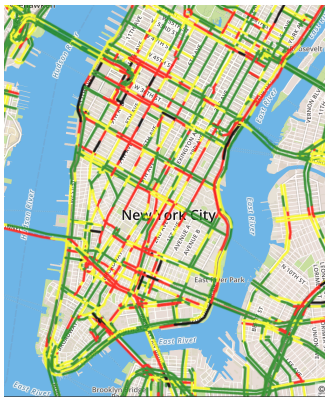
(1) Design navigation apps for better robustness

$$\delta_{\ell m}^{-1}\dot{r}_{\ell m} = r_{\ell m}\left(\sum_q r_{\ell q}\pi_q - \pi_m\right)$$

$\delta_{\ell m} \to$ congestion-dependent reaction



(2) Design control policies with system-level performance



Given: Real-Time traffic conditions

Objective: Minimize city overall congestion

Control: Automated Intersections

G. Bianchin and F. Pasqualetti, "Gramian-based optimization for the analysis and control of traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019

Organization of Dissertation and Talk Outline

Application to traffic and transportation

Robustness of networks with app routing

Application to robotics

Trajectory planning for secure navigation Part II

Motivation

Countless number of applications

Flexibility, complex tasks

Unknown and adversarial environments

Robustness and security are needed for autonomy

- Needed in autonomous operations
- Accidents in adversarial settings

The Register®

Biting the hand that feeds IT

US spy drone hijacked with GPS spoof hack, report says

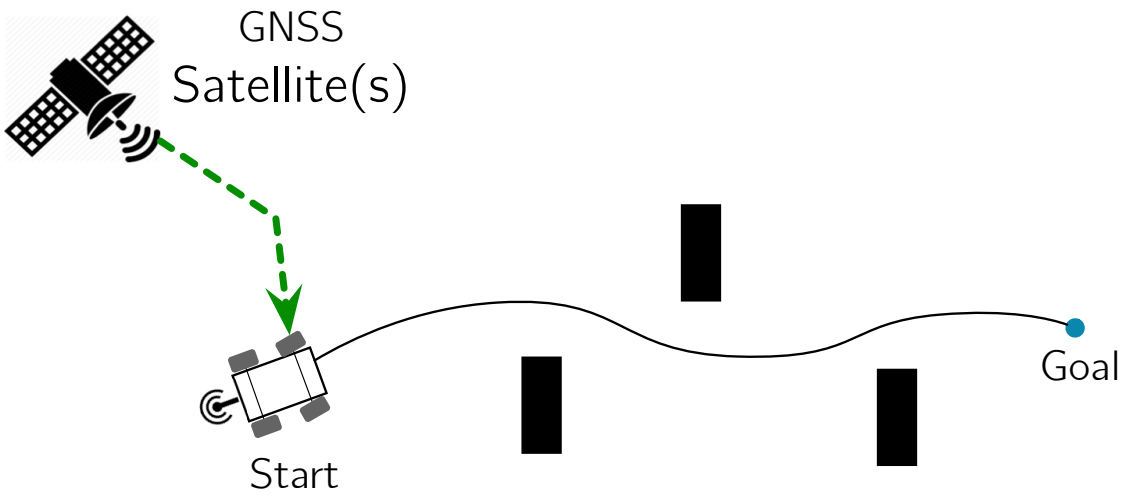
Electronic warfare comes of age – in Iran

By Dan Goodin 15 Dec 2011 at 23:27 158 SHARE ▼

The US stealth drone broadcast last week on Iranian state television was captured by spoofing its GPS coordinates, a hack that tricked the bird into landing in Iranian territory instead of where it was programmed to touch down, *The Christian Science Monitor* reported.

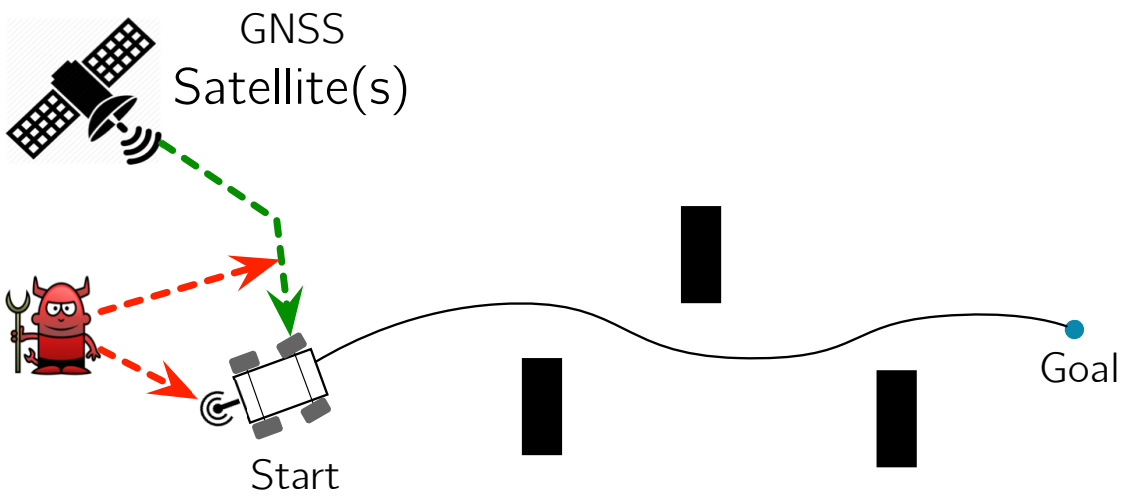
The 1700-word article cited an unnamed Iranian engineer who said he's studying the inner workings of the American bat-wing RQ-170 Sentinel that recently went missing over Iranian airspace. He said the spoofing technique made the craft "land on its own where we wanted it to, without having to crack the remote-control signals and communications" from the US control center.

Secure Navigation in Adversarial Environments



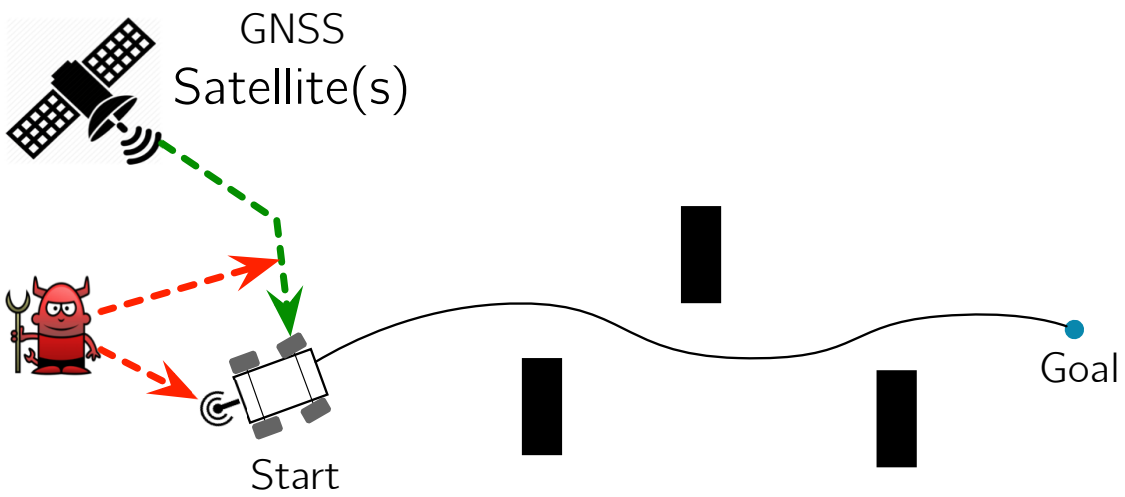
- Navigate from start position to goal
- Despite unknown and arbitrary attacks
- Vulnerability: wireless communication
- e.g. GPS spoofing

Secure Navigation in Adversarial Environments

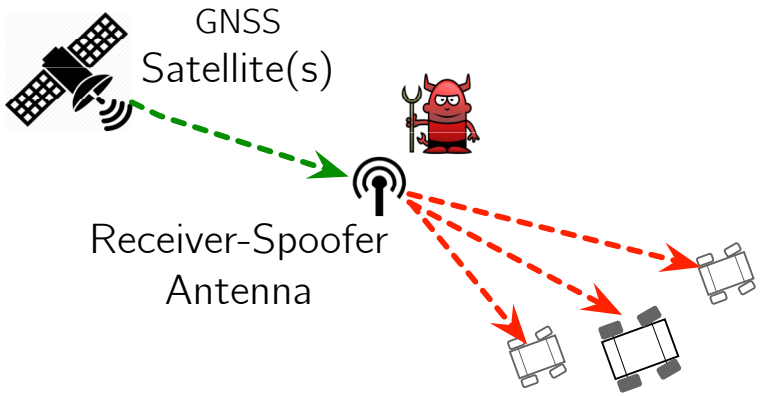


- Navigate from start position to goal
- Despite unknown and arbitrary attacks
- Vulnerability: wireless communication
- e.g. GPS spoofing

Secure Navigation in Adversarial Environments



- Navigate from start position to goal
- Despite unknown and arbitrary attacks
- Vulnerability: wireless communication
- e.g. GPS spoofing



Models for Attacks

Nominal System

2D double-integrator:

$$\dot{p}_{\mathsf{n}} = v_{\mathsf{n}}$$

$$\dot{v}_{\mathsf{n}} = u_{\mathsf{n}}$$

Models for Attacks

Nominal System

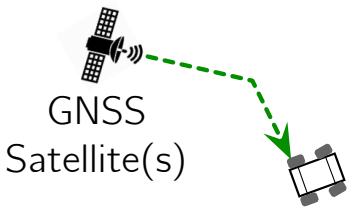
2D double-integrator:

$$\dot{p}_{\mathsf{n}} = v_{\mathsf{n}}$$

$$\dot{v}_{\mathsf{n}} = u_{\mathsf{n}}$$

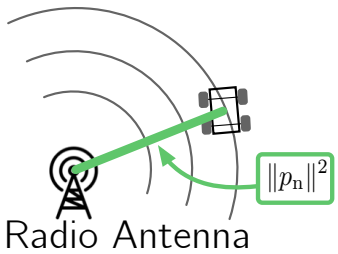
GPS

$$y_{\mathsf{n}}^{\mathsf{GPS}} = p_{\mathsf{n}}$$



RSSI

$$y^{\mathsf{RSSI}} = \|p_{\mathsf{n}}\|^2$$



Models for Attacks

Nominal System

2D double-integrator:

$$\dot{p}_{\mathsf{n}} = v_{\mathsf{n}}$$

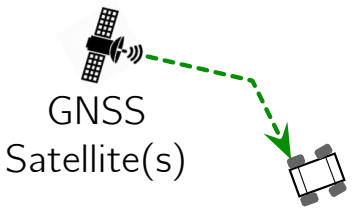
$$\dot{v}_{\mathsf{n}} = u_{\mathsf{n}}$$

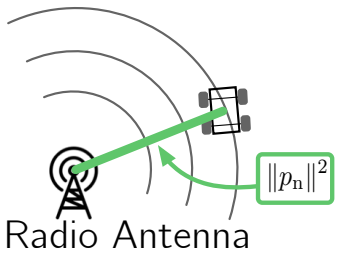
GPS

$$y_{\mathsf{n}}^{\mathsf{GPS}} = p_{\mathsf{n}}$$

RSSI

$$y^{\mathsf{RSSI}} = \|p_{\mathsf{n}}\|^2$$





System Under Attack

Control inputs are compromised

$$\dot{p} = v$$

$$\dot{v} = u$$

GPS

$$y^{\mathsf{GPS}} = p + u^{\mathsf{SPOOF}}$$

RSSI

$$y^{\mathsf{RSSI}} = \|p_{\mathsf{n}}\|^2$$

Models for Attacks

Nominal System

2D double-integrator:

$$\dot{p}_{\mathsf{n}} = v_{\mathsf{n}}$$

$$\dot{v}_{\mathsf{n}} = u_{\mathsf{n}}$$

System Under Attack

Control inputs are compromised

$$\dot{p} = v$$

$$\dot{v} = u$$

GPS

$$y^{\mathsf{GPS}} = p + u^{\mathsf{SPOOF}}$$

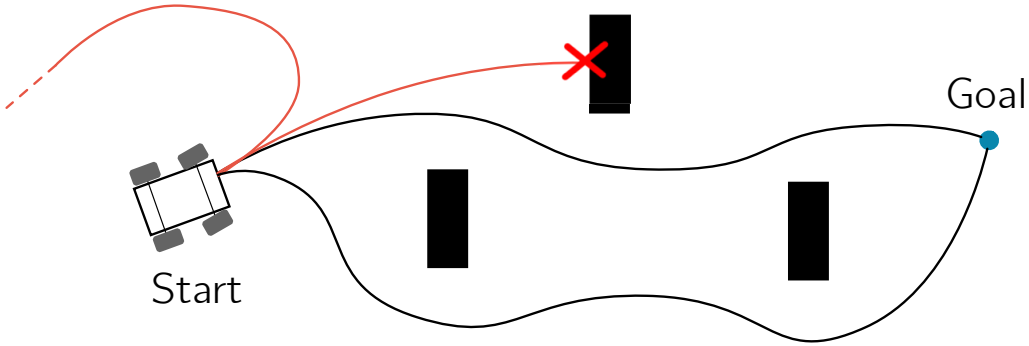
RSSI

$$y^{\mathsf{RSSI}} = \|p_{\mathsf{n}}\|^2$$

Trajectory planner: choose u_{n}

Attacker: choose (u, u^{SPOOF})

Can We Make Robotic Navigation Secure?



1. Can **attacker** modify trajectory and escape detectability?

2. Can **planner** design secure trajectories?

Undetectable Attacks

An **Undetectable Attack** is:

Pair (u, u^{SPOOF}) that is not "visible" from the sensors:

$$y_{\text{n}}^{\text{GPS}} = y^{\text{GPS}}, \qquad y_{\text{n}}^{\text{RSSI}} = y^{\text{RSSI}} \qquad \text{(at all times)}$$

(Bianchin, Liu, Pasqualetti, Automatica '20)

An attack is undetectable **if and only if**

$$u^{\top} p = u_{\text{n}}^{\top} p_{\text{n}} + \|v_{\text{n}}\|^2 - \|v\|^2$$
$$u^{\text{SPOOF}} = p_{\text{n}} - p$$

- Undetectable attacks exist
- Can be computed given nominal state

Undetectable Attacks

An **Undetectable Attack** is:

Pair (u, u^{SPOOF}) that is not "visible" from the sensors:

$$y_{\text{n}}^{\text{GPS}} = y^{\text{GPS}}, \qquad y_{\text{n}}^{\text{RSSI}} = y^{\text{RSSI}} \qquad \text{(at all times)}$$

(Bianchin, Liu, Pasqualetti, Automatica '20)

An attack is undetectable **if and only if**

$$u^{\top} p = u_{\text{n}}^{\top} p_{\text{n}} + \|v_{\text{n}}\|^2 - \|v\|^2$$
$$u^{\text{SPOOF}} = p_{\text{n}} - p$$

- Undetectable attacks exist
- Can be computed given nominal state

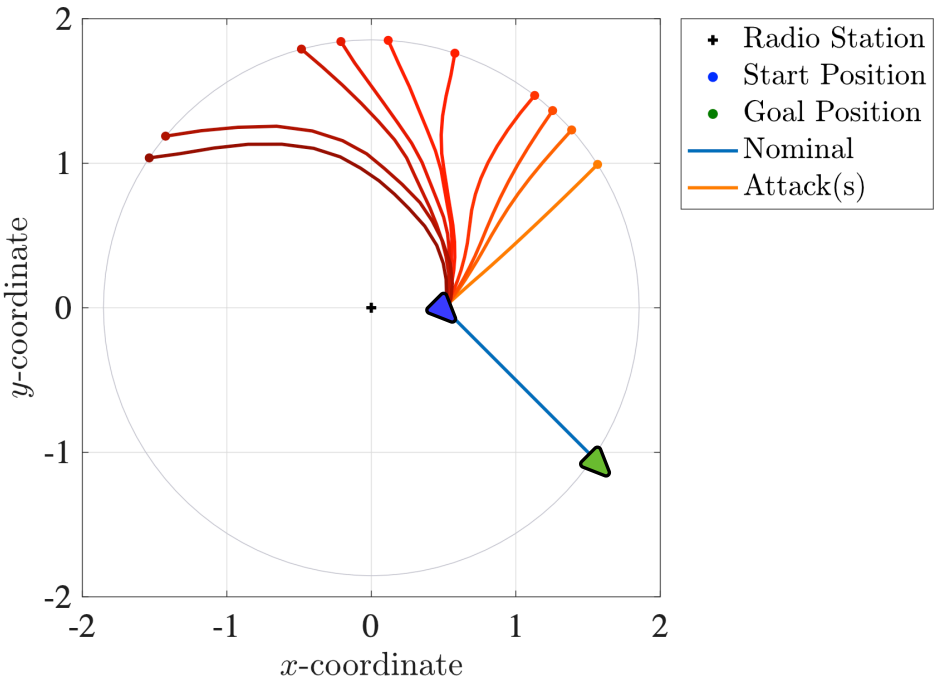
Undetectable Attacks (2)

(Bianchin, Liu, Pasqualetti, Automatica '20)

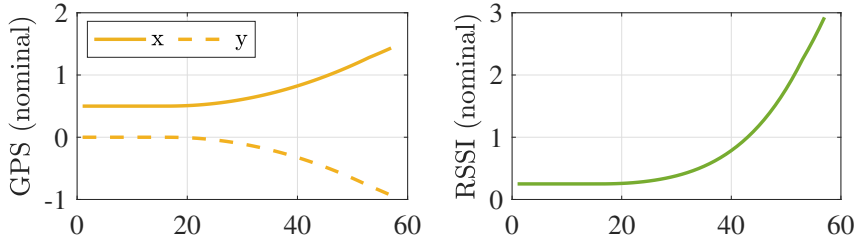
An attack is undetectable **if and only if**

$$u^\top p = u_{\mathsf{n}}^\top p_{\mathsf{n}} + \|v_{\mathsf{n}}\|^2 - \|v\|^2$$

$$u^{\mathsf{SPOOF}} = p_{\mathsf{n}} - p$$



Nominal sensor readings:



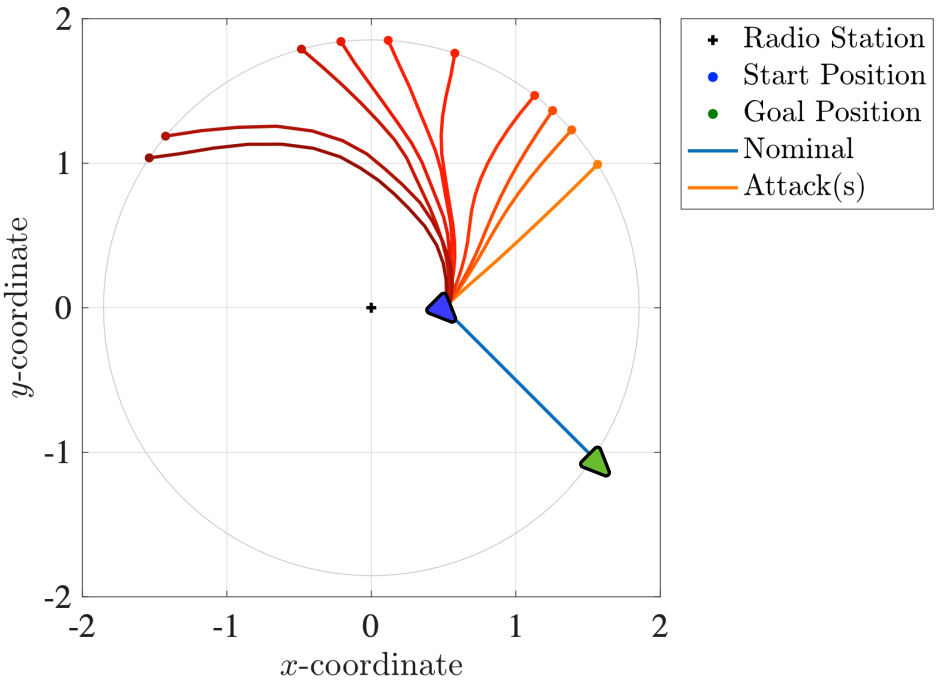
Undetectable Attacks (2)

(Bianchin, Liu, Pasqualetti, Automatica '20)

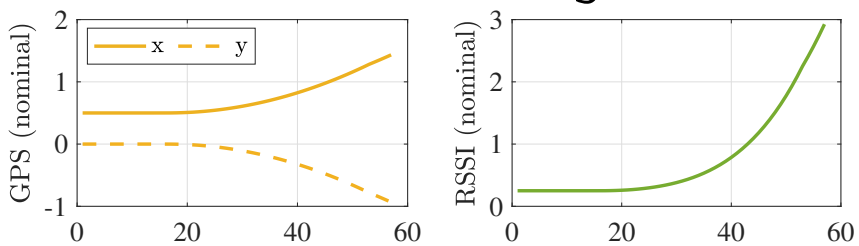
An attack is undetectable **if and only if**

$$u^\top p = u_{\mathsf{n}}^\top p_{\mathsf{n}} + \|v_{\mathsf{n}}\|^2 - \|v\|^2$$

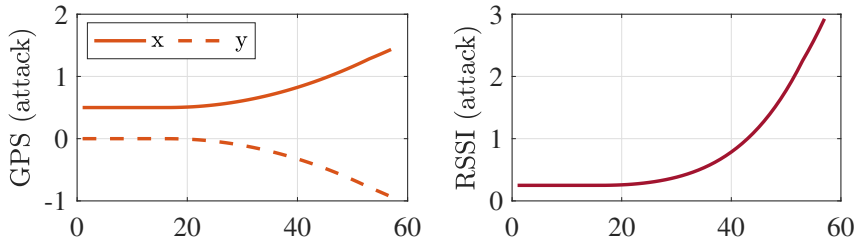
$$u^{\mathsf{SPOOF}} = p_{\mathsf{n}} - p$$



Nominal sensor readings:

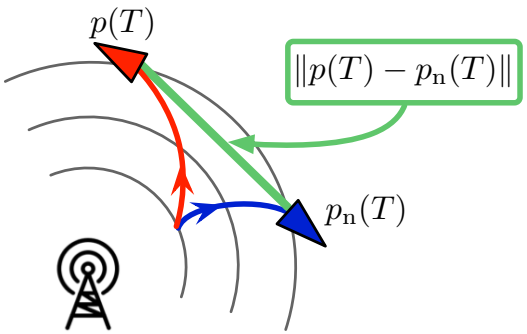


Attacked sensor readings:



Attack Planning

$$\begin{aligned}(\text{attacker}) = \max_{u} \quad & \|p(T) - p_{\mathrm{n}}(T)\| \\ \text{s.t.} \quad & u = (\text{undetectable}) \\ & \dot{x} = (\text{double integrator}) \\ & \|u\| \le u_{\max}\end{aligned}$$



(Liu, Bianchin, Pasqualetti, Automatica '20)

$$u = a_{\mathrm{r}} p + w$$

where $w^{\top} p = 0$, and

$$a_{\mathrm{r}} = (u_{\mathrm{n}}^{\top} p_{\mathrm{n}} + \|v_{\mathrm{n}}\|^2 - \|v\|^2)\|p\|^{-2}$$

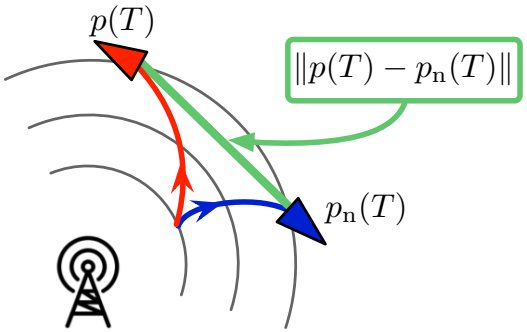
$$w = -\operatorname{sgn}(\lambda^{\top} B W x)\sqrt{u_{\max}^2/\|p\|^2 - a_{\mathrm{r}}{}^2} W x$$

and $(\lambda, x) =$ solve a two-point BVP

- a_{r}= radial accel.
- w= tangential accel.
- Bang-Bang control
- Can be computed by **solving BVP offline**

Attack Planning

$$\begin{aligned}\text{(attacker)} = \max_{u} \quad & \|p(T) - p_{\text{n}}(T)\| \\ \text{s.t.} \quad & u = \text{(undetectable)} \\ & \dot{x} = \text{(double integrator)} \\ & \|u\| \le u_{\max}\end{aligned}$$



(Liu, Bianchin, Pasqualetti, Automatica '20)

$$u = a_{\text{r}} p + w$$

where $w^{\top} p = 0$, and

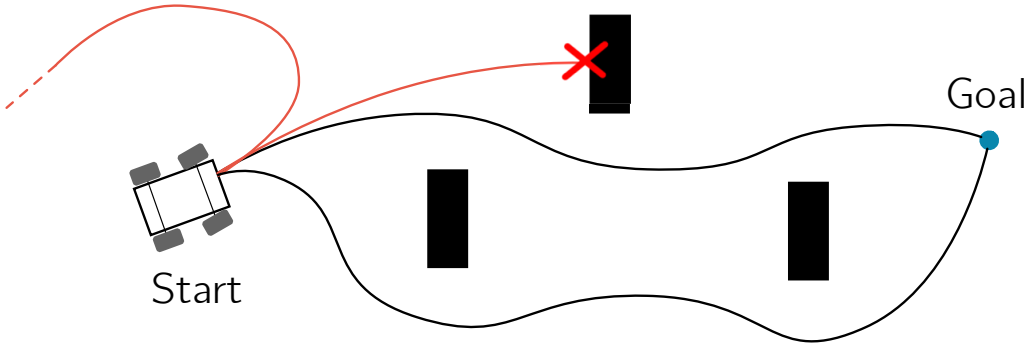
$$a_{\text{r}} = (u_{\text{n}}^{\top} p_{\text{n}} + \|v_{\text{n}}\|^2 - \|v\|^2)\|p\|^{-2}$$

$$w = -\operatorname{sgn}(\lambda^{\top} BWx)\sqrt{u_{\max}^2/\|p\|^2 - a_{\text{r}}{}^2}\, Wx$$

and (λ, x) = solve a two-point BVP

- a_{r}= radial accel.
- w= tangential accel.
- Bang-Bang control
- Can be computed by **solving BVP offline**

Can We Make Robotic Navigation Secure?



1. Can **attacker** modify trajectory and escape detectability?

✓ Yes, under appropriate design

2. Can **planner** design secure trajectories?

Secure Trajectories

A **Secure Trajectory** is:

1. Navigate from start to destination
2. Attacks can be detected

Assume max. acceleration is bounded: $\|u_n\| \leq u_{\max}$

(Liu, Bianchin, Pasqualetti, Automatica '20)

A trajectory is secure **if and only if**

$$u_n = \kappa \frac{p_n}{\|p_n\|} u_{\max} \qquad \kappa(t) \in \{-1, 1\}$$

1. Nominal controls affect existence of attacks
2. State-feedback law, maximum energy at all times

Secure Trajectories

A **Secure Trajectory** is:

1. Navigate from start to destination
2. Attacks can be detected

Assume max. acceleration is bounded: $\|u_{\mathsf{n}}\| \leq u_{\max}$

(Liu, Bianchin, Pasqualetti, Automatica '20)

A trajectory is secure **if and only if**

$$u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \qquad \kappa(t) \in \{-1, 1\}$$

1. Nominal controls affect existence of attacks
2. State-feedback law, maximum energy at all times

Secure Trajectories

A **Secure Trajectory** is:

1. Navigate from start to destination
2. Attacks can be detected

Assume max. acceleration is bounded: $\|u_{\mathsf{n}}\| \leq u_{\max}$

(Liu, Bianchin, Pasqualetti, Automatica '20)

A trajectory is secure **if and only if**

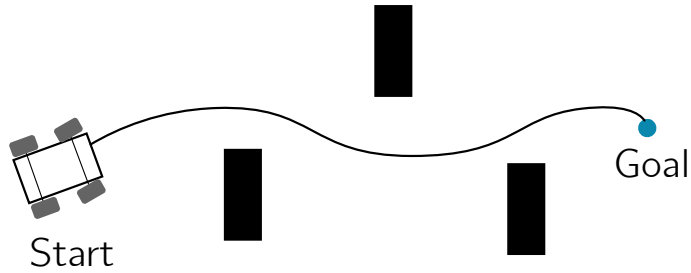
$$u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \qquad \kappa(t) \in \{-1, 1\}$$

1. Nominal controls affect existence of attacks
2. State-feedback law, maximum energy at all times

Computing Minimum-Time Secure Trajectories

Given (start) and (goal) positions:

$$\begin{aligned}(\text{planner}) = \min_{\kappa, T} \quad & T + \|p_{\mathsf{n}}(T) - p_{\mathsf{F}}\| \\ \text{s.t.} \quad & u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \\ & \dot{x}_{\mathsf{n}} = (\text{double integrator})\end{aligned}$$



(Liu, Bianchin, Pasqualetti, Automatica '20)

$$\kappa^* = -\operatorname{sgn}(\lambda^\top B p_{\mathsf{n}}), \qquad T^* = \xi$$

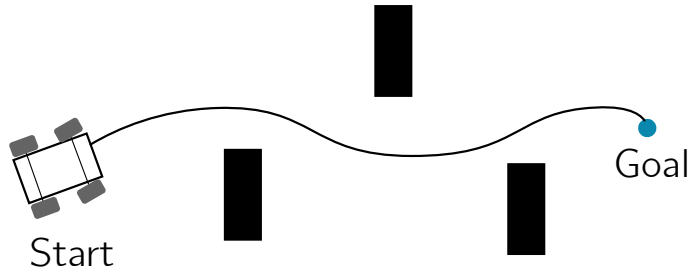
where $(\lambda, p_{\mathsf{n}}, \xi) =$ solve a two-point BVP

- Open-loop input
- Bang-Bang control

Computing Minimum-Time Secure Trajectories

Given (start) and (goal) positions:

$$\begin{aligned}(\text{planner}) = \min_{\kappa, T} \quad & T + \|p_{\mathsf{n}}(T) - p_{\mathsf{F}}\| \\ \text{s.t.} \quad & u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \\ & \dot{x}_{\mathsf{n}} = (\text{double integrator})\end{aligned}$$



(Liu, Bianchin, Pasqualetti, Automatica '20)

$$\kappa^* = -\operatorname{sgn}(\lambda^\top B p_{\mathsf{n}}), \qquad T^* = \xi$$

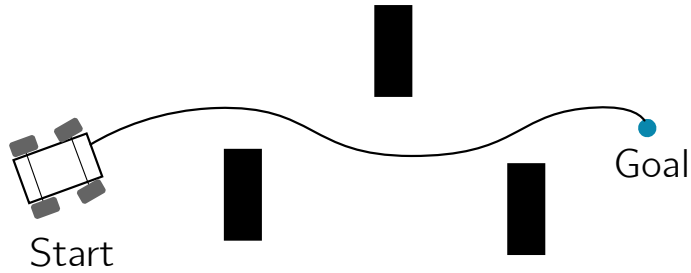
where $(\lambda, p_{\mathsf{n}}, \xi) =$ solve a two-point BVP

- Open-loop input
- Bang-Bang control

Computing Minimum-Time Secure Trajectories

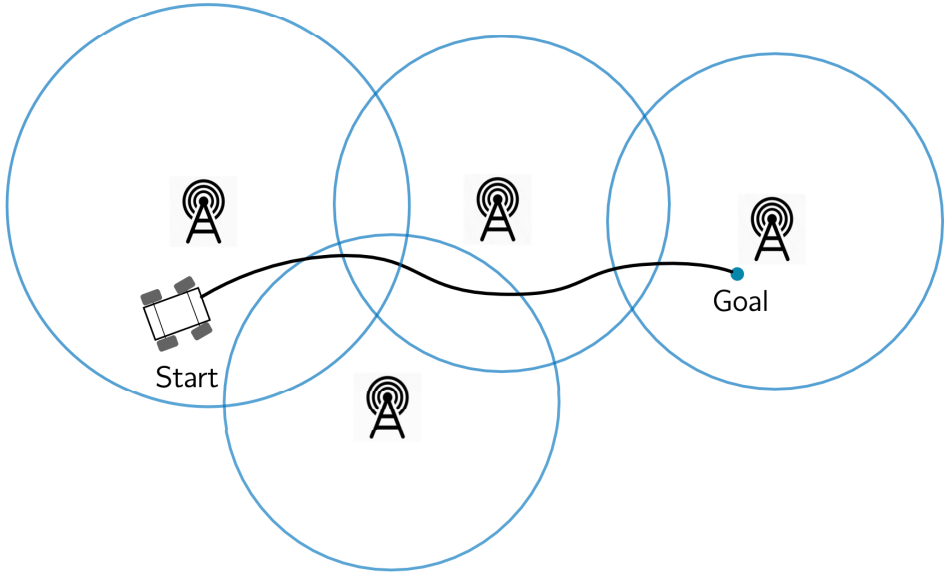
Given (start) and (goal) positions:

$$\begin{aligned}(\text{planner}) = \min_{\kappa, T} \quad & T + \|p_{\mathsf{n}}(T) - p_{\mathsf{F}}\| \\ \text{s.t.} \quad & u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \\ & \dot{x}_{\mathsf{n}} = (\text{double integrator})\end{aligned}$$



- Optimization $\rightarrow$ final time is unknown
- Optimization $\rightarrow$ final position may not be reachable exactly

Secure Navigation: Waypoint Design

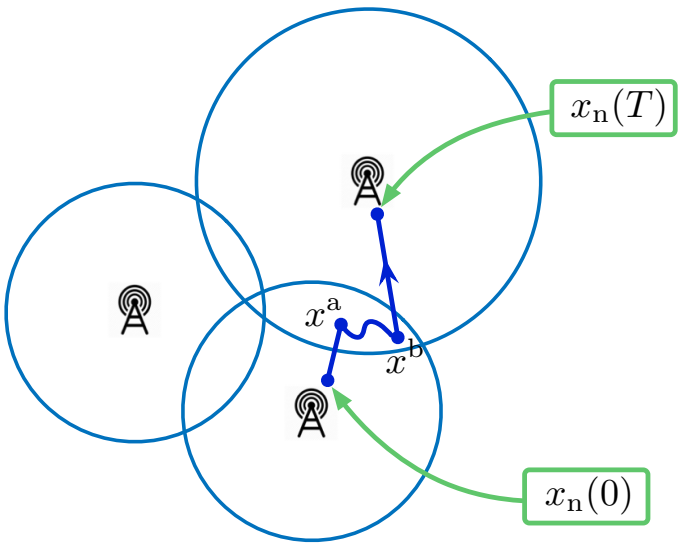


(Bianchin, Liu, Pasqualetti, L-CSS '20)

How do we plan trajectories when RSSI readings change over time?

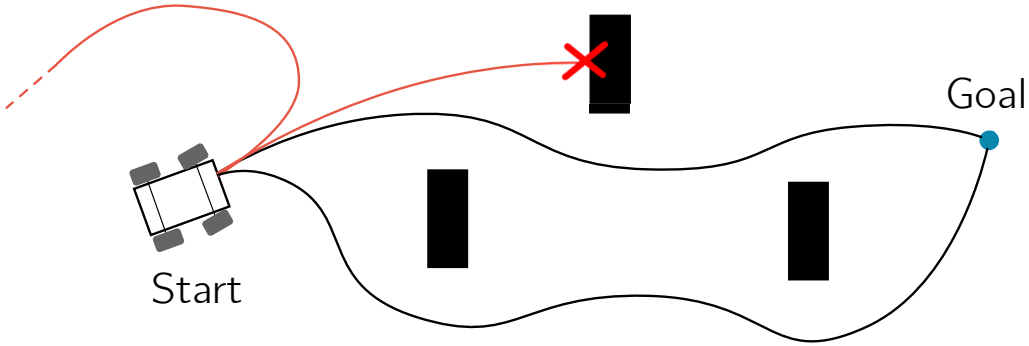
Secure Navigation: Waypoint Design

Given: $x_{\mathrm{n}}(0)$ and $x_{\mathrm{n}}(T)$
Goal: Navigate from $x_{\mathrm{n}}(0)$ to $x_{\mathrm{n}}(T)$



1. Chose $x_{\mathrm{n}}^{\mathrm{a}}$ s.t.
   - $\operatorname{Rank} \mathcal{R}(x_{\mathrm{n}}^{\mathrm{a}}) \geq 2$
   - $x_{\mathrm{n}}^{\mathrm{a}} \in \mathcal{S}(x_{\mathrm{n}}(0))$
2. Navigate from $x_{\mathrm{n}}(0)$ to $x_{\mathrm{n}}^{\mathrm{a}}$ via secure trajectories
3. Chose $x_{\mathrm{n}}^{\mathrm{b}}$ s.t.
   - $\operatorname{Rank} \mathcal{R}(x_{\mathrm{n}}^{\mathrm{b}}) \geq 2$
   - $x_{\mathrm{n}}^{\mathrm{b}} \in \mathcal{S}(x_{\mathrm{n}}(T))$
4. Navigate from $x_{\mathrm{n}}^{\mathrm{a}}$ to $x_{\mathrm{n}}^{\mathrm{b}}$
5. Navigate from $x_{\mathrm{n}}^{\mathrm{b}}$ to $x_{\mathrm{n}}(T)$ via secure trajectories

Can We Make Robotic Navigation Secure?



1. Can **attacker** modify trajectory and escape detectability?

✓ Yes, under appropriate design

2. Can **planner** design secure trajectories?

✓ Yes, there is a tradeoff between security and control energy

Systems with nonlinear dynamics

(control inputs/states) $\leftrightarrow$ (security)

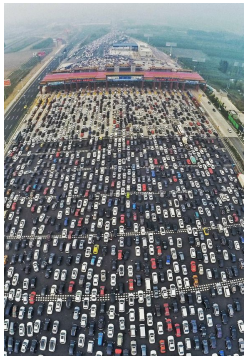
Conclusions and Directions

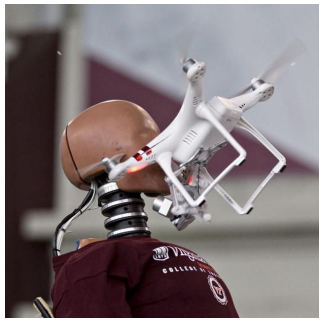
Increasing complexities and performance needs

Conclusions and Directions

Increasing complexities and performance needs

(performance)
$\neq$
(robustness)

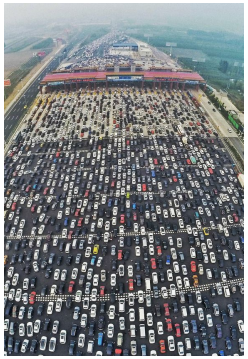


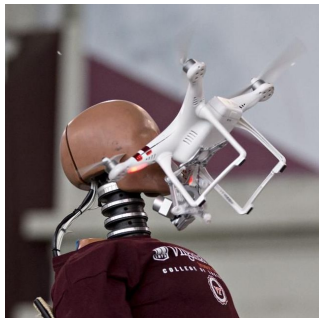


Conclusions and Directions

Increasing complexities and performance needs

(performance)
$\neq$
(robustness)





(source: Facebook)

Networks provide redundancy

Data = facts

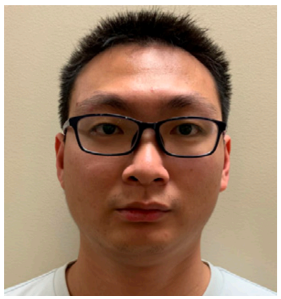
Models+control+optimization

Thanks: Made This Possible

Angy

Tommy

L + C



Yin Chen

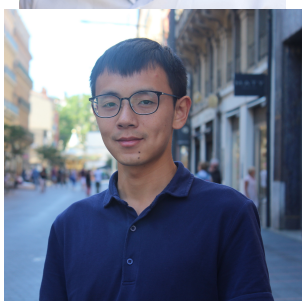
Giacomo

Vaibhav

M+C

Riccardo

Sid+Adrian



Thanks: Could Not Make It

Publications

Application to Traffic and Transportation

- G. Bianchin and F. Pasqualetti, "The impact of routing apps in traffic networks: Emerging oscillations and robust information design," *IEEE Transactions on Automatic Control*, 2020. in preparation
- G. Bianchin and F. Pasqualetti, "Gramian-based optimization for the analysis and control of traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019

Application to Robotics and Complex Networks

- G. Bianchin, Y.-C. Liu, and F. Pasqualetti, "Secure navigation of robots in adversarial environments," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 1–6, 2020
- Y.-C. Liu, G. Bianchin, and F. Pasqualetti, "Secure trajectory planning against undetectable spoofing attacks," *Automatica*, vol. 112, p. 108655, 2020
- G. Bianchin, P. Frasca, A. Gasparri, and F. Pasqualetti, "The observability radius of networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3006–3013, 2017

Publications (2)

Conference proceedings

- G. Bianchin, F. Pasqualetti, and S. Kundu, "Resilience of traffic networks with partially controlled routing," in *American Control Conference*, (Philadelphia, PA, USA), pp. 2670–2675, July 2019

- G. Bianchin and F. Pasqualetti, "A network optimization framework for the analysis and control of traffic dynamics and intersection signaling," in *IEEE Conf. on Decision and Control*, (Miami, FL, USA), pp. 1017–1022, Dec. 2018

- G. Bianchin and F. Pasqualetti, "Time-delay attacks in network systems," in *Cyber-Physical Systems Security*, pp. 157–174, Springer International Publishing, 2018

- T. Menara, G. Bianchin, M. Innocenti, and F. Pasqualetti, "On the number of strongly structurally controllable networks," in *American Control Conference*, (Seattle, WA, USA), pp. 340–345, May 2017

- G. Bianchin, P. Frasca, A. Gasparri, and F. Pasqualetti, "The observability radius of network systems," in *American Control Conference*, (Boston, MA, USA), pp. 185–190, July 2016

- G. Bianchin, F. Pasqualetti, and S. Zampieri, "The role of diameter in the controllability of complex networks," in *IEEE Conf. on Decision and Control*, (Osaka, Japan), pp. 980–985, Dec. 2015

Control-Theoretic Methods for the Robustness of Network Systems

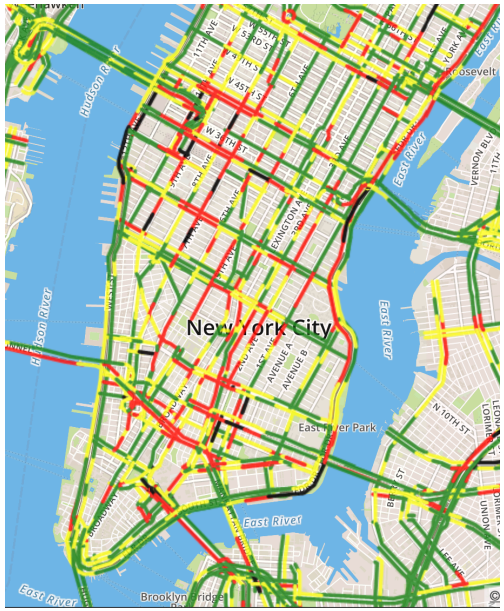
Gianluca Bianchin

Department of Mechanical Engineering
University of California, Riverside

Ph.D. Final Defense | March 3, 2020

Extras: Robustness in Traffic Networks

Directions: Network-Wide Control of Intersections



Given: Real-Time traffic conditions

Objective: Minimize city overall congestion

Control: Automated Intersections

Network-wide control

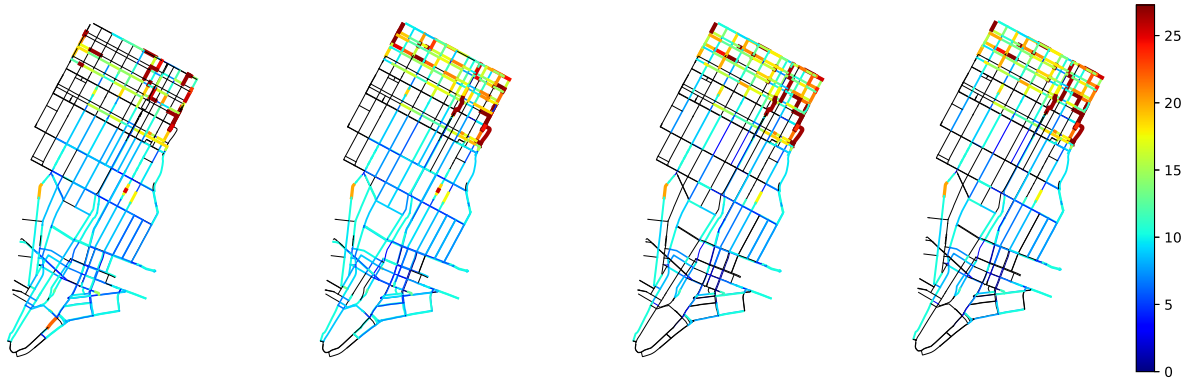
- MPC-based
- Massive optimization, limited horizons

Single Intersection Control

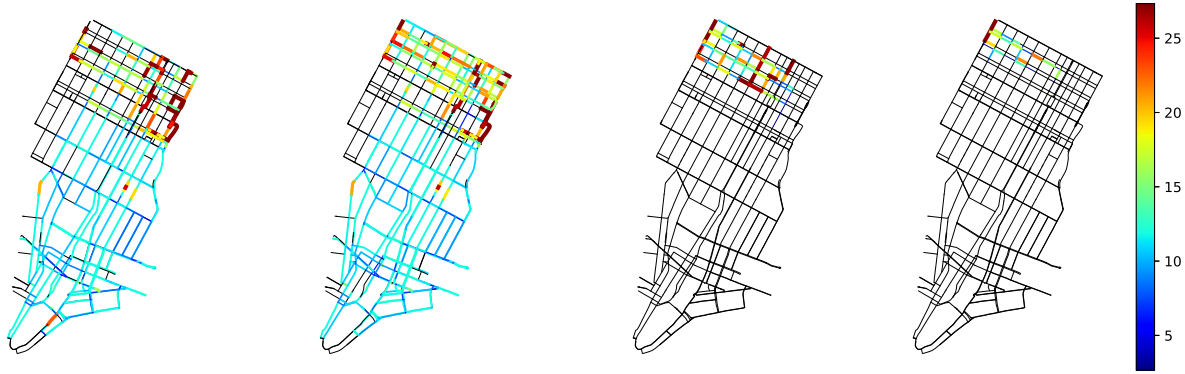
- Max-Pressure, SCOOT, ...
- Suboptimal at network level

Test Case: Manhattan, NY

Control Individual intersections



Network-wide Optimization



- Tractable techniques, ensure system-level optimality
- Outperform state-of-the-art intersection control methods by 20%

Directions (2): Network-Wide Control of Intersections (2)

For linear system $\dot{x} = Ax + Bu$, the **controllability Gramian** is

$$W = \int_0^T e^{-A\sigma} B B^\top e^{-A^\top \sigma} \, d\sigma$$

Minimum-energy control:

$$u(t) = B^\top e^{A^\top (T-t)} W^{-1} x_T$$

$W \rightarrow$ measure of the degree of controllability

(Bianchin, Pasqualetti T-ITS '19)

The following optimization problems are equivalent:

min $\int_0^\infty$ (Netw. congestion) dt

s.t. (Feasible traffic lights)

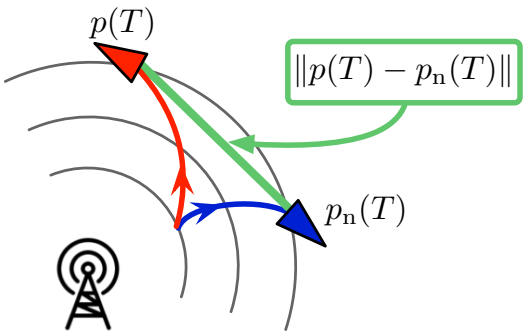
min $\mathrm{Trace}(W)$

s.t. (Approximate dynamics)
(Feasibility constraints)

Extras: Secure Navigation of Robots

Attack Planning

$$\begin{aligned}\text{(attacker)} = \max_{u} \quad & \|p(T) - p_{\mathrm{n}}(T)\| \\ \text{s.t.} \quad & \dot{x} = \text{(double integrator)} \\ & u = \text{(undetectable)} \\ & \|u\| \le u_{\max}\end{aligned}$$



(Liu, Bianchin, Pasqualetti, Automatica '20)

$$u = a_{\mathrm{r}} p + w$$

where $w^{\mathsf{T}} p = 0$, and

$$a_{\mathrm{r}} = (u_{\mathrm{n}}^{\mathsf{T}} p_{\mathrm{n}} + \|v_{\mathrm{n}}\|^2 - \|v\|^2)\|p\|^{-2}$$

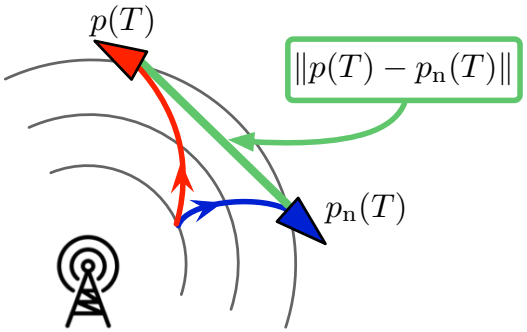
$$w = -\operatorname{sgn}(\lambda^{\mathsf{T}} BWx)\sqrt{u_{\max}^2/\|p\|^2 - a_{\mathrm{r}}{}^2}\, Wx$$

and (λ, x) = solve a two-point BVP

- a_{r}= radial accel.
- w= tangential accel.
- Bang-Bang control
- Can be computed by **solving BVP offline**

Computing Optimal Undetectable Attacks

$$\begin{aligned} \text{(attacker)} = \max_{u} \quad & \|p(T) - p_{\mathsf{n}}(T)\| \\ \text{s.t.} \quad & \dot{x} = \text{(double integrator)} \\ & u = \text{(undetectable)} \\ & \|u\| \leq u_{\max} \end{aligned}$$



For single integrator robots ($A = 0$), BVP can be solved explicitly:

(Bianchin, Liu, Pasqualetti, L-CSS '20)

$$u = a_{\mathsf{r}}\, p + w$$

where $w^{\mathsf{T}} p = 0$, and

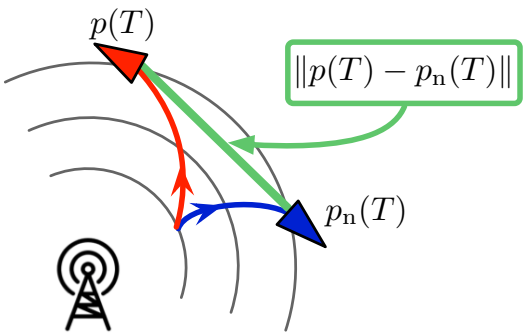
$$a_{\mathsf{r}} = u_{\mathsf{n}}^{\mathsf{T}} p_{\mathsf{n}} \|p\|^{-2}$$

$$w = -\gamma \sqrt{u_{\max}^2 / \|p\|^2 - a_{\mathsf{r}}{}^2}\, W x$$

$\phi \leftarrow$ Angle between $x(t)$ and $-x_{\mathsf{n}}(T)$;
if $\phi = 0$ **then**
  $\gamma \leftarrow 0$;
else if $0 < \phi \leq \pi$ **then**
  $\gamma \leftarrow 1$;
else
  $\gamma \leftarrow -1$;

Computing Optimal Undetectable Attacks

$$\begin{aligned} \max_{u} \quad & \|p(T) - p_{\mathrm{n}}(T)\| \\ \text{s.t.} \quad & \dot{x} = Ax + Bu \\ & u = \text{undetectable} \\ & \|u\| \leq u_{\max} \end{aligned}$$



In summary,

- Double-integrator: two-point BVP (solve numerically, burdensome)
- Single-integrator: feedback law (closed-form, efficient)

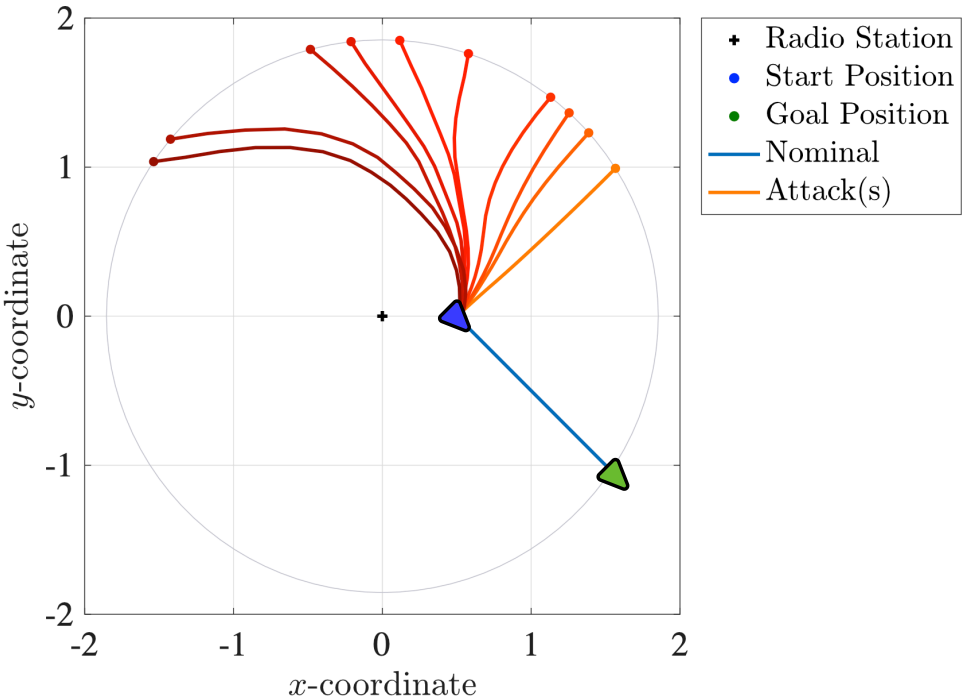
Designing Controls that are Secure

(Liu, Bianchin, Pasqualetti, Automatica '20)

A trajectory is secure **if and only if**

$$u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \qquad \kappa(t) \in \{-1, 1\}$$

What if we **use controls that are not secure**?

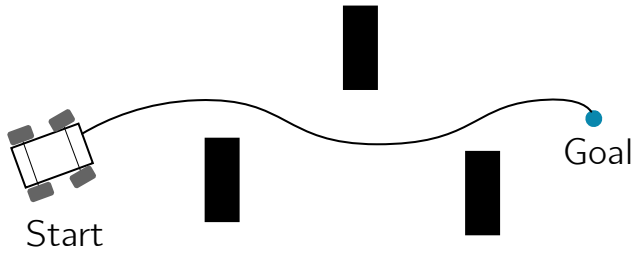


There is a tradeoff between control energy and security

Computing Minimum-Time Secure Trajectories

Given initial and final positions p_{I}, p_{F}

$$\begin{aligned} \min_{\kappa,T} \quad & T \ + \ \|p_{\mathsf{n}}(T) - p_{\mathsf{F}}\| \\ \text{s.t.} \quad & \dot{x}_{\mathsf{n}} = Ax_{\mathsf{n}} + Bu_{\mathsf{n}} \\ & u_{\mathsf{n}} = \kappa \frac{p_{\mathsf{n}}}{\|p_{\mathsf{n}}\|} u_{\max} \end{aligned}$$

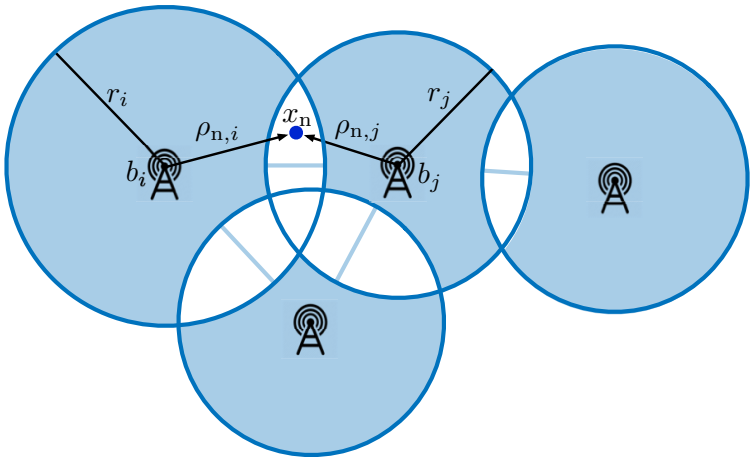


Emerging Difficulty:

What are the configurations $\mathcal{S}$ that are reachable via secure inputs?

1. In general: no systematic way to characterize $\mathcal{S}$
2. Single-Integrator dynamics: $\mathcal{S} = \{x \ : \ x = \alpha x_{\mathsf{n}}(0), \alpha \in \mathbb{R}_{>0}\}$

Existence of Undetectable Attacks



- If robot is in the range r_i of station i

$$\|x_{\mathsf{n}} - b_i\| \leq r_i$$

- Then, i-th RSSI reading is available

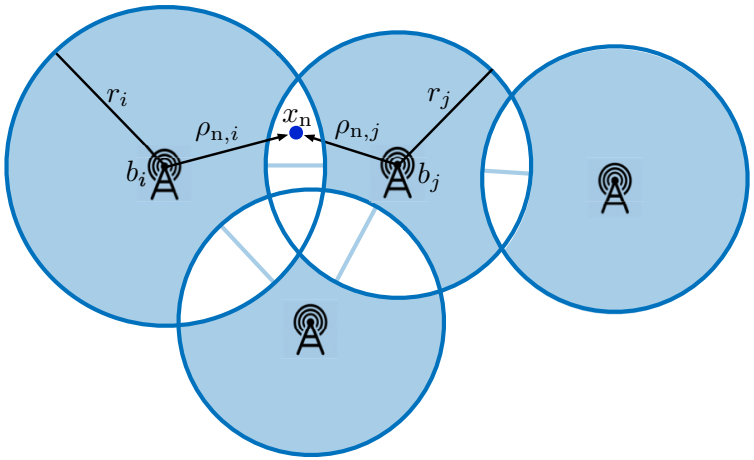
$$y_{\mathsf{n},i}^{\mathsf{RSSI}} = \|x_{\mathsf{n}} - b_i\|^2, \quad y_{\mathsf{n}}^{\mathsf{GNSS}} = x_{\mathsf{n}}$$

There exist undetectable attacks **only if**

$$\mathrm{Rank}\left(\begin{bmatrix} x_{\mathsf{n}} - b_{i_1}, & x_{\mathsf{n}} - b_{i_2}, & \cdots & x_{\mathsf{n}} - b_{i_s} \end{bmatrix}\right) < 2$$

at some time t

Existence of Undetectable Attacks



- If robot is in the range r_i of station i

$$\|x_{\mathsf{n}} - b_i\| \leq r_i$$

- Then, i-th RSSI reading is available

$$y_{\mathsf{n},i}^{\mathsf{RSSI}} = \|x_{\mathsf{n}} - b_i\|^2, \quad y_{\mathsf{n}}^{\mathsf{GNSS}} = x_{\mathsf{n}}$$

There exist undetectable attacks **only if**

$$\mathrm{Rank}(\begin{bmatrix} x_{\mathsf{n}} - b_{i_1}, & x_{\mathsf{n}} - b_{i_2}, & \cdots & x_{\mathsf{n}} - b_{i_s} \end{bmatrix}) < 2$$

at some time t