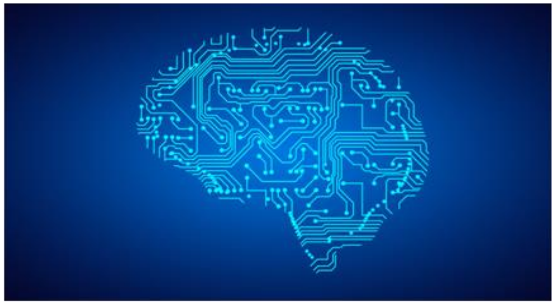
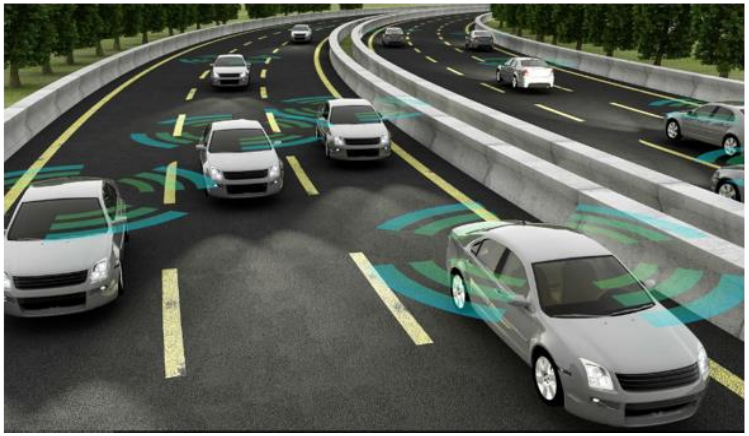
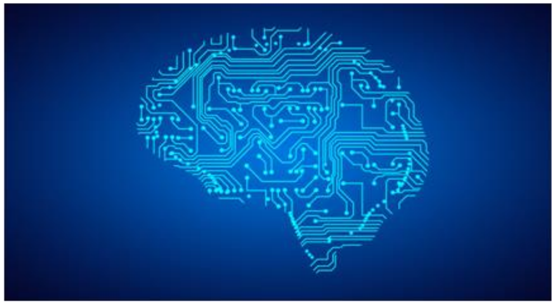
Calibrating Car Following Models

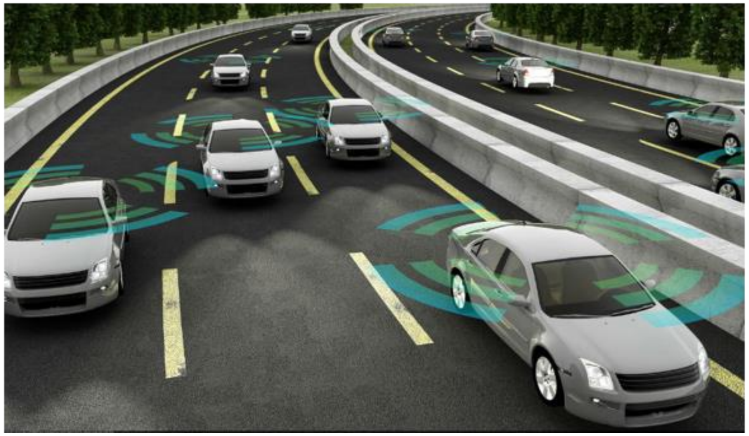




Calibrating Car Following Models

Calibration:

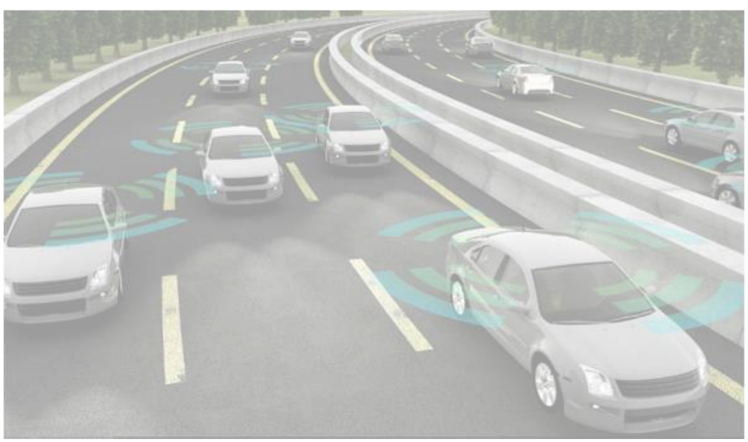




Calibrating Car Following Models

Calibration:

Given some data and a model, find the model parameters that best fit the data



Outline

- Macro and Micro refresher

Outline

- Macro and Micro refresher
- Problem and Motivation

Outline

- Macro and Micro refresher
- Problem and Motivation
- Calibration as Optimization Problem

Outline

- Macro and Micro refresher
- Problem and Motivation
- Calibration as Optimization Problem
- Adjoint method

Outline

- Macro and Micro refresher
- Problem and Motivation
- Calibration as Optimization Problem
- Adjoint method
- Preliminary Results and Discussion

Macroscopic Models

- Cell Transmission Model (CTM)/Lighthill-Whitham-Richard (LWR)

Macroscopic Models

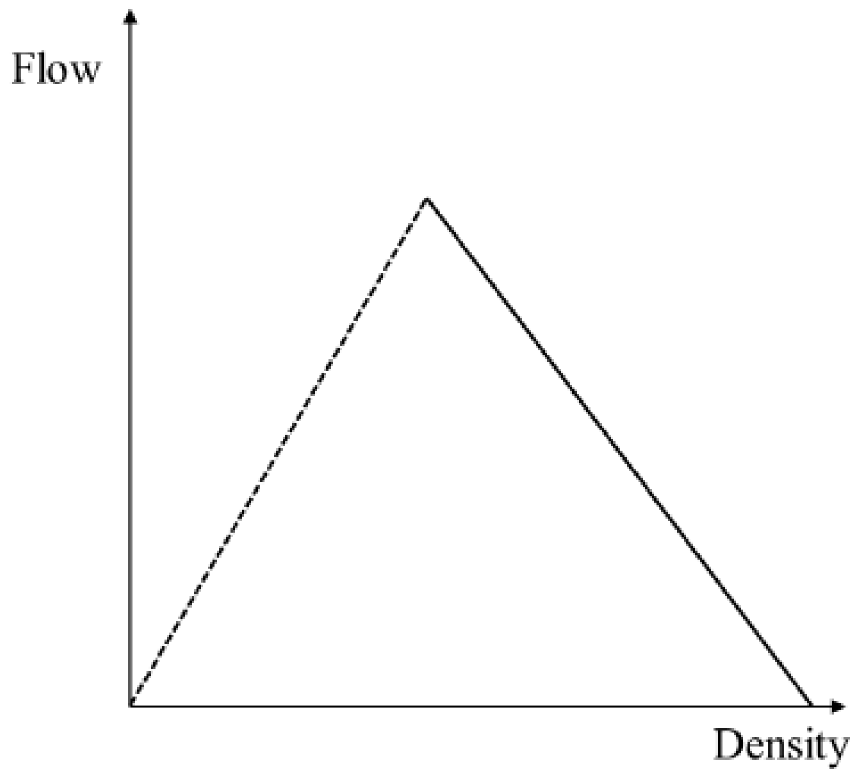
- Cell Transmission Model (CTM)/Lighthill-Whitham-Richard (LWR)
- Fundamental Diagram

Macroscopic Models

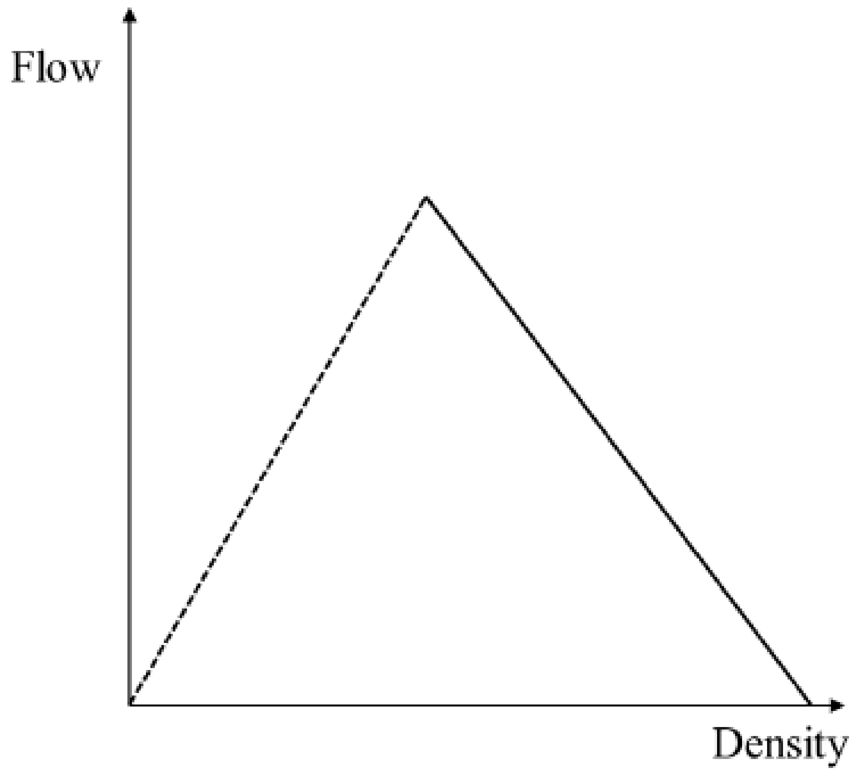
- Cell Transmission Model (CTM)/Lighthill-Whitham-Richard (LWR)
- Fundamental Diagram
- Hyperbolic conservation law; Link to hydrodynamics

Macroscopic Models

- Cell Transmission Model (CTM)/Lighthill-Whitham-Richard (LWR)

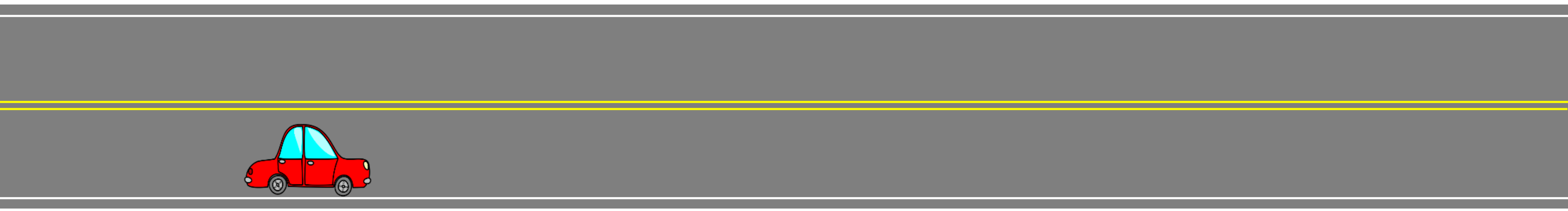


Macroscopic Models



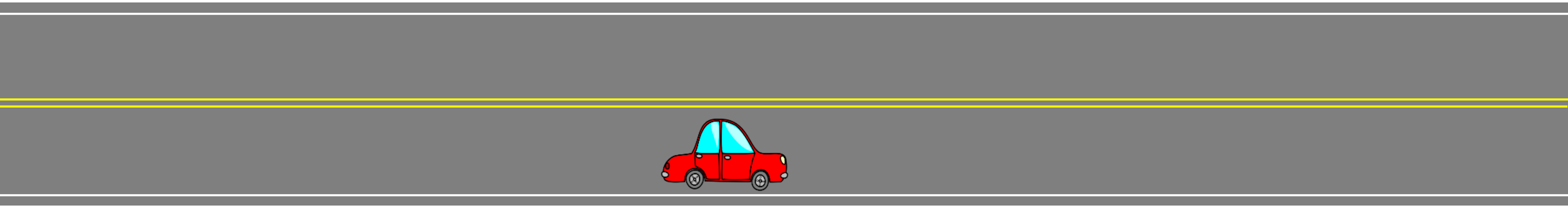
MFD and conservation laws agreed upon framework for macro- modeling.

Car-following = Microsimulation



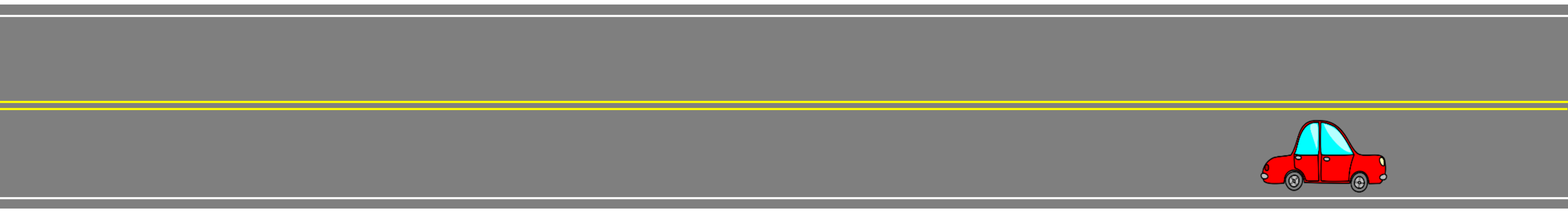
Suppose we know the trajectory of a “lead vehicle”

Car-following = Microsimulation



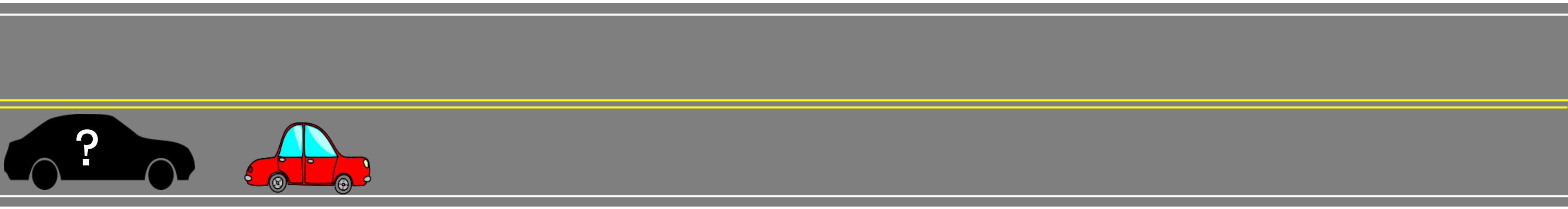
Suppose we know the trajectory of a “lead vehicle”

Car-following = Microsimulation



Suppose we know the trajectory of a "lead vehicle"

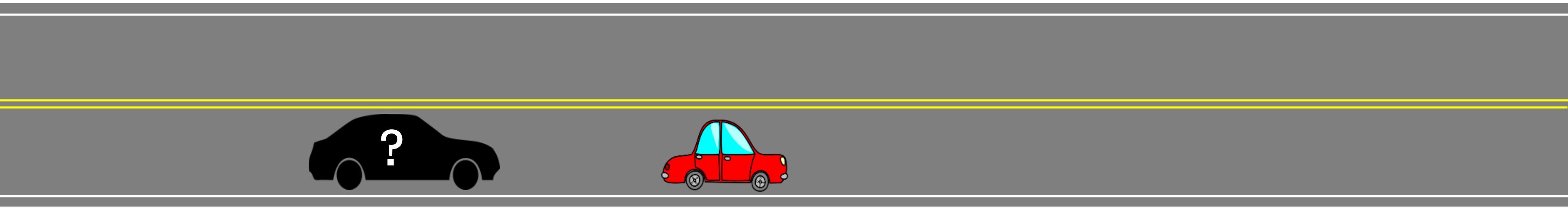
Car-following = Microsimulation



Suppose we know the trajectory of a "lead vehicle"

Car-following model predicts trajectory of "following vehicle"

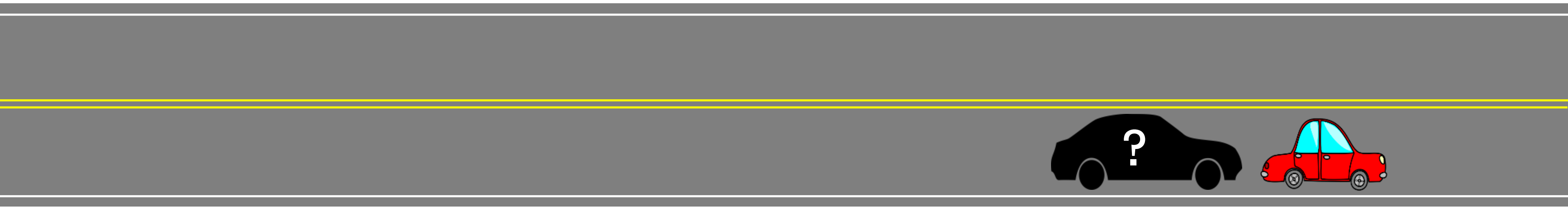
Car-following = Microsimulation



Suppose we know the trajectory of a “lead vehicle”

Car-following model predicts trajectory of “following vehicle”

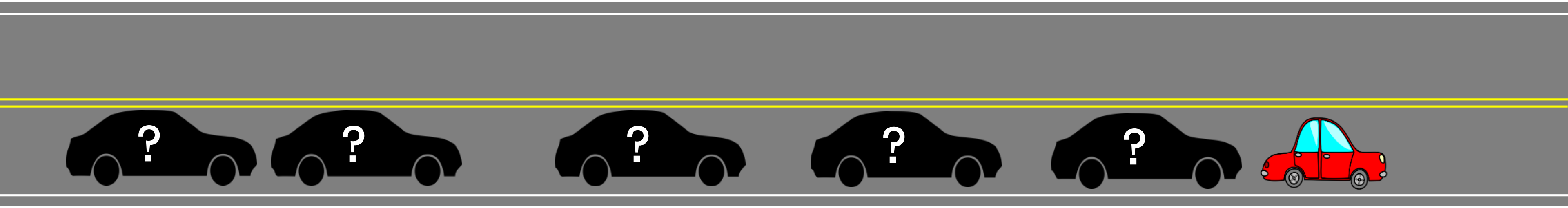
Car-following = Microsimulation



Suppose we know the trajectory of a "lead vehicle"

Car-following model predicts trajectory of "following vehicle"

Car-following = Microsimulation



Suppose we know the trajectory of a “lead vehicle”

Car-following model predicts trajectory of “following vehicle”

Problem Statement

- We have a dataset with many vehicle trajectories

Problem Statement

- We have a dataset with many vehicle trajectories

- We have some car-following model with unknown parameters

Problem Statement

- We have a dataset with many vehicle trajectories
- We have some car-following model with unknown parameters
- Find the parameters that result in the best fit of the model to the data

Problem Statement

- We have a dataset with many vehicle trajectories
- We have some car-following model with unknown parameters
- Find the parameters that result in the best fit of the model to the data
- Nonlinear Nonconvex program. Formalisms later

Why Study Car-following?

- Reason 1

- Reason 2

Why Study Car-following?

- Reason 1

  Micro-models have different strengths compared to macro-

- Reason 2

Why Study Car-following?

- Reason 1

  Micro-models have different strengths compared to macro-

- Reason 2

  Many new sources for trajectory data

Why Study Car-following?

- Reason 1 Micro-models have different strengths compared to macro-
- Micro- can be better for short links; low densities

Why Study Car-following?

- Reason 1 Micro-models have different strengths compared to macro-
- Micro- can be better for short links; low densities
- Micro- gives more flexibility for inhomogeneous traffic

Why Study Car-following?

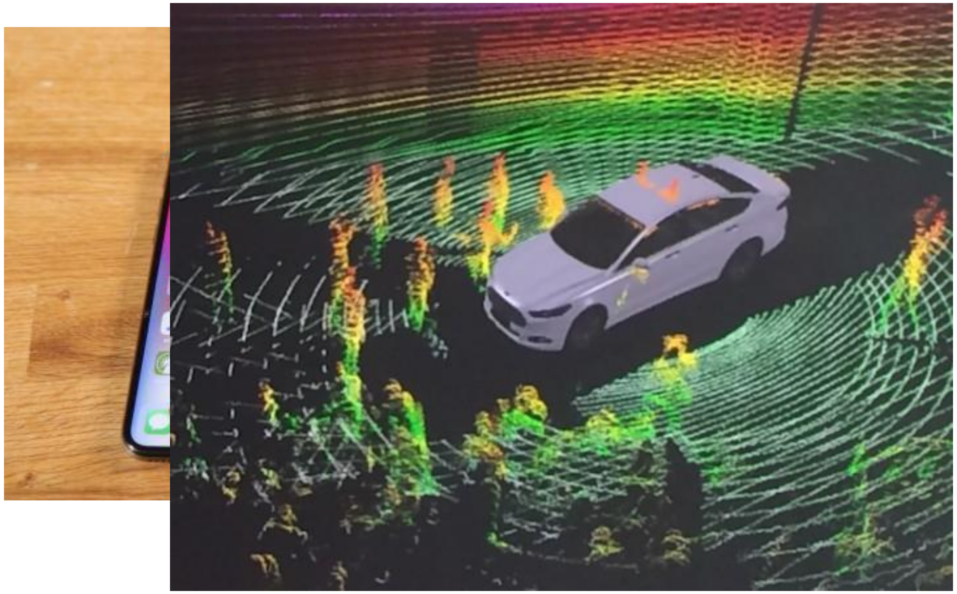
- Reason 2 Many new sources for trajectory data

- GPS (smart phones)

Photo credit: CNBC

Why Study Car-following?

- Reason 2 Many new sources for trajectory data

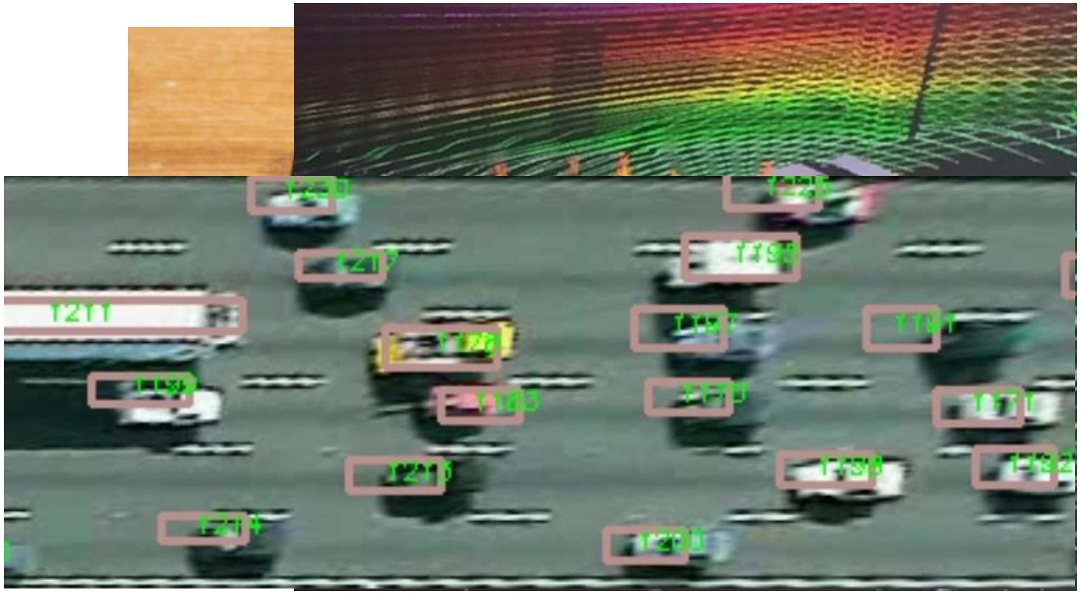


- GPS (smart phones)
- Lidar/Radar (A.V.)

Photo credit: CNBC, cleantechnica

Why Study Car-following?

- Reason 2 Many new sources for trajectory data



- GPS (smart phones)
- Lidar/Radar (A.V.)
- Computer Vision (traffic cameras)

Photo credit: CNBC, cleantechnica, FHWA

Formulation

Some notation

- $x_n(t)$: vector consisting of position and speed of vehicle n, at time t, calculated from the car-following model.
- $h_n(x_n(t), x_n(t-\tau_n), x_{n-1}(t-\tau_n), p_n)$: generic car-following model such that $\dot{x}_n = h_n$.
- p_n: a vector consisting of model parameters describing vehicle n.
- $\hat{x}_n(t)$: "true" measurements of vehicle n, taken from data.

Formulation

Define Calibration as the following optimization problem:

$$
\begin{aligned}
\min_{p} \quad & F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt \\
\text{s.t.} \quad & \dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0, \quad t \in [t_i, T_{i-1}], \quad i = 1, \ldots, n \\
& x_i(t_i) - (\hat{x}_i(t_i) + a_i) = 0 \quad i = 1, \ldots, n \\
& b_{\text{low}} \leq p \leq b_{\text{high}}
\end{aligned}
$$

Formulation

Define Calibration as the following optimization problem:

Maximize Goodness of fit (minimize loss)

$$\min_{p} \quad F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt$$

$$\text{s.t.} \quad \dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0, \quad t \in [t_i, T_{i-1}], \quad i = 1, \ldots, n$$

$$x_i(t_i) - (\hat{x}_i(t_i) + a_i) = 0 \quad i = 1, \ldots, n$$

$$b_{\text{low}} \leq p \leq b_{\text{high}}$$

Formulation

Define Calibration as the following optimization problem:

Maximize Goodness of fit (minimize loss)

Vehicles must follow model

$$\min_{p} \quad F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt$$

$$\text{s.t.} \quad \dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0, \quad t \in [t_i, T_{i-1}], \quad i = 1, \ldots, n$$

$$x_i(t_i) - (\hat{x}_i(t_i) + a_i) = 0 \quad i = 1, \ldots, n$$

$$b_{\text{low}} \leq p \leq b_{\text{high}}$$

Formulation

Define Calibration as the following optimization problem:

Maximize Goodness of fit (minimize loss)

$$\min_{p} \quad F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt$$

Vehicles must follow model

$$\text{s.t.} \quad \dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0, \quad t \in [t_i, T_{i-1}], \quad i = 1, \dots, n$$

$$x_i(t_i) - (\hat{x}_i(t_i) + a_i) = 0 \quad i = 1, \dots, n$$

$$b_{\text{low}} \leq p \leq b_{\text{high}}$$

Initial conditions (for model)

Formulation

Define Calibration as the following optimization problem:

Maximize Goodness of fit (minimize loss)

Vehicles must follow model

$$\min_{p} \quad F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt$$

$$\text{s.t.} \quad \dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0, \quad t \in [t_i, T_{i-1}], \quad i = 1, \ldots, n$$

$$x_i(t_i) - (\hat{x}_i(t_i) + a_i) = 0 \quad i = 1, \ldots, n$$

$$b_{\text{low}} \leq p \leq b_{\text{high}}$$

Initial conditions (for model)

Box constraints on parameters

Video Interlude

Next Generation Simulation (NGSim) dataset created by FHWA.

1 camera of 1 location (U.S. I-80) for 15 minutes

Adjoint Method

Adjoint method allows us to find the gradient of this optimization problem in an efficient way

Adjoint Method

Adjoint method allows us to find the gradient of this optimization problem in an efficient way

We can also find the gradient using finite differences. Adjoint method is better. Why?

Adjoint Method

Adjoint method allows us to find the gradient of this optimization problem in an efficient way

We can also find the gradient using finite differences. Adjoint method is better. Why?

Adjoint method is faster, especially when the model has a lot of parameters.

Adjoint Method

Adjoint method allows us to find the gradient of this optimization problem in an efficient way

We can also find the gradient using finite differences. Adjoint method is better. Why?

Adjoint method is faster, especially when the model has a lot of parameters.

Adjoint method can also be more accurate.

Adjoint Method

Adjoint method is for optimization problems constrained by differential equations

Adjoint: O(1)

Finite Differences: O(m)

m is number of parameters.

Adjoint Method

Objective function:

$$F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t) dt$$

Adjoint Method

Objective function:

$$F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t)dt$$

Model:

$$\dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0$$

Adjoint Method

Objective function:

$$F = \sum_{i=1}^{n} \int_{t_i}^{T_{i-1}} f(x_i, \hat{x}_i, t) dt$$

Model:

$$\dot{x}_i(t) - h_i(x_i(t), x_{i-1}(t), p_i) = 0$$

Augmented Objective function:

$$L = \int_{t_n}^{T_{n-1}} dt \Big[f(x_n, \hat{x}_n, t) + \lambda^T(t) \big(\dot{x}_n(t) - h_n(x_n(t), x_{n-1}(t), p_n) \big) \Big]$$

Adjoint Method

Augmented Objective function:

$$L = \int_{t_n}^{T_{n-1}} dt \left[f(x_n, \hat{x}_n, t) + \lambda^T(t) \big(\dot{x}_n(t) - h_n(x_n(t), x_{n-1}(t), p_n) \big) \right]$$

Adjoint variables (think of as Lagrange multiplier)

Adjoint Method

Augmented Objective function:

$$L = \int_{t_n}^{T_{n-1}} dt \left[f(x_n, \hat{x}_n, t) + \lambda^T(t) \big(\dot{x}_n(t) - h_n(x_n(t), x_{n-1}(t), p_n) \big) \right]$$

Adjoint variables (think of as Lagrange multiplier)

$$\frac{dF}{dp} = \frac{dL}{dp}$$

Adjoint Method

Augmented Objective function:

$$L = \int_{t_n}^{T_{n-1}} dt \left[f(x_n, \hat{x}_n, t) + \lambda^T(t)\big(\dot{x}_n(t) - h_n(x_n(t), x_{n-1}(t), p_n)\big) \right]$$

Adjoint variables (think of as Lagrange multiplier)

$$\frac{dF}{dp} = \frac{dL}{dp}$$

1. Solve model for x
2. Solve adjoint system for Lambda
3. Calculate gradient

Numerical Experiment

Goal: To compare different ways of solving calibration problem.

1. Gradient-free

2. Gradient-based using finite differences

3. Gradient-based using adjoint method

Numerical Experiment

	Finite BFGS	Adjoint BFGS	NM Simplex	Differential Evolution
# Objective Evaluations	972	34	1965	3246
# Gradient Evaluations	0 (118)	6	0	0
Final Objective Value	1723	3509	1613	2914
Runtime (s)	9.64	1.23	24.69	32.74

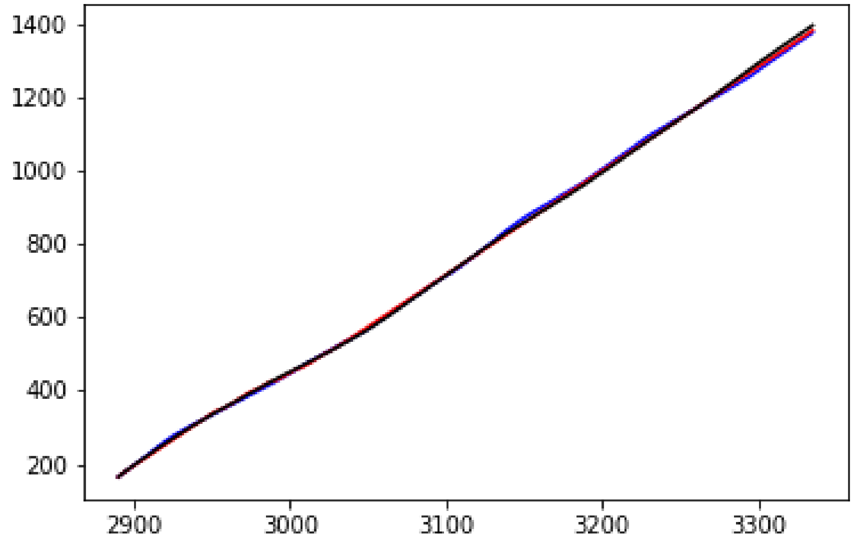
Numerical Experiment

	Finite BFGS	Adjoint BFGS	NM Simplex	Differential Evolution
# Objective Evaluations	972	34	1965	3246
# Gradient Evaluations	0 (118)	6	0	0
Final Objective Value	1723	3509	1613	2914
Runtime (s)	9.64	1.23	24.69	32.74

	Objective Function	Finite Differences	Adjoint method
Runtime (s)	0.011	0.047	0.032

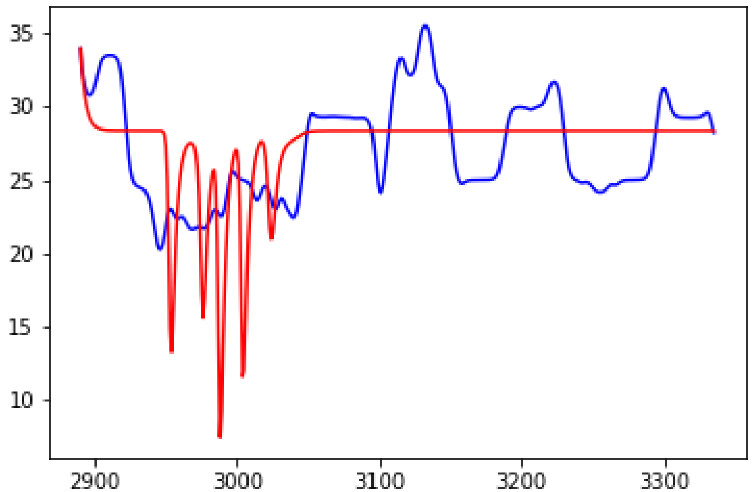
Numerical Experiment

	Finite BFGS	Adjoint BFGS	NM Simplex	Differential Evolution
# Objective Evaluations	972	34	1965	3246
# Gradient Evaluations	0 (118)	6	0	0
Final Objective Value	1723	3509	1613	2914
Runtime (s)	9.64	1.23	24.69	32.74



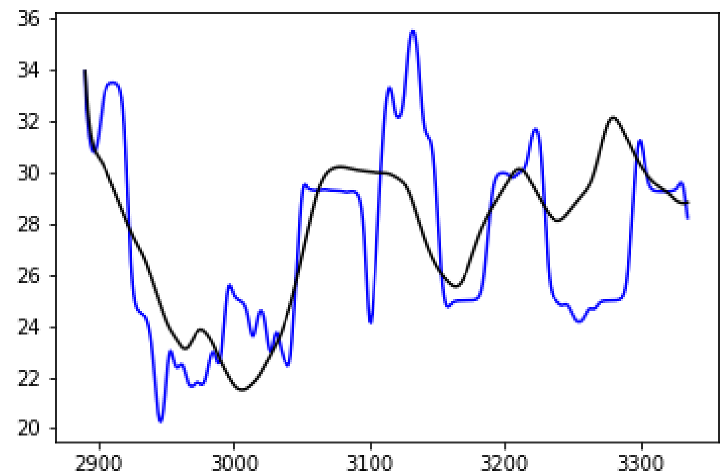
Numerical Experiment

Finite Dif. BFGS vs Adjoint BFGS



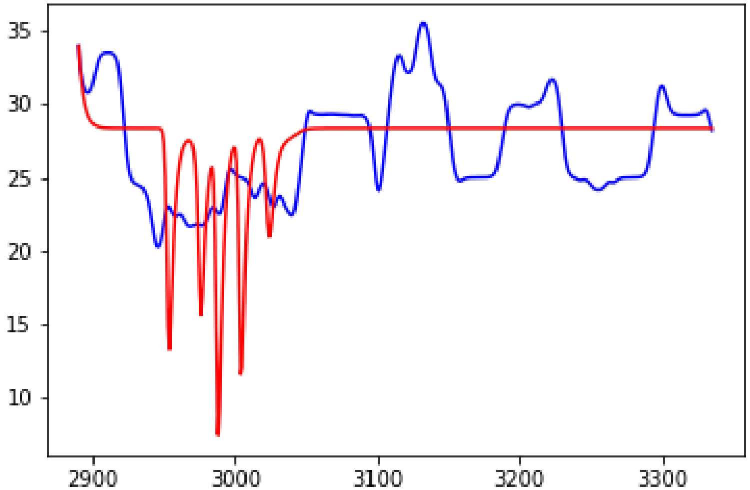
Speed-Time Planes

Blue is measurements (data)



Numerical Experiment

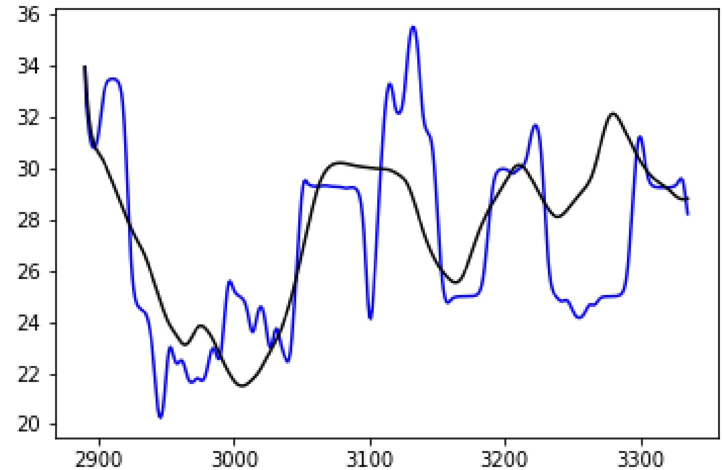
Finite Dif. BFGS vs Adjoint BFGS



Speed-Time Planes

Blue is measurements (data)

Finite Differences has given strange result because of inaccurate gradient



Conclusion

- Can use adjoint method to calculate gradient for calibration problem efficiently

Conclusion

- Can use adjoint method to calculate gradient for calibration problem efficiently
- Works for models posed as ODE and DDE; works for Hessian; works for arbitrary platoon size

Conclusion

- Can use adjoint method to calculate gradient for calibration problem efficiently
- Works for models posed as ODE and DDE; works for Hessian; works for arbitrary platoon size
- Can use gradient-based algorithms to solve calibration problem faster compared to grad-free

Conclusion

- Can use adjoint method to calculate gradient for calibration problem efficiently
- Works for models posed as ODE and DDE; works for Hessian; works for arbitrary platoon size
- Can use gradient-based algorithms to solve calibration problem faster compared to grad-free
- Choice of optimization algorithm affects results; lower objective value not necessarily better!