Cooperative Collision Avoidance via Proximal Message Passing

Hao Yi Ong and J. Christian Gerdes

Stanford University

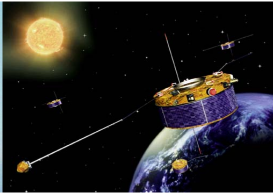
July 3, 2015

Outline

Cooperative collision avoidance (CCA)

- multivehicle problem
  - stage-cost function (fuel, trajectory deviation, etc.)
  - linearized dynamics
  - shared collision avoidance constraints





Complications

- practically
  - vehicles has limited network information, sensing, actuation, and computation capabilities
  - dynamic network: changing number of vehicles and obstacles
- mathematically
  - CCA as mathematical program is nonconvex
  - large problem size ($>$100 vehicles)
  - efficient algorithm needed for online implementation

Goals

application-agnostic, robust method for

- arbitrary-scale optimization
  - dynamic control of large-scale network
- decentralized optimization
  - vehicles/devices coordinate to solve large problem by passing relatively small messages
- model predictive control (MPC)
  - fast enough for real-time use on embedded systems

Previous approaches

- mixed-integer programming [M+12a]
- distributed algorithm
  - satellite swarms [M+12b]
  - cooperative robots [B+13]
- former works well for problem with few vehicles, while latter capitalizes on application-specific problem structures

Proximal message passing

- builds on network energy management by Kraning et al. [K^+13]
  - convex objective and constraints only
  - finite horizon problem
  - offline solution
- extensions handle
  - nonconvexity
  - infinite horizon problem
  - real-time application

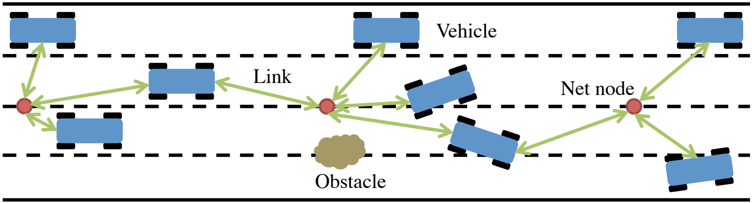
Proximal message passing

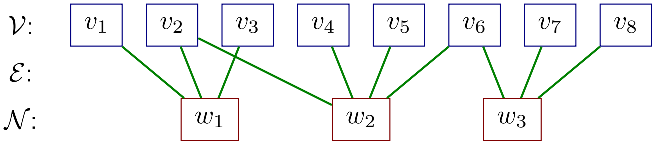
- decentralized method to solve CCA for real-time MPC applications
  - each vehicle plans its own state trajectory and control inputs
  - vehicles coordinate via simple positional message exchanges with neighbors
  - "plug-and-play" concept for more/less vehicles and obstacles
- can handle **enormous** problems

Outline

Network model

- *network* consists of vehicles $\mathcal{V}$, communication links $\mathcal{E}$, and nets $\mathcal{W}$
- *nets* represent communication relays or transponders

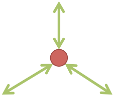




Vehicles

- models vehicles with linearizable dynamics
- variables are state-control pairs $(x_v, u_v) \in \mathbf{R}^{nT} \times \mathbf{R}^{mT}$
- objective function $f_v(x_v, u_v) : \mathbf{R}^{nT} \times \mathbf{R}^{mT} \to \mathbf{R} \cup \{+\infty\}$
  - fuel consumption, trajectory tracking, smooth trajectory,...
  - $+\infty$ used to encode vehicle constraints

Nets



- models collision warning and guidance software with transponder
- objective function $\mu_w\left(x_{\mathcal{V}_w}\right) : \mathbf{R}^{n|\mathcal{V}_w|T} \to \mathbf{R} \cup \{+\infty\}$
- nonconvex constraints of the form

$$\left\|p_{i,\tau} - p_{j,\tau}\right\|_2 \geq D$$

  - multiple balls to cover forbidden region around vehicle
  - can also use minimum covering ellipsoid or any other shape

Cooperative collision avoidance problem

cooperative collision avoidance problem (CCA):

$$\text{minimize} \quad \sum_{v \in \mathcal{V}} f_v\left(x_v, u_v\right) + \sum_{w \in \mathcal{W}} \mu_w\left(x_{\mathcal{V}_w}\right)$$

- variables are vehicle state-control pairs $(x, u) \in \mathbf{R}^{|\mathcal{V}| \times nT} \times \mathbf{R}^{|\mathcal{V}| \times mT}$
- linear dynamics and other constraints are contained in vehicle objectives f_v
- nonconvex collision avoidance constraints are contained in net objectives μ_w

Outline

Convex-concave procedure (CCP)

- express problem with 'difference of convex' functions

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

  where f_i and g_i are convex

- iterative procedure: convexify nonconvex parts at k^{th} iteration by replacing $g(x)$ with

$$\hat{g}(x) = g\left(x^{(k)}\right) + \nabla g\left(x^{(k)}\right)^T \left(x - x^{(k)}\right)$$

Convex-concave procedure

- collision avoidance constraint function is a "negative ball"

$$D - \|p_{i,\tau} - p_{j,\tau}\|_2 \leq 0$$

- first term is constant (convex); second term is concave
- linearize second term in objective to get halfspace

$$D - g^T (p_{i,\tau} - p_{j,\tau}) \leq 0,$$

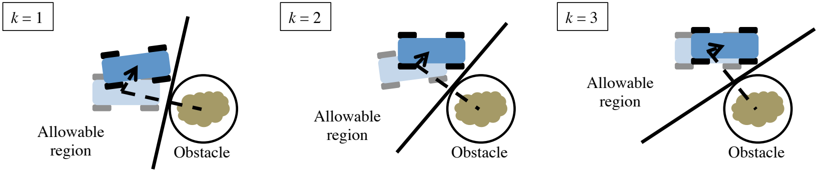
where

$$g = \frac{p^k_{i,\tau} - p^k_{j,\tau}}{\left\|p^k_{i,\tau} - p^k_{j,\tau}\right\|_2}$$

Convex-concave procedure

intuition: ball for forbidden region is approximated by a halfspace tangent to the ball

- in each iteration, vehicle position is optimized within halfspace constraint and halfspace approximation is updated thereafter



Problem decomposition

convexified CCA:

$$\text{minimize} \quad \sum_{v \in \mathcal{V}} f_v\left(x_v, u_v\right) + \sum_{w \in \mathcal{W}} g_w^k\left(x_{\mathcal{V}_w}\right)$$

issues:

- problem dimension is still large
- decomposition needs to allow vehicle and obstacle plug-and-play

Consensus optimization

idea: decompose each iteration of convexified CCA into small, manageable subproblems using ADMM [B+10]

convexified CCA in consensus form:

$$\begin{array}{ll} \text{minimize} & \sum_{v \in \mathcal{V}} f_v\left(x_v, u_v\right) + \sum_{w \in \mathcal{W}} g_w^k\left(\tilde{x}_{\mathcal{V}_w}\right) \\ \text{subject to} & x = \tilde{x} \end{array}$$

- f_v and g_w^k encode vehicle objectives and convexified net constraints in the k^{th} iteration of CCP
- $x = \tilde{x}$ are *consensus* constraints that couple solutions to independent f_v and g_w^k minimizations

Proximal message passing algorithm

- define "price" term to be running sum of inconsistencies:

$$y := y + (x - \tilde{x})$$

- repeat until convergence:

1. **prox controller update:** in parallel, minimize each vehicle's f_v plus vehicle's price component y_v
2. **avoidance projection update:** in parallel, minimize each net's g_w^k plus net's price component y_w
3. **price update:** in parallel, update price with the difference between prox controller and avoidance projection solutions

Proximal message passing algorithm

repeat until convergence:

1. *prox controller update*

$$\left(x_v^{\kappa+1}, u_v^{\kappa+1}\right) := \underset{(x_v, u_v)}{\operatorname{argmin}} \left(f_v\left(x_v, u_v\right) + (\rho/2)\left\|x_v - \left(\tilde{x}_v^\kappa - y_v^\kappa\right)\right\|_2^2\right)$$

   in parallel, for each vehicle
   ($\rho > 0$; RHS is **proximal operator** of f_v at $\tilde{x}_v^\kappa - y_v^\kappa$)

2. *avoidance projection update*

$$\tilde{x}_{\mathcal{V}_w}^{\kappa+1} := \underset{\tilde{x}_{\mathcal{V}_w}}{\operatorname{argmin}} \left(g_w^\kappa\left(\tilde{x}_{\mathcal{V}_w}\right) + (\rho/2)\left\|\tilde{x}_{\mathcal{V}_w} - \left(x_{\mathcal{V}_w}^\kappa + y_{\mathcal{V}_w}^\kappa\right)\right\|_2^2\right)$$

   in parallel, for each net
   ($\rho > 0$; RHS is **projection operator** onto convexified constraints at $x_{\mathcal{V}_w}^\kappa + y_{\mathcal{V}_w}^\kappa$)

3. *scaled price update*

$$y_{\mathcal{V}_w}^{\kappa+1} := y_{\mathcal{V}_w}^\kappa + \left(x_{\mathcal{V}_w}^\kappa - \tilde{x}_{\mathcal{V}_w}^\kappa\right)$$

   in parallel, for each net

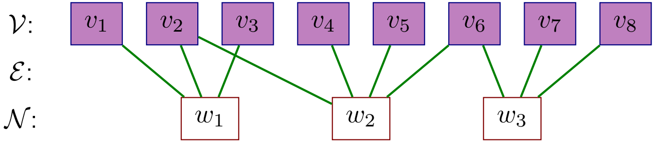
Convergence

for proximal message passing objective functions just has to be convex closed proper (can be nondifferentiable, take on $+\infty$ values, ...)

- *residual convergence*: $x \to \tilde{x}$ (trajectories deconflicted)
- *objective convergence*: $f\left(x^\kappa, u^\kappa\right) + g\left(\tilde{x}^\kappa\right) \to f_0^\star$ (controls are locally optimal)
- *price convergence*: $y^\kappa \to y^\star$ (optimal prices are found)

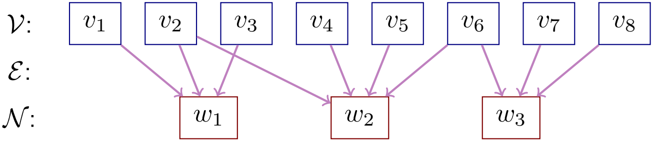
Example

vehicles compute new tentative trajectories



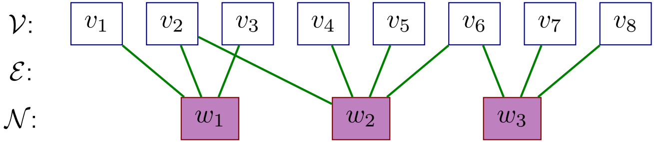
Example

vehicles send tentative trajectories to neighboring nets



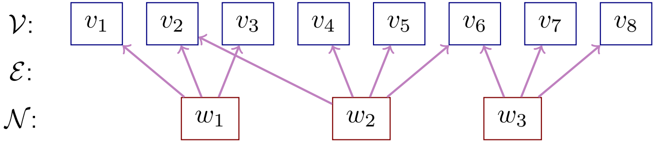
Example

nets deconflict trajectories; update prices



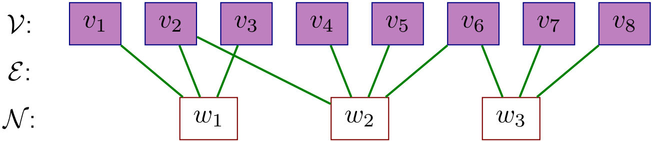
Example

nets send updated prices, trajectories to neighboring vehicles



Example

repeat



Proximal message passing algorithm

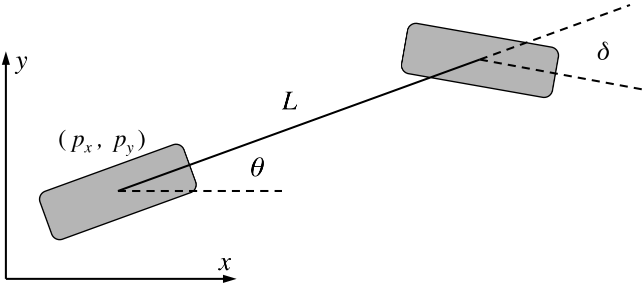
- each vehicle and net only has knowledge of its own objective function
- for each vehicle type, need to implement own minimizer
  - can have different types of vehicles with different dynamics, constraints (motorcycle, cars,...)
- all message passing is local, between vehicles and adjacent nets
- no global coordination other than iteration synchronization

Outline

Vehicle model

simple kinematic bicycle model of the car

$$\dot{p}_x = \nu \cos(\theta)\,, \qquad \dot{p}_y = \nu \sin(\theta)\,, \qquad \dot{\theta} = \frac{\nu}{L}\tan(\theta)$$



Model predictive control

consider MPC algorithm

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{v\in\mathcal{V}(t)} \left(\lambda \sum_{\tau=0}^{T-2} \left(\delta_{v,\tau+1} - 2\delta_{v,\tau} + \delta_{v,\tau-1}\right)^2 \right. \\
& \displaystyle\qquad \left. + \sum_{\tau=1}^{T} \left\| p_\tau^{\mathsf{veh}} - \bar{p}_\tau^{\mathsf{veh}} \right\|_2^2 \right) + \sum_{w\in\mathcal{W}(t)} \mu_w\left(x_{\mathcal{V}_w}\right) \\
\text{subject to} & x_{v,0} = x_v^{\mathsf{init}}, \\
& x_{v,\tau+1} = A_{v,\tau} x_{v,\tau} + B_{v,\tau}\delta_{v,\tau} + c_{v,\tau}, \\
& \forall v \in \mathcal{V}\left(t\right),\ \ \tau = 0,\ldots,T-1,
\end{array}
$$

- solves CCA only for the next T discrete time steps
- locally optimizes for smooth steering and reference tracking
- repeatedly executes first input of control sequence

Model predictive control

transform into familiar convexified CCA in consensus form

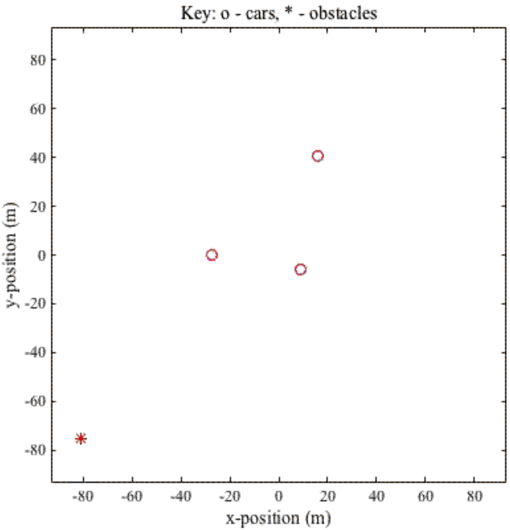
$$\text{minimize} \quad \sum_{v \in \mathcal{V}} f_v\left(x_v, u_v\right) + \sum_{w \in \mathcal{W}} g_w^k\left(x_{\mathcal{V}_w}\right)$$

- f_v locally optimizes for smooth steering and reference tracking and encodes vehicle dynamics constraints
- g_w^k encodes convexified constraints at the k^{th} CCP iteration

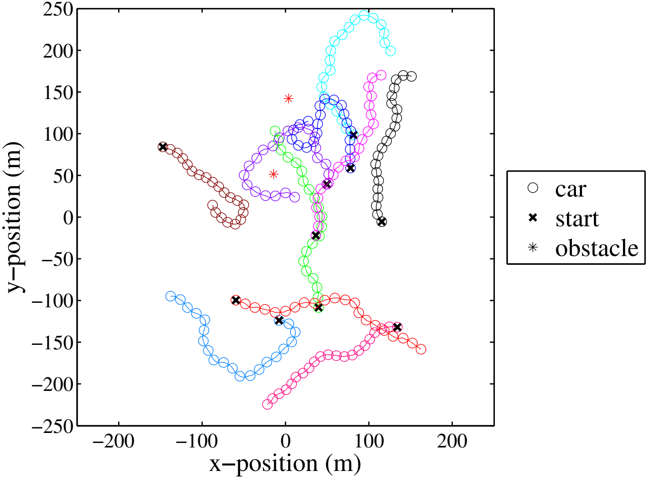
Network topology and simulation

- number of vehicles ranges from 5 to 500
- number of nets ranges from 0 to 200
- MPC: 4 sec time horizon, 0.1 sec intervals
- network topology (initial positions, reference trajectories, obstacle locations) chosen randomly
  - vehicles linked to nets if in close proximity

Example: Proximal message passing



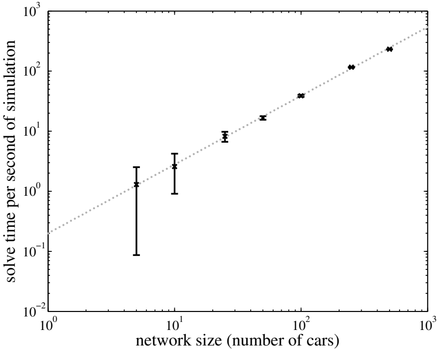
Example: 10 vehicles



Serial single-thread implementation

- use CVXGEN to generate custom `C` optimization code
- examples run on a 2.7 Ghz Intel i7 with 8 GB of RAM
- objective values at convergence of message passing algorithm always $<$1% suboptimal when compared with CVX for small network sizes for each (centralized) CCP iteration

Solve time scaling (serial)



- fit exponent is 1.145; solve time scales as $O\left(|\mathcal{V}|^{1.145}\right)$
- with fully decentralized computation, average of 35 ms per MPC iteration for **any network size**

Outline

Summary

- developed a completely decentralized local optimization method for guidance and control of multivehicle networks
- decentralized computation allows for millisecond solve times **independent** of network size
- when combined with MPC, can be used for real-time network operation
- envision a robust plug-and-play system for any vehicle types

Future work

- robust, asynchronous message passing algorithm
  - work with unreliable communication or network protocols
- more sophisticated vehicle models
- shared constraints beyond collision avoidance

References

- S. Boyd et al.
  Distributed optimization and statistical learning via the alternating direction method of multipliers.
  Foundations and Trends in Optimization, 1(3):1–122, 2010.
- J. Bento et al.
  A message-passing algorithm for multi-agent trajectory planning.
  In *Advances in Neural Information Processing Systems*, pages 521–529, 2013.
- M. Kraning et al.
  Dynamic network energy management via proximal message passing.
  Foundations and Trends in Optimization, 1(2):70–122, 2013.

References

- D. Mellinger et al.
  Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams.
  In *IEEE Conf. on Robotics and Automation*, pages 477–483, May 2012.
- D. Morgan et al.
  Model predictive control of swarms of spacecraft using sequential convex programming.
  In *AIAA/AAS Astrodynamics Specialist Conf.*, Aug. 2012.