Optimal Control and Optimization Methods for Multi-Robot Systems

Javier Alonso-Mora, Ketan Savla and Daniela Rus

Tutorial on Multi-robot systems @ RSS 2015

July 2015
Future: many robots performing many tasks
We aim at optimal solutions for multi-robots

- Optimal control and optimization methods

- Attractive since:
  - they provide guarantees in the optimality of the solution
  - applicable to efficiently solve a wide range of problems
    - thanks to advances in the field of constrained optimization
    - and an increase in computational power of robotic platforms
Optimization is everywhere
Overview of this talk

- We give an overview of the required tools

- We focus on four canonical problems for multi-robot systems

- We describe some of the works by the community

Disclaimers

- Focus on motion planning / control / task assignment
- Broad field – we will miss some things
- Large body of works – if you feel we are missing some important reference, please let us know, We’ll gladly add them
- Contact: jalonsom@mit.edu
- We are working on a tutorial/review
Overview

- Introduction

1. Optimal control and optimization tools
   - Optimal control & dynamic programming
   - Constrained optimization
   - Combinatorial optimization

2. Problem definition & overview of state of the art

Summary
Optimal control & dynamic programming

- Given a controlled dynamical system
  - State $x(t)$, control input $u(t)$
  - Continuous
    \[ \dot{x} = f(x, u), \quad x(0) = x^0 \]
  - Discrete
    \[ x(t + 1) = Ax(t) + Bu(t) \]
- A running cost \[ r(x(t), u(t)) \]
- Find the optimal control inputs
Optimal control & dynamic programming

- Optimal control [discrete, infinite horizon]

\[
\begin{align*}
\text{minimize} & \quad J = \sum_{t=0}^{\infty} r(x(t), u(t)) \\
\text{subject to} & \quad u(t) \in U, \ x(t) \in X, \quad t = 0, 1, \ldots \\
& \quad x(t+1) = Ax(t) + Bu(t), \quad t = 0, 1, \ldots \\
& \quad x(0) = x^0
\end{align*}
\]

- Dynamic programming solves for a value function satisfying Bellman equation

Running cost
State and control constraints
Controlled dynamical model
Initial state
Model predictive control

- Model predictive control

\[
\begin{align*}
\text{minimize} & \quad \sum_{\tau=t}^{t+T} r(x(\tau), u(\tau)) \\
\text{subject to} & \quad u(\tau) \in U, \quad x(\tau) \in X, \quad \tau = t, \ldots, t + T \\
& \quad x(\tau + 1) = Ax(\tau) + Bu(\tau), \quad \tau = t, \ldots, t + T \\
& \quad x(0) = x^0
\end{align*}
\]

- Solve for a time horizon $T$ and apply the first command, repeat at $t+1$

- Can be solved implicitly or explicitly (regions)
Constrained optimization

- For a set of variables

\[ \mathbf{x} \in \mathbf{X} \]

- Find the optimal value that minimizes

\[
\mathbf{x}^* := \arg \min_{\mathbf{x}} f(\mathbf{x})
\]

subject to

\[
\begin{align*}
g_i(\mathbf{x}) & \leq 0 & \forall i & \in \{1, \ldots, n_{ineq}\} \\
h_i(\mathbf{x}) &= 0 & \forall i & \in \{1, \ldots, n_{eq}\}
\end{align*}
\]

Depending on the “shape” of \( f(\mathbf{x}) \), \( g_i(\mathbf{x}) \) and \( h_i(\mathbf{x}) \) different problems are formulated.
Constrained optimization

- **Convex optimization with continuous variables**
  - Linear programming LP: $w_n x_1 + \ldots + w_n x_n$
  - Quadratic programming QP: $w_n x_1^2 + \ldots + w_n x_n^2$
  - Semi-definite programming SDP

- convex optimization methods are (roughly) always global, always fast
Constrained optimization

- for general nonconvex problems
  - *local optimization methods* are fast, but need not find global solution (and even when they do, cannot certify it)
  - *global optimization methods* find global solution (and certify it), but are not always fast (indeed, are often slow)

Prof. S. Boyd, EE364b, Stanford University
Constrained optimization

- **Non-convex** optimization with **continuous** variables \( \mathbf{x} \in \mathbb{R}^v \)
  - Search techniques [global]
  - Gradient-based methods [local]
  - Sequential convex programming **SCP** [local]
Constrained optimization

- **Non-convex** optimization with **continuous** variables $x \in \mathbb{R}^v$
  - Sequential convex programming **SCP** [local]

- A local optimization method for nonconvex problems that leverages convex optimization
  - Convex portions of a problem are handled ‘exactly’ and efficiently
Constrained optimization

- **Non-convex optimization with continuous variables** $x \in \mathbb{R}^v$
  - Sequential convex programming **SCP** [local]

- a local optimization method for nonconvex problems that leverages convex optimization
  - convex portions of a problem are handled ‘exactly’ and efficiently

- **SCP** is a **heuristic**
  - it can fail to find optimal (or even feasible) point
  - results can (and often do) depend on starting point
  (can run algorithm from many initial points and take best result)

- **SCP** often works well, *i.e.*, finds a feasible point with good, if not optimal, objective value
Constrained optimization

- **Optimization with integer variables**
  - Integer linear program as network flow
  - Mixed integer program MIP [global]

- **Combinatorial optimization**
  - Traveling salesman problem TSP
    - small problems solved via MIP, large problems solved with heuristics

\[
x_j \in \mathbb{N}, \quad x_j \in \{0, 1\}
\]

**Branch-and-Bound**

Each node in branch-and-bound is a new MIP

EFFICIENT - GLOBAL OPTIMUM

INEFFICIENT - GLOBAL OPTIMUM
Constrained optimization: overview

- Convex optimization with continuous variables
  - LP / QP / SDP

- Non-convex optimization with continuous variables
  - Gradient-based methods [local]
  - Sequential convex programming SCP [local]

- Optimization with integer variables
  - Mixed integer program MIP [global]
  - Integer linear program as network flow
  - Combinatorial optimization
Overview

- Introduction
- 1. Optimal control and optimization tools

2. Problem definition & state of the art
   - Multi-robot motion planning
   - Formation planning
   - Task assignment
   - Surveillance and monitoring

- Summary
Multi-robot motion planning: problem definition

- Compute robot trajectories such that
  - Drive robots initial to final configuration
  - Avoid static and dynamic obstacles
  - Avoid inter-robot collisions
  - Respect dynamic model of the robot
    - Kinematic model, velocity/acceleration limits….
Multirobot motion planning: problem definition

- Global planning
  - Trajectory from initial to final state

- Local planning (collision avoidance)
  - Trajectory from initial state up to a short time horizon
Multi-robot motion planning: problem definition

- Global planning
  - Trajectory from initial to final state

- Local planning (collision avoidance)
  - Trajectory from initial state up to a short time horizon
MMP: global planning

- "Traditional" approaches
  - Assign priorities and sequentially compute trajectories
MMP: global planning

- “Traditional” approaches
  - Assign priorities and sequentially compute trajectories
  - Compute robot paths and adjust velocity profiles
MMP: global planning

- "Traditional" approaches
  - Assign priorities and sequentially compute trajectories
  - Compute robot paths and adjust velocity profiles

- Optimization-based trajectory generation (examples)
  - "Near"-optimal approaches
    - Continuous space: Mixed Integer Program [Mellinger et al, 2012]
    - Discrete graph: Integer Linear Program [Yu and Rus, 2015]
  
  - Locally optimal approaches
    - Continuous obstacle-free: SCP [Augugliaro et al, 2012]
    - Continuous with obstacles: SCP [Chen et al, 2015]
    - Continuous 2D: Message passing [Bento et al, 2013]
MMP: centralized global planning

- Optimal trajectories, continuous, with dynamics [Mellinger et al, 2012]
- Formulated as a Mixed Integer Program
MMP: centralized global planning

- Optimal trajectories, continuous, with dynamics [Mellinger et al, 2012]
- Formulated as a Mixed Integer Program
  - Trajectory = piecewise polynomial functions over \( n_w \) time intervals using Legendre polynomial basis functions \( P_{pw}(t) \)
  - Minimize the integral of the square of the norm of the snap (the second derivative of acceleration, \( k_r = 4 \))
MMP: centralized global planning

- Optimal trajectories, continuous, with dynamics [Mellinger et al, 2012]
- Formulated as a Mixed Integer Program
  - Trajectory = piecewise polynomial functions over $n_w$ time intervals using Legendre polynomial basis functions $P_{pw}(t)$
  - Minimize the integral of the square of the norm of the snap (the second derivative of acceleration, $k_r = 4$)
  - Integer constraints for obstacle avoidance
    - At least one of the linear constraints defined by the faces of the obstacle separates the obstacle from the robot volume

$$n_{of} \cdot r_T(t_k) \leq s_{of} + M b_{ofk} \quad \forall f = 1, \ldots, n_f(o)$$
$$b_{ofk} = 0 \text{ or } 1 \quad \forall f = 1, \ldots, n_f(o)$$
$$\sum_{f=1}^{n_f(o)} b_{ofk} \leq n_f(o) - 1$$

- Optimal, but computationally expensive
MMP: centralized global planning

- Near-optimal planning on a discrete graph [Yu and Rus, 15]
- Formulated as an Integer Linear Program (efficient)
MMP: centralized global planning

- Near-optimal planning on a discrete graph [Yu and Rus, 15]
- Formulated as an Integer Linear Program (efficient)
MMP: centralized global planning

- Locally optimal, continuous, 2D, holonomic, parallelizable
- ADMM – 3 weight message passing [J. Bento et al, 2013]
MMP: centralized global planning

- Locally optimal trajectories in free space, with dynamics
- Sequential convex programming (efficient) [Augugliaro et al, 2012]
  - The optimization variable $\chi \in \mathbb{R}^{3NK}$ consists of the vehicles’ accelerations at each time step $k$
  - The optimality criterion is the sum of the total thrust at each time step
  - Convex constraints: physical properties of vehicles’
  - Non-convex constraints: collision avoidance:

$$\|p_i[k] - p_j[k]\|_2 \geq R, \quad \forall i, j, \quad i \neq j, \quad \forall k$$

- Linearized around the current solution results in QP:

$$\text{minimize} \quad \chi^T P \chi + q^T \chi + r$$
$$\text{subject to} \quad A_{eq} \chi = b_{eq}$$
$$A_{in} \chi \preceq b_{in},$$
MMP: centralized global planning

- Locally optimal trajectories in free space, with dynamics
- Sequential convex programming (efficient) [Augugliaro et al, 2012]
MMP: centralized global planning

- Locally optimal trajectories, with dynamics
- Sequential convex programming (efficient) [Chen et al, 2015]
Multi-robot motion planning: problem definition

- Global planning
  - Trajectory from initial to final state

- Local planning (collision avoidance)
  - Trajectory from initial state up to a short time horizon
MMP: collision avoidance

- Velocity obstacles with motion constraints [Alonso-Mora et al. 2010]
  - Set of motion primitives towards linear trajectories (reference velocity)
  - Collision avoidance constraints in reference velocity space
MMP: collision avoidance

- Velocity obstacles with motion constraints [Alonso-Mora et al. 2010]
  - Set of motion primitives towards linear trajectories (reference velocity)
  - Collision avoidance constraints in reference velocity space

\[ \| (p_i + u_i t) - (p_j + u_j t) \| > r_i + r_j \]

\[ \forall t \in [0, \tau] \]
MMP: collision avoidance

- Velocity obstacles with motion constraints  [Alonso-Mora et al. 2010]
  - Set of motion primitives towards linear trajectories (reference velocity)
  - Collision avoidance constraints in reference velocity space

\[ \| (p_i + u_i t) - (p_j + u_j t) \| > r_i + r_j \]
\[ \forall t \in [0, \tau] \]
\[ \frac{p_i - p_j}{t} + (u_i - u_j) \| > \frac{r_i + r_j}{t} \]

Distributed with assumption on \( u_j \)

- Static: \( u_j = 0 \)
- Constant velocity: \( u_j = v_j \)
- Both decision-making:
  - Collaborative
  \[ \Delta v_i = \lambda \Delta v_{ij} \]
MMP: collision avoidance

- Velocity obstacles with motion constraints  [Alonso-Mora et al. 2010]
  - Set of motion primitives towards linear trajectories (reference velocity)
  - Collision avoidance constraints in reference velocity space

\[
\| (p_i + u_i t) - (p_j + u_j t) \| > r_i + r_j
\]
\[
\forall t \in [0, \tau]
\]
\[
\| \frac{p_i - p_j}{t} + (u_i - u_j) \| > \frac{r_i + r_j}{t}
\]

Distributed with assumption on \( u_j \)

- Static: \( u_j = 0 \)
- Constant velocity: \( u_j = v_j \)
- Both decision-making:
  - Collaborative
    \[
    \Delta v_i = \lambda \Delta v_{ij}
    \]

- This gives a distributed convex optimization with linear constraints
MMP: collision avoidance

- Optimal control [Hoffmann and Tomlin 2008]

- Model predictive control [Shim, Kim and Sastry 2003]

- Convex optimization in velocity space [van den Berg et al. 2009]
  - Extension to account for robot dynamics [Alonso-Mora et al. 2010]
  - Also applied to aerial vehicles [Alonso-Mora et al. 2015]
Formation control/planning: problem definition

- Maintain desired inter-robot distances defining the formation
Formation control

- Obstacle-free environments
  - Centralized optimal coverage with assignment [Alonso-Mora et al. 2012]
  - Leader follower with optimal control [Ji, Muhammad and Egerstedt 2006]
  - Distributed QP with leader follower [Turpin, Michael and Kumar 2012]
  - Model Predictive Control [Dunbar and Murray 2002]
  - Distributed consensus [Montijano and Mosteo 2014]
Formation planning: with obstacles

- Convex optimization
  - SDP, circular formation, triangulate space [Derenick and Spletzer 2007]
  - SDP for circular obstacles [Derenick, Spletzer and Kumar 2010]
  - Centralized LP in velocity space [Karamouzas and Guy 2015]
  - Distributed QP in velocity space [Alonso-Mora et al. 2015]
    - Constraints: Avoidance + min/max inter-robot distance
Formation planning: with obstacles

- Distributed convex optimization [Alonso-Mora et al. 2015]
  - Compute a new velocity
    minimize (deviation to target global motion of the object)
    s.t. Collision avoidance constraints [velocity obstacles]
    Shape maintenance constraints: min / max distance

Force sensing used to indicate intention and to coordinate
Constraints convexified & partitioned assuming cooperation
Formation planning

- Distributed convex optimization [Alonso-Mora et al. 2015]
Formation planning

- **Convex optimization**
  - SOP, circular formation, triangulate space [Derenick and Spletzer 2007]
  - SDP for circular obstacles [Derenick, Spletzer and Kumar 2010]
  - Centralized LP in velocity space [Karamouzas and Guy 2015]
  - Distributed QP in velocity space [Alonso-Mora et al. 2015]

- **Non-convex optimization**
  - Off-line global MIP for sub-groups [Kushleyev, Mellinger and Kumar 2012]
  - On-line local sequential convex programming [Alonso-Mora et al. 2015]
Formation planning

- Centralized off-line MIP subgroups [Kushleyev, Mellinger and Kumar 2012]

MIQP trajectory planning for subgroups of fixed formation

Distributed formation control within the subgroup

[Mellinger, Kushleyev, Kumar, 2012]

[Turpin, Michel and Kumar 2011]
Formation planning

- Centralized off-line MIP subgroups [Kushleyev, Mellinger and Kumar 2012]
Formation planning

- Centralized local real-time SCP [Alonso-Mora et al. 2015]

Goal – Target motion

Compute largest convex obstacle-free area
Iterative QP + SDP

Compute optimal formation parameters via a nonlinear constrained optimization SCP

Optimal robot commands

\[
x^*_i = w_t ||t - g(t_f)||^2 + w_s ||s - \bar{s}||^2 + w_q ||q - \bar{q}||^2 + c_i
\]

\[
\begin{align*}
\text{s.t. } & C_1^i = \{A(t + s \text{ rot}(q, f_{0,i})) \leq b\} \\
& C_2 = \{s d^i_0 \geq 2 \max(r, h)\} \\
& C_3 = \{||q||^2 = 1\}
\end{align*}
\]
Formation planning

- Centralized local real-time SCP [Alonso-Mora et al. 2015]
Formation planning

- Centralized local real-time SCP [Alonso-Mora et al. 2015]
Take home message

- **Convex optimization with continuous variables**
  - LP/QP/SDP
    - Very fast, global optimum
    - But, most problems are not convex

- **Non-convex optimization with continuous variables**
  - Sequential convex programming SCP [local]
    - Fast but local, often works well, but no strict guarantees

- **Non-convex optimization with binary variables**
  - Mixed Integer Program MIP [global]
    - Slow but eventually will find the global optimum

\[ x \in \mathbb{R}^v \]

\[ x_j \in \{0, 1\} \]
Surveillance and monitoring: problem definition
Surveillance and monitoring: problem definition

- Consider $m$ robots at $p = \{p_1, \ldots, p_m\}$
- Environment is partitioned into $v = \{v_1, \ldots, v_m\}$
- Cost:
  \[
  \mathcal{H}(p, v) = \sum_{i=1}^{m} \int_{v_i} f(\|x-p_i\|) \varphi(x) \, dx
  \]
  \begin{itemize}
  \item $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ density
  \item $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ penalty function
  \end{itemize}
- Voronoi partition $\{V_1, \ldots, V_m\}$ generated by points $\{p_1, \ldots, p_m\}$

\[
V_i(p) = \{x \in \Omega \mid \|x-p_i\| \leq \|x-p_j\|, \forall j \neq i\}
\]
Surveillance and monitoring: problem definition

\[
\mathcal{H}(p, v) = \sum_{i=1}^{m} \int_{v_i} f(\|x - p_i\|) \varphi(x) \, dx
\]

**Theorem (Alternating Algorithm, Lloyd '57)**

1. at fixed positions, optimal partition is Voronoi
2. at fixed partition, optimal positions are "generalized centers"
3. alternate \(v-p\) optimization

\[\longrightarrow \text{local optimum} = \text{center Voronoi partition}\]
Surveillance and monitoring

- Spatial distribution known
  - Gradient descent – alternating algorithm [Lloyd 1982]

- Spatial distribution unknown
  - Adaptive algorithms [Schwager, Rus and Slotine 2009]
  - Motion constraints [Savla and Frazzoli 2010]
  - Persistent surveillance [Smith et al. 2011]
  - Adapting to sensing/actuation [Pierson et al. 2015]
Multi-robot coverage

- Adapting to sensing/actuation [Pierson et al. 2015]

In the following experiment, robot 2 (red) has a lower sensor health. Its Voronoi cell will shrink over time to compensate.
Task assignment: problem definition

- Taxonomy [Gerkey 2004]
Task assignment: Single task per robot

3. THE GENERAL ASSIGNMENT PROBLEM

Suppose \( n \) individuals \((i = 1, \ldots, n)\) are available for \( n \) jobs \((j = 1, \ldots, n)\) and that a rating matrix \( R = (r_{ij}) \) is given, where the \( r_{ij} \) are positive integers, for all \( i \) and \( j \). An assignment consists of the choice of one job \( j_i \) for each individual \( i \) such that no job is assigned to two different men. Thus, all of the jobs are assigned and an assignment is a permutation

\[
\begin{pmatrix}
1 & 2 & \ldots & n \\
\hat{j}_1 & \hat{j}_2 & \ldots & \hat{j}_n
\end{pmatrix}
\]

of the integers 1, 2, \ldots, \( n \). The General Assignment Problem asks:

For which assignments is the sum

\[ r_{1\hat{j}_1} + r_{2\hat{j}_2} + \ldots + r_{n\hat{j}_n} \]

of the ratings largest?

- Optimal [Kuhn 1955]
- Suboptimal: auction [Bertsekas 1992]
- Concurrent assignment and planning [Turpin, Michael, Kumar 2014]
Task assignment: vehicle routing

- A potentially large number of tasks to be satisfied by a set of robots

- Static vehicle routing [Toth and Vigo 2001]
  - Traveling salesman problem
    - Small problems can be solved via a MIP
    - Large problems are typically solved with heuristics (tabu search)

- Dynamic vehicle routing [Bertsimas and van Ryzin 1991]
  - Introduced queuing theory (Arrival process: spatio-temporal Poisson)
Task assignment: vehicle routing

- A potentially large number of tasks are to be satisfied by a set of robots

- Static vehicle routing [Toth and Vigo, 2001]
  - Traveling salesman problem
    - Large problems are typically solved with heuristics (tabu search)

- Dynamic vehicle routing [Bertsimas and van Ryzin, 1991]
  - Introduced queuing theory
  - Motivated many extensions
    - time constraints [Pavone et al, 2009]
    - service priorities [Smith et al, 2009]
    - adaptive and decentralized algorithms [Arsie et al, 2009]
    - complex vehicle dynamics [Savla et al. 2008]
    - limited sensing range [Enright and Frazzoli, 2006]
    - mobility on demand and rebalancing [Smith et al, 2013]
An optimal spatially-unbiased heavy-load policy

- Voronoi partition + single robot TSP [Frazzoli and Bullo, CDC04]
Combination of optimization methods

- Animation display with multiple robots [Alonso-Mora et al. 2012]
- Optimal coverage, goal assignment and collision avoidance
Combination of optimization methods

- Animation display with multiple robots [Alonso-Mora et al. 2012]
- Optimal coverage, goal assignment and collision avoidance
Overview

- Introduction
- 1. Optimal control and optimization tools
- 2. Problem definition & overview of state of the art

Summary
Optimal control / optimization techniques can play an important role in the design and operation of multi-robot systems.

We provided an overview of these techniques in the context of four major classes of multi-robot problems: multi-robot motion planning, formation planning, task assignment, and surveillance & monitoring.

Optimization methods can also be found in other areas, such as cooperative localization and mapping.
Questions?

- Optimal control / optimization techniques can play an important role in the design and operation of multi-robot systems.
- Please send me more refs. and we will add them!
- Contact: J. Alonso-Mora: jalonsom@mit.edu

Acknowledgements:
- Ketan Salva - joint work for this presentation/paper
- P. Beardsley, F. Bullo, E. Frazzoli, D. Rus, R. Siegwart