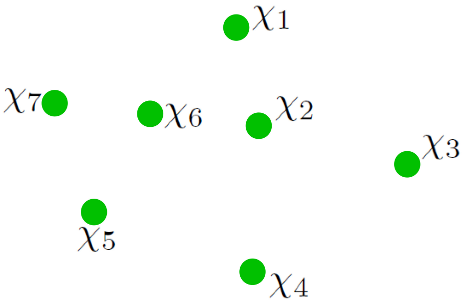
ALADIN—An Algorithm for Distributed Non-Convex Optimization and Control

Boris Houska, Yuning Jiang, Janick Frasch,

Rien Quirynen, Dimitris Kouzoupis, Moritz Diehl

ShanghaiTech University, University of Magdeburg, University of Freiburg

Motivation: sensor network localization

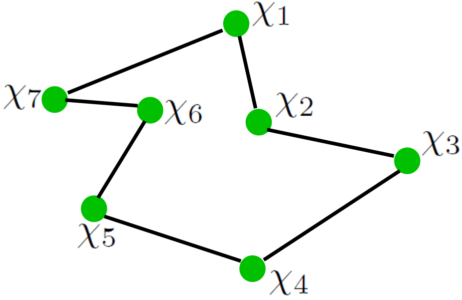


Decoupled case:

- each sensor takes measurement η_i of its position χ_i and solves

$$\forall i \in \{1, \ldots, 7\}, \qquad \min_{\chi_i} \|\chi_i - \eta_i\|_2^2 \ .$$

Motivation: sensor network localization

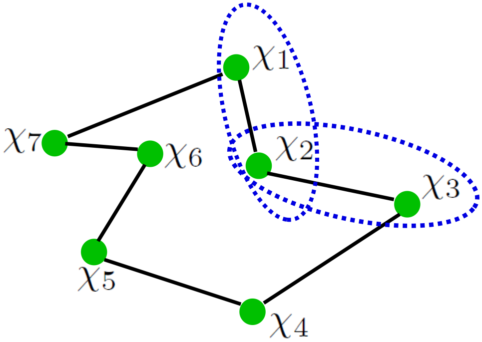


Coupled case:

- sensors additionally measure the distance to their neighbors

$$\min_{\chi} \sum_{i=1}^{7} \left\{ \|\chi_i - \eta_i\|_2^2 + \left(\|\chi_i - \chi_{i+1}\|_2 - \bar{\eta}_i\right)^2 \right\} \qquad \text{with } \chi_8 = \chi_1$$

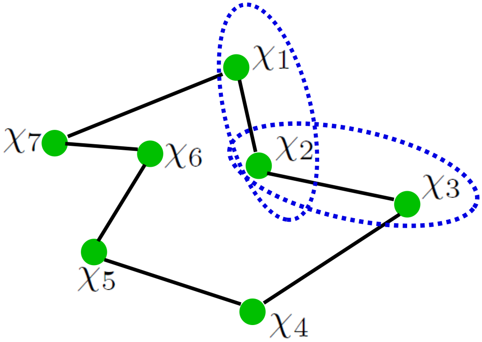
Motivation: sensor network localization



Equivalent formulation:

- set $x_1 = (\chi_1, \zeta_1)$ with $\zeta_1 = \chi_2$,
- set $x_2 = (\chi_2, \zeta_2)$ with $\zeta_2 = \chi_3$,
- and so on

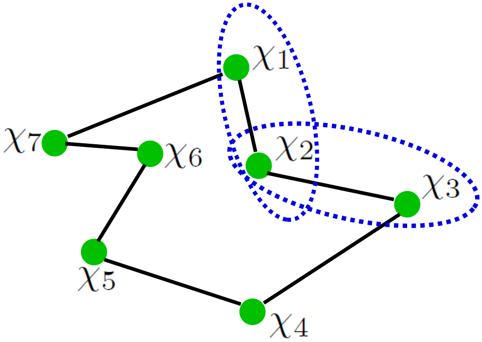
Motivation: sensor network localization



Equivalent formulation:

- set $x_1 = (\chi_1, \zeta_1)$ with $\zeta_1 = \chi_2$,
- set $x_2 = (\chi_2, \zeta_2)$ with $\zeta_2 = \chi_3$,
- and so on

Motivation: sensor network localization



Equivalent formulation:

- set $x_1 = (\chi_1, \zeta_1)$ with $\zeta_1 = \chi_2$,
- set $x_2 = (\chi_2, \zeta_2)$ with $\zeta_2 = \chi_3$,
- and so on

Motivation: sensor network localization

Equivalent formulation (cont.):

- new variables $x_i = (\chi_i, \zeta_i)$
- separable non-convex objectives

$$f_i(x_i) = \frac{1}{2}\|\chi_i - \eta_i\|_2^2 + \frac{1}{2}\|\zeta_i - \eta_{i+1}\|_2^2 + \frac{1}{2}\left(\|\chi_i - \zeta_i\|_2 - \bar{\eta}_i\right)^2$$

- affine coupling, $\zeta_i = \chi_{i+1}$, can be written as

$$\sum_{i=1}^{7} A_i x_i = 0 \; .$$

Motivation: sensor network localization

Equivalent formulation (cont.):

- new variables $x_i = (\chi_i, \zeta_i)$
- separable non-convex objectives

$$f_i(x_i) = \frac{1}{2}\|\chi_i - \eta_i\|_2^2 + \frac{1}{2}\|\zeta_i - \eta_{i+1}\|_2^2 + \frac{1}{2}\left(\|\chi_i - \zeta_i\|_2 - \bar{\eta}_i\right)^2$$

- affine coupling, $\zeta_i = \chi_{i+1}$, can be written as

$$\sum_{i=1}^{7} A_i x_i = 0 \; .$$

Motivation: sensor network localization

Equivalent formulation (cont.):

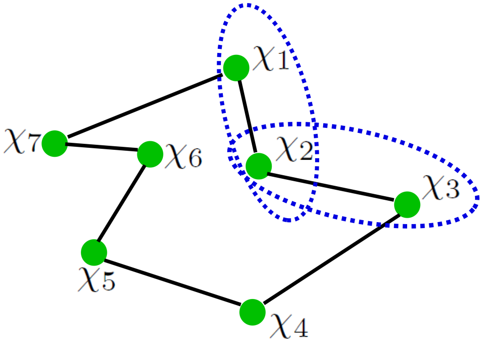
- new variables $x_i = (\chi_i, \zeta_i)$
- separable non-convex objectives

$$f_i(x_i) = \frac{1}{2}\|\chi_i - \eta_i\|_2^2 + \frac{1}{2}\|\zeta_i - \eta_{i+1}\|_2^2 + \frac{1}{2}\left(\|\chi_i - \zeta_i\|_2 - \bar{\eta}_i\right)^2$$

- affine coupling, $\zeta_i = \chi_{i+1}$, can be written as

$$\sum_{i=1}^{7} A_i x_i = 0 \ .$$

Motivation: sensor network localization



Optimization problem:

$$\min_x \sum_{i=1}^{7} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{7} A_i x_i = 0 \ .$$

Aim of distributed optimization algorithms

- Find local minimizers of

$$\min_{x} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

Aim of distributed optimization algorithms

- Find local minimizers of

$$\min_{x} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

Aim of distributed optimization algorithms

- Find local minimizers of

$$\min_{x} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

Aim of distributed optimization algorithms

- Find local minimizers of

$$\min_{x} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

Overview

- **Theory**
  - Distributed optimization algorithms
  - ALADIN
- **Applications**
  - Sensor network localization
  - MPC with long horizons

Distributed optimization problem

- Find local minimizers of

$$\min_x \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b \; .$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

Dual decomposition

- Main idea: solve dual problem

$$\max_{\lambda}\; d(\lambda) \quad \text{with} \quad d(\lambda) = \min_{x} \sum_{i=1}^{N} \{f_i(x_i) + \lambda^\mathsf{T} A_i x_i\} - \lambda^\mathsf{T} b$$

  Evaluation of d can be parallelized.

- Applicable if f_is are (strictly) convex
- For non-convex f_i: duality gap possible

- H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, 1963.

Dual decomposition

- Main idea: solve dual problem

$$\max_{\lambda} \; d(\lambda) \quad \text{with} \quad d(\lambda) = \sum_{i=1}^{N} \min_{x_i} \{f_i(x_i) + \lambda^\mathsf{T} A_i x_i\} - \lambda^\mathsf{T} b$$

  Evaluation of d can be parallelized.

- Applicable if f_is are (strictly) convex
- For non-convex f_i: duality gap possible

- H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, 1963.

Dual decomposition

- Main idea: solve dual problem

$$\max_{\lambda} \; d(\lambda) \quad \text{with} \quad d(\lambda) = \sum_{i=1}^{N} \min_{x_i} \{f_i(x_i) + \lambda^\mathsf{T} A_i x_i\} - \lambda^\mathsf{T} b$$

  Evaluation of d can be parallelized.

- Applicable if f_is are (strictly) convex
- For non-convex f_i: duality gap possible

- H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, 1963.

Dual decomposition

- Main idea: solve dual problem

$$\max_{\lambda} \; d(\lambda) \quad \text{with} \quad d(\lambda) = \sum_{i=1}^{N} \min_{x_i} \{f_i(x_i) + \lambda^\mathsf{T} A_i x_i\} - \lambda^\mathsf{T} b$$

  Evaluation of d can be parallelized.

- Applicable if f_is are (strictly) convex
- For non-convex f_i: duality gap possible

- H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, 1963.

ADMM (consensus variant)

Alternating Direction Method of Multipliers

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$; $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda_i^\top A_i y_i + \frac{\rho}{2} \left\| A_i(y_i - x_i) \right\|_2^2 \; .$$

2. Implement dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.

3. Solve coupled QP

$$\min_{x^+} \; \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| A_i(y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\top A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^N A_i x_i^+ = b \; .$$

4. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$.

D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations, 1976.

ADMM (consensus variant)

Alternating Direction Method of Multipliers

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$; $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda_i^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| A_i(y_i - x_i) \right\|_2^2 \; .$$

2. Implement dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.

3. Solve coupled QP

$$\min_{x^+} \; \sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| A_i(y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\mathsf{T} A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i^+ \; = \; b \; .$$

4. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$.

- D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations, 1976.

ADMM (consensus variant)

Alternating Direction Method of Multipliers

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$; $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda_i^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| A_i(y_i - x_i) \right\|_2^2 \; .$$

2. Implement dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.

3. Solve coupled QP

$$\min_{x^+} \; \sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| A_i(y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\mathsf{T} A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i^+ \; = \; b \; .$$

4. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$.

- D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations, 1976.

ADMM (consensus variant)

Alternating Direction Method of Multipliers

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$; $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda_i^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| A_i (y_i - x_i) \right\|_2^2 \; .$$

2. Implement dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i (y_i - x_i)$.

3. Solve coupled QP

$$\min_{x^+} \; \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| A_i (y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\mathsf{T} A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^N A_i x_i^+ \; = \; b \; .$$

4. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$.

- D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations, 1976.

ADMM (consensus variant)

Alternating Direction Method of Multipliers

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$; $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \ f_i(y_i) + \lambda_i^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| A_i(y_i - x_i) \right\|_2^2 \ .$$

2. Implement dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.

3. Solve coupled QP

$$\min_{x^+} \ \sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| A_i(y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\mathsf{T} A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i^+ \ = \ b \ .$$

4. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$.

- D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations, 1976.

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ = -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ \; = \; -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ \; = \; -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ \;=\; -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ \; = \; -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Limitations of ADMM

1) Convergence rate of ADMM is very scaling dependent.

2) ADMM may be divergent, if f_is are nonconvex. Example:

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

- unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$.
- For $\rho = \frac{3}{4}$ all sub-problems are strictly convex.
- ADMM is divergent; $\lambda^+ \; = \; -2\lambda$.

- **This talk: addresses Problem 2), mitigates Problem 1)**

Overview

- **Theory**
  - Distributed optimization algorithms
  - ALADIN
- **Applications**
  - Sensor network localization
  - MPC with long horizons

ALADIN (full step variant)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \|y_i - x_i\|^2_{\Sigma_i} \; .$$

2. Compute $g_i = \nabla f_i(y_i)$, choose $H_i \approx \nabla^2 f_i(y_i)$, and solve coupled QP

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \;\mid\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (full step variant)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \; .$$

2. Compute $g_i = \nabla f_i(y_i)$, choose $H_i \approx \nabla^2 f_i(y_i)$, and solve coupled QP

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \; \mid \; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (full step variant)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \; .$$

2. Compute $g_i = \nabla f_i(y_i)$, choose $H_i \approx \nabla^2 f_i(y_i)$, and solve coupled QP

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \;\; \big| \;\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (full step variant)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \; .$$

2. Compute $g_i = \nabla f_i(y_i)$, choose $H_i \approx \nabla^2 f_i(y_i)$, and solve coupled QP

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \;\mid\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

Special cases

- For $\rho \to \infty$:

$$\text{ALADIN} \;\equiv\; \text{SQP}$$

- For $0 < \rho < \infty$:

If $H_i = \rho A_i^\top A_i$, $\Sigma_i = A_i^\top A_i$, then

$$\text{ALADIN} \;\equiv\; \text{ADMM}$$

- For $\rho = 0$:

$$\text{ALADIN} \;\equiv\; \text{Dual Decomposition } (+ \text{ Inexact Newton})$$

Special cases

- For $\rho \to \infty$:

$$\text{ALADIN} \;\equiv\; \text{SQP}$$

- For $0 < \rho < \infty$:

If $H_i = \rho A_i^\mathsf{T} A_i$, $\Sigma_i = A_i^\mathsf{T} A_i$, then

$$\text{ALADIN} \;\equiv\; \text{ADMM}$$

- For $\rho = 0$:

$$\text{ALADIN} \;\equiv\; \text{Dual Decomposition } (+ \text{ Inexact Newton})$$

Special cases

- For $\rho \to \infty$:

$$\text{ALADIN} \;\equiv\; \text{SQP}$$

- For $0 < \rho < \infty$:

If $H_i = \rho A_i^\mathsf{T} A_i$, $\Sigma_i = A_i^\mathsf{T} A_i$, then

$$\text{ALADIN} \;\equiv\; \text{ADMM}$$

- For $\rho = 0$:

$$\text{ALADIN} \;\equiv\; \text{Dual Decomposition } (+ \text{ Inexact Newton})$$

Local convergence

Assumptions

- f_is are twice continuously differentiable
- minimizer (x^*, λ^*) regular KKT point (LICQ + SOSC satisfied)
- ρ satisfies $\nabla^2 f_i(y_i) + \rho\Sigma_i \succ 0$

Theorem Full-step variant of ALADIN converges locally

1. with quadratic convergence rate, if $H_i = \nabla^2 f_i(y_i) + \mathbf{O}(\|y_i - x^*\|)$
2. with linear converges rate, if $\|H_i - \nabla^2 f_i(y_i)\|$ is sufficiently small

Local convergence

Assumptions

- f_is are twice continuously differentiable
- minimizer (x^*, λ^*) regular KKT point (LICQ + SOSC satisfied)
- ρ satisfies $\nabla^2 f_i(y_i) + \rho\Sigma_i \succ 0$

Theorem Full-step variant of ALADIN converges locally

1. with quadratic convergence rate, if $H_i = \nabla^2 f_i(y_i) + \mathbf{O}(\|y_i - x^*\|)$
2. with linear converges rate, if $\|H_i - \nabla^2 f_i(y_i)\|$ is sufficiently small

Globalization

Definition (L1-penalty function)

- We say that x^+ is a descent step if $\Phi(x^+) < \Phi(x)$ for

$$\Phi(x) \; = \; \sum_{i=1}^{N} f_i(x_i) + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i x_i - b \right\|_1 \; ,$$

  $\overline{\lambda}$ sufficiently large.

Globalization

Rough sketch:

- As long as $\nabla^2 f_i(x_i) + \rho\Sigma_i \succ 0$ the proximal objectives

$$\tilde{f}_i(y_i) = f_i(y_i) + \frac{\rho}{2} \|y_i - x_i\|_{\Sigma_i}^2$$

  are strictly convex in a neighborhood of the current primal iterates x_i.

- If we don't update x, y can be enforced to converge to the solution z of the convex auxiliary problem

$$\min_z \sum_{i=1}^{N} \tilde{f}_i(z_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i z_i = b\,.$$

- Strategy: if x^+ is not a descent direction, we skip the primal update until y is a descent and set $x^+ = y$ (similar to proximal methods)
- This strategy leads to a globalization routine for ALADIN if $H_i \succ 0$.

Globalization

Rough sketch:

- As long as $\nabla^2 f_i(x_i) + \rho\Sigma_i \succ 0$ the proximal objectives

$$\tilde{f}_i(y_i) = f_i(y_i) + \frac{\rho}{2}\left\|y_i - x_i\right\|_{\Sigma_i}^2$$

  are strictly convex in a neighborhood of the current primal iterates x_i.

- If we don't update x, y can be enforced to converge to the solution z of the convex auxiliary problem

$$\min_z \sum_{i=1}^{N} \tilde{f}_i(z_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i z_i = b \; .$$

- Strategy: if x^+ is not a descent direction, we skip the primal update until y is a descent and set $x^+ = y$ (similar to proximal methods)

- This strategy leads to a globalization routine for ALADIN if $H_i \succ 0$.

Globalization

Rough sketch:

- As long as $\nabla^2 f_i(x_i) + \rho\Sigma_i \succ 0$ the proximal objectives

$$\tilde{f}_i(y_i) = f_i(y_i) + \frac{\rho}{2}\left\|y_i - x_i\right\|_{\Sigma_i}^2$$

  are strictly convex in a neighborhood of the current primal iterates x_i.

- If we don't update x, y can be enforced to converge to the solution z of the convex auxiliary problem

$$\min_z \; \sum_{i=1}^{N} \tilde{f}_i(z_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i z_i = b \; .$$

- Strategy: if x^+ is not a descent direction, we skip the primal update until y is a descent and set $x^+ = y$ (similar to proximal methods)

- This strategy leads to a globalization routine for ALADIN if $H_i \succ 0$.

Globalization

Rough sketch:

- As long as $\nabla^2 f_i(x_i) + \rho\Sigma_i \succ 0$ the proximal objectives

$$\tilde{f}_i(y_i) = f_i(y_i) + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2$$

  are strictly convex in a neighborhood of the current primal iterates x_i.

- If we don't update x, y can be enforced to converge to the solution z of the convex auxiliary problem

$$\min_z \; \sum_{i=1}^N \tilde{f}_i(z_i) \quad \text{s.t.} \quad \sum_{i=1}^N A_i z_i = b \; .$$

- Strategy: if x^+ is not a descent direction, we skip the primal update until y is a descent and set $x^+ = y$ (similar to proximal methods)
- This strategy leads to a globalization routine for ALADIN if $H_i \succ 0$.

Inequalities

- Distributed NLP with inequalities:

$$\min_{x} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^{N} A_i x_i = b \\ h_i(x_i) \leq 0 \ . \end{array} \right.$$

- Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}^{n_h}$ potentially non-convex.
- Matrices $A_i \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$ given.
- Problem: N is large.

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \| y_i - x_i \|^2_{\Sigma_i} \quad \text{s.t.} \quad h_i(y_i) \leq 0 \;\mid\; \kappa_i \,.$$

2. Set $g_i = \nabla f_i(y_i) + \nabla h_i(y_i)\kappa_i$, $H_i \approx \nabla^2 \left(f_i(y_i) + \kappa_i^\mathsf{T} h_i(y_i) \right)$, solve

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \;\mid\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \|y_i - x_i\|_{\Sigma_i}^2 \quad \text{s.t.} \quad h_i(y_i) \le 0 \mid \kappa_i \, .$$

2. Set $g_i = \nabla f_i(y_i) + \nabla h_i(y_i)\kappa_i$, $H_i \approx \nabla^2 \left(f_i(y_i) + \kappa_i^\mathsf{T} h_i(y_i) \right)$, solve

$$\min_{\Delta y} \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \mid \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \|y_i - x_i\|_{\Sigma_i}^2 \quad \text{s.t.} \quad h_i(y_i) \leq 0 \;\mid\; \kappa_i \;.$$

2. Set $g_i = \nabla f_i(y_i) + \nabla h_i(y_i)\kappa_i$, $H_i \approx \nabla^2 \left(f_i(y_i) + \kappa_i^\mathsf{T} h_i(y_i) \right)$, solve

$$\min_{\Delta y} \; \sum_{i=1}^N \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^N A_i(y_i + \Delta y_i) = b \;\mid\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input:

- Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, $\rho > 0$, $\epsilon > 0$.

Repeat:

1. Solve decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \quad \text{s.t.} \quad h_i(y_i) \leq 0 \;\mid\; \kappa_i \; .$$

2. Set $g_i = \nabla f_i(y_i) + \nabla h_i(y_i)\kappa_i$, $H_i \approx \nabla^2 \left(f_i(y_i) + \kappa_i^\mathsf{T} h_i(y_i) \right)$, solve

$$\min_{\Delta y} \; \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i (y_i + \Delta y_i) = b \;\mid\; \lambda^+$$

3. Set $x \leftarrow y + \Delta y$ and $\lambda \leftarrow \lambda^+$.

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Remarks:

- If approximation $C_i \approx C_i^* = \nabla h_i(y_i)$ is available, solve QP

$$\min_{\Delta y, s} \quad \sum_{i=1}^{N} \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} + \lambda^\mathsf{T} s + \frac{\mu}{2} \|s\|_2^2$$

$$\text{s.t.} \qquad \left\{ \begin{array}{rcl} \sum_{i=1}^{N} A_i (y_i + \Delta y_i) & = & b + s \quad \Big| \; \lambda_{\mathrm{QP}} \\ C_i \Delta y_i & = & 0 \,, \quad i \in \{1, \ldots, N\} \,. \end{array} \right.$$

  with $g_i = \nabla f_i(y_i) + (C_i^* - C_i)^\mathsf{T} \kappa_i$ and $\mu > 0$ instead.

- If H_i and C_i constant, pre-compute matrix decompositions

ALADIN (with inequalities)

Augmented Lagrangian based Alternating Direction Inexact Newton Method

Remarks:

- If approximation $C_i \approx C_i^* = \nabla h_i(y_i)$ is available, solve QP

$$
\begin{aligned}
\min_{\Delta y, s} \quad & \sum_{i=1}^{N} \left\{ \tfrac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} + \lambda^\mathsf{T} s + \tfrac{\mu}{2} \left\| s \right\|_2^2 \\
\text{s.t.} \quad & \left\{ \begin{array}{rcl} \sum_{i=1}^{N} A_i \left(y_i + \Delta y_i \right) & = & b + s \quad \Big| \ \lambda_{\mathrm{QP}} \\ C_i \Delta y_i & = & 0 \,, \quad i \in \{1, \ldots, N\} \,. \end{array} \right.
\end{aligned}
$$

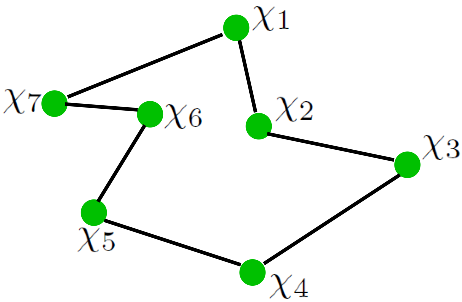
  with $g_i = \nabla f_i(y_i) + (C_i^* - C_i)^\mathsf{T} \kappa_i$ and $\mu > 0$ instead.

- If H_i and C_i constant, pre-compute matrix decompositions

Overview

- **Theory**
  - Distributed Optimization Algorithms
  - ALADIN
- **Applications**
  - Sensor network localization
  - MPC with long horizons

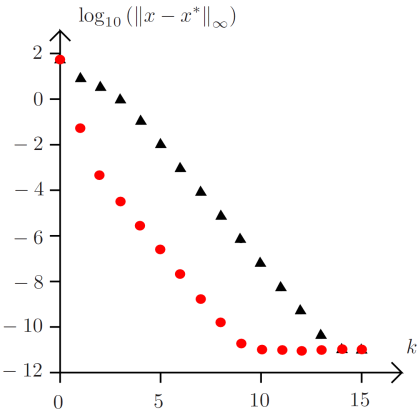
Sensor network localization



Case study:

- 25000 sensors measure positions and distances in a graph
- additional inequality constraints, $\|\chi_i - \eta_i\|_2 \leq \bar{\sigma}$, remove outliers.

ALADIN versus SQP



- 10^5 primal and $7.5 * 10^4$ dual optimization variables
- implementation in `JULIA`

Overview

- **Theory**
  - Distributed Optimization Algorithms
  - ALADIN
- **Applications**
  - Sensor network localization
  - MPC with long horizons

Nonlinear MPC

Repeat:

- Wait for state measurement $\hat{x}$.
- Solve

$$\min_{x,u} \quad \sum_{i=0}^{m-1} l(x_i, u_i) + M(x_m)$$

$$\text{s.t.} \quad \begin{cases} x_{i+1} = f(x_i, u_i) \\ x_0 = \hat{x} \\ (x_i, u_i) \in \mathbb{X} \times \mathbb{U}\ , \end{cases}$$

- Send u_0 to the process.

ACADO Toolkit

Mathematical Formulation

$$
\begin{aligned}
\min_{x,u} \quad & \int_0^T x^2 + u^2 \mathrm{d}t \\
& \dot{x} = f(x,u) \\
\text{s.t.} \quad & x(0) = x_0 \\
& -1 \leq u \leq 1\ .
\end{aligned}
$$

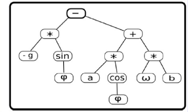
ACADO Syntax

```
DifferentialState      x;
Control                u;

DifferentialEquation f;
f << dot(x) == u + ...;

ocp.minLagrangeTerm( x*x+u*u );
ocp.subjectTo( f );
ocp.subjectTo( -1 <= u <= 1 );
```

Symbolic Structure Detection



Algorithm

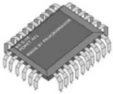
- Multiple Shooting
- Real-Time Gauss Newton
- Online Active Set Strategy

Optimized C-Code

```
r[1] = a[15]*c[17] + a[16]*c[19] + ... ;
r[2] = sin(a[1]*a[2]) + a[4] + ... ;
r[3] = cos(r[1])/exp(c[4])+ r[1] +... ;
```

Customized Solver Implemented on Chip/FPGA:

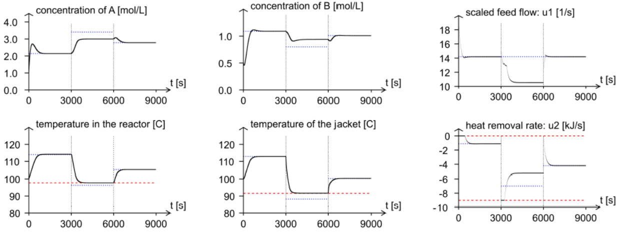
Measurement x_0



Optimal Decision u*

MPC Benchmark

Nonlinear Model, 4 States, 2 Controls

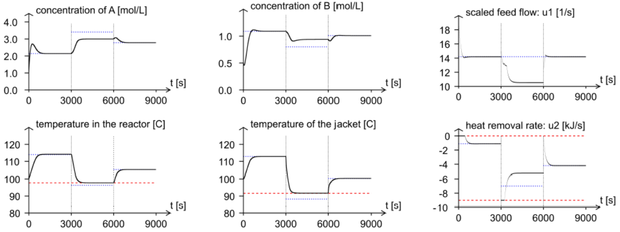


Run-time of ACADO code generation (not distributed)

	CPU time	Percentage
Integration & sensitivities	$117\ \mu$s	$65\,\%$
QP (Condensing + qpOASES)	$59\ \mu$s	$33\,\%$
Complete real-time iteration	$181\ \mu$s	$100\,\%$

MPC Benchmark

Nonlinear Model, 4 States, 2 Controls



Run-time of ACADO code generation (not distributed)

	CPU time	Percentage
Integration & sensitivities	$117\ \mu$s	$65\,\%$
QP (Condensing + qpOASES)	$59\ \mu$s	$33\,\%$
Complete real-time iteration	$181\ \mu$s	$100\,\%$

MPC with ALADIN

ALADIN Step 1: solve decoupled NLPs

- choose a short horizon $n = \frac{m}{N}$ and solve

$$\min_{y,v} \quad \Psi_j(y_0^j) + \sum_{i=0}^{n-1} l(y_i^j, v_i^j) + \Phi_{j+1}(y_n^j)$$

$$\text{s.t.} \quad \left\{ \begin{array}{l} y_{i+1}^j = f(y_i^j, v_i^j) \ , i = 0, ..., n-1 \\ y_i^j \in \mathbb{X} \ , \ v_i^j \in \mathbb{U} \ . \end{array} \right.$$

- Arrival and end costs depend on ALADIN iterates,

$$\begin{array}{rclcrcl} \Psi_0(y) & = & I(\hat{x}, y) & & \Phi_N(y) & = & M(y) \\ \Psi_j(y) & = & -\lambda_j^\mathsf{T} y + \frac{\rho}{2} \|y - z_j\|_P^2 & & \Psi_j(y) & = & \lambda_j^\mathsf{T} y + \frac{\rho}{2} \|y - z_j\|_P^2 \end{array}$$

MPC with ALADIN

ALADIN Step 1: solve decoupled NLPs

- choose a short horizon $n = \frac{m}{N}$ and solve

$$\min_{y,v} \quad \Psi_j(y_0^j) + \sum_{i=0}^{n-1} l(y_i^j, v_i^j) + \Phi_{j+1}(y_n^j)$$

$$\text{s.t.} \quad \left\{ \begin{array}{l} y_{i+1}^j = f(y_i^j, v_i^j)\ , i = 0, ..., n-1 \\ y_i^j \in \mathbb{X}\ ,\ v_i^j \in \mathbb{U}\ . \end{array} \right.$$

- Arrival and end costs depend on ALADIN iterates,

$$\begin{array}{rclrcl} \Psi_0(y) & = & I(\hat{x}, y) & \Phi_N(y) & = & M(y) \\ \Psi_j(y) & = & -\lambda_j^\mathsf{T} y + \frac{\rho}{2} \|y - z_j\|_P^2 & \Psi_j(y) & = & \lambda_j^\mathsf{T} y + \frac{\rho}{2} \|y - z_j\|_P^2 \end{array}$$

MPC with ALADIN

ALADIN Step 2: solve coupled QP

- As we have no constraints, QP $=$ LQR problem
- solve all matrix-valued Ricatti equations offline
- solve online QP online by backward-forward sweep

Code export

- export all online operations as optimized C-code
- NLP solver: explicit MPC (rough heuristic)
- one ALADIN iteration per sampling time, skip globalization

MPC with ALADIN

ALADIN Step 2: solve coupled QP

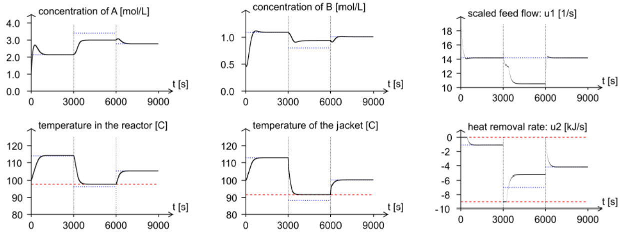
- As we have no constraints, QP $=$ LQR problem
- solve all matrix-valued Ricatti equations offline
- solve online QP online by backward-forward sweep

Code export

- export all online operations as optimized C-code
- NLP solver: explicit MPC (rough heuristic)
- one ALADIN iteration per sampling time, skip globalization

ALADIN + code generation (results by Yuning Jiang)

Same nonlinear model as before, $n = 2$

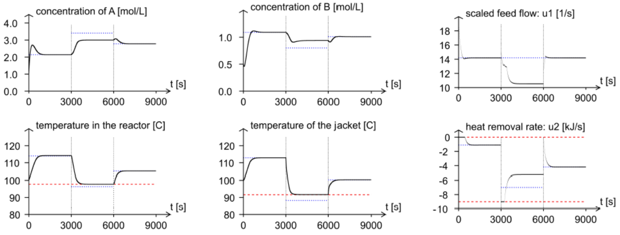


Run-time of real-time ALADIN, 10 processors

	CPU time	Percentage
Parallel explicit MPC	6 μs	54 %
QP sweeps	3 μs	27 %
communication overhead	2 μs	18 %

ALADIN + code generation (results by Yuning Jiang)

Same nonlinear model as before, $n = 2$



Run-time of real-time ALADIN, 10 processors

	CPU time	Percentage
Parallel explicit MPC	$6\ \mu$s	$54\,\%$
QP sweeps	$3\ \mu$s	$27\,\%$
communication overhead	$2\ \mu$s	$18\,\%$

Conclusions

ALADIN Theory

- can solve nonconvex distributed optimization problems,

$$\min_x \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b \ ,$$

  to local optimality.

- contains SQP, ADMM, and Dual Decomposition as special case
- local convergence analysis similar to SQP; globalization possible

Conclusions

ALADIN Theory

- can solve nonconvex distributed optimization problems,

$$\min_x \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b \ ,$$

  to local optimality.

- contains SQP, ADMM, and Dual Decomposition as special case

- local convergence analysis similar to SQP; globalization possible

Conclusions

ALADIN Theory

- can solve nonconvex distributed optimization problems,

$$\min_x \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b \ ,$$

  to local optimality.

- contains SQP, ADMM, and Dual Decomposition as special case
- local convergence analysis similar to SQP; globalization possible

Conclusions

ALADIN Applications

- large-scale distributed sensor network:

  ALADIN outperforms SQP and ADMM

- small-scale embedded MPC:

  ALADIN can be used to alternate between explicit & online MPC

Conclusions

ALADIN Applications

- large-scale distributed sensor network:

  ALADIN outperforms SQP and ADMM

- small-scale embedded MPC:

  ALADIN can be used to alternate between explicit & online MPC

References

- B. Houska, J. Frasch, M. Diehl. An Augmented Lagrangian Based Algorithm for Distributed Non-Convex Optimization. SIOPT, 2016.
- D. Kouzoupis, R. Quirynen, B. Houska, M. Diehl. A block based ALADIN scheme for highly parallelizable direct optimal control. ACC, 2016.