Model Predictive Control and Reinforcement Learning
Lecture 14: Recent Developments in Nonlinear and Robust MPC Algorithms

Joschka Boedecker and Moritz Diehl
joint work with Jonathan Frey and Florian Messerer

University Freiburg

July 30, 2021

Overview

1. Real-Time Sequential Convex Programming (SCP)
2. The Software Packages BLASFEO and acados
3. A Fast Algorithm for Closed-Loop Robustified MPC

Overview

1. Real-Time Sequential Convex Programming (SCP)
2. The Software Packages BLASFEO and acados
3. A Fast Algorithm for Closed-Loop Robustified MPC

Nonlinear MPC solves Parametric Nonlinear Programs (pNLP)

Discrete time NMPC Problem (an NLP)

$$
\begin{aligned}
\min_{s,a} \ & \textstyle\sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\
\text{s.t.} \quad & s_0 = \bar{s}_i \\
& s_{k+1} = f(s_k, a_k) \\
& 0 \geq h(s_k, a_k), \ k = 0, \ldots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

Variables $s = (s_0, \ldots)$ and $a = (a_0, \ldots, a_{N-1})$ can be summarized in vector $x = (s, a) \in \mathbb{R}^{n_x}$. Assume c, E, h, r convex.

Nonlinear MPC solves Parametric Nonlinear Programs (pNLP)

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{s,a} \ & \textstyle\sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\ \text{s.t.} \quad & s_0 = \bar{s}_i \\ & s_{k+1} = f(s_k, a_k) \\ & 0 \geq h(s_k, a_k), \ k = 0, \dots, N-1 \\ & 0 \geq r(s_N) \end{aligned}$$

Variables $s = (s_0, \dots)$ and $a = (a_0, \dots, a_{N-1})$ can be summarized in vector $x = (s, a) \in \mathbb{R}^{n_x}$. Assume c, E, h, r convex.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_x}} \ & F(x) \\ \text{s.t.} \ & G(x, \bar{s}_i) = 0 \\ & H(x) \geq 0 \end{aligned}$$

Assume F, H convex.

Nonlinear MPC solves Parametric Nonlinear Programs (pNLP)

Discrete time NMPC Problem (an NLP)

$$
\begin{aligned}
\min_{s,a} \quad & \sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\
\text{s.t.} \quad & s_0 = \bar{s}_i \\
& s_{k+1} = f(s_k, a_k) \\
& 0 \geq h(s_k, a_k), \; k = 0, \ldots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

Variables $s = (s_0, \ldots)$ and $a = (a_0, \ldots, a_{N-1})$ can be summarized in vector $x = (s, a) \in \mathbb{R}^{n_x}$.
Assume c, E, h, r convex.

Nonlinear Program (NLP)

$$
\begin{aligned}
\min_{x \in \mathbb{R}^{n_x}} \quad & F(x) \\
\text{s.t.} \quad & G(x, \bar{s}_i) = 0 \\
& H(x) \geq 0
\end{aligned}
$$

Assume F, H convex.

MPC policy: $\pi_{\text{MPC}}(\bar{s}_i) := a_0^*(\bar{s}_i)$ and closed loop $\bar{s}_{i+1} = f(\bar{s}_i, \pi_{\text{MPC}}(\bar{s}_i)) + \epsilon_i$ (disturbance)

Sequential Convex Programming (SCP)

Convex Subproblem at x^j

$$x^{j+1} \in \arg\min_{x \in \mathbb{R}^{n_x}} F(x)$$
$$\text{s.t. } G_{\mathrm{L}}(x, \bar{s}_i; x^j) = 0$$
$$H(x) \geq 0$$

Assume F, H convex.

Sequential Convex Programming (SCP)

OCP-QP at $s^j = (s_0^j, \ldots)$ and $a^j = (a_0^j, \ldots)$

$$
\begin{aligned}
\min_{s,a} \; & \textstyle\sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\
\text{s.t.} \quad & s_0 = \bar{s}_i \\
& s_{k+1} = f_{\mathrm{L}}(s_k, a_k; s_k^j, a_k^j) \\
& 0 \geq h(s_k, a_k), \; k = 0, \ldots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

Convex Subproblem at x^j

$$
\begin{aligned}
x^{j+1} \in \arg \min_{x \in \mathbb{R}^{n_x}} \; & F(x) \\
\text{s.t.} \; & G_{\mathrm{L}}(x, \bar{s}_i; x^j) = 0 \\
& H(x) \geq 0
\end{aligned}
$$

Assume F, H convex.

Distinguish three indices: $i-$th MPC problem, $j-$th SCP iteration, $k-$th time step in horizon

Real-Time Iteration (RTI)

Solve only one convex subproblem per samling time $(i = j)$, for latest state measurement $\bar{s}_{i+1}$.

Convex OCP (often a QP)

$$
\begin{aligned}
\min_{s,a} \; & \textstyle\sum_{k=0}^{N-1} c(s_k, a_k) + E(s_N) \\
\text{s.t.} \quad & s_0 = \bar{s}_i \\
& s_{k+1} = f_{\mathrm{L}}(s_k, a_k; s_k^{i-1}, a_k^{i-1}) \\
& 0 \geq h(s_k, a_k), \; k = 0, \dots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

Convex Subproblem at x^j

$$
\begin{aligned}
x^i \in \arg \min_{x \in \mathbb{R}^{n_x}} \; & F(x) \\
\text{s.t.} \; G_{\mathrm{L}}(x, \bar{s}_i; x^{i-1}) &= 0 \\
H(x) &\geq 0
\end{aligned}
$$

Parametric convex program with parameter $\bar{s}_{i+1}$

RTI MPC policy: $\pi_{\mathrm{RTI}}(\bar{s}_i) := a_0^i(\bar{s}_i)$

Overview

1 Real-Time Sequential Convex Programming (SCP)

2 The Software Packages BLASFEO and acados

3 A Fast Algorithm for Closed-Loop Robustified MPC

BLASFEO – Basic Linear Algebra For Embedded Optimization

Newton-type optimization needs linear algebra for matrix multiplications and factorizations etc. In embedded optimization, matrices often have fixed dimensions and consist of dense blocks.

BLASFEO contains fast linear algebra subroutines tailored for embedded optimization.

- Uses Panel-Major format. No reformatting in order to improve small scale performance.
- Hand tailored assembly kernels for different CPU architectures like `x86_64` and `Aarch64`
- Exploits vectorization with SIMD instruction by ISA extensions like `NEON`, `SSE3`, `AVX`, ...
- Supports high-end and low-end CPU like `Cortex A53` with in-order front-end.
- Compatible with conventional `BLAS API`

[BLASFEO: Basic Linear Algebra Subroutines For Embedded Optimization. G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, M. Diehl, ACM Transactions on Mathematical Software (TOMS) (2018)]

Philosophy & History

- Successor of the `ACADO Toolkit` (which used code generation also for linear algebra)
- Principles of `acados`:
  - efficiency – `BLASFEO`, `HPIPM`, C
  - flexibility – general formulation
  - modularity – encapsulation
  - portability – self-contained C library with little dependencies
- Model functions based on code generation using CasADi
- Problem formulation in high-level interface (`Python`, MATLAB, Octave)
- Generate corresponding C code for problem specific solver
  - uses only `acados` C interface
  - first developed in Python interface
  - used for S-function generation – Simulink interface
- solver interfaces for
  - OCP structured NLP and QP
  - Initial value problems for ODEs and DAEs – integrators

QP solver types and sparsity – an overview

QP solver types and their way to handle sparsity:

	Active-Set	Interior-Point	First-Order
dense	qpOASES	HPIPM	
sparse	PRESAS	CVXGEN, OOQP	FiOrdOs, OSQP
OCP structure	qpDUNES, ASIPM	HPMPC, HPIPM, ASIPM, FORCES	

underlined: available in acados + support in Simulink
gray: not interfaced in acados – partly proprietary

efficient condensing from HPIPM:

- condensing: OCP structured $\rightarrow$ dense, expand solution
- partial condensing: OCP structured with horizon N $\rightarrow$ OCP structured with horizon $N_2 < N$, expand solution, $N_2 \hat{=}$ qp_solver_cond_N

Integration methods in acados

- solve Initial Value Problems (IVP) for
  - Ordinary Differential Equations (ODE)
  - Differential-Algebraic Equations (DAE)
  - $+$ sensitivity propagation (derivative of result with respect to initial state, control input)
- `sim_method` in MATLAB, supports `'erk'`, `'irk'`, `'irk_gnsf'`
- size of Butcher table: `sim_method_num_stages`
- time step is divided into `sim_method_num_steps` intervals
- ERK: explicit Runge-Kutta
  - integration order `sim_method_num_stages` $= 1, 2, 4$
- IRK: implicit Runge-Kutta
  - Gauss-Legendre Butcher tableaus
  - integration order $2 \cdot$ `sim_method_num_stages`
- GNSF-IRK: implicit structure-exploiting Runge-Kutta method
  - Detecting and Exploiting Generalized Nonlinear Static Feedback Structures in DAE Systems for MPC, J. Frey, R. Quirynen, D. Kouzoupis, G. Frison, J. Geisler, A. Schild, M. Diehl, ECC 2019

Overview

1 Real-Time Sequential Convex Programming (SCP)

2 The Software Packages BLASFEO and acados

3 A Fast Algorithm for Closed-Loop Robustified MPC

Literature

PA1: Survey of Sequential Convex Programming and Generalized Gauss-Newton Methods. F. Messerer, K. Baumgrtner, M. Diehl, ESAIM: Proceedings and Surveys (2021)

PA2: BLASFEO: Basic Linear Algebra Subroutines For Embedded Optimization. G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, M. Diehl, ACM Transactions on Mathematical Software (TOMS) (2018)

PA3: acados – a modular open-source framework for fast embedded optimal control. R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, M Diehl, Math. Prog. Comp (accepted)

PA4: An Efficient Algorithm for Tube-based Robust Nonlinear Optimal Control with Optimal Linear Feedback. Florian Messerer and Moritz Diehl, CDC 2021 (accepted)

PDFs are on course page:
https://syscop.de/teaching/ss2021/model-predictive-control-and-reinforcement-learning