MPC for planning and control

From modeling and solver

Basic concept

RL

MPC

model

predict

control

LQR

Robust MPC

$$\min_{\{u_{k+j}\}} \max_{\{w_{k+j}\}} : \sum_{j=0}^{N-1} x_{k+j}^{\mathrm{T}} Q x_{k+j} + u_{k+j} R u_{k+j} + J_N(x_{k+N})$$

s.t.

$$x_{k+1} = A x_k + B u_k + G w_k$$
$$x_{k+j} \in X, \forall w_{k+j} \in W, j \in \mathbf{N}_{[0,N-1]}$$
$$u_{k+j} \in U, \forall w_{k+j} \in W, j \in \mathbf{N}_{[0,N-1]}$$
$$x_{k+N} \in X_N$$

Stochastic MPC

$$\min_{\{\pi_{k+j}\}_{j=0}^{N-1}} \mathbb{E}_{x_k}\Big[\sum_{j=0}^{N-1} J(x_{k+j}, \pi_{k+j}) + J_N(x_{k+N})\Big]$$

s.t.

$$x_{k+1} = A x_k + B u_k + G w_k$$
$$Pr(E_x x_{k+j} \le \mathbf{1}) \ge 1 - \epsilon, \quad j \in \mathbf{N}_{[0,N-1]}$$
$$Pr(E_u u_{k+j} \le \mathbf{1}) \ge 1 - \epsilon, \quad j \in \mathbf{N}_{[0,N-1]}$$
$$T_f(\cdot) \le 0$$

Real-Time MPC

Problem Statement

$$J = \sum_{N-1}^{0}(x^T Q x + u^T R u)$$

$$u_0^* = \min_{x_k, u_k} \sum_{k=0}^{N} (x_k - x_r)^T Q (x_k - x_r) + \sum_{k=0}^{N-1} u_k^T R u_k$$

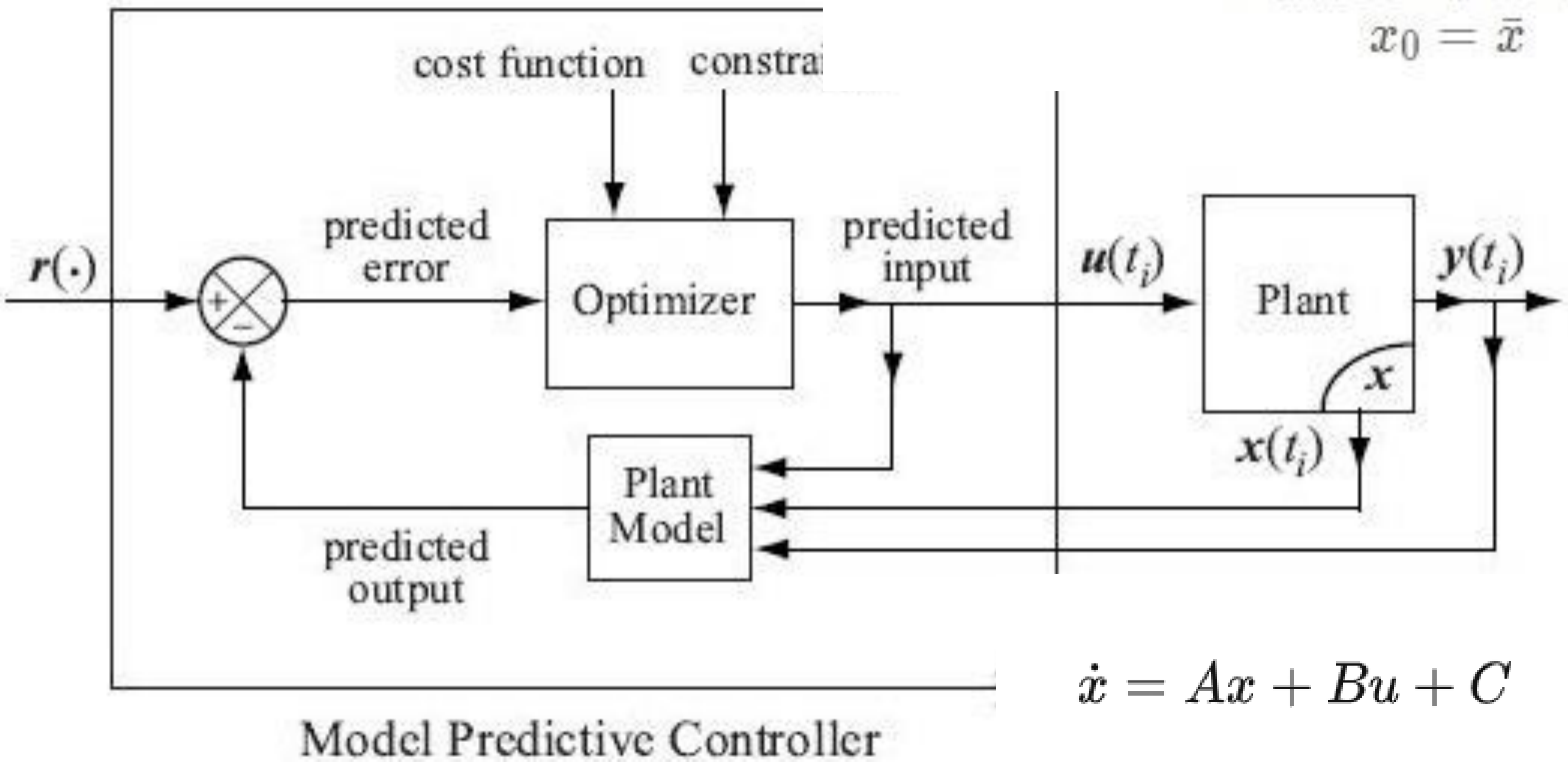
$$x_{k+1} = A x_k + B u_k$$

$$x_{\min} \leq x_k \leq x_{\max}$$

$$u_{\min} \leq x_k \leq u_{\max}$$

$$x_0 = \bar{x}$$

We have inequality here



$$\dot{x} = Ax + Bu + C$$

The Riccati Equation - Discrete Time

This is the Riccati matrix- difference equation. Solve it for $\{P(k),\ k \in \{0,\dots,N\}\}$

$$
\begin{aligned}
P(k) &= Q + A'P(k+1)A \\
&\qquad - A'P(k+1)B\big(R + B'P(k+1)B\big)^{-1}B'P(k+1)A, \\
P(N) &= Q_f.
\end{aligned}
$$

If $N = \infty$ the steady state solution P replaces $P(k)$. This P is the unique positive define solution found by the algebraic Riccati equation,

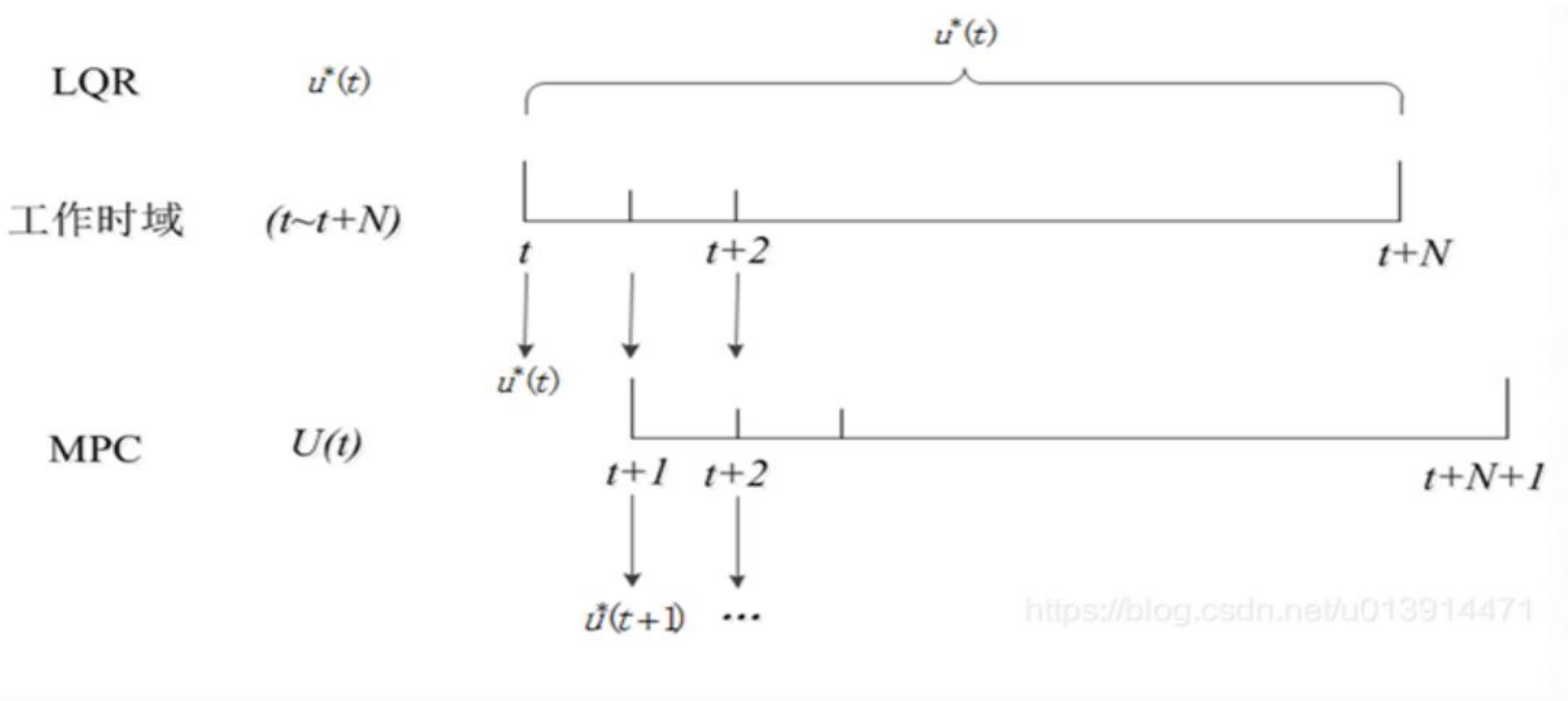
$$P = Q + A'PA - A'PB(R + B'PB)^{-1}B'PA.$$

The optimal control is:

$$u(k) = \Big(-\big(R+B'P(k+1)B\big)^{-1}B'P(k+1)A\Big)x(k), \text{ or } \quad u(k) = \Big(-\big(R+B'PB\big)^{-1}B'PA\Big)x(k).$$

How LQR works?

MPC VS LQR in self-driving



Model : the same
Solve : Matrix algbra vs SQP
Constraints: MPC can deal with inequality constraint

MPC VS RL

Dynamic programming
Value function formulation

$$V(s) = \inf_a [\psi(s,a) + \gamma \mathbb{E}_\xi [V(g(s,a,\xi))]]$$

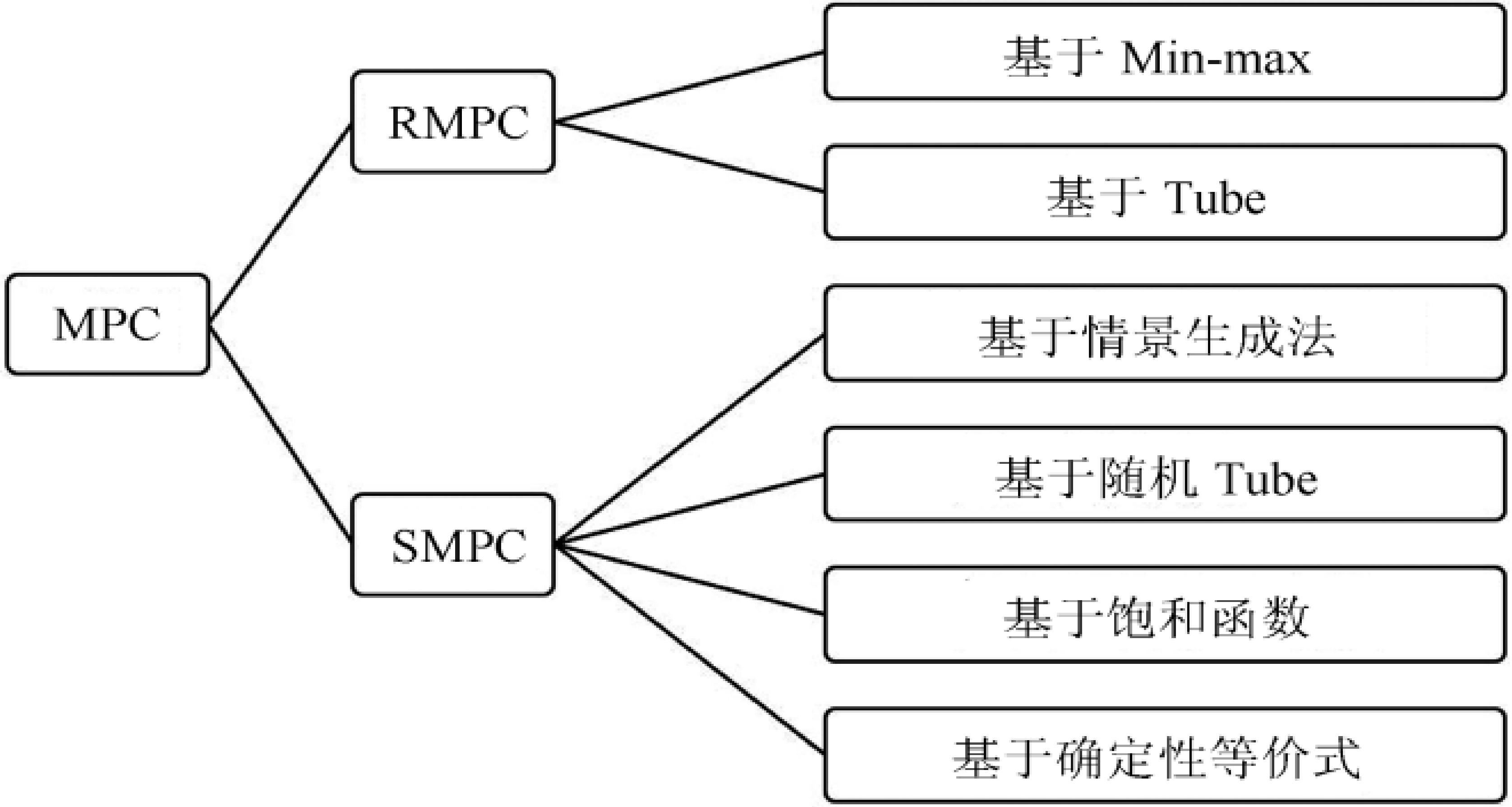
Q function formulation
(1)

$$Q(s,a) = \psi(s,a) + \gamma \mathbb{E}_\xi \left[\inf_v Q(g(s,a,\xi), v) \right]$$

Optimal policy (assuming all well defined)

$$\begin{aligned} \pi^*(s) &= \arg\min_a Q(s,a) \\ &= \arg\min_a \{\psi(s,a) + \gamma \mathbb{E}_\xi [V(g(s,a,\xi))]\} \end{aligned}$$

RobusMPC VS StochasticMPC



OSQP for MPC

https://osqp.org/docs/solver/index.html

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u \end{array}$$

Algorithm 1

1: **given** initial values x^0, z^0, y^0 and parameters $\rho > 0$, $\sigma > 0$, $\alpha \in (0, 2)$

2: **repeat**

3: $(\tilde{x}^{k+1}, \nu^{k+1}) \leftarrow$ solve linear system $\begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1} I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1} y^k \end{bmatrix}$

4: $\tilde{z}^{k+1} \leftarrow z^k + \rho^{-1}(\nu^{k+1} - y^k)$

5: $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1 - \alpha) x^k$

6: $z^{k+1} \leftarrow \Pi\left(\alpha \tilde{z}^{k+1} + (1 - \alpha) z^k + \rho^{-1} y^k\right)$

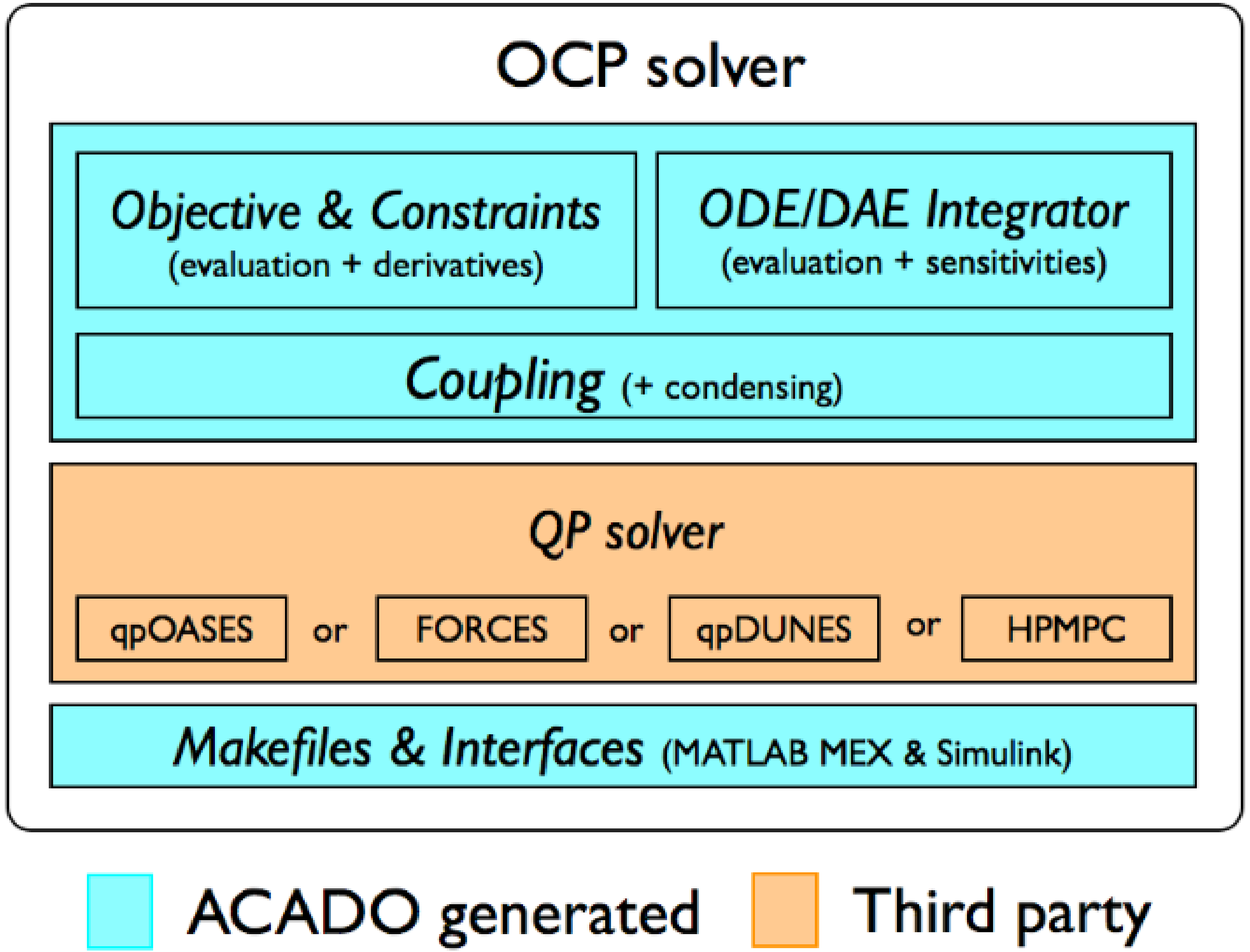
7: $y^{k+1} \leftarrow y^k + \rho\left(\alpha \tilde{z}^{k+1} + (1 - \alpha) z^k - z^{k+1}\right)$

8: **until** termination criterion is satisfied

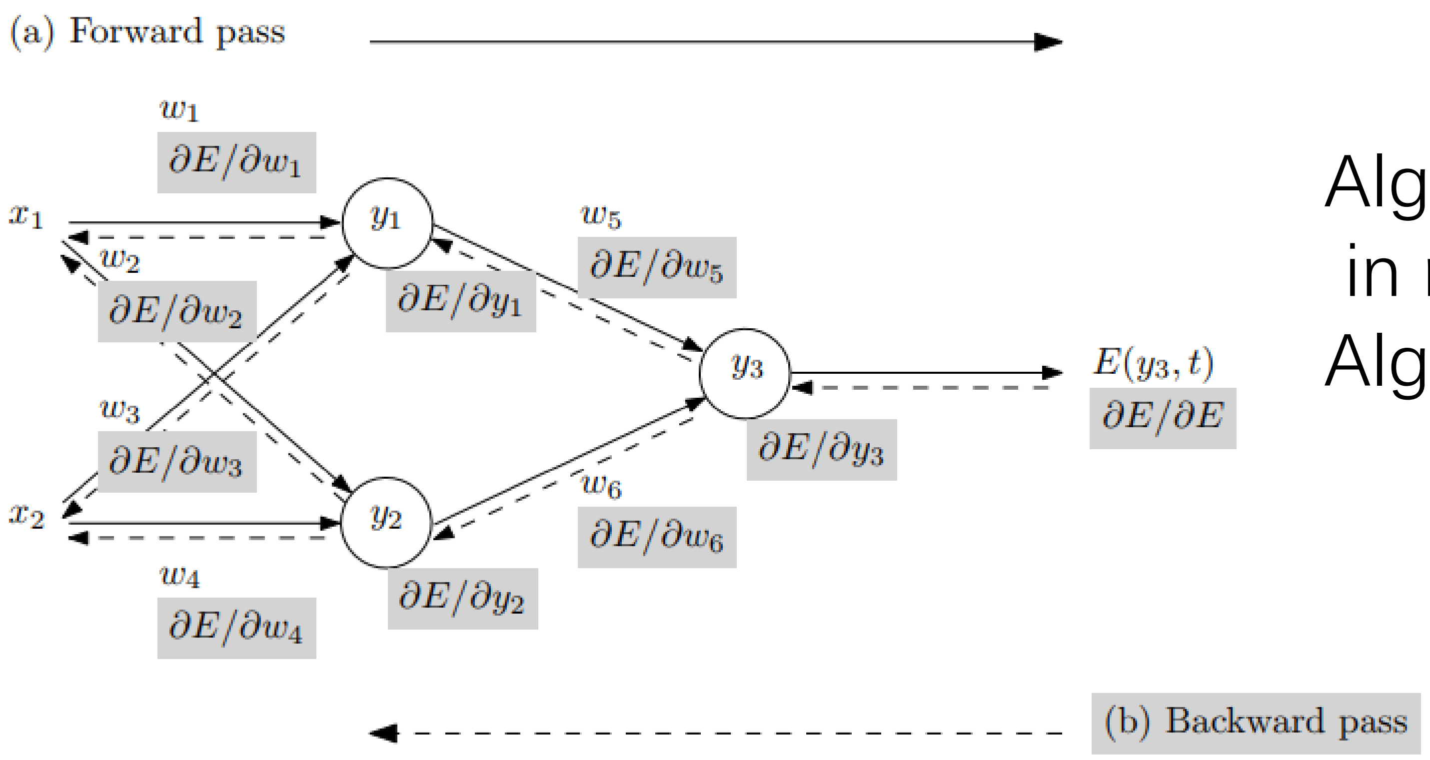
OSQP for MPC

Argument	Description	Allowed values	Default value
rho	ADMM rho step	0 < rho	0.1
sigma	ADMM sigma step	0 < sigma	0.000001
max_iter	Maximum number of iterations	0 < max_iter (integer)	4000
eps_abs *	Absolute tolerance	0 <= eps_abs	0.001
eps_rel *	Relative tolerance	0 <= eps_rel	0.001
eps_prim_inf *	Primal infeasibility tolerance	0 <= eps_prim_inf	0.0001
eps_dual_inf *	Dual infeasibility tolerance	0 <= eps_dual_inf	0.0001
alpha *	ADMM overrelaxation parameter	0 < alpha < 2	1.6
linsys_solver	Linear systems solver type		qdldl
delta *	Polishing regularization parameter	0 < delta	0.000001
polish *	Perform polishing	True/False	FALSE
polish_refine_iter *	Refinement iterations in polish	0 < polish_refine_iter (integer)	3
warm_start *	Perform warm starting	True/False	TRUE

MPC code geration
Toolbox :https://acado.github.io/features.html



NMPC: Numerical Optimal Control Toolbox :https://acado.github.io/features.html



Algorithmic differentiation in reverse mode: Forward Algorithmic differentiation

NMPC: Numerical Optimal Control
Toolbox: CasADi for Optimization

- ◆e direct multiple-shooting method
- ◆Direct Single Shooting
- ◆Hessian Approximations
- ◆constrained Gauss-Newton method
- ◆Sequential Quadratic Programming
- ◆Nonlinear IP Methods
- ◆Sequential Approaches and Sparsity Exploitation

NewSolver:https://faculty.sist.shanghaitech.edu.cn/faculty/boris/paper/AladinChapter.pdf

a) preprocessing	`checkInput(), setDefaultOpts()`
b) problem/sensitivity setup	`createLocSolAndSens()`
ALADIN main loop	`iterateAL()`
in parallel: c) solve local NLPs	`parallelStep(), BFGS(),`
in parallel: d) evaluate sensitivities	`parallelStepInnerLoop(),`
in parallel: e) Hessian approx./regularization	`updateParam(), regularizeH()`
f) solve the coordination QP	`createCoordQP(), solveQP(),solveQPdec()`
g) compute primal/dual step	`computeALstep()`
h) postprocessing	`displaySummary(), displayTimers()`

Figure 3: Structure of `run_ALADIN()` in ALADIN-α.

NewSolver: https://faculty.sist.shanghaitech.edu.cn/faculty/boris/paper/AladinChapter.pdf

---

Algorithm 1: Basic ALADIN

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, scaling matrices $\Sigma_i \in \mathbb{S}^n_{++}$ and a termination tolerance $\varepsilon > 0$.

Repeat:

1. Solve for all $i \in \{1, \ldots, N\}$ the decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda^\top A_i y_i + \frac{1}{2} \|y_i - x_i\|^2_{\Sigma_i} \; .$$

2. Set $g_i = \nabla f_i(y_i)$ and $H_i \approx \nabla^2 f_i(y_i)$.
3. Solve the coupled equality constrained QP

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

4. Set $x \leftarrow x^+ = y + \Delta y$ and $\lambda \leftarrow \lambda^+$ and continue with Step 1.

Three challenges：C1

A hard, non-convex optimization problems

- Geometric constraints
- Kinematic constraints
- Dynamic constraints

coupled

complicated an optimization problem

decoupled

Decoupled motion planning approaches

① path planning problem： geometric path that satisfies the geometric constraints.

② path following problem： taking into account all remaining constraints.

Three challenges：C2

Involve constraints that must hold during the complete motion time by through time gridding.

Drawbacks

- Constraints may be violated between the grid points,
- leads to a high number of constraints.

Piecewise_jerk_path_planning :max iteration reached

Three challenges：C3

The environment is generally uncertainDrawbacks

TO DO：to update the motion trajectory in real time, based on the most recent world information.

Solution1

https://github.com/meco-group/omg-tools

1.Proposing a B-spline parameterization for the motion trajectories.
2.Time gridding is avoided by exploiting the properties of B-splines to guarantee constraint satisfaction at all times m by using time-varying separating hyperplanes

Solution1

https://github.com/meco-group/omg-tools

Optimization problem

$$
\begin{aligned}
\underset{q(\cdot),T}{\text{minimize}} \quad & T \\
\text{subject to} \quad & q(0) = q_{\text{start}} \ , \ q(T) = q_{\text{end}} \\
& \dot{q}(0) = 0 \ , \ \dot{q}(T) = 0 \\
& \ddot{q}(0) = 0 \ , \ \ddot{q}(T) = 0 \\
& \dot{q}_{\min} \leq \dot{q}(t) \leq \dot{q}_{\max} \\
& \ddot{q}_{\min} \leq \ddot{q}(t) \leq \ddot{q}_{\max} \\
& \text{dist}(\text{veh}(t) \ , \ \text{obs}(t)) \geq \epsilon \\
& \forall t \in [0, T].
\end{aligned}
$$

linear combination of B-spline

$$s(t) = \sum_{i=1}^{n} c_i \cdot B_i(t).$$

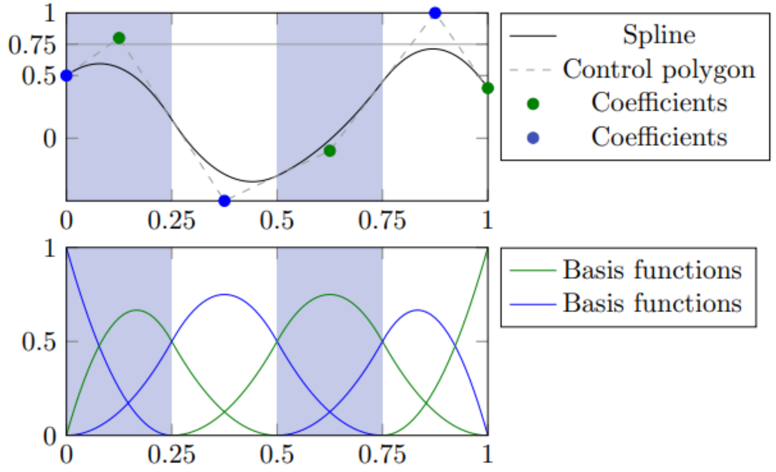


Fig. 1: Graphical illustration of a spline as a linear combination of B-spline basis functions

Solution1

https://github.com/meco-group/omg-tools

$$a(t)^T v_i(t) - b(t) \geq 0,\ \ i = 1 \ldots 4$$
$$a(t)^T q(t) - b(t) \leq -r_{\text{veh}}$$
$$||a(t)||_2 \leq 1$$
$$\forall t \in [0, T].$$

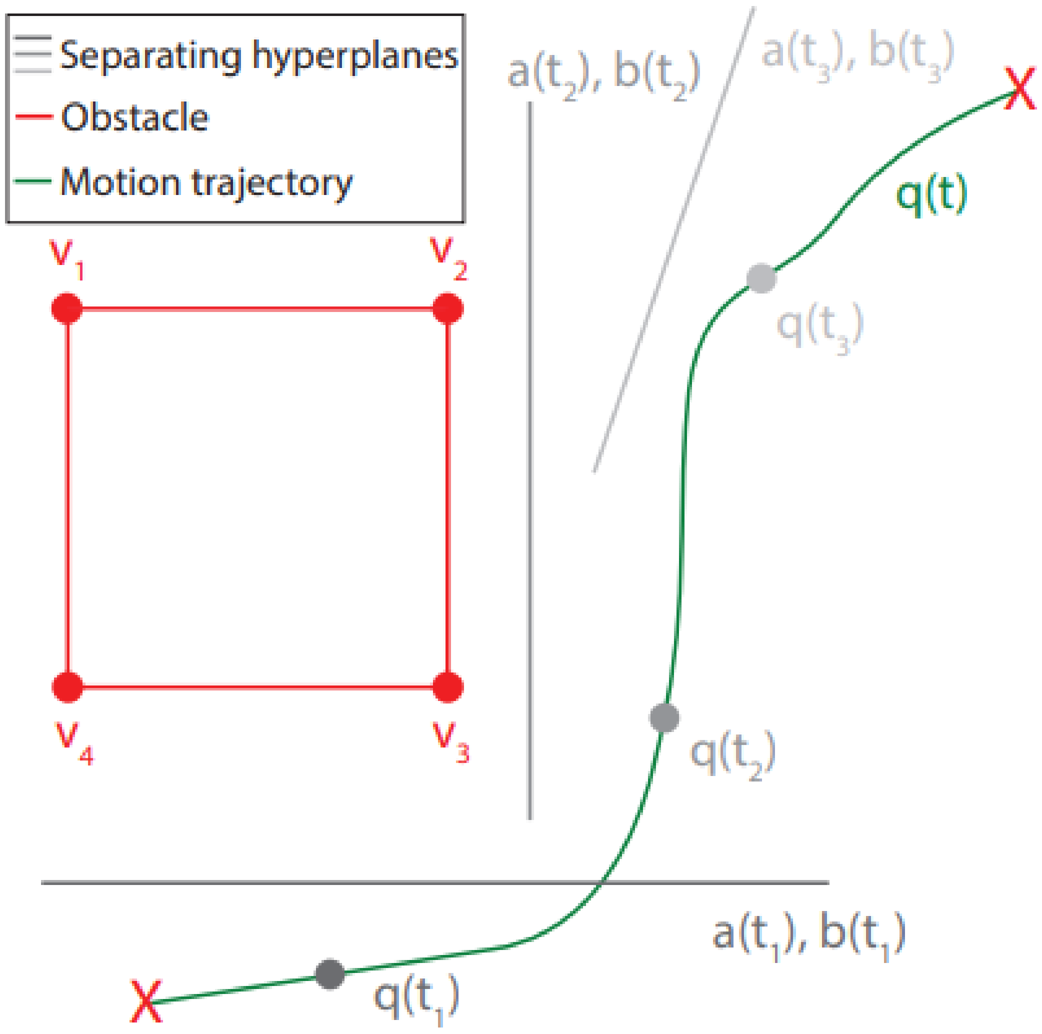


Fig. 2: Separating hyperplane theorem