

ACADOS / ACADO 2.0 / ???
Reference

February 9, 2016

Chapter 1

Interfaces

1.1 OCP QP Interface

The interface describes an Optimal Control Problem (OCP) Quadratic Programming (QP) problem in the form

$$\min_{x,u} \sum_{n=0}^{N-1} \frac{1}{2} \begin{bmatrix} u_n \\ x_n \\ 1 \end{bmatrix}^T \begin{bmatrix} R_n & S_n & r_n \\ S_n^T & Q_n & q_n \\ r_n^T & q_n^T & 1 \end{bmatrix} \begin{bmatrix} u_n \\ x_n \\ 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_N \\ 1 \end{bmatrix}^T \begin{bmatrix} Q_N & q_N \\ q_N^T & 1 \end{bmatrix} \begin{bmatrix} x_N \\ 1 \end{bmatrix} \quad (1.1)$$

$$s.t. \quad x_{n+1} = A_n x_n + B_n u_n + b_n, \quad n = 0, \dots, N-1 \quad (1.2)$$

$$\underline{u}_n \leq u_n \leq \bar{u}_n, \quad n = 0, \dots, N-1 \quad (1.3)$$

$$\underline{x}_n \leq x_n \leq \bar{x}_n, \quad n = 0, \dots, N \quad (1.4)$$

$$\underline{d}_n \leq C_n x_n + D_n u_n \leq \bar{d}_n, \quad n = 0, \dots, N-1 \quad (1.5)$$

$$\underline{d}_N \leq C_N x_N \leq \bar{d}_N \quad (1.6)$$

The C code interface looks like

```
int ocp_qp_solver(  
    int N, int *nx, int *nu, int *nb, int *ng,  
    double **A, double **B, double **b,  
    double **Q, double **S, double **R, double **q, double **r,  
    int **idxb, double **lb, double **ub,  
    double **C, double **D, double **ld, double **ud,  
    double **x, double **u,  
    struct ocp_qp_solver_args *args, double *work);
```

where

N [input] is the horizon length.

nx [input] is the vector of the state sizes n_u at the different stages, such that
nx[n] is the state size at stage n.

- nu** [input] is the vector of the input sizes n_x at the different stages, such that $\mathbf{nu}[\mathbf{n}]$ is the input size at stage \mathbf{n} .
- nb** [input] is the vector of the bound sizes n_b at the different stages, such that $\mathbf{nb}[\mathbf{n}]$ is the bound size at stage \mathbf{n} . The value of $\mathbf{nb}[\mathbf{n}]$ is smaller or equal to $\mathbf{nx}[\mathbf{n}] + \mathbf{nu}[\mathbf{n}]$.
- ng** [input] is the vector of the general polytopic constraint sizes n_g at the different stages, such that $\mathbf{ng}[\mathbf{n}]$ is the general polytopic constraint size at stage \mathbf{n} .
- A** [input] is the vector of size N of the pointers to the first element of the matrices A_n , such that $\mathbf{A}[\mathbf{n}]$ is the pointer to the first element of the matrix A_n , and $\mathbf{A}[\mathbf{n}][0]$ is the first element of the matrix A_n . The matrix referenced by the pointer $\mathbf{A}[\mathbf{n}]$ is stored in column-major (or Fortran-like) order, in a vector of $\mathbf{nx}[\mathbf{n}+1] \times \mathbf{nx}[\mathbf{n}]$ double-precision floating-point numbers.
- B** [input] is the vector of size N of the pointers to the first element of the matrices B_n , such that $\mathbf{B}[\mathbf{n}]$ is the pointer to the first element of the matrix B_n , and $\mathbf{B}[\mathbf{n}][0]$ is the first element of the matrix B_n . The matrix referenced by the pointer $\mathbf{B}[\mathbf{n}]$ is stored in column-major (or Fortran-like) order, in a vector of $\mathbf{nx}[\mathbf{n}+1] \times \mathbf{nu}[\mathbf{n}]$ double-precision floating-point numbers.
- b** [input] is the vector of size N of the pointers to the first element of the vectors b_n , such that $\mathbf{b}[\mathbf{n}]$ is the pointer to the first element of the vector b_n , and $\mathbf{b}[\mathbf{n}][0]$ is the first element of the vector b_n . The vector referenced by the pointer $\mathbf{b}[\mathbf{n}]$ is stored in a vector of $\mathbf{nx}[\mathbf{n}+1] \times 1$ double-precision floating-point numbers.
- Q** [input] is the vector of size $N + 1$ of the pointers to the first element of the matrices Q_n , such that $\mathbf{Q}[\mathbf{n}]$ is the pointer to the first element of the matrix Q_n , and $\mathbf{Q}[\mathbf{n}][0]$ is the first element of the matrix Q_n . The matrix referenced by the pointer $\mathbf{Q}[\mathbf{n}]$ is stored in column-major (or Fortran-like) order, in a vector of $\mathbf{nx}[\mathbf{n}] \times \mathbf{nx}[\mathbf{n}]$ double-precision floating-point numbers.
- S** [input] is the vector of size N of the pointers to the first element of the matrices S_n , such that $\mathbf{S}[\mathbf{n}]$ is the pointer to the first element of the matrix S_n , and $\mathbf{S}[\mathbf{n}][0]$ is the first element of the matrix S_n . The matrix referenced by the pointer $\mathbf{S}[\mathbf{n}]$ is stored in column-major (or Fortran-like) order, in a vector of $\mathbf{nu}[\mathbf{n}] \times \mathbf{nx}[\mathbf{n}]$ double-precision floating-point numbers.
- R** [input] is the vector of size N of the pointers to the first element of the matrices R_n , such that $\mathbf{R}[\mathbf{n}]$ is the pointer to the first element of the matrix R_n , and $\mathbf{R}[\mathbf{n}][0]$ is the first element of the matrix R_n . The matrix

referenced by the pointer $\mathbf{R}[\mathbf{n}]$ is stored in column-major (or Fortran-like) order, in a vector of $\mathbf{nu}[\mathbf{n}] \times \mathbf{nu}[\mathbf{n}]$ double-precision floating-point numbers.

q [input] is the vector of size $N + 1$ of the pointers to the first element of the vectors q_n , such that $\mathbf{q}[\mathbf{n}]$ is the pointer to the first element of the vector q_n , and $\mathbf{q}[\mathbf{n}][0]$ is the first element of the vector q_n . The vector referenced by the pointer $\mathbf{q}[\mathbf{n}]$ is stored in a vector of $\mathbf{nx}[\mathbf{n}] \times 1$ double-precision floating-point numbers.

r [input] is the vector of size N of the pointers to the first element of the vectors r_n , such that $\mathbf{r}[\mathbf{n}]$ is the pointer to the first element of the vector r_n , and $\mathbf{r}[\mathbf{n}][0]$ is the first element of the vector r_n . The vector referenced by the pointer $\mathbf{r}[\mathbf{n}]$ is stored in a vector of $\mathbf{nu}[\mathbf{n}] \times 1$ double-precision floating-point numbers.

idxb [input] is the vector of size $N + 1$ of the pointers to the first element of the integer vectors $idxb_n$ describing the indexes of the corresponding upper and lower bounds in **lb** and **ub**, such that $\mathbf{idxb}[\mathbf{n}]$ is the pointer to the index of the first bound at stage n , and $\mathbf{idxb}[\mathbf{n}][0]$ is index of the first bound at stage n . The indexes in $\mathbf{idxb}[\mathbf{n}]$ correspond to the position of the constrained components in the variables vector $\begin{bmatrix} u_n \\ x_n \end{bmatrix}$: therefore a bound on the first input component has index 0, a bound on the last input component has index $\mathbf{nu}[\mathbf{n}] - 1$, a bound on the first state component has index $\mathbf{nu}[\mathbf{n}]$ and a bound on the last state component has index $\mathbf{nu}[\mathbf{n}] + \mathbf{nx}[\mathbf{n}] - 1$. The vector referenced by the pointer $\mathbf{idxb}[\mathbf{n}]$ is stored in a vector of $\mathbf{nb}[\mathbf{n}] \times 1$ integer numbers.

lb [input] is the vector of size $N + 1$ of the pointers to the first element of the vectors $\begin{bmatrix} u_n \\ x_n \end{bmatrix}$, such that $\mathbf{lb}[\mathbf{n}]$ is the pointer to the first element of the vector $\begin{bmatrix} u_n \\ x_n \end{bmatrix}$, and $\mathbf{lb}[\mathbf{n}][0]$ is the first element of the vector $\begin{bmatrix} u_n \\ x_n \end{bmatrix}$. The vector referenced by the pointer $\mathbf{lb}[\mathbf{n}]$ is stored in a vector of $\mathbf{nb}[\mathbf{n}] \times 1$ double-precision floating-point numbers.

ub [input] is the vector of size $N + 1$ of the pointers to the first element of the vectors $\begin{bmatrix} \bar{u}_n \\ \bar{x}_n \end{bmatrix}$, such that $\mathbf{ub}[\mathbf{n}]$ is the pointer to the first element of the vector $\begin{bmatrix} \bar{u}_n \\ \bar{x}_n \end{bmatrix}$, and $\mathbf{ub}[\mathbf{n}][0]$ is the first element of the vector $\begin{bmatrix} \bar{u}_n \\ \bar{x}_n \end{bmatrix}$. The vector referenced by the pointer $\mathbf{ub}[\mathbf{n}]$ is stored in a vector of $\mathbf{nb}[\mathbf{n}] \times 1$ double-precision floating-point numbers.

C [input] is the vector of size $N + 1$ of the pointers to the first element of the matrices C_n , such that $\mathbf{C}[\mathbf{n}]$ is the pointer to the first element of the matrix C_n , and $\mathbf{C}[\mathbf{n}][0]$ is the first element of the matrix C_n . The matrix

referenced by the pointer `C[n]` is stored in column-major (or Fortran-like) order, in a vector of `ng[n] × nx[n]` double-precision floating-point numbers.

D [input] is the vector of size N of the pointers to the first element of the matrices D_n , such that `D[n]` is the pointer to the first element of the matrix D_n , and `D[n][0]` is the first element of the matrix D_n . The matrix referenced by the pointer `D[n]` is stored in column-major (or Fortran-like) order, in a vector of `ng[n] × nu[n]` double-precision floating-point numbers.

ld [input] is the vector of size $N + 1$ of the pointers to the first element of the vectors \underline{d}_n , such that `ld[n]` is the pointer to the first element of the vector \underline{d}_n , and `ld[n][0]` is the first element of the vector \underline{d}_n . The vector referenced by the pointer `ld[n]` is stored in a vector of `ng[n] × 1` double-precision floating-point numbers.

ud [input] is the vector of size $N + 1$ of the pointers to the first element of the vectors \bar{d}_n , such that `ud[n]` is the pointer to the first element of the vector \bar{d}_n , and `ud[n][0]` is the first element of the vector \bar{d}_n . The vector referenced by the pointer `ud[n]` is stored in a vector of `ng[n] × 1` double-precision floating-point numbers.

x [output] is the vector of size $N + 1$ of the pointers to the first element of the vectors x_n , such that `x[n]` is the pointer to the first element of the vector x_n , and `x[n][0]` is the first element of the vector x_n . The vector referenced by the pointer `x[n]` is stored in a vector of `nx[n] × 1` double-precision floating-point numbers.

u [output] is the vector of size $N + 1$ of the pointers to the first element of the vectors u_n , such that `u[n]` is the pointer to the first element of the vector u_n , and `u[n][0]` is the first element of the vector u_n . The vector referenced by the pointer `u[n]` is stored in a vector of `nu[n] × 1` double-precision floating-point numbers.

args [input] is the pointer to a structure of type `ocp_qp_solver_args` that defines the arguments (as e.g. maximum number of iterations, minimum step size, ...) passed to the specific solver.

work [workspace] is the pointer to the working space used by the specific solver. The working space size (in doubles) is returned by a call to the function `ocp_qp_SOLVERNAME_workspace_double(int N, int *nx, int *nu, int *nb, int *ng, struct ocp_qp_solver_args *args)`, where `SOLVERNAME` is the name of the specific solver.

Furthermore, the function returns an `int`, that is defined in the following enum (TODO change the names to something better!!!):

ACADOS_OK Solution successfully found.

ACADOS_MAXITER Maximum number of iterations reached.

ACADOS_MINSTEP Minimum step size reached (in IPs, probably unfeasible problem).

1.1.1 Examples

MPC problem

In the MPC problem, the initial state is fixed. This is modelled by choosing $\mathbf{nx}[0]=0$, i.e. not considering the initial state as an optimization variable. As a consequence, e.g. the matrix $\mathbf{A}[0]$ has size $\mathbf{nx}[1] \times 0$, the matrix $\mathbf{Q}[0]$ has size 0×0 , and the vector $\mathbf{q}[0]$ has size 0×1 . The information about the known value of x_0 and the matrix A_0 are used to compute the value of the vector $\mathbf{b}[0]$, that is initialized to $b_0 + A_0 \cdot x_0$.