Problem Formulation

$$
\begin{aligned}
&\underset{x(\cdot),\,u(\cdot),\,z(\cdot),\,s(\cdot),\,s^{\mathrm{e}}}{\min} && \texttt{/* Cost Function */} \\
& && \int_0^T l(x(\tau),u(\tau),z(\tau),p) + \frac{1}{2}\begin{bmatrix} s_{\mathrm{l}}(\tau) \\ s_{\mathrm{u}}(\tau) \\ 1 \end{bmatrix}^\top \begin{bmatrix} Z_{\mathrm{l}} & 0 & z_{\mathrm{l}} \\ 0 & Z_{\mathrm{u}} & z_{\mathrm{u}} \\ {z_{\mathrm{l}}}^\top & {z_{\mathrm{u}}}^\top & 0 \end{bmatrix} \begin{bmatrix} s_{\mathrm{l}}(\tau) \\ s_{\mathrm{u}}(\tau) \\ 1 \end{bmatrix} \mathrm{d}\tau \, + \\
& && \quad m(x(T),z(T),p) + \frac{1}{2}\begin{bmatrix} s_{\mathrm{l}}^{\mathrm{e}} \\ s_{\mathrm{u}}^{\mathrm{e}} \\ 1 \end{bmatrix}^\top \begin{bmatrix} Z_{\mathrm{l}}^{\mathrm{e}} & 0 & z_{\mathrm{l}}^{\mathrm{e}} \\ 0 & Z_{\mathrm{u}}^{\mathrm{e}} & z_{\mathrm{u}}^{\mathrm{e}} \\ {z_{\mathrm{l}}^{\mathrm{e}}}^\top & {z_{\mathrm{u}}^{\mathrm{e}}}^\top & 0 \end{bmatrix} \begin{bmatrix} s_{\mathrm{l}}^{\mathrm{e}} \\ s_{\mathrm{u}}^{\mathrm{e}} \\ 1 \end{bmatrix} \\
& && \texttt{/* Initial value */} \\
&\text{s.t.} && \underline{x}_0 \le J_{\mathrm{bx},0}\, x(0) \le \bar{x}_0, \\
& && \texttt{/* Dynamics */} \\
& && f_{\mathrm{impl}}(x(t),\dot{x}(t),u(t),z(t),p) = 0, && t \in [0,\,T), \\
& && \texttt{/* Path Constraints with lower bound */} \\
& && \underline{h} \le h(x(t),u(t),p) + J_{\mathrm{sh}} s_{\mathrm{l,h}}(t), && t \in [0,\,T), \\
& && \underline{x} \le J_{\mathrm{bx}} x(t) + J_{\mathrm{sbx}} s_{\mathrm{l,bx}}(t), && t \in (0,\,T), \\
& && \underline{u} \le J_{\mathrm{bu}} u(t) + J_{\mathrm{sbu}} s_{\mathrm{l,bu}}(t), && t \in [0,\,T), \\
& && \underline{g} \le C x(t) + D u(t) + J_{\mathrm{sg}} s_{\mathrm{l,g}}(t), && t \in [0,\,T), \\
& && s_{\mathrm{l,h}}(t), s_{\mathrm{l,bx}}(t), s_{\mathrm{l,bu}}(t), s_{\mathrm{l,g}}(t) \ge 0, && t \in [0,\,T), \\
& && \texttt{/* Path Constraints with upper bound */} \\
& && h(x(t),u(t),p) - J_{\mathrm{sh}} s_{\mathrm{u,h}}(t) \le \bar{h}, && t \in [0,\,T), \\
& && J_{\mathrm{bx}} x(t) - J_{\mathrm{sbx}} s_{\mathrm{u,bx}}(t) \le \bar{x}, && t \in (0,\,T), \\
& && J_{\mathrm{bu}} u(t) - J_{\mathrm{sbu}} s_{\mathrm{u,bu}}(t) \le \bar{u}, && t \in [0,\,T), \\
& && C x(t) + D u(t) - J_{\mathrm{sg}} s_{\mathrm{u,g}} \le \bar{g}, && t \in [0,\,T), \\
& && s_{\mathrm{u,h}}(t), s_{\mathrm{u,bx}}(t), s_{\mathrm{u,bu}}(t), s_{\mathrm{u,g}}(t) \ge 0, && t \in [0,\,T), \\
& && \texttt{/* Terminal Constraints with lower bound */} \\
& && \underline{h}^{\mathrm{e}} \le h^{\mathrm{e}}(x(T),p) + J_{\mathrm{sh}}^{\mathrm{e}} s_{\mathrm{l,h}}^{\mathrm{e}}, \\
& && \underline{x}^{\mathrm{e}} \le J_{\mathrm{bx}}^{\mathrm{e}} x(T) + J_{\mathrm{sbx}} s_{\mathrm{l,bx}}^{\mathrm{e}}, \\
& && \underline{g}^{\mathrm{e}} \le C^{\mathrm{e}} x(T) + J_{\mathrm{sg}}^{\mathrm{e}} s_{\mathrm{l,g}}^{\mathrm{e}} \le \bar{g}^{\mathrm{e}}, \\
& && s_{\mathrm{l,h}}^{\mathrm{e}}, s_{\mathrm{l,bx}}^{\mathrm{e}}, s_{\mathrm{l,bu}}^{\mathrm{e}}, s_{\mathrm{l,g}}^{\mathrm{e}} \ge 0, \\
& && \texttt{/* Terminal Constraints with upper bound */} \\
& && h^{\mathrm{e}}(x(T),p) - J_{\mathrm{sh}}^{\mathrm{e}} s_{\mathrm{h}}^{\mathrm{e}} \le \bar{h}^{\mathrm{e}}, \\
& && J_{\mathrm{bx}}^{\mathrm{e}} x(T) - J_{\mathrm{sbx}} s_{\mathrm{bx}}^{\mathrm{e}} \le \bar{x}^{\mathrm{e}}, \\
& && C^{\mathrm{e}} x(T) - J_{\mathrm{sg}}^{\mathrm{e}} s_{\mathrm{g}}^{\mathrm{e}} \le \bar{g}^{\mathrm{e}} \\
& && s_{\mathrm{u,h}}^{\mathrm{e}}, s_{\mathrm{u,bx}}^{\mathrm{e}}, s_{\mathrm{u,bu}}^{\mathrm{e}}, s_{\mathrm{u,g}}^{\mathrm{e}} \ge 0,
\end{aligned}
$$

with

- state vector $x : \mathbb{R} \rightarrow \mathbb{R}^{n_{\mathrm{x}}}$
- control vector $u : \mathbb{R} \rightarrow \mathbb{R}^{n_{\mathrm{u}}}$
- algebraic state vector $z : \mathbb{R} \rightarrow \mathbb{R}^{n_{\mathrm{z}}}$
- model parameters $p \in \mathbb{R}^{n_{\mathrm{p}}}$
- slacks for path constraints $s_{\mathrm{l}}(t) = (s_{\mathrm{l,bu}}, s_{\mathrm{l,bx}}, s_{\mathrm{l,g}}, s_{\mathrm{l,h}}) \in \mathbb{R}^{n_{\mathrm{s}}}$ and $s_{\mathrm{u}}(t) = (s_{\mathrm{u,bu}}, s_{\mathrm{u,bx}}, s_{\mathrm{u,g}}, s_{\mathrm{u,h}}) \in \mathbb{R}^{n_{\mathrm{s}}}$
- slacks for terminal constraints $s_{\mathrm{l}}^{\mathrm{e}}(t) = (s_{\mathrm{l,bx}}^{\mathrm{e}}, s_{\mathrm{l,g}}^{\mathrm{e}}, s_{\mathrm{l,h}}^{\mathrm{e}}) \in \mathbb{R}^{n_{\mathrm{s}}^{\mathrm{e}}}$ and $s_{\mathrm{u}}^{\mathrm{e}}(t) = (s_{\mathrm{u,bx}}^{\mathrm{e}}, s_{\mathrm{u,g}}^{\mathrm{e}}, s_{\mathrm{u,h}}^{\mathrm{e}}) \in \mathbb{R}^{n_{\mathrm{s}}^{\mathrm{e}}}$

Some of the following restrictions may apply to matrices in the formulation:

DIAG
: Diagonal

SPUM
: Horizontal slice of a permuted unit matrix

SPUME
: Like **SPUM**, but with empty rows intertwined

1 Dynamics

The system dynamics can be formulated in different ways in `acados`.

Implicit Dynamics

The most general way to provide a continuous time ODE in `acados`is to define the function $f_{\mathrm{impl}} : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{z}} \times \mathbb{R}^{n_\mathrm{p}} \rightarrow \mathbb{R}^{n_\mathrm{x}+n_\mathrm{z}}$ which is fully implicit DAE formulation describing the system as:

$$f_{\mathrm{impl}}(x, \dot{x}, u, z, p) = 0.$$

We offer to discretize f_{impl} with a classic implicit Runge-Kutta (`irk`) or a structure exploiting implicit Runge-Kutta method (`irk_gnsf`).

Explicit Dynamics

Alternatively, we offer an explicit Runge-Kutta integrator (`erk`), which can be used with explicit ODE models, i.e. models of the form

$$f_{\mathrm{expl}}(x, u, p) = \dot{x}$$

Discrete Dynamics

Another option is to provide a discrete function that maps the state from shooting node i to the next shooting node, i.e. a function

$$x_{i+1} = f_{\mathrm{disc}}(x_i, u_i, p_i)$$

Mathematical Expression	string identifier	data type	required
f_{impl} respectively f_{expl}	`dyn_expr_f`	`CasADi` expression	yes
f_{disc}	`dyn_exp_phi`	`CasADi` expression	yes
-	`dyn_type`	string (`explicit` or `implicit`)	yes

2 Cost

There are different `acados` modules to model the cost functions.

- $l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_z} \to \mathbb{R}$ is the Lagrange objective term.
- $m : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}$ is the Mayer objective term.

to decide which one is used set `cost_type` for l, `cost_type_e` for m.

Setting the slack penalties is done in the same way for all cost modules, namely:

Mathematical Expression	string identifier	data type	required
Z_l	`cost_Zl`	double, **DIAG**	no
Z_u	`cost_Zu`	double, **DIAG**	no
z_l	`cost_zl`	double	no
z_u	`cost_zu`	double	no
Z_l^e	`cost_Zl_e`	double, **DIAG**	no
Z_u^e	`cost_Zu_e`	double, **DIAG**	no
z_l^e	`cost_zl_e`	double	no
z_u^e	`cost_zu_e`	double	no

Moreover, you can specify `cost_Z`, to set $Z_\mathrm{l}, Z_\mathrm{u}$ to the same values, i.e. use a symmetric L2 slack penalty. Similarly, `cost_z`, `cost_Z_e`, `cost_z_e` can be used to set symmetric slack L1 penalties, respectively penalties for the terminal slack variables.

Cost module: `auto`

Set `cost_type` to `auto` (default). In this case we detect if the cost function specified is a linear least squares term and transcribe it in the corresponding form. Otherwise, it is formulated using the external cost module. Note: slack penalties are optional and we plan to detected them form the expressions in future versions.

Mathematical Expression	string identifier	data type	required
l	`cost_expr_ext_cost`	`CasADi` expression	yes
m	`cost_expr_ext_cost_e`	`CasADi` expression	yes

Cost module: `external`

Set `cost_type` to `ext_cost`.

Mathematical Expression	string identifier	data type	required
l	`cost_expr_ext_cost`	`CasADi` expression	yes
m	`cost_expr_ext_cost_e`	`CasADi` expression	yes

Cost module: `linear least squares`

Set `cost_type` to `linear_ls`.
The Lagrange cost term has the form

$$l(x,u,z) = \frac{1}{2} \left\| \underbrace{V_x x + V_u u + V_z z}_{y} - y_\mathrm{ref} \right\|_W^2$$

with matrices V_x, V_u, V_z, W of appropriate dimensions.
Similarly, the Mayer cost term has the form

$$m(x,u,z) = \frac{1}{2} \left\| \underbrace{V_x^{\mathrm{e}} x}_{y^{\mathrm{e}}} - y_{\mathrm{ref}}^{\mathrm{e}} \right\|_{W^{\mathrm{e}}}^2$$

with matrices $V_x^{\mathrm{e}}, W^{\mathrm{e}}$ of appropriate dimensions.

Mathematical Expression	string identifier	data type	required
V_x	`cost_Vx`	double	yes
V_u	`cost_Vu`	double	yes
V_z	`cost_Vz`	double	yes
W	`cost_W`	double	yes
y_{ref}	`cost_y_ref`	double	yes
V_x^{e}	`cost_Vx_e`	double	yes
W^{e}	`cost_W_e`	double	yes
$y_{\mathrm{ref}}^{\mathrm{e}}$	`cost_y_ref_e`	double	yes

Cost module: `nonlinear least squares`

Set `cost_type` to `nonlinear_ls`.
The cost function has the same form as in the linear least squares module.
The only difference is that y, respectively y^{e} are defined as `CasADi` expressions, instead of the matrices V_x, V_u, V_z, respectively V_x^{e}

Mathematical Expression	string identifier	data type	required
y	`cost_expr_y`	`CasADi` expression	yes
W	`cost_W`	double	yes
y_{ref}	`cost_y_ref`	double	yes
y^{e}	`cost_expr_y_e`	`CasADi` expression	yes
$y_{\mathrm{ref}}^{\mathrm{e}}$	`cost_y_ref_e`	double	yes

3 Constraints

3.1 Initial State

Note: An initial state is not required. For example for MHE problems it should not be set.

Simple syntax for initial constraint $x(0) = \bar{x}_0$:

Mathematical Expression	string identifier	data type	required
$\bar{x}_0$	`constr_x0`	double	no

Extended syntax:

3.2 Path Constraints

Mathematical Expression	string identifier	data type	required
$\underline{x}_0$	`constr_lbx_0`	double	no
$\bar{x}_0$	`constr_ubx_0`	double	no
$J_{\mathrm{bx},0}$	`constr_Jbx_0`	double	no

Mathematical Expression	string identifier	data type	required
J_{bx}	`constr_Jbx`	double, **SPUM**	no
$\underline{x}$	`constr_lbx`	double	no
$\bar{x}$	`constr_ubx`	double	no
J_{bu}	`constr_Jbu`	double, **SPUM**	no
$\underline{u}$	`constr_lbu`	double	no
$\bar{u}$	`constr_ubu`	double	no
C	`constr_C`	double	no
D	`constr_D`	double	no
$\underline{g}$	`constr_lg`	double	no
$\bar{g}$	`constr_ug`	double	no
h	`constr_expr_h`	`CasADi` expression	no
$\underline{h}$	`constr_lh`	double	no
$\bar{h}$	`constr_uh`	double	no
J_{sbx}	`constr_Jsbx`	double, **SPUME**	no
J_{sbu}	`constr_Jsbu`	double, **SPUME**	no
J_{sg}	`constr_Jsg`	double, **SPUME**	no
J_{sbx}	`constr_Jsh`	double, **SPUME**	no

3.3 Terminal Constraints

Mathematical Expression	string identifier	data type	required
J_{bx}	`constr_Jbx_e`	double, **SPUM**	no
$\underline{x}^{\mathrm{e}}$	`constr_lbx_e`	double	no
$\bar{x}^{\mathrm{e}}$	`constr_ubx_e`	double	no
C^{e}	`constr_C_e`	double	no
$\underline{g}^{\mathrm{e}}$	`constr_lg`	double	no
$\bar{g}^{\mathrm{e}}$	`constr_ug`	double	no
h^{e}	`constr_expr_h_e`	`CasADi` expression	no
$\underline{h}^{\mathrm{e}}$	`constr_lh_e`	double	no
$\bar{h}^{\mathrm{e}}$	`constr_uh_e`	double	no
J_{sbx}	`constr_Jsbx`	double, **SPUME**	no
$J_{\mathrm{sg}}^{\mathrm{e}}$	`constr_Jsg_e`	double, **SPUME**	no
$J_{\mathrm{sbx}}^{\mathrm{e}}$	`constr_Jsh_e`	double, **SPUME**	no

3.4 External links

`https://docs.google.com/spreadsheets/d/1rVRycLnCyaWJLwnV47u30Vokp7vRu68og3OhlDbSjDU/edit?usp=sharing`