acados Workshop

Albert-Ludwigs-Universität Freiburg – April 2021

Exercises: Getting started with acados in Matlab

In this exercise you will adapt the acados MATLAB getting started examples given in the folder `examples/acados_matlab_octave/getting_started` to get to familiar with acados.

Important Resources

- https://docs.acados.org/
- acados problem formulation PDF: https://github.com/acados/acados/blob/master/docs/problem_formulation/problem_formulation_ocp_mex.pdf
- Python API - documents all options in template interface: https://docs.acados.org/python_api
- acados Matlab cheat sheet

Exercises

Warm-Up

1. Run the `minimal_example_ocp.m` in https://github.com/acados/acados/blob/master/examples/acados_matlab_octave/getting_started/minimal_example_ocp.m
2. Change the QP Solver, e.g. `qpOASES`, `HPIPM` with different condensing options, `OSQP`.
3. Change the integration method, vary the number of intermediate integration steps.
4. Regard the formulation of the cost function. Take a look at the output of the automatic structure detection for the cost function. Compare it with the cost formulation stated in the problem formulation PDF.

Parameters and Model-Plant Mismatch

In this exercise, we reformulate the pendulum on cart model to include the mass of the cart as a parameter. Within the closed-loop simulation, we investigate how the control performance changes depending on the model-plant mismatch due to a wrong parameter value.

1. Run the script `simulink_example_advanced.m`.
   The script first calls the script `minimal_example_ocp.m` and then generates the problem specific C code, as well as the S-functions. Check `ocp.acados_ocp_nlp_json` and compare with the Python API (see *Important Resources*).
2. Adapt the files `pendulum_on_cart_model.m` and `minimal_example_ocp.m` to include the mass of the cart M as a parameter.
   Note: Make M a `CasADi` symbolic variable and pass it to `ocp_model` using something like `ocp_model.set('sym_p',...)`

3. Solve the OCP for different values of M:
   Note: Once you created the solver with the parametric model, parameters can be updated using `ocp.set( )` routine. The routine can be called as: `ocp.set(field, value, [stage])`, where the last argument is optional.
   For example: `ocp.set('p', 1.0)` sets the parameter to 1 for all shooting nodes. The same thing can be done more explicitly using:

```
for ii = 0 : N
    ocp.set('p', 1, ii);
end
```

4. Run again the script `simulink_example_advanced.m` and make the corresponding changes to the `.slx` file such that you can set the parameters for the integrator and the OCP solver independently.
   Hint: Check the output in your Matlab terminal for information on the input and output ports of the S-function block.

5. Try different values for the model and plant parameter and check if the controller is still able to stabilize the system.

Timings

1. Adapt the file `minimal_example_ocp.m` to solve the OCP multiple times and store the computation time for each run.
   Hint: Use the member functions `store_iterate()` and `load_iterate()` to initialize the solver with the same values before each run.

2. Compare the timings with the performance of the general purpose solver `IPOPT` which is used by default in `CasADi`.