Three page Instructions for using MPC_for_environmental_management

Welcome. Unzip the package to a location where you have read/write access and can launch Matlab/Python from the terminal.

The parent directory: `MPC_for_environmental_management` will be your home-base for the experiment (I've set things up so that you call functions with the `pwd` command, if you can't work from the `MPC_for_environmental_management` directory just replace `pwd` with a string containing the absolute path to `MPC_for_environmental_management`).

Pre-requisites:
The assumption is that you are going to attempt to get the minimum annual extent of the Arctic sea ice to track a particular path beginning at the control start year (for example 2017). You will need to be able to get from your model:

1) The year
2) Mean Northern Hemisphere temperature
3) Mean Southern Hemisphere temperature
4) Minimum annual extent of Arctic sea ice in millions of square km
5) The total annual Northern Hemisphere SO_2 emissions (Gt)
6) The total annual Southern Hemisphere SO_2 emissions (Gt) (will be zero)
7) The mid-year forcing due to greenhouse gas (GHG) (Wm^{-2})
8) The incoming SW radiative forcing (Wm-2)
9) The outgoing SW radiative forcing (Wm-2)
10) The net TOA radiative forcing (Wm-2)
11) The year total Northern Hemisphere volcanic sulphate emissions (Gt)
12) The year total Southern Hemisphere volcanic sulphate emissions (Gt)

You will also need to be able to force your model with SO_2 emissions specified as the total weekly mass of SO_2 to inject into the Northern hemisphere stratosphere for the first 21 weeks of the calendar year.

Spin-up
The MPC needs to spin up on some data from your model. So, before you begin the control experiment, collect from your model the items (1) to (12) above starting at a pre-industrial-ish year (ideally 1860 or 1900 at a push) and ending the year before the control starts (say 2016). Put this data in a Matlab matrix called `spin_up_data` with 12 columns each column coinciding with (1) to (12) above. Save `spin_up_data` into a file called `spin_up_data` i.e., in Matlab do:
```
save spin_up_data spin_up_data
```

You now need to make a note of the following:

1) The Northern Hemisphere unperturbed temperature in degC (for HadGEM2 this is about 13.1).
2) The Southern Hemisphere unperturbed temperature in degC (for HadGEM2 this is about 13.5).
3) The unperturbed minimum Arctic sea ice extent in millions of square km (for HadGEM2 this is about 5.5)

In the `MPC_for_environmental_management` directory open `IAGP_model_setup` in the initializations directory. This is where all the parameters for the project are located. For now just find lines 58 to 60 and change for your unperturbed temperature and extent values (1) to (3) above and

save the changes. You might also want to change lines 53 and 54: `Param.ice_stabilisation_level = 5;` and
`Param.ice_stabilisation_year = 2040;` These set the target ice extent and stabilization year.

Now, while in the `MPC_for_environmental_management` directory do:

```
IAGP_directory_setup(pwd,'my_first_project')
```

This will make the required sub-directories, then:

```
copyfile([pwd filesep 'spin_up_data.mat'],...
    [pwd filesep, 'my_first_project' filesep 'inputs_and_outputs'])
```

This will move the previously saved spin-up data to where it needs to be. Then:

```
Param = IAGP_model_setup(pwd,'my_first_project');
Data = IAGP_data_setup_ONE_OFF(pwd,'my_first_project')
```

This will ask for confirmation to write data as it could potentially overwrite the data if the project already exists (answer y and return). Then:

```
[Data, Param] = IAGP_batch_process_spin_up_data(pwd,… 'my_first_project', Data, Param);
```

This runs the MPC over the spin up data year by year and finally makes the first SO_2 emissions text file needed to specify SO_2 emissions for your model for 'next' year. The file is called `output_file_<year>` where `<year>` is the simulation year to apply the SO_2. Each of the functions above may take a few seconds to run.

Python/bash script (1)

You now need a Python script that can retrieve the `output_file_<year>` text file from the `MPC_for_Ben/my_first_project/inputs_and_outputs` directory and convert it into the required format needed by your model. The text file is in this format:

```
2017
1,0.000000
2,0.000000
3,0.000000
4,0.000000
5,0.000000
6,0.000000
7,0.000000
8,0.000000
9,0.000000
10,0.000000
11,0.000000
12,0.000000
13,0.000000
14,0.000000
15,0.000000
16,0.000000
17,0.000000
18,0.000000
19,0.000000
20,0.000000
21,0.000000
```

The first line is the year, then 21 week number -- SO_2 emissions pairs. You will need to make from this the input to your model that applies the amount of SO_2 specified in each week (starting Jan 1st). The value is a mass so we assume your model can disperse mass into some layer of the stratosphere etc. (this is what we did with HadGEM2).

You then set your model running for a year. At the end of a year you need:

Python/bash script (2)
This script must be able to extract from your model's last years' run, the 12 values described in the first bullet list and place them into a comma-separated text file called `input_file_<year>` that looks exactly like the example below (except the year and values will change obviously). This one's called `input_file_2025` and it looks like:

```
year,2025
NH_temperature,14.2017
SH_temperature,14.6371
minimum_sea_ice_extent,3.8525
NH_SO2_emission,10.033
SH_SO2_emission,0
f_ghg,3.3135
f_sw_down,341.5
f_sw_up,99.035
net_toa,0.67255
NH_volcano_size,0
SH_volcano_size,0
```

NB. There is no blank line at the beginning and it's important to get the names exactly right.

The `NH_SO2_emission` is the amount of SO_2 that was applied (in Gt).

The Python script must place this file in the `MPC_for_environmental_management /my_first_project/inputs_and_outputs` directory

You then need to set the MPC function running to process this new data and generate next years SO_2 file. To do this run (I'm guessing this will be an automated call):

```
[Data, emis] = IAGPperformMPC(the_year, pwd, 'my_first_project', [], [], [], 1, 1);
```

Make sure it looks exactly like this -- the three empty values are needed to tell the script to load things from memory (you can pass them in when running in test mode), and the two 1's on the end are flags to tell the function to save the state data and make the SO_2 file. You don't need to keep the return values `Data` and `emis`. The function saves all the internal states and files so you can run Matlab like a batch file.

The function should print out results and eventually (20 seconds or so) make the `output_file_<year>` text file for next year. You are then back to using Python script (1).

Repeat this cycle until the end of the experiment.

Comment
I've tested this a million times and it works great --but you know how programming can be-- we may need to work together for a while to iron out any issues. It is also easy to work on more than one project at a time. Just choose a different project name and start from the beginning. Everything will be stored in a different sub-directory. By default, the plotting option is enabled. This writes a postscript file each year. If your machine doesn't like .ps files then change the flag to 0 in line 33 of the IAGP_model_setup function.
All the best and good luck, hope we can get some nice results.
Dave
d.t.leedal@lancaster.ac.uk