Instructions for installing and running the judge

Overview

There is one judge for each arena. The purpose of the judge is to observe the robot in the arena, determine its position, and calculate whether the targets are visited by the robot. It records and displays the elapsed time of the trajectory. Only one judge is required for all the teams at one arena.

The judge requires calibration parameters in its algorithms. There is already a set of default calibration parameters that can be used, but if required, a re-calibration is possible via a graphical interface.

Files required

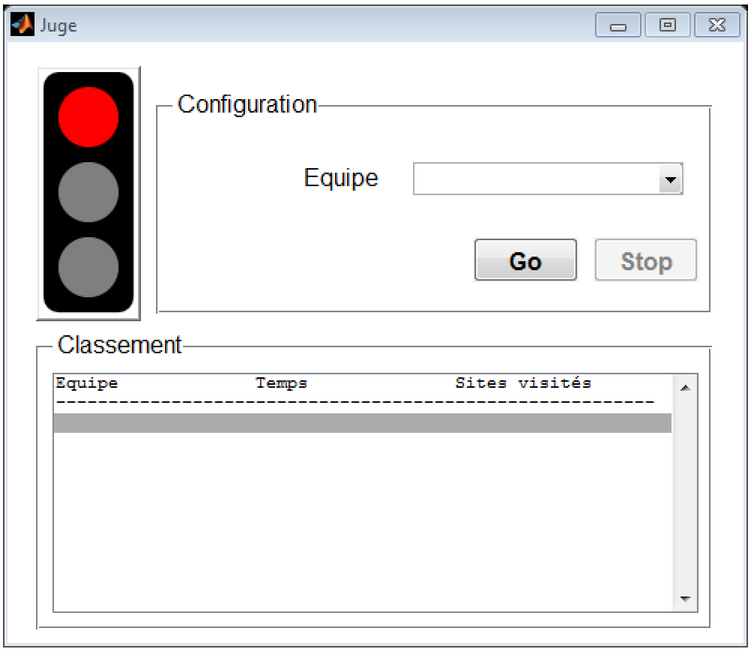
The judge is an application that runs on a PC. To use the judge, you need to get the files from the directory matlab/judge from the repository.

How to run the judge

1. Set up the arena with the robot at the start position and *N* number of targets.

2. Make sure the judge camera is connected to your PC.

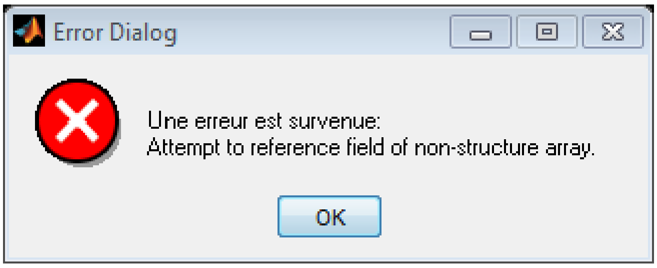
3. Open the script `launch_judge.m`. This script calls the function `judge_control` with the default parameters. `result_file` is the name of file where the results are stored. If this file does not exist, it will be created automatically. `n_sites` is the number of targets (*N*) in the arena. The default `calibration_file` is `judge_thresholds_calibration.mat.` Run the script. If all is in place, you should see this window:



```
Equipe                   Temps                    Sites visités
------------------------------------------------------------
```

Go to Step 3.

However, if you get this error dialog when calling `judge_control`:



Une erreur est survenue:
Attempt to reference field of non-structure array.

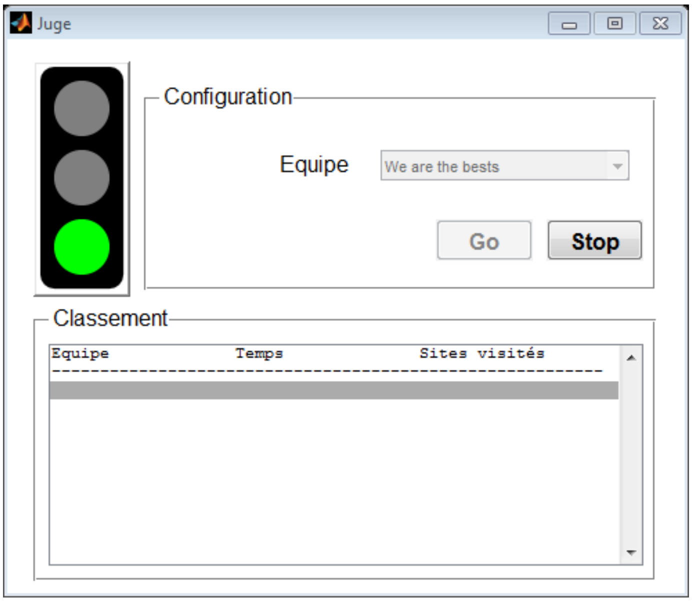
accompanied by an error like this in the command window:

```
Attempt to reference field of non-structure array.
Error in Judge/set.CalibrationFile (line 95)
            set(obj.DetectionObj, 'MatIm2worldBluebox', [tmp.calibration_data.matrices.robot2world], ...
Error in matlab.system.SystemProp/parseInputs (line 875)


Error in matlab.system.SystemProp/sysObjSetProperties (line 294)


Error in Judge (line 72)
            setProperties(obj,length(varargin),varargin{:});
Error in judge_control>judge_control_OpeningFcn (line 99)
    judge = Judge('Nsites',handles.nsites,'CalibrationFile',calibrationFile);
Error in gui_mainfcn (line 220)
    feval(gui_State.gui_OpeningFcn, gui_hFigure, [], guidata(gui_hFigure), varargin{:});
Error in judge_control (line 42)
    gui_mainfcn(gui_State, varargin{:});
```

this means that the variables in your calibration file are empty. In this case, you can generate a default calibration file by following the instructions here. Once the calibration file is generated, run the script. You should then see the “Juge” window.

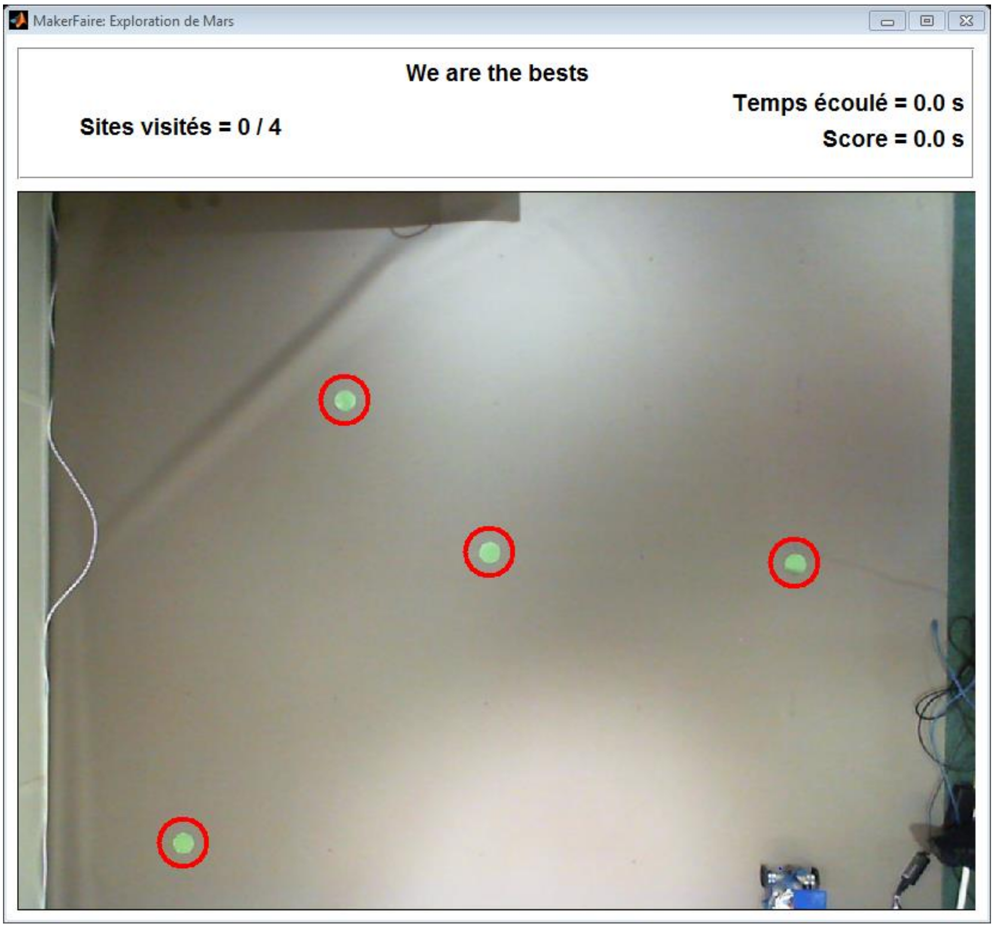
4. Choose your team name from the “Equipe” drop-down bar.

5. Click on “Go”. The traffic light will go from red to yellow initially. If the traffic light then goes to green, all is ready and you should see the following windows:



Equipe: We are the bests

```
Equipe                    Temps                    Sites visités
-------------------------------------------------------------
```



We are the bests

Sites visités = 0 / 4

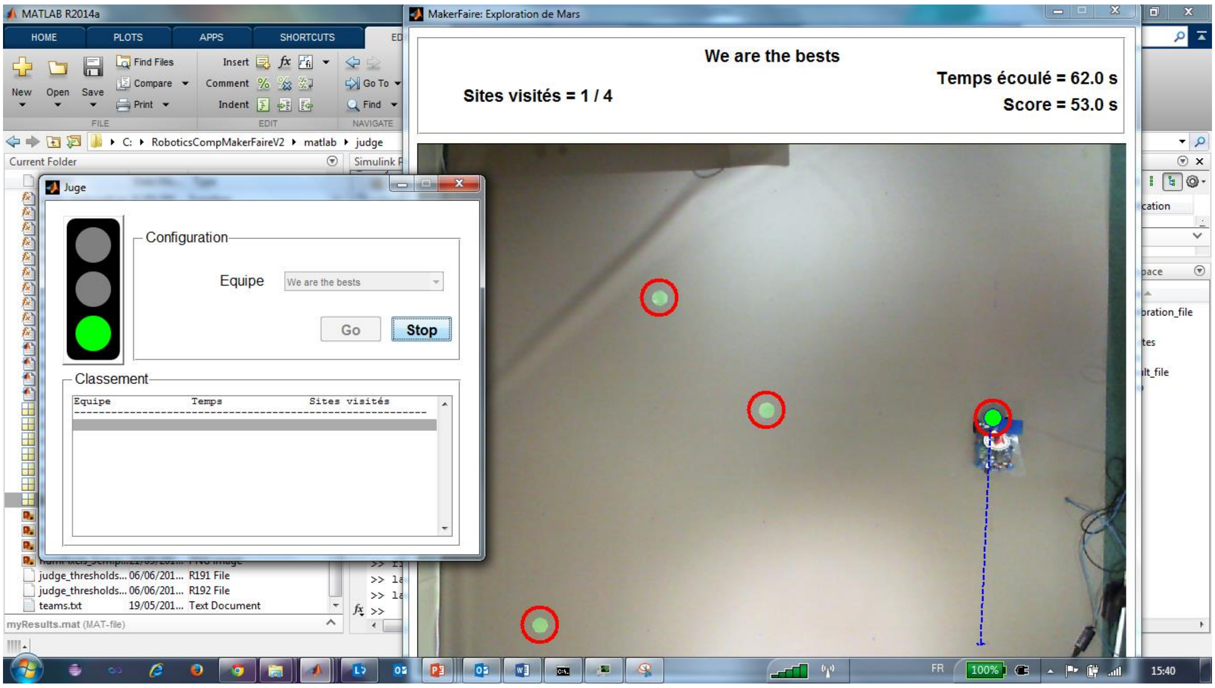
Temps écoulé = 0.0 s

Score = 0.0 s

Go to Step 6.

However, if the traffic light goes back to red, an error message should have been displayed in the Command Window in MATLAB. This should tell you why there is a problem.

6. When the traffic light has gone green, you can start your robot in the arena. As soon as the robot starts to move, its trajectory will be displayed as a blue dotted line, and the counter in the judge UI will increment, indicating the elapsed time. If a site is judged to be visited, the target will turn bright green. The time corresponding to the last target visited will also be displayed:



We are the bests

Sites visités = 1 / 4

Temps écoulé = 62.0 s

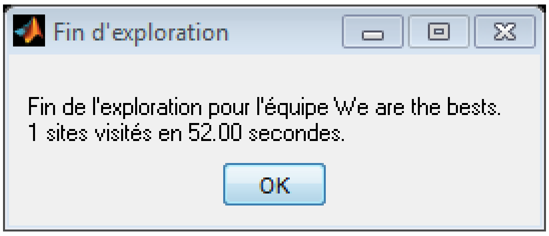
Score = 53.0 s

Equipe: We are the bests

7. The judge will stop when :

    a. All the targets are considered visited.
    b. The robot has finished its trajectory but not all the sites are considered visited. In this case, you can click on “Stop” which will stop the counter. The time recorded for this trajectory corresponds to the time when the last target was visited.
    c. The maximum time allowed has elapsed, i.e. 180s.

    You should see a window like this:



Fin d'exploration

Fin de l'exploration pour l'équipe We are the bests.
1 sites visités en 52.00 secondes.

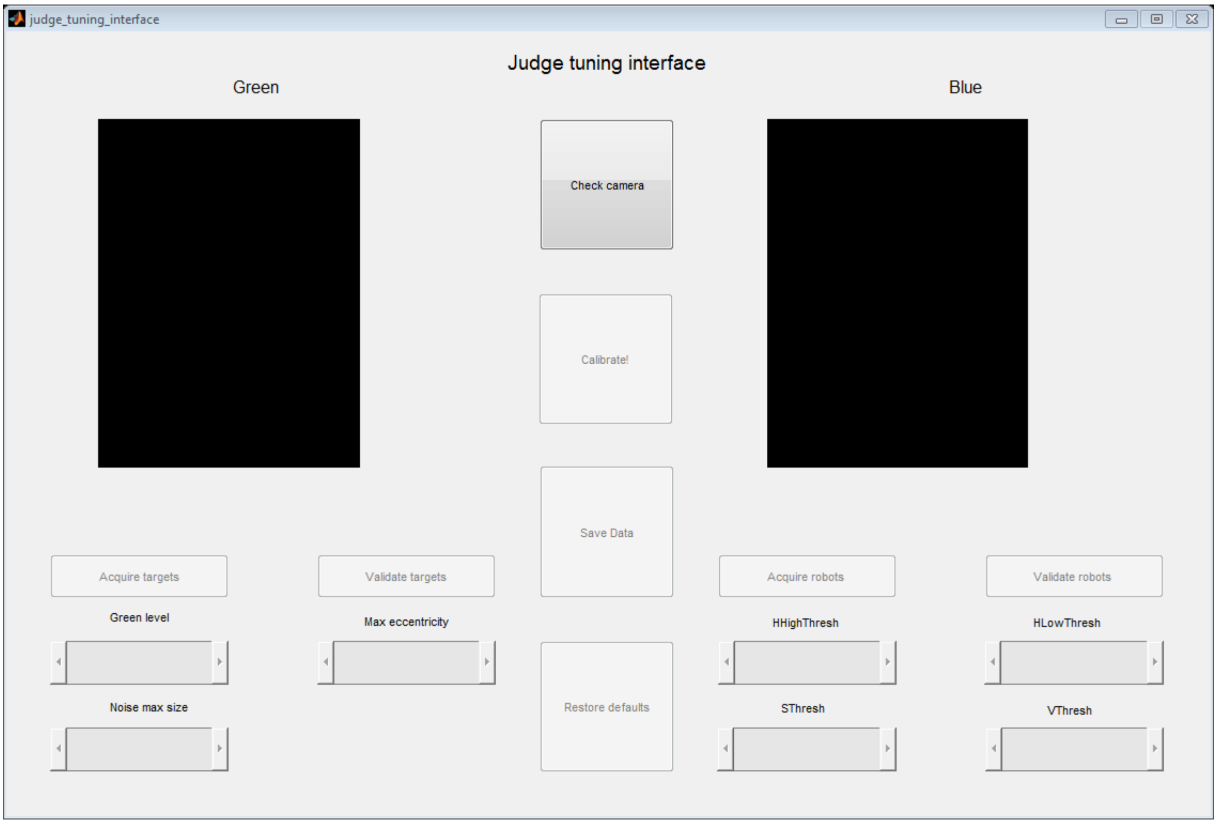
    You will also see the result in the table “Classement” in the judge UI.

8. You can now move onto the next teams by repeating Steps 3-6.

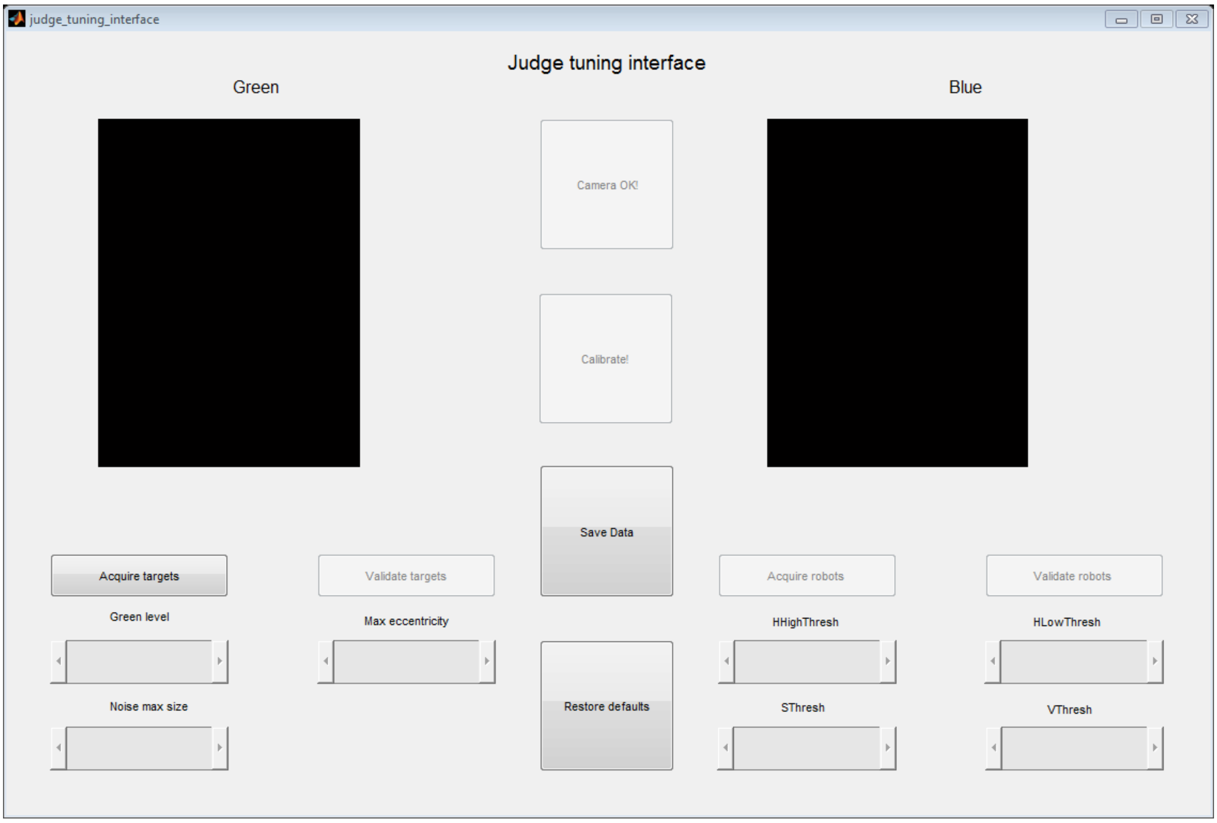
Calibration

Generate a default calibration file

1. Launch `judge_tuning_interface` (from the command window).



Click on “Check camera”. This checks the camera connection. If the connection is OK, the text will change to “Camera OK!”, this button will grey out, and the “Acquire targets” button beneath the “Green” figure will be activated (as shown below). You can go to Step 2.



If this button doesn't grey out, you need to check your camera connection. You can do this by using either the webcam support from MATLAB or from the Image Acquisition Toolbox. Assuming there is only one webcam connected to your PC, using the MATLAB webcam support, do:

```
>> cam = webcam; % to initialize the camera
>> preview(cam); % to get a display of the video stream.
>> closePreview(cam); % to stop and close the display.
>> clear('cam');
```

Using the Image Acquisition Toolbox, do:

```
>> imaqreset;
>> info = imaqhwinfo % this should give you a list of installed
adaptors, including winvideo
>> info = imaqhwinfo('winvideo') % should give you one deviceID
which corresponds to the camera
>> obj = imaq.VideoDevice('winvideo', 1, 'MJPG_800x600'); %
instantiates the video object
>> preview(obj) % to get a display of the video stream
>> closepreview(obj) % to stop and close the display
>> delete(obj) % deletes the video obj
>> clear obj % clears the obj
```

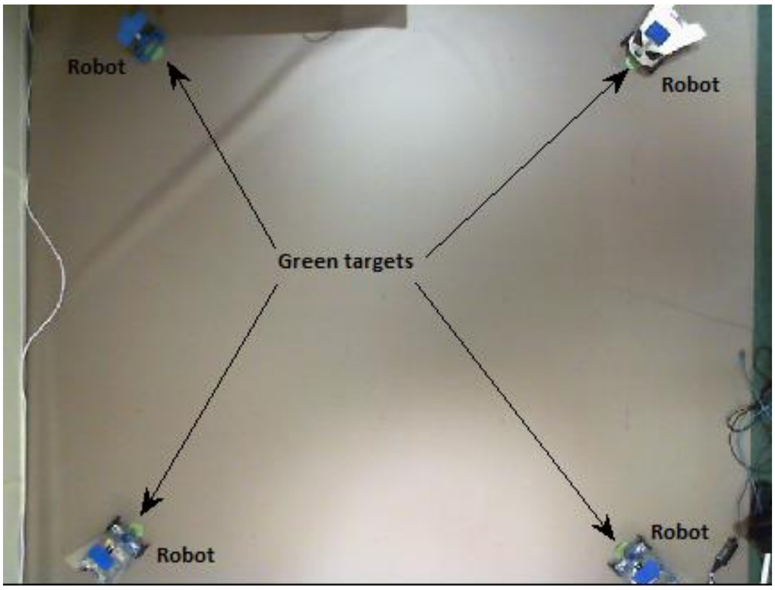
If either of this lets you connect to the camera, you can proceed to Step 2.

2. Click on the button "Restore Defaults".

3. Click on the button "Save Data". This gives to the default parameters in the file `judge_thresholds_calibration.mat`.

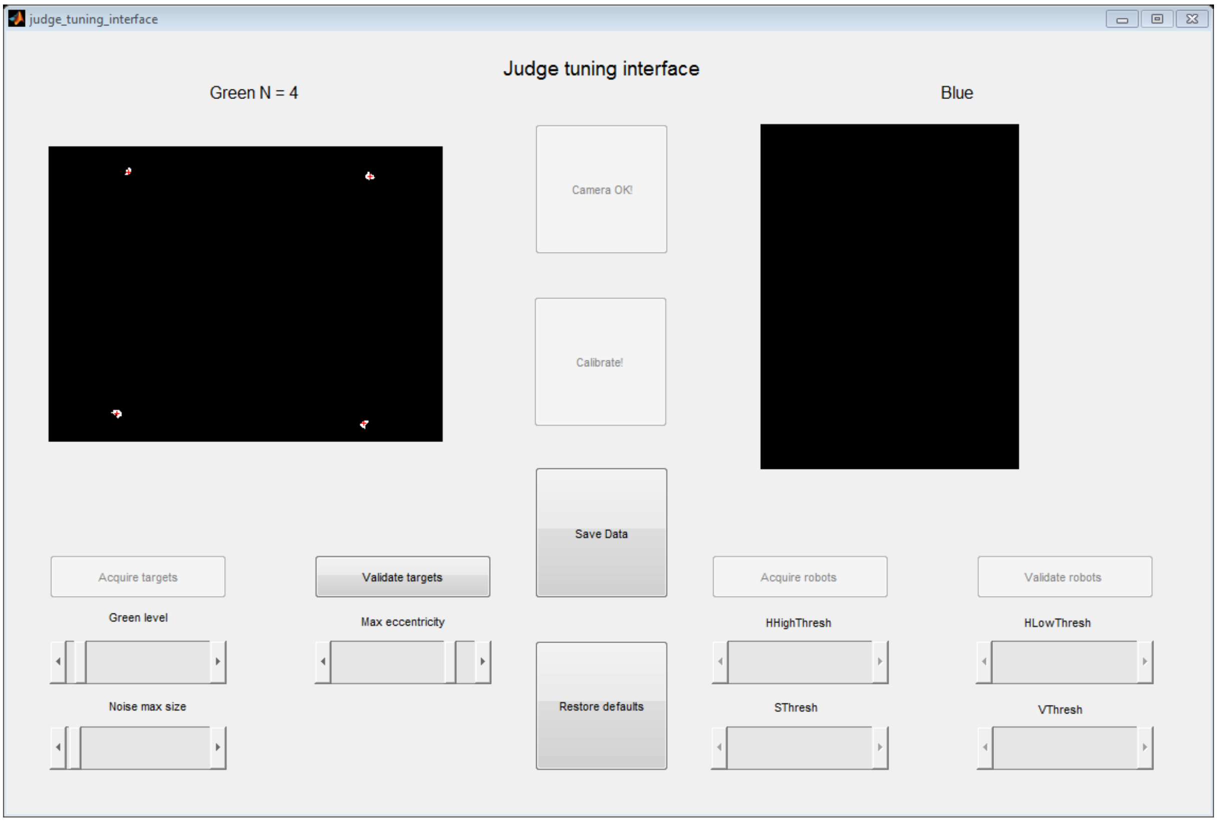
Re-calibrate the judge

1. Place 4 green targets at the four corners of the arena such that they mark out a 2mx2m square. Place 4 robots (with their blue "hats") on the targets such that the blue hats are exactly above the targets. The robots should be facing the centre of the arena. This configuration allows the judge camera to see all the targets and all the robots simultaneously:



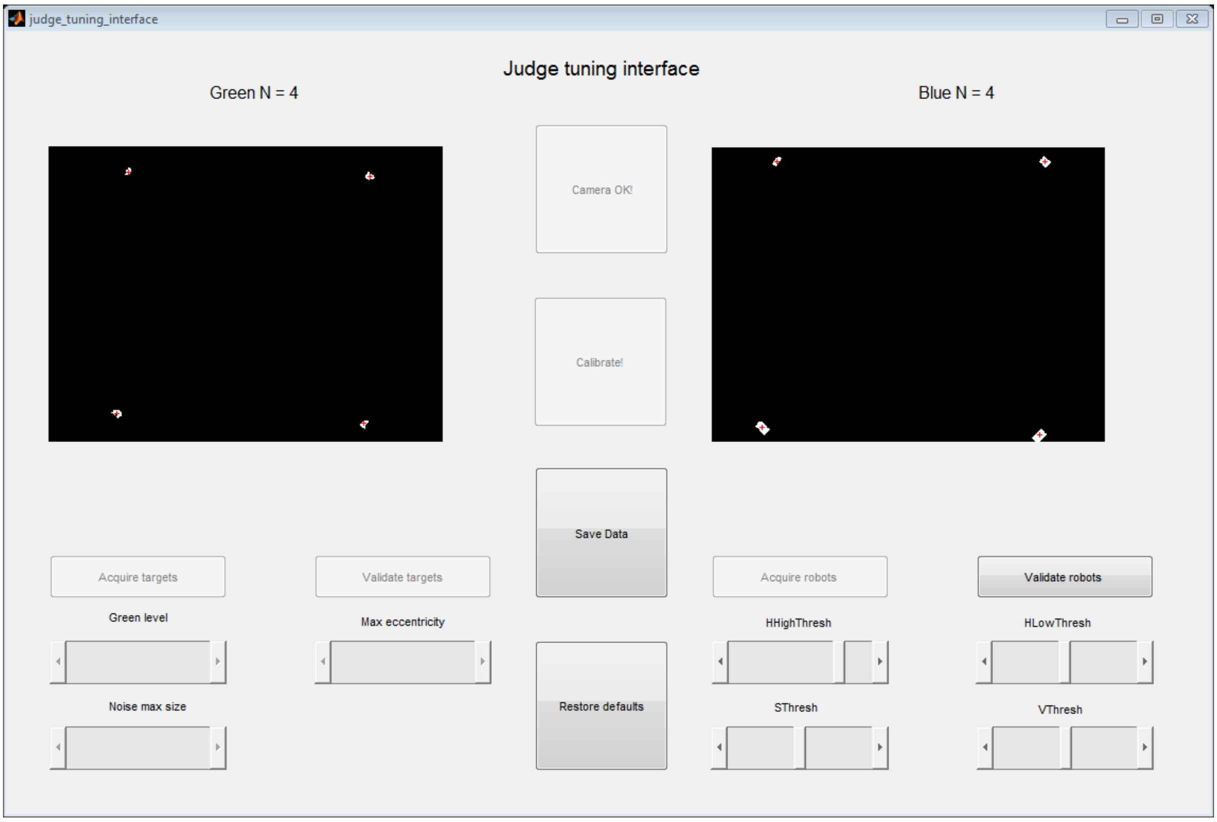
2. Follow Step 1 in this section.

3. Click on the button "Acquire targets". Four and only four green targets should be detected. These are shown as four white blobs on a black background, corresponding to the actual target positions. If required, use the sliders "Green level", "Max eccentricity" and "Noise max size" to change the detection parameters to obtain the biggest possible blobs without any false detections. When the detections are correct, the title should read "Green N = 4".



Once you are satisfied with the detections, click on "Validate targets".

4. Click on the button “Acquire robots”. Four and only four blue robots should be detected. These are shown as four white blobs on a black background, corresponding to the actual robot positions. If required, use the four slider bars below the button to adjust the detection parameters, in order to obtain the largest possible blobs without any false detections. When the detections are correct, the title should read “Blue N = 4”.



Once you are satisfied with the detections, click on “Validate robots”.

5. The “Calibrate” button in the middle should now be active. Click on it to do a geometric calibration of the camera.

6. Once the calibration is finished, click on “Save Data”. This saves all the parameters in the file `judge_thresholds_calibration.mat`.