University of Konstanz
Department of Mathematics and Statistics
Summer Term 2015

Sebastian Gruler
María López Quijorna
Markus Schweighofer

Polynomial Optimization – Computer Project 3

This is a continuation of Computer Project 2. The aim of the project is to observe and to visualize how the accuracy increases while passing from the basic moment relaxation to a higher moment relaxation. Moreover, you will learn how to relax a polynomial equality constraint to a linear vector equality (i.e., a finite system of linear equations). This is just a simple version of the relaxation of a polynomial inequality to a linear matrix inequality that you already know.

Copy the four files you have created in Computer Project 2 inside a new directory named `pop3narendra` where `narendra` must be replaced by your given name in lowercase letters. These files should be expanded and modified during this project as follows:

The MuPAD procedure `robot` should take an additional parameter `k`:

```
robot:=proc(R,k)
```

Consequently, the MATLAB function `robot` should also take an additional parameter $k \in \{0, 1\}$. Calling `robot(R,0)` should produce the basic moment relaxation that has already been implemented in the last project. Calling `robot(R,1)` should produce a higher moment relaxation of the polynomial equalities and inequalities associated to the "robot" R. It is up to you how to define this higher moment relaxation but it should achieve a better approximation quality than the basic relaxation defined last time while it should still terminate after a moderate time. Below we give some suggestions and hints of what to do.

The MATLAB script `robotdemo.m` should again contain examples of how to apply the robot procedure with interesting robots. You should comment on the outcomes and observations in this script file. Your tutor should again have a lot of fun while executing it!

It is perfectly allowed to collaborate with other students. However, the finalization, annotation and submission of the project has to be done by each participant individually. Comments should be concise and in English language.

As for ideas of how to obtain a suitable higher moment relaxation, consider as an example again the robot $R_0 := \left(\begin{smallmatrix} 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{smallmatrix}\right)$ with the associated polynomial constraints

$$x_{3,1}^2 + x_{3,2}^2 = 1,\ 1 \leq (x_{3,1} - 1)^2 + x_{3,2}^2 \leq 3.$$

You could expand the polynomial equation $x_{3,1}^2 + x_{3,2}^2 - 1 = 0$ into a family of polynomial equations parametrized by $a, b, c \in \mathbb{R}$ by multiplying it with $ax_{3,1} + bx_{3,2} + c$:

$$(ax_{3,1} + bx_{3,2} + c)(x_{3,1}^2 + x_{3,2}^2 - 1) = 0 \qquad (a, b, c \in \mathbb{R}).$$

Separating coefficients and monomials, this can be rewritten as the family

$$\begin{pmatrix} a & b & c \end{pmatrix} (x_{3,1}^2 + x_{3,2}^2 - 1) \begin{pmatrix} x_{3,1} \\ x_{3,2} \\ 1 \end{pmatrix} = 0 \qquad (a, b, c \in \mathbb{R})$$

which is of course equivalent to the *polynomial vector equality*

$$(x_{3,1}^2 + x_{3,2}^2 - 1) \begin{pmatrix} x_{3,1} \\ x_{3,2} \\ 1 \end{pmatrix} = 0$$

which can now be linearized or equivalently can be written as a system

$$x_{3,1}^3 + x_{3,1}x_{3,2}^2 - x_{3,1} = 0,\ x_{3,1}^2 x_{3,2} + x_{3,2}^3 - x_{3,2} = 0,\ x_{3,1}^2 + x_{3,2}^2 - 1 = 0$$

of polynomial equalities and then be linearized. Instead of multiplying with $ax_{3,1} + bx_{3,2} + c$ you could of course multiply with the general degree one polynomial in *all* the variables $x_{i,j}$ (not just $x_{3,1}$ and $x_{3,2}$) which will result in a bigger SDP. You could even decide to multiply with the general degree two polynomial in some or all variables or with a "general" polynomial of a special kind which you conjecture to be a reasonable choice.

As for the inequalities, take $(x_{3,1} - 1)^2 + x_{3,2}^2 - 1 \geq 0$ as an example: You could leave it as it is (like in Project 2), multiply it with the square $(ax_{3,1} + bx_{3,2} + c)^2$ of $ax_{3,1} + bx_{3,2} + c$ and then proceed as in the lecture to get an LMI. Again, you could replace $ax_{3,1} + bx_{3,2} + c$ by other choices as above.

In the MATLAB script `robotdemo.m`, you should again define several example robots and illustrate their behavior. Include in your demo a loop where an angle φ runs from 0 to 180° by steps of 5° and optimize the scalar product of the (relaxed) position of the robot's end effector with the unit vector $\begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix}$ (thus simulating a robot's move from the right to the left). In each iteration of the loop draw *two* robots in the same image: Use *red color* for the basic moment relaxation (obtained by setting $k = 0$) and *blue color* for the higher moment relaxation you have implemented (obtained by setting $k = 1$). In this demo, you should comment on what you think about the quality of the relaxations

Due by Friday, July 10th, 2015, 11:11 am. The four files (1)—(4) must be sent attached to an electronic mail to Sebastian Gruler[1], María López Quijorna[2] and Markus Schweighofer[3]. The best solution will be rewarded by a glass of wheat beer in the university beer garden.

[1] `http://www.math.uni-konstanz.de/~gruler/`

[2] `http://www.math.uni-konstanz.de/~lopez/`

[3] `http://www.math.uni-konstanz.de/~schweigh/`