

Explicitly Parameterized Solutions of Parametric Cone Programs

June 3, 2014

1 Parametric Programming

1.1 Parametric Cone Programs

Parametric programming considers optimization problems of which the data are affected by one or more parameters, and analyzes the solution of the problem in function of the parameters. Let us consider the following parametric cone program:

$$\begin{aligned} \Pi(\theta) : \quad & \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad h(\theta)^\top x \\ & \text{subject to} \quad F(\theta)x + g(\theta) \preceq_K 0, \end{aligned} \tag{1}$$

where the problem data depend on parameters $\theta \in \mathbb{R}^t$. For the proper cone K , (combinations of) the nonnegative orthant, the second-order cone and the positive semidefinite cone are considered. The set of all parameter values for which $\Pi(\theta)$ is feasible is denoted by Θ_{feas} :

$$\Theta_{\text{feas}} = \{\theta \in \mathbb{R}^t : \exists x, F(\theta)x + g(\theta) \preceq_K 0\}.$$

The optimal value function $p^* : \Theta_{\text{feas}} \rightarrow \mathbb{R}$ associates with every θ the optimal value of $\Pi(\theta)$, and the corresponding optimal set is given by the optimal set mapping $X^* : \Theta_{\text{feas}} \rightarrow 2^{\mathbb{R}^n}$. More generally, we will consider ϵ -suboptimal set mappings $X_\epsilon^* : \Theta_{\text{feas}} \rightarrow 2^{\mathbb{R}^n}$ which give the ϵ -suboptimal set of $\Pi(\theta)$ as a function of θ :

$$X_\epsilon^*(\theta) = \{x : F(\theta)x + g(\theta) \preceq_K 0, h(\theta)^\top x \leq p^*(\theta) + \epsilon\}.$$

An optimizer function $x^* : \Theta_{\text{feas}} \rightarrow \mathbb{R}^n$ associates to each parameter $\theta \in \Theta_{\text{feas}}$ an optimizer $x^*(\theta) \in X^*(\theta)$.

Relying of the fact that the considered cones K are self-dual, the dual cone program of $\Pi(\theta)$ amounts to

$$\begin{aligned} \Delta(\theta) : \quad & \underset{y}{\text{maximize}} \quad g(\theta)^\top y \\ & \text{subject to} \quad F(\theta)^\top y + h(\theta) = 0 \\ & \quad \quad \quad y \succeq_{K^*} 0. \end{aligned} \tag{2}$$

Its optimal value function is denoted d^* , its optimal set mapping Y^* and its ϵ -suboptimal set mappings Y_ϵ^* .

< GP: Above, Θ_{feas} is purely linked to the feasibility of the primal while feasibility of the dual also matters.

Notation: Θ_{pf} , Θ_{df} ? >

< GP: Should we mention strict feasibility here? Notation: Θ_{psf} , Θ_{dsf} ? >

Assumptions

Below, we present an approach for computing approximate solutions of the considered parametric cone programs on a subset Θ of Θ_{feas} . Our approach relies on the following assumptions:

A1. The problem data depend polynomially on θ . That is, $h(\theta)$, $F(\theta)$ and $g(\theta)$ are polynomial functions.

< GP: This dependency may be considered piecewise polynomial, but do we have applications for this? >

A2. The considered set Θ is a hyperrectangle in \mathbb{R}^t . *< GP: In a second step we may extend hyperrectangles to general polyhedra. >*

A3. For all $\theta \in \Theta$ both the primal program $\Pi(\theta)$ and the dual program $\Delta(\theta)$ are strictly feasible.

The last assumption guarantees that for all $\theta \in \Theta$

$$-\infty < p^*(\theta) = d^*(\theta) < \infty,$$

and that both $p^*(\theta)$ and $d^*(\theta)$ are attained (i.e. both $X^*(\theta)$ and $Y^*(\theta)$ are non-empty).

< GP: We mainly need these last conditions, so maybe state them as assumption 3, and state below that this can be achieved using Slater's constraint qualification? >

1.2 Literature Survey

Research on parametric programming traces back to the 50s [9,11], for as soon as people in operations research were enabled to solve real decision problems by Dantzig's simplex method, they realized the dependency of the outcome on the numerical problem data. Contrary to sensitivity analysis, which investigates the effect of small data perturbations, parametric programming analyzes the solution of the optimization problem for the full range of parameters affecting the numerical data. In doing so, researchers resort to so-called critical regions: sets of parameter values for which the solution of the optimization problem features similar properties (e.g. a specific active set for LPs). Hence, parametric programming concerns characterizing and analyzing the critical regions, as well as the solution (i.e. the optimal value function and the optimal set mapping) both within and across critical regions. Apart from few exceptions, neither the critical regions nor the solution have an analytical expression, which renders the analyses technical and abstract. The exceptions most studied in the literature are parametric LPs of which either the right-hand side of the constraints or the linear objective depends affinely on the parameters [6,7]. Early on it was realized that the critical regions are convex polyhedra and that within a region the solution depends affinely on the parameters. Hence both the optimal value function and the optimizer function are piecewise affine. By the 70s a complete explicit description of the case with multiple parameters was derived [8] and active research on reducing the complexity of computing the solution and on dealing with degeneracies (i.e. allowing non-singleton optimal sets for certain parameter values) continued [3] *< GP: add more references from [1] >*. If both the linear objective and the right-hand side of the constraints depend affinely on the parameters, the critical regions remain convex polyhedra, the optimizer function piecewise affine, yet the optimal value function becomes piecewise quadratic.

In the beginning of the 20th *< GH: 21st century? >* century parametric programming was boosted by the advent of explicit model predictive control (MPC) strategies [1]. MPC is a control strategy that solves an optimization problem at every time sample to compute the next control action. These on line optimization problems only differ in the current sensor measurements, which affect the problem data. Explicit MPC solves the corresponding parametric program off line, yielding the optimal controls as an explicit functions of the sensor measurements. The parametric program is a QP where the parameters affect the linear part of the objective function as well as the right hand side of the linear constraints. As shown in [2], the critical regions are convex *< GP: ?? >* polyhedra, the optimizer function is piecewise affine and the optimal value function is continuous and piecewise quadratic. These initial results sparked wide interest in parametric QPs, with particular focus on efficient computation of the parametric solution and dealing with degeneracies. *< GP: add more references from [1] >*

Although the parametric programming problems mentioned above do admit an explicit characterization of the solution, in the worst case the number of critical regions increases exponentially with the number of constraints. This, for instance, inhibits the wide application of explicit MPC where the number of constraints is determined by the number of states, inputs and the prediction horizon. Therefore, recent research has turned to approximate solutions of parametric problems, primarily with a view to MPC applications. In [5] an approximate solution for parametric LPs with parameter dependent right hand sides is presented. The parameter domain is split up into simplices, and for parameter values within a simplex an upper and lower

bound on the optimal value function is constructed. Simplices are split up into smaller ones until the required accuracy is reached. [2] extended this procedure to general convex parametric programs, where the objective and constraint functions are jointly convex in the optimization variables and the parameters.

Recently, [10] presented an alternative approach to construct an approximate solution of the parametric QPs emerging in MPC. The optimizer function is explicitly parameterized as a polynomial function of the parameters. The coefficients are computed to guarantee feasibility for all parameter values and to minimize either the ℓ_∞ or the ℓ_1 norm of the resulting optimal value function approximation. The semi-infinite constraints emerging in this formulation are transformed to finitely many LMIs using sum of squares.

1.3 Explicitly Parameterized Solutions

In this report we adopt an approach similar to [10]. We construct an approximate primal optimizer function $\hat{x}(\theta)$ and dual optimizer function $\hat{y}(\theta)$ within *a priori* chosen sets of basis functions $b_{x,i}(\theta)$ and $b_{y,i}(\theta)$ on Θ :

$$\hat{x}(\theta, C_x) = C_x b_x(\theta), \quad (3a)$$

$$\hat{y}(\theta, C_y) = C_y b_y(\theta), \quad (3b)$$

with $b_x \theta = (b_{x,i}(\theta))$ and $b_y \theta = (b_{y,i}(\theta))$. *< GP: Can we talk about basis functions? A basis for what type of functions? (polynomials: only a basis for analytical functions and hence, will generally fail to capture the true optimizer functions, generally only C^1 , even if we let the number of basis functions go to ∞). Particularly for the multi-parametric case, how to choose the basis functions such that asymptotically we will include the true optimizer functions? >* The approximate optimizer functions (3), give rise to approximate optimal value functions of the following form:

$$\hat{p}(\theta, C_x) = h(\theta)^\top C_x b_x(\theta),$$

$$\hat{d}(\theta, C_y) = g(\theta)^\top C_y b_y(\theta).$$

In computing the coefficients C_x , we impose $\hat{x}(\theta)$ to be a feasible point of $\Pi(\theta)$ for all $\theta \in \Theta$. Hence, we enforce that for all $\theta \in \Theta$, there exists $\epsilon(\theta) < \infty$ such that

$$\hat{x}(\theta, C_x) \in X_{\epsilon(\theta)}^*(\theta).$$

To obtain the best possible approximation, we minimize $\int_\Theta \epsilon(\theta) d\theta$ while respecting the containment constraint above for all $\theta \in \Theta$. Since

$$\int_\Theta \epsilon(\theta) d\theta = \int_\Theta \hat{p}(\theta, C_x) - p^*(\theta) d\theta = \int_\Theta \hat{p}(\theta, C_x) d\theta - \int_\Theta p^*(\theta) d\theta$$

we end up with the following semi-infinite cone program

$$\begin{aligned} \hat{\Pi} : \quad & \underset{C_x}{\text{minimize}} \quad \int_\Theta h(\theta)^\top C_x b_x(\theta) d\theta \\ & \text{subject to} \quad F(\theta) C_x b_x(\theta) + g(\theta) \preceq_K 0, \quad \forall \theta \in \Theta. \end{aligned} \quad (4)$$

Similarly for the dual:

$$\begin{aligned} \hat{\Delta} : \quad & \underset{C_y}{\text{maximize}} \quad \int_\Theta g(\theta)^\top C_y b_y(\theta) d\theta \\ & \text{subject to} \quad F(\theta)^\top C_y b_y(\theta) + h(\theta) = 0, \quad \forall \theta \in \Theta \\ & \quad \quad \quad C_y b_y(\theta) \succeq_K 0, \quad \forall \theta \in \Theta. \end{aligned} \quad (5)$$

Let C_x^* and C_y^* denote optimal points of $\hat{\Pi}$ respectively $\hat{\Delta}$ then

$$\hat{d}(\theta, C_y^*) \leq d^*(\theta) \leq p^*(\theta) \leq \hat{p}(\theta, C_x^*), \quad \forall \theta \in \Theta,$$

and within the chosen parameterization, the average gap $\hat{p}(\theta, C_x^*) - \hat{d}(\theta, C_y^*)$ over Θ is minimal.

To add: Conversion to tractable optimization problem by using B-splines as basis functions: semi-inf. equality constraints by equalizing B-spline coefficients / semi-inf. inequality constraints by enforcing inequality on the coefficients.

In that last step we rely on properties of the B-spline bases functions (positivity and partition of unity):

$$b(\theta) \geq 0, \forall \theta \in \Theta \quad \mathbf{1}^\top b \equiv 1.$$

These properties imply at each value θ , $\hat{x}(\theta)$ and $\hat{y}(\theta)$ are a convex combination of their coefficients. *< GP: Technically, we only need the Bspline positivity for convex cone programs. >*

To include somewhere: Reasons for choosing a B-spline parameterization

- Great approximating power. Whereas polynomials form a basis for analytical functions, in most cases neither the optimizer function nor the optimal value function of the considered parametric programs is analytical. *< GP: From the viewpoint of approximation the true solution, a simplicial subdivision of the parameter domain would be more appropriate than a hyperrectangular one ... >*
- Transformation of semi-infinite constraints to a finite dimensional set of constraints. Generally good trade-off between conservatism and computational complexity (in contrast to sum-of-squares, particularly for matrix constraints), and 2 ways to reduce conservatism: knot insertion and degree augmentation. *< GP: We should include some guidelines for choosing between these two and on where to insert knots. >*
- It allows for various extensions:
 - The polynomial parameter dependency can be extended to piecewise polynomial. The only complication is that there will be more constraints in the tractable reformulation.
 - The problem formulation can be further generalized to include some general convex functions. For instance, if $h(x, \theta)$ is convex in x for every $\theta \in \Theta$, then

$$\int_{\Theta} h(\hat{x}(\theta), \theta) d\theta$$

is convex in the coefficients C_x . Hence, general convex objective functions are readily allowed. In addition, if $g(x)$ is convex, then with a Bspline parameterization of $x(\theta)$:

$$g\left(\sum_{i=1}^m C_{x,i} b_{x,i}(\theta)\right) \leq \sum_{i=1}^m g(C_{x,i}) b_{x,i}(\theta) \quad \forall \theta \in \Theta.$$

Hence,

$$g(C_{x,i}) \leq 0, \forall i = 1, \dots, m \quad \Rightarrow \quad g(x(\theta)) \leq 0, \forall \theta \in \Theta.$$

1.4 Discussion / Notes / Thoughts

- GP: Should we adopt a more general problem formulation, where we replace the constraint in (1) by

$$\mathcal{F}(x, \theta) + F_0(\theta) \preceq_K 0$$

with $\mathcal{F}(x, \theta)$ a mapping linear in x and polynomial in θ ? The dual problem then involves $\mathcal{F}^{adj}(y, \theta)$. Also, it might be more appropriate to write the explicit parameterizations in the sum form:

$$\hat{x}(\theta) = \sum_{i=1}^{N_x} C_{x,i} b_{x,i}(\theta).$$

This would be more appropriate when dealing with matrices...

- GP: The parameter dependencies $h(\theta)$, $F(\theta)$ and $g(\theta)$ can be extended to be piece-wise polynomial. What is the effect on the overall complexity (e.g. generally more knots and consequently, more constraints at the end) ???
- GP: I'm rather sure that the problem formulation can be extended to rational parameter dependencies $h(\theta)$, $F(\theta)$ and $g(\theta)$. Ways to convert the semi-infinite constraints to a finite number of generalized inequalities: nurbs, S-procedure, descriptor forms ...
- GP: I'm rather sure that, when considering rational parameter dependencies of the data, the solutions $\hat{x}(\theta)$ and $\hat{y}(\theta)$ should still be parameterized as a linear combination of basis functions. Hence, no general rational functions with a free numerator and a free denominator. What would be appropriate rational basis functions ???
- GP: Which forms of Θ are allowed (polyhedra, semi-algebraic sets ...) and how to derive such inner approximations of the set of parameter values for which $\Pi(\theta)$ is strictly feasible? First idea: solve the parametric phase-1 type program:

$$\begin{aligned} & \underset{x,t}{\text{minimize}} \quad t \\ & \text{subject to} \quad F(\theta)x + g(\theta) \preceq_K t\mathbf{1} \end{aligned}$$

2 Applications

- explicit MPC
- time-optimal point-to-point motion
- trade-off curves
- combined structure control design
 - Similar to LPV control for static parameter, but with different objective (ℓ_1 instead of ℓ_∞).
 - What is the effect of the choice of the objective function ??? Also, what is the value of the ℓ_1 objective for varying parameters (suppose for instance that the worst case ℓ_2 gain gets only slightly larger, but better performance over large parts of the parameter domain)???

2.1 Case study: combined structure control design for earthquake isolation

Consider the 3-DOF model of [4], that should be prevented against earthquakes. In this case study, the objective is to design a state-feedback controller with a good tradeoff between disturbance rejection and actuator effort, while optimizing a time-invariant structural parameter. Viewing this problem as an LPV state-feedback synthesis problem with time-invariant parameter, solving only one LMI optimization problem results in accurate approximations of the performance as a function of the structural parameter. In this way, a (nearly) optimal structural parameter can be efficiently selected.

The structural parameter is selected as spring coefficient k_2 , see [4]. Assuming that $k_2 \in [\underline{k}_2, \bar{k}_2]$, and defining $\theta := (k_2 - \underline{k}_2)/(\bar{k}_2 - \underline{k}_2)$, the system model is expressed in state-space form as

$$\begin{cases} \dot{x}(t) &= A(\theta)x(t) + B_u u(t) + B_w w(t), \\ z(t) &= C_z x(t), \end{cases}$$

where $A(\theta) \in \mathbb{R}^{6 \times 6}$ has an affine dependency on $\theta \in [0, 1]$, and $B_u \in \mathbb{R}^{6 \times 3}$, $B_w \in \mathbb{R}^{6 \times 1}$ and $C_z \in \mathbb{R}^{3 \times 6}$ are constant matrices. A state-feedback controller $K(\theta) \in \mathbb{R}^{3 \times 6}$ is to be designed, such that the closed-loop system

$$\begin{cases} \dot{x}(t) &= (A(\theta) + B_u K(\theta))x(t) + B_w w(t), \\ z(t) &= C_z x(t), \\ u(t) &= K(\theta)x(t), \end{cases}$$

meets a \mathcal{H}_2 performance from w to z and from w to u for each parameter value $\theta \in [0, 1]$.

We are interested in optimizing γ as a function of parameter θ (instead of optimizing worst-case performance). This is achieved by solving the convex optimization problem

$$\begin{aligned} & \text{minimize} && \int_0^1 \text{Trace}\{Z(\theta)\}d\theta, \\ & \text{subject to} && \begin{bmatrix} Q(\theta)A(\theta)^T + A(\theta)Q(\theta) + B_uL(\theta) + L(\theta)^TB_u^T & B_w \\ B_w^T & -I \end{bmatrix} \prec 0, \\ & && C_zQ(\theta)C_z^T - \mu I \prec 0, \\ & && \begin{bmatrix} Q(\theta) & L(\theta)^T \\ L(\theta) & Z(\theta) \end{bmatrix} \succ 0. \end{aligned}$$

To make this optimization problem numerically tractable, it is necessary to impose a specific parameterization on the LMI variables, and subsequently use some relaxation technique to derive a finite set of sufficient LMIs.

The LMI variables are parameterized as polynomial splines of some preselected degree, and can thus be expressed in terms of Bspline basis functions and coefficients. For example, the variable $Q(\theta)$ is parameterized as

$$Q(\theta) = \sum_{i=1}^m C_{Q,i} b_{Q,i}(\theta),$$

where $b_{Q,i}$ and $C_{Q,i}$, $i = 1, \dots, m$ are Bspline basis functions and constant matrix coefficients, respectively. A polynomial parameterization results as a special case.

Solving the resulting optimization problem yields an upper bound $\hat{p}(\theta)$ on the optimal value function $p^*(\theta)$, while solving the associated dual problem

$$\begin{aligned} & \text{maximize} && \int_0^1 \text{Trace}\{U_{12}(\theta)B_w^T\} + \text{Trace}\{U_{12}(\theta)^TB_w\} - \text{Trace}\{U_{22}(\theta)\} - \mu \text{Trace}\{V(\theta)\}d\theta, \\ & \text{subject to} && U(\theta) := \begin{bmatrix} U_{11}(\theta) & U_{12}(\theta) \\ U_{12}(\theta)^T & U_{22}(\theta) \end{bmatrix} \succ 0, \\ & && V(\theta) \succ 0, \\ & && \begin{bmatrix} A(\theta)^TU_{11}(\theta) + U_{11}(\theta)A(\theta) + C_z^TV(\theta)C_z & U_{11}(\theta)B_u \\ B_u^TU_{11}(\theta) & I \end{bmatrix} \succ 0. \end{aligned}$$

provides a lower bound $\hat{d}(\theta)$:

$$\hat{d}(\theta) \leq d^*(\theta) \leq p^*(\theta) \leq \hat{p}(\theta), \quad \theta \in [0, 1].$$

The optimal \mathcal{H}_2 performance is obtained for fixed values of θ by gridding the parameter space using 100 equidistant points on the interval $[\underline{k}_2, \bar{k}_2] = [14546500, 58186000]$. The resulting tradeoff curve is shown in Figure 1, revealing an optimal value $k_2 \approx 4.9 \cdot 10^7$. The computation of this tradeoff curve requires solving 100 convex optimization problems. On the other hand, parametric programming allows the computation of an upper and a lower bound on the exact solution by only solving one convex optimization problem.

To assess the benefit of parametric programming for this example, we start by assuming a polynomial dependency of all the LMI variables on the parameter θ . See Table 1 for the selected polynomial degrees. We take $q = 2$. Subsequently, the application of Polya relaxations is compared to knot insertion, of which the results are given in Table 2. It is remarked that application of a Polya relaxation of degree k yields the same increase in numerical complexity as the insertion of k knots, which is confirmed by the computation times. In addition, note that both the upper bounds and the lower bounds resulting from knot insertion are tighter for all cases.

To reduce conservatism, a first possibility is to increase the polynomial degree of the LMI variables. Another option is to assume the LMI variables to be polynomial splines, which boils down to increasing the number of knots instead of the polynomial degree of these variables. A comparison is shown in Table 3.

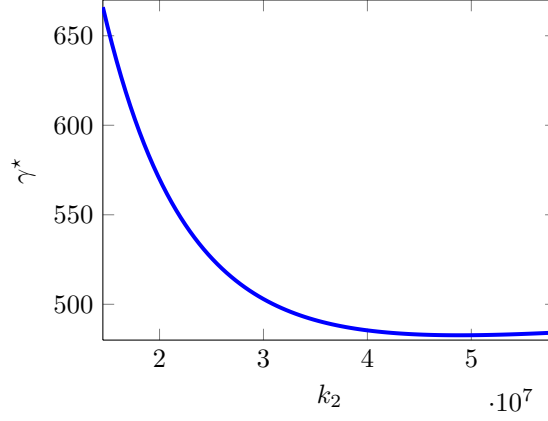


Figure 1: The optimal \mathcal{H}_2 performance as a function of k_2 .

Table 1: Selected polynomial degrees of the LMI variables.

LMI variable	degree
$Q(\theta), U(\theta), V(\theta)$	q
$F(\theta), L(\theta)$	$q + 1$

Table 2: Polya vs. knot insertion for polynomial LMI variables with $q = 2$.

degree Polya relaxation	0	1	2	4	8
primal objective	560.8	554.2	543.6	537.2	532.8
computation time [sec]	2.12	0.80	0.98	1.05	1.44
dual objective	445.9	451.4	462.4	468.6	472.3
computation time [sec]	1.28	0.69	0.76	0.84	1.10
inserted knots	0	1	2	4	8
primal objective	560.8	550.3	532.2	529.2	528.0
computation time [sec]	0.77	0.82	0.87	1.05	1.46
dual objective	445.9	467.8	472.0	474.8	475.8
computation time [sec]	1.23	0.70	0.79	0.88	1.06

Table 3: Higher polynomial degree vs. more knots of LMI variables (no Polya relaxation or knot insertion)

Polynomial degree q (0 knots)	2	3	4	5
primal objective	560.8	536.2	530.2	526.5
computation time [sec]	2.24	0.95	0.94	1.00
dual objective	445.9	481.1	495.4	501.0
computation time [sec]	1.31	0.78	0.84	1.02
number of knots ($q = 2$)	0	1	2	3
primal objective	560.8	532.0	522.6	518.4
computation time [sec]	2.13	0.88	0.96	1.13
dual objective	445.9	487.2	497.7	503.5
computation time [sec]	1.34	0.83	0.93	1.06

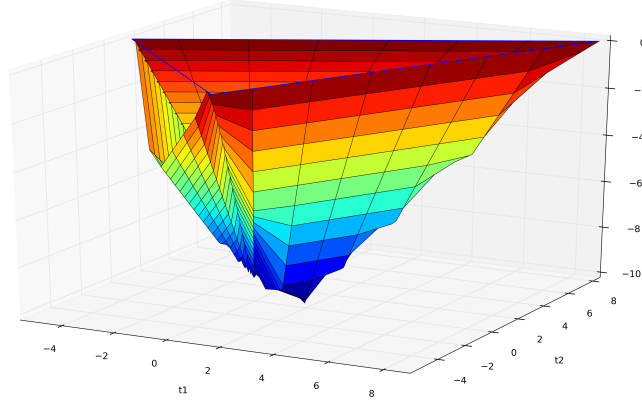


Figure 2: The approximated objective function for a degree 1 tensor product spline with 21 equidistantly spaced knots in α and 51 in β .

2.2 Case study: A multiparametric lp with polyhedral parameter set

Consider the following multiparametric LP reported in [6] and subsequently treated in [3]:

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^2}{\text{minimize}} && h(\theta)^\top - 2x_1 - x_2 \\
 & \text{subject to} && x_1 + 3x_2 \leq 9 - 2\theta_1 + \theta_2 \\
 & && 2x_1 + x_2 \leq 8 + \theta_1 - 2\theta_2 \\
 & && x_1 \leq 4 + \theta_1 + \theta_2 \\
 & && x_1, x_2 \geq 0,
 \end{aligned}$$

on $\Theta \in [-10, 10] \times [-10, 10]$.

It can be shown that for this problem Θ_{feas} is a triangular set. Hence, the optimization variables cannot be parameterized by a tensor spline in θ_1 and θ_2 as this would render the problem infeasible. Instead, a parameterization is chosen that maps a rectangular domain to a polyhedral set.

Consider a spline parameterization $(p_{\theta_1}(\alpha), p_{\theta_2}(\alpha))$ of the boundary of Θ_{feas} in terms of a scalar parameter $\alpha \in [0, 1]$ *< WVL: What about higher dimensions? Is it straightforward to parameterize such polyhedral sets? >*. To ‘fill’ the polydral set an additional variable $\beta \in [0, 1]$ is introduced. By defining $\theta_1(\alpha, \beta) = p_{\theta_1}(\alpha)\beta$ and $\theta_2(\alpha, \beta) = p_{\theta_2}(\alpha)\beta$, any (α, β) now maps to a feasible θ_1, θ_2 . Similarly, x can also be parameterized as a tensor spline in α and β .

Figure 2 shows the approximated objective function for a tensor spline of degree 1. The accordance with the true objective function is good, except near the switch in critical regions, as shown in figure 3.

References

- [1] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In Lalo Magni, Davide Martino Raimondo, and Frank Allgöwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 345–369. Springer Berlin Heidelberg, 2009.

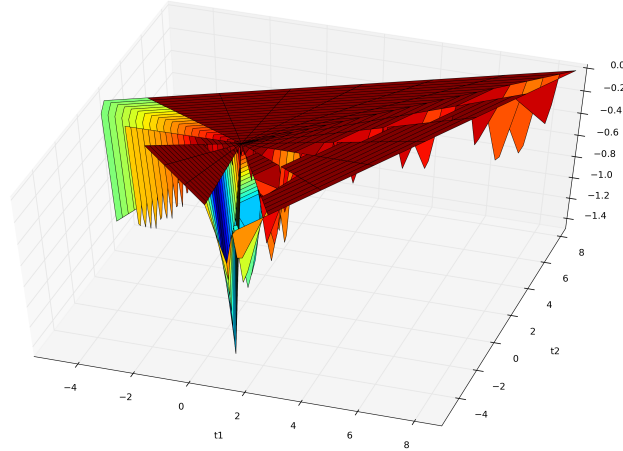


Figure 3: The error between the true value of the objective function and our approximation.

- [2] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002.
- [3] F. Borrelli, A. Bemporad, and M. Morari. Geometric algorithm for multiparametric linear programming. *Journal of Optimization Theory and Applications*, 118(3):515–540, 2003.
- [4] J. F. Camino, M. C. de Oliveira, and R. E. Skelton. ”Convexifying” linear matrix inequality methods for integrating structure and control design. *Journal of Structural Engineering*, 129:978–988, 2003.
- [5] C. Filippi. An algorithm for approximate multiparametric linear programming. *Journal of Optimization Theory and Applications*, 120(1):73–95, 2004.
- [6] T. Gal. *Postoptimal Analyses, Parametric Programming and Related Topics*. McGraw-Hill Inc., 1979.
- [7] T. Gal. Linear parametric programming – a brief survey. In Anthony V. Fiacco, editor, *Sensitivity, Stability and Parametric Analysis*, volume 21 of *Mathematical Programming Studies*, pages 43–68. Springer Berlin Heidelberg, 1984.
- [8] Tomas Gal and Josef Nedoma. Multiparametric linear programming. *Management Science*, 18(7):pp. 406–422, 1972.
- [9] S. I. Gass and Thomas L. Saaty. Parametric objective function (part 2)- generalization. *Journal of the Operations Research Society of America*, 3(4):pp. 395–401, 1955.
- [10] Yasuaki Oishi. Simplified approaches to polynomial design of model predictive controllers. In *Control Applications (CCA), 2013 IEEE International Conference on*, pages 960–965, Aug 2013.
- [11] Thomas Saaty and Saul Gass. Parametric objective function (part 1). *Journal of the Operations Research Society of America*, 2(3):316–319, 1954.