

Modeling and solving uncertain optimization problems in YALMIP

Johan Löfberg

*Department of Electrical Engineering, Linköping University,
Linköping, Sweden, (e-mail: johanl@isy.liu.se)*

Abstract: A considerable amount of optimization problems arising in the control and systems theory field can be seen as special instances of robust optimization. Much of the modeling effort in these cases is spent on converting an uncertain problem to a robust counterpart without uncertainty. Since many of these conversions follow standard procedures, it is amenable to software support. This paper presents the robust optimization framework in the modeling language YALMIP, which carries out the uncertainty elimination automatically, and allows the user to concentrate on the high-level model instead.

Keywords: Uncertainty descriptions; Robust linear matrix inequalities; Convex optimization

1. INTRODUCTION

The basic problem addressed in this paper is uncertain optimization problems.

$$\begin{aligned} \min_x \max_w f(x, w) \\ \text{s.t. } g(x, w) \preceq 0 \quad \forall w \in \mathcal{W} \end{aligned}$$

To keep notation at a minimum, we write $g(x, w) \preceq 0$ here, but keep in mind that these constraints may include elementwise inequalities, equality, integrality, second-order and semidefinite cone constraints.

In short, there are two ways to address this class of problems. Historically, probabilistic approaches where constraints are added by sampling from the uncertainty have been common. The disadvantage with this approach is of course that it only gives an (optimistic) approximation, and thus no guaranteed solutions. It is however possible to obtain statistical confidence results, in particular in the convex case Calafiore and Campi (2005).

Recently, a more strict approach has become popular. The paradigm here is to convert the problem to a certain problem, by in some way removing the uncertainty, using methods such as explicit maximization, duality properties, or conservative relaxation methods. These approaches have to a large extent gained popularity due to developments in the convex and conic optimization field Ben-Tal and Nemirovski (1998); Ben-Tal and Nemirovskii (2002). This paradigm, commonly referred to as a worst-case approach, is what we concentrate on in this paper.

Many robust optimization problems falls into standard cases, which can be converted to certain counterparts by standard but error-prone and cumbersome reformulations. Our goal is to supply an extension to the modeling language YALMIP Löfberg (2004) for modeling robust optimization problems in an intuitive format, and let the software package take care of the reformulations. In other words, it is the robust optimization correspondence of the convex programming reformulations in the "nonlinear

operator"-framework in YALMIP and the similar "disciplined convex programming" framework in CVX Grant et al. (2007). The outcome of the feature presented in this paper is a new optimization problem which solves the worst-case scenario.

$$\begin{aligned} \min_{x,y} \tilde{f}(x, y) \\ \text{s.t. } \tilde{g}(x, y) \leq 0 \end{aligned}$$

Typically, the problem is changed considerably from the original problem, indicated by the new objective, constraints and additional variables, introduced in order to eliminate the uncertainty. While the original uncertain problem, for instance, is a linear program, the robust counterpart can become a semidefinite program. Even worse, the robust counterpart may not even be a tractable problem. In the following sections, we will outline the basic ideas in the proposed robust optimization extension, and introduce the uncertainty scenarios that are supported.

2. NOTATION

Matrices will generically be denoted using capital letters while vectors are in small letters. Inner product $\text{trace}(A^T B)$ will be written as $A \bullet B$. Cone constraints are indicated using the \succeq operator. The cone can be either the positive orthant cone, second order cone, or the cone of semidefinite matrices. To simplify notation, conic constraints in dual form $C - \sum_i^m A_i w_i \succeq 0$ will be written in the operator form $C - \mathcal{A}^T(w) \succeq 0$. The primal form conic constraints, $A_i \bullet X = b_i, i = 1 \dots m$, will be written compactly as $\mathcal{A}(X) = b$. The cones might be direct products of several cones, possibly of different type, and the data is partitioned accordingly. For simplicity though, one may think of all cone constraints as having only one element.

3. GENERAL FRAMEWORK

The first and most important idea in the robust optimization framework in YALMIP is that uncertain mod-

els should be modeled using exactly the same syntax as certain models. Hence, the only addition to the modeling language is a new command `uncertain`, which declares a set of variables as uncertain.

When YALMIP encounters a model with uncertain variables, three main steps occur. To begin with, the variables explicitly declared as uncertain are detected, and constraints involving only these variables are separated from the model. These constraints constitute (the initial) uncertainty description.

YALMIP then applies an expansion of the remaining model, to model all advanced nonlinear operators, such as absolute values and norms, typically using graph representations. Note that the expansion of expressions that only involve uncertain variables might generate new variables and constraints, which have to be added to the list of uncertain variables, and to the uncertainty description. As an example, if the uncertainty description is defined in the modeling language as $|w|_\infty \leq 1$, the set of uncertain variables and associated uncertainty set after the expansion is performed will be $\{(w, t) : -t \leq w \leq t, t \leq 1\}$. The introduction of the auxiliary, and strictly speaking redundant, variable t is a consequence of the way YALMIP models nonlinear operators, such as norms.

A complication in terms of implementation is that the modeling language has to treat the uncertainty as a constant during the convexity propagation and expansion. For instance, the constraint $|xw| \leq 1$ is not convex, and could thus lead to problems when convexity analysis is performed by YALMIP to decide on how to model the absolute value operators. However, the standard graph model $-1 \leq xw \leq 1$ will be derived, since the modeling language temporarily treats w as a constant and thus sees an affine term inside the absolute value operator. After this expansion has been made, the model is conceptually in YALMIP standard form (no high-level operators such as norms, absolute values, etc.), albeit parameterized in the w variables.

At this point, the robustification should take place. Depending on the class of uncertain constraints, and the uncertainty model, different approaches are used. The process of removing the uncertainty and deriving a robust counterpart is called the filtering step, and there are currently five filters implemented, as outlined in the next section.

Once the filter has been applied, a standard YALMIP model with no uncertainty has been generated, and can be solved using any installed solver suitable for the problem or manipulated further by the user.

Important to understand is that the uncertainty modeling and the derivation of robust counterparts is a feature that is (essentially) completely integrated in the infrastructure of YALMIP. Hence, nothing prevents a user to, e.g., define uncertain sum-of-squares problems with combinatorial constraints. The modeling languages performs the sum-of-squares compilation, uncertainty removal, and addition of combinatorial constraints separately in a modular fashion.

4. THE FILTERS

The mechanism of converting a problem with uncertainty, to the corresponding certain counterpart, is called the filtering step in the robust optimization framework in YALMIP.

At the moment, five different filters are implemented. The goal is to extend this list in future versions, but the current set of scenarios are considered the most important in practice.

4.1 Duality filter

The duality filter is applicable to elementwise constraints with coefficients linearly parameterized in the uncertain variable, and the uncertainty constrained to an intersection of linear, second order and semidefinite cones. Consider a single elementwise constraint

$$(Aw + b)^T x + (c^T w + d) \leq 0 \quad \forall w : E - \mathcal{F}^T(w) \succeq 0$$

By writing the left-hand term as $(A^T x + c)^T w + (b^T x + d)$, it follows from duality theory of conic optimization that the maximum of this function, over w , is less than or equal to 0 if and only if¹ there exist a Z such that the following condition holds Ben-Tal and Nemerovskii (2002)

$$E \bullet Z + b^T x + d \leq 0, \mathcal{F}(Z) = A^T x + c, Z \succeq 0$$

This is a very general and useful result. A major drawback is however that the filter can lead to a substantial increase of problem size, since a new variable Z , and the associated constraints, have to be introduced for every uncertain constraint. Hence, when possible, more specialized filters should be used.

Note that the strong duality arguments are employed in the w -space. Hence, additional complicating constraints on x , such as integral variables or other convex or nonconvex constraints, does not influence the correctness of the method. Of course, this holds also for the remaining four filters.

4.2 Enumeration filter

A classical uncertainty case is constraints where the parameterization is linear in the uncertainty, and the uncertainty is constrained to a polytopic set. To simplify notation, let $\mathcal{A}_w(x)$ denote the parameterized operator $\sum A_i(w)x_i$ where each matrix A_i is linearly parameterized in w . Consider the following uncertain conic constraint

$$(\mathcal{C}^T(w) + D) + (\mathcal{B}^T(x) + \mathcal{A}_w^T(x)) \preceq 0 \quad \forall w : Ew \leq f$$

This is the case that arise, e.g., in stability analysis of polytopic systems Boyd et al. (1994), where w corresponds to parameters in an uncertain system and x corresponds to the variables parameterizing a Lyapunov matrix.

From convexity, it follows that it is sufficient to study the vertices of the polytope $Ew \leq f$. Hence, if we let $\{w_i\}$ denote the vertex enumeration of the uncertainty polytope,

¹ Assuming strict complementarity, which in our case means that the uncertainty set has a strict interior.

the robustified constraint is given by the intersection of the conic constraint evaluated at the vertices $\{w_i\}$.

$$(C^T(w_i) + D) + (B^T(x) + A_{w_i}^T(x)) \preceq 0$$

The problem with this approach is obvious; simple polytopes can generate intractably many vertices. As a trivial example, the unit-cube in \mathbb{R}^n has 2^n vertices.

4.3 Explicit maximization

Specializing the problem structure further, we arrive at a case where we actually can perform the maximization over the uncertainty analytically. Consider an elementwise constraint linearly parameterized in an uncertainty which is constrained to a norm-ball.

$$(c^T w + d) + (Aw + b)^T x \leq 0 \quad \forall w : |w|_p \leq 1$$

The maximum over w can be derived by using the fact that $\max_{|w|_p \leq 1} q^T w$ is $|q|_{p^*}$ where $|\cdot|_{p^*}$ denotes the dual norm Boyd and Vandenberghe (2004). Hence, the robustified constraint is

$$(b^T x + d) + |c + A^T x|_{p^*} \leq 0$$

The current implementation only exploits this result for the (possibly scaled and translated) ∞ -norm case, where the dual norm is the 1-norm. Other standard conic-representable cases (1-norm and 2-norm) are dealt with using enumeration and duality filters. More general cases are however expected to be supported in the explicit filter in a future release.

4.4 Pólya filter

The filters above are all exact, in the sense that the robustified constraints are both sufficient and necessary for the original constraints to be robustly satisfied. Unfortunately, there are not many more cases where simple sufficient and necessary counterparts are available Ben-Tal and Nemirovski (1998). Instead, one has to rely on conservative approximations. One common case where a simple conservative result is available is polynomially parameterized elementwise or semidefinite constraints, with the uncertainty constrained to a simplex.

$$p(x, w) \preceq 0 \quad \forall w : \sum_{i=1}^m w_i = 1, w \geq 0$$

Since w is constrained to a simplex, the constraint is trivially equivalent to $p(x, w) (\sum_{i=1}^m w_i)^N \preceq 0$ for arbitrary N . If we assume that $p(x, w)$ is homogeneous² in w , the polynomial $p(x, w) (\sum_{i=1}^m w_i)^N \preceq 0$, when seen as a polynomial in w with coefficients parameterized in x , is non-positive if all coefficients are non-positive. This follows trivially since w is non-negative. Hence, a sufficient condition is

$$\text{coefficients}_w \{ p(x, w) (\sum_{i=1}^m w_i)^N \} \preceq 0$$

² This is not a restriction, since any polynomial can be rendered homogeneous on a simplex by multiplying monomial terms with suitable powers of $\sum_{i=1}^m w_i$.

Note that this trivially also holds for the symmetric matrix polynomial case, i.e., when $p(x, w)$ is a matrix polynomial, with matrix coefficients functions of x .

The reason we denote the filter Pólya, is due to a result of Pólya, stating that for a finite sufficiently large N , the condition is necessary³ Hardy et al. (1952). However, a bound on this sufficiently large N is typically unreasonably large, and depends on the parameterized coefficients, so the necessity result is of no direct use to us. Instead, the user has to specify N and hope that the relaxation is sufficiently tight. Necessity in the matrix case has recently been shown in Scherer (2005).

An important feature of this approach, compared to more advanced schemes using recent developments in convex optimization based relaxations of polynomial problems, such as sum-of-squares and moment relaxations, is that an elementwise constraint leads to elementwise constraints in the robust counterpart. In other words, the problem class does not change. Nevertheless, since YALMIP has support for sum-of-squares reformulations and moment relaxations, future versions may have support for stronger relaxations, at the cost of problem complexity.

4.5 Elimination filter

If everything else fails, our last resort is to constrain the decision variables such that the uncertainties disappear from the constraint. Consider a polynomially parameterized constraint, with arbitrary uncertainty description.

$$p(x, w) \preceq 0 \quad \forall w \in \mathcal{W}$$

If we see this as a polynomial in w with coefficients parameterized in x , a trivial sufficient condition is obtained by constraining the coefficients to be zero.

$$p(x, w) \preceq 0, \text{coefficients}_w \{ p(x, w) \} = 0$$

Although this condition is trivial, it is actually rather useful in some situations, as mentioned in one of the examples in the next section. Additionally, the elimination filter is the only option when an equality constraint involves uncertain variables.

4.6 Comparison of filter complexity

As a simple illustration of the three first approaches to derive a robust counterpart, consider a robust linear programming problem with the constraint ($A \in \mathbb{R}^{m \times n}$)

$$A(x + w) \leq b \quad \forall -1 \leq w \leq 1$$

This set of uncertain constraints can be robustified using the duality approach, the enumeration approach, and the explicit maximization. The explicit maximization approach will lead to the robustified constraint $Ax + |A| \mathbf{1} \leq b$, where $\mathbf{1}$ denotes a vector of ones and $|A|$ is elementwise absolute value. In other words, the size of the problem remains the same. The enumeration approach would lead

³ To be precise, the theorem concerns a strict inequality of a homogeneous polynomial, and the restriction on the simplex can be relaxed to any set in the positive orthant not including the origin.

to the constraints $A(x + v_i) \leq b$, where v_i denotes the 2^n vertices of the unit cube. Hence, the original set of constraints are replicated 2^n times, thus leading to a quickly growing problem size.

Finally, the duality based approach leads to a model with $2nm + n$ variables, nm equality constraints, and $m + 2nm$ inequality constraints⁴

From this simple comparison, it should be clear that the choice of filter can make a huge impact on the resulting problem. The current implementation always tries to apply the explicit maximization scheme first. If this filter not is applicable, it resorts to the enumeration scheme if possible. The duality based scheme is only used if necessary, or if an explicit choice is made via an option structure that is available for the user to guide the conversion.

5. EXAMPLES

To illustrate the use of the software package, we will solve a couple of uncertain optimization problems, arising in control theory and robust optimization in general.

5.1 Robust linear programming

As a first example, we simply implement the robust linear programming problem from Section 4.6, minimize $c^T x$ subject to $A(x + w) \leq b$ where $|w| \leq 1$.

```
x = sdpvar(n,1);
w = sdpvar(n,1);

C = [A*(x+w) <= b,
     abs(w) <= 1,
     uncertain(w)]
0 = c'*x

solvesdp(C,0);
```

For a user familiar with YALMIP, the only new modeling construct is the command `uncertain` which declares a variable uncertain. The actual uncertainty set will be extracted automatically from the model. After extracting the uncertainty model and analyzing the uncertain constraints, a suitable filter will be applied to derive the robust counterpart, which will be solved with a suitable solver, in this case any installed linear programming solver. In this case, the explicit maximization filter will be applied. Note that according to the discussion in Section 3, the uncertainty model will be expanded to the model $\{(w, t) : -t \leq w \leq t, t \leq 1\}$. This uncertainty set only implicitly defines a simple box-bounded set. However, since this case is so common, and the use of the absolute value operator is common among users, YALMIP implements some specialized code to analyze the uncertainty set to detect the redundant variables t and project the problem to the standard box-bounded case.

⁴ For every row a_i , we have to maximize $a_i^T w$ subject to $-1 \leq w \leq 1$. Since the dimension of w is n , there are $2n$ inequalities in the uncertainty constraint set. There will thus be $2n$ dual variables for each row. The dual variables are constrained by n equality constraints and $2n$ inequality constraints. Summing up and adding the original constraints and variables leads to the result.

5.2 LPV Stabilization

Our task is to compute a state-feedback $u = Kx$ for the parameter-varying system $\dot{x} = A(\rho)x + Bu$, where A is linearly parameterized in the variable ρ , which is constrained to a simplex. Without going into details, a controller that minimizes an upper bound on $\int x^T Q x + u^T R u$ can be found by solving the following uncertain semidefinite program in the variable L and the inverse Lyapunov matrix Y Boyd et al. (1994).

$$\begin{aligned} & \max \text{trace}(Y) \\ & \begin{bmatrix} -(AY + BL) - (AY + BL)^T & Y & L^T \\ Y & Q^{-1} & 0 \\ L & 0 & R^{-1} \end{bmatrix} \succeq 0 \end{aligned}$$

The feedback matrix can be recovered as $K = LY^{-1}$. This problem can be solved easily using our framework, since it fits into the enumeration scenario. However, to make matters more challenging, we complicate the problem slightly. To decrease conservativity, we use a parameterized inverse Lyapunov function $Y = \sum \alpha_i Y_i$. The problem with this parameterization is that the product $A(\rho)Y(\rho)$ yields bilinear terms. Hence, the enumeration scheme cannot be used. However, if the matrix $A(\rho)$ has a particular structure, some terms in Y can still be parameterized, without giving rise to any bilinear terms. This is where the elimination filter comes into play. By simply using a full parameterization, and letting YALMIP derive the robust counterpart, YALMIP will automatically constrain the structure of Y so that no bilinear terms are generated. After this elimination is done, the remaining uncertainty is dealt with using enumeration. A slightly less conservative approach can be obtained by using the Pólya filter instead. This approach will allow nonlinear terms, and will use the conservative relaxation outlined in the previous section. The following code illustrates how we would solve a problem in the case when ρ is two-dimensional, for a system with n states and m inputs, using a Pólya filter with $N = 1$.⁵

```
rho = sdpvar(2,1)
A = A1*rho(1) + A2*rho(2);

Y0 = sdpvar(n,n);
Y1 = sdpvar(n,n);
Y2 = sdpvar(n,n);
Y = Y0 + rho(1)*Y1 + rho(2)*Y2
L = sdpvar(n,m);

S = -A*Y-B*L;
C = [[S+S'      Y      L';
      Y      inv(Q)  zeros(n,m);
      L      zeros(m,n) inv(R)] > 0]
C = [C, 0 <= rho, sum(rho) == 1]
C = [C, uncertain(rho)]
0 = -trace(Y);
options = sdpsettings('robust.polya',1);
solvesdp(C,0,options)
```

⁵ A complete implementation can be found in the YALMIP wiki control.ee.ethz.ch/~joloef/wiki/pmwiki.php?n=Examples.LPV

5.3 Uncertain sum-of-squares

As a final example, we solve a problem where we show-case the integration of different modules in YALMIP. Here, illustrated by combining the robust optimization module with the sum-of-squares capabilities of YALMIP, described in detail in Löfberg (2008).

A nonlinear system is described by the following differential equation.

$$\begin{aligned}\dot{x}_1 &= -\frac{3}{2}x_1^3 - \frac{1}{2}x_1^2 - x_2 \\ \dot{x}_2 &= 6x_1 - wx_2\end{aligned}$$

The model is not known exactly, due to the uncertain parameter $w \in \mathcal{W} = \{w : 3 \leq w \leq 5\}$. Our goal is to show that the nonlinear system is stable for any $w \in \mathcal{W}$, and our approach to do this is to construct a polynomial Lyapunov function, and prove robust (asymptotic) stability using sum-of-squares techniques.

The uncertain variable w does not pose a problem for standard sum-of-squares techniques. Including information about uncertainty in the differential equation can be done relatively easily in a sum-of-squares framework by, e.g., suitable application of the positivstellensatz, Parrilo (2003). However, we will apply a robust optimization approach instead.

To prove stability, we introduce a polynomial Lyapunov function $V(x) = c^T v(x)$. For stability, we require

$$\begin{aligned}V(x) &> 0 \quad \forall x \neq 0 \\ \dot{V}(x) &< 0 \quad \forall x \neq 0, \quad w \in \mathcal{W}\end{aligned}$$

A sum-of-squares approach tries to find symmetric matrices Q_1 and Q_2 such that $V(x) = h(x)^T Q_1 h(x)$ and $\dot{V}(x) = -h(x)^T Q_2 h(x)$, given a polynomial basis $h(x)$. By writing the sum-of-squares problem in image form Parrilo (2003), the semidefinite problem that arise will include two constraints, $Q_1(c) \succ 0$ and $Q_2(c, w) \succ 0 \forall w \in \mathcal{W}$.

We begin by defining the variables and basic expressions involved in the problem. Without any deeper thought, we use a fourth order polynomial, and bound the Lyapunov function and its negative derivative from below by a quadratic function to ensure that the functions are positive definite (instead of using positive definite matrix constraints, which strictly speaking is impossible in practice, and can be a conservative way to impose strict definiteness of the polynomials⁶)

```
sdpvar x1 x2 w
f = [-1.5*x1^2-0.5*x1^3-x2;
     6*x1-w*x2];
x = [x1;x2];
[V,c] = polynomial(x,4);
dVdt = jacobian(V,x)*f;
r = x'*x;
```

The sum-of-squares constraints and the uncertainty model are defined and the problem is solved. If feasible, robust asymptotic stability is proven.

⁶ Methods to impose strict inequalities in a sum-of-squares setting is a delicate issue beyond the scope of this discussion.

```
C = [uncertain(w), 3<=w<=5];
C = [C, sos(V-r), sos(-dVdt-r)];
solvesdp(C, [], [], c);
```

Behind the scenes, YALMIP will derive the matrices $Q_1(c)$ and $Q_2(c, w)$. Since $Q_2(c, w)$ is linear in w and w is described by a polytope, the enumeration filter is applicable and is used to eliminate the uncertainty.

6. CONCLUSION

A software framework for robust optimization has been presented. The modeling language in YALMIP allows users to concentrate on the application model, while YALMIP takes care of reformulations required to remove uncertainty in the problem and compute robust solutions.

The implementation is currently limited to a small number of standard uncertainty cases, albeit they have to be considered the most common cases found in practice. Additional scenarios will be available in future versions.

In addition to more uncertainty scenarios, the framework will hopefully be extended to support a broader class of optimization problems, such as general convex problems, and uncertain geometric programs. The current version is primarily meant for problems with (mixed-integer) conic representable constraints.

REFERENCES

- A. Ben-Tal and A. Nemirovskii. Robust optimization - methodology and applications. *Mathematical Programming (Series B)*, 92:453–480, 2002.
- A. Ben-Tal and A. Nemirovskii. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, November 1998.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. SIAM, Philadelphia, Pennsylvania, 1994.
- G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming (Series B)*, 102(1):25–46, January 2005.
- M. C. Grant, S. Boyd, and Y. Ye. *CVX User's manual*. Stanford University, 2007. Available from <http://www.stanford.edu/~boyd/cvx>.
- G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, second edition, 1952.
- J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, Accepted for publication, 2008.
- J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. Available from control.ee.ethz.ch/~joloef/wiki/pmwiki.php.
- P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming Ser. B*, 96(2):293–320, 2003.
- C.W. Scherer. Relaxations for robust linear matrix inequality problems with verifications for exactness. *SIAM Journal on Matrix Analysis and Applications*, 27(2): 365–395, 2005.