SyNRAC: A Maple-Package for Solving Real Algebraic Constraints

Hirokazu Anai and Hitoshi Yanami

Information Technology Core Laboratories, Fujitsu Laboratories Ltd.
Kamikodanaka 4-1-1, Nakahara-ku, Kawasaki 211-8588, Japan
anai@jp.fujitsu.com, yanami@flab.fujitsu.co.jp

Abstract. In this paper we present a maple-package, named SyNRAC, for solving real algebraic constraints derived from various engineering problems. Our main tool is real quantifier elimination and we focus on its application to robust control design problems.

1 Introduction

SyNRAC is a maple-package aiming to be a comprehensive toolbox composed of a collection of solvers for real algebraic constraints derived from various engineering problems. SyNRAC stands for a Symbolic-Numeric toolbox for Real Algebraic Constraints. The solvers to be addressed include mainly symbolic ones and also symbolic-numeric ones to improve efficiency of symbolic approaches. Hence they can deal with parametric and nonconvex constraints.

The focus of the implemented algorithms is on practically effective quantifier elimination (QE) for certain industrial/engineering problems and simplification of quantifier-free formulas. Therefore SyNRAC provides a yet another implementation of quantifier elimination, a thing which is still missing in maple. Currently the following algorithms are available in SyNRAC:

- a special QE by the Sturm-Habicht sequence for sign definite condition (§2.1)
- a special QE by virtual substitution for linear formulas (§2.2)
- some simplifications of quantifier-free formulas

Furthermore, based on SyNRAC we are going to develop some toolboxes tailored for specific application fields, e.g., robust control design, on MATLAB, which would be novel tools that provide new systematic design procedures for engineers. Taking maple/MATLAB as a platform has the following advantages:

- maple-packages are automatically incorporated into MATLAB, which is widely used in engineering, via its "Symbolic Math Toolbox"
- They provide a good environment to realize symbolic-numeric solvers (floating-point arithmetic, many numerical packages for, e.g., optimization)

We note that this work is strongly motivated by one of the authors' previous works concerning practically effective applications of QE to robust control design problems [1,2,3]. They show that when we solve practical control problems it is effective to use the scheme that well combines reduction of problems to particular formulas and usage of QE algorithms specialized to such particular formulas.

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 828–837, 2003.

2 Special Quantifier Elimination Methods

Our current focus of implemented solvers is on QE algorithms specialized to particular types of input formulas that are sufficiently efficient for practical problems. Two special QE algorithms explained below are now available in SyNRAC.

Moreover simplification of quantifier-free formulas is important. Formulas which occur during a special QE algorithm tend to get extremely large, deeply nested and highly redundant. Utilizing simplification algorithms combined with a special QE algorithm contributes to improve not only readability of the resulting formula but efficiency of the computation (see [4,5] for possible simplifications). Automatic formula simplifiers are implemented in REDLOG[1] and QEPCAD.[2] Implementation of such simplifications in SyNRAC is ongoing.

2.1 Special QE Using the Sturm-Habicht Sequence

A special QE method based on the Sturm-Habicht sequence for the first-order formula $\forall x\ f(x) > 0$, where $f(x) \in \mathbb{R}[x]$ was proposed in [4]. The algorithm is desired to be modified for checking a *sign definite condition (SDC)*:

$$\forall x > 0,\ f(x) > 0\ , \tag{1}$$

since a quite wide range of the important problems in robust control can be reduced to SDC (see §3.1). We briefly sketch a special QE algorithm using the Sturm-Habicht sequence for the SDC (see [1] for details).

Definition 1. *Let P, Q be polynomials in $\mathbb{R}[x]$; $P = \sum_{k=0}^{n} a_k x^k$, $Q = \sum_{k=0}^{m} b_k x^k$, where n, m are non-negative integers. For $i = 0, 1, \ldots, \ell = min(n, m)$ we define the subresultant $Sres_i(P, n, Q, m)$ associated to P, n, Q and m of index i as $\sum_{j=0}^{i} M_j^i(P, Q) x^j$, where $M_j^i(P, Q)$ is the determinant of the matrix composed by the columns $1, 2, \ldots,\ n + m - 2i - 1$ and $n + m - i - j$ in the matrix $s_i(P, n, Q, m)$:*

$$s_i(P,n,Q,m) := \overbrace{\begin{pmatrix} a_n & \cdots & a_0 & & \\ & \ddots & & \ddots & \\ & & a_n & \cdots & a_0 \\ b_m & \cdots & b_0 & & \\ & \ddots & & \ddots & \\ & & b_m & \cdots & b_0 \end{pmatrix}}^{n+m-i} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} m-i \\ \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} n-i \end{matrix} .$$

Let $v = n + m - 1$ and $\delta_k = (-1)^{\frac{k(k+1)}{2}}$. The Sturm-Habicht sequence associated to P and Q is defined as the list of polynomials $\{SH_j(P, Q)\}_{j=0,\ldots,v+1}$ given by $SH_{v+1}(P, Q) = P$, $SH_v(P, Q) = P'Q$, $SH_j(P, Q) = \delta_{v-j} \cdot Sres_j(P, p, P'Q, v)$ for $j = 0, 1, \ldots, v - 1$, where $P' = \frac{dP}{dx}$. In particular, $\{SH_j(P, 1)\}_{j=0,\ldots,v+1}$ is called the Sturm-Habicht sequence of P. We simply denote it by $\{SH_j(P)\}$.

[1] REDLOG is a QE package based on virtual substitution (§2.2) on REDUCE.

[2] QEPCAD is a general QE package that is applicable to all first-order formulas based on cylindrical algebraic decomposition (CAD) [6,7].

The Sturm-Habicht sequence can be used for real root counting in almost the same way as the Sturm sequence. Moreover it has better properties in terms of specialization and computational complexity (see [8,9] for details).

Theorem 1 (González-Vega et al.[9]). *Let* $P(x) \in \mathbb{R}[x]$ *and* $\{g_0(x), \ldots, g_k(x)\}$ *be a set of polynomials obtained from* $\{SH_j(P(x))\}$ *by deleting the identically zero polynomials. Let* $\alpha, \beta \in \mathbb{R} \cup \{-\infty, +\infty\}$ *s.t.* $\alpha < \beta$. *We define* $W_{SH}(P; \alpha)$ *as the number of sign variations on* $\{g_0(\alpha), \ldots, g_k(\alpha)\}$. *Then* $W_{SH}(P; \alpha, \beta) \equiv W_{SH}(P; \alpha) - W_{SH}(P; \beta)$ *gives the number of real roots of* P *in* $[\alpha, \beta]$.

We denote the principal j-th Sturm-Habicht coefficient of $SH_j(f)$, i.e., the coefficient of degree j of $SH_j(f)$, by $st_j(f)$ and the constant term of $SH_j(f)$ by $ct_j(f)$ for all j. Then we have

$$\begin{aligned} W_{SH}(f; 0, +\infty) &= W_{SH}(f; 0) - W_{SH}(f; +\infty) \\ &= V(\{ct_n(f), \ldots, ct_0(f)\}) - V(\{st_n(f), \ldots, st_0(f)\}) \ , \end{aligned} \quad (2)$$

where $V(\{a_i\})$ stands for the number of sign changes over the sequence $\{a_i\}$. The SDC (1) holds if and only if both W_{SH} $(f; 0, +\infty) = 0$ and $st_n(f) > 0$ hold. Hence an equivalent condition to the SDC (1) can be obtained as follows: Consider all the possible sign combinations over the polynomials $ct_i(f)$,$st_i(f)$ (there are at most $3^{2(n+1)-3} = 3^{2n-1}$ patterns since $ct_0(f) = st_0(f)$, $st_n(f) > 0$, $st_{n-1}(f) > 0$); Choose all sign conditions that satisfy $W_{SH}(f; 0, +\infty) = 0$ by (2); Construct semi-algebraic sets generated by $ct_i(f)$, $st_i(f)$ for the selected sign conditions and combine them as a union. The obtained condition is of the form of a union of semi-algebraic sets. They are expected to contain many empty sets. We can prune some impossible sign combinations beforehand (see [1]). All procedures mentioned above have been implemented in SyNRAC.

2.2 Linear QE by Virtual Substitution

We present another special QE algorithm, i.e., quantifier elimination for *linear formulas.* A linear formula is a formula whose atomic subformulas are all linear with respect to its quantified variables. In 1993 Weispfenning [10] proposed a QE algorithm for linear formulas. Loos and Weispfenning [11] have presented more efficient algorithms. We explain the essence of their algorithms.

Let $Q_1 x_1 \cdots Q_n x_n \varphi$ be a linear formula, where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free formula. By using the equivalence $\forall x \varphi(x) \Longleftrightarrow \neg(\exists x \neg\varphi(x))$, we can change the formula into an equivalent formula of the form

$$(\neg)\exists x_1 \cdots (\neg)\exists x_n (\neg)\varphi \ .$$

The negation '$\neg$' that precedes a quantifier-free formula can be easily eliminated (use De Morgan's law and rewrite the atomic subformulas), which is not essential part of quantifier elimination. In addition to that, a practical problem is mostly given by an *existential formula,* i.e., a formula of the form $\exists x_1 \cdots \exists x_n \varphi$. We assume from now on that the input is an existential formula. Thus our main purpose is to eliminate the quantified variable $\exists x$ in $\exists x \varphi$ with φ quantifier-free.

Definition 2. *Let φ be a quantifier-free formula, $x \in X$ a variable, and S a set of terms, where each term $t \in S$ does not contain x. Then S is called an elimination set for $\exists x\varphi$ if the equivalence*

$$\exists x\varphi \Longleftrightarrow \bigvee_{t\in S} \varphi(x//t)$$

holds, where $\varphi(x//t)$ is the formula obtained by a modified substitution.[3]

It is known that for any given linear formula $\exists x\varphi$ as above, there exists an elimination set for the formula.

Lemma 1 (Weispfenning [10]). *Let φ be a linear quantifier-free formula, x a quantified variable in φ, and $\Psi = \{a_i x - b_i \ \rho_i \ 0 | i \in I, \rho_i \in \{=, \neq, \leq, <\}\}$ the set of atomic subformulas in φ. Then*

$$S = \{\frac{b_i}{a_i}, \frac{b_i}{a_i} \pm 1 | i \in I\} \cup \{\frac{1}{2}(\frac{b_i}{a_i} + \frac{b_j}{a_j}) | i, j \in I, i \neq j\}$$

is an elimination set for $\exists x\varphi$, where S is regarded as a set of linear terms.

By using Lemma 1, we can eliminate all the quantifiers in a given formula; eliminate them one by one from inside.

Loos and Weispfenning [11] have found smaller elimination sets than in Lemma 1 and succeeded in improving the algorithm because smaller elimination sets help prevent the number of atomic subformulas from getting larger during the elimination process. All the QE algorithms mentioned here have been implemented in SyNRAC. Furthermore an algorithm that returns a *sample point* if the input formula turns out to be 'true' has also been implemented in SyNRAC.

3 Fixed-Structure Robust Controller Synthesis

As applications of SyNRAC we focus on fixed-structure robust controller synthesis problems: Controller synthesis problems are to choose controller parameters so that given specifications are satisfied. It is strongly desired that the fixed-order controller design problems be resolved in practical problems which operate under the constraint of fixed structure. However, it is a long-standing open problem in robust control and the lack of effective results on the problem has prevented modern design methods from being applicable to such practical problems.

Recently methods based on QE have been proposed for such robust controller synthesis and multi-objective design problems [12,13,14,15,1]. The design scheme they used is, so called *a parameter space approach*, as follows:

1. Determine the structure of the controller and select the design parameters, e.g., x_1 and x_2 in the PI-controller $x_1 + \frac{x_2}{s}$ are the parameters
2. Reduce the specifications to the equivalent first-order formulas

[3] There is a procedure assigning the expression $\varphi(x/t)$ obtained from φ by substituting t for x a formula equivalent to it. We denote the resulting formula by $\varphi(x//t)$.

3. Compute the admissible regions of the design parameters for all specifications by applying QE to the obtained first-order formulas
4. Superpose the admissible regions in the parameter space and take the parameters in the intersections

However, since they in general need to use general QE algorithm based on CAD, they have a drawback on computational complexity. For the efficient computation it is effective to use the scheme that well combines reduction of target problems to particular classes of formulas and usage of special QE algorithms for the particular input formulas. Actually this scheme is successfully applied to several open problems in robust control; the following are such examples.

3.1 Control Synthesis Based on SDC

Many important design specifications in robust control such as H_∞ norm constraints, gain/phase margins and stability radius specification, which are frequently used as indices for the robustness of feedback control systems, can be recast as SDCs (see [16,17,18,3,19,1]). A special quantifier elimination method using the Sturm-Habicht sequence has sufficient practicability for the SDCs derived from practical control problems [1,3]: For example, an H_∞ norm constraint of a strictly proper transfer function $P(s) = n(s)/d(s)$ given by

$$\|P(s)\|_\infty := \sup_\omega |P(j\omega)| < 1$$

is equivalent to $\forall\omega\ d(j\omega)d(-j\omega) > n(j\omega)n(-j\omega)$. Since we can find a function $f(\omega^2)$ which satisfies $f(\omega^2) = d(j\omega)d(-j\omega) - n(j\omega)n(-j\omega) > 0$, letting $x = \omega^2$ lead to SDC. Similarly, a finite frequency H_∞ norm defined by

$$\|P(s)\|_{[\omega_1,\omega_2]} := \sup_{\omega_1 \le \omega \le \omega_2} |P(j\omega)| < 1$$

can be recast as the condition $f(x) \neq 0$ in $[\ -\omega_2^2, -\omega_1^2\ ]$, which is reduced to SDC for $f(z)$ by a bilinear transformation $z = -(x + \omega_2^2)/(x + \omega_1^2)$.

3.2 Linear Programming Approach to Control Design

Recently it has been reported in [2] that other important problems in robust control, which are recast as parametric linear programming (LP) problems, can be resolved with sufficient efficiency for practical use by using a special QE method based on virtual substitution [10]: If there is no feasible controller parameter value for a given specification, it is required to relax the given specification within acceptable levels. A systematic approach to estimating how we can relax a design specification is achieved by applying QE to parametric LP.

We briefly review robust controller synthesis via LP [20]. Consider a feedback control system shown in Fig.1, where $\mathbf{p} = [p_1, p_2, \ldots, p_s]$ is the vector of uncertain real parameters in the plant G and $\mathbf{x} = [x_1, x_2, \ldots, x_t]$ is the vector of real parameters of the controller C. Assume that the controller considered here is of

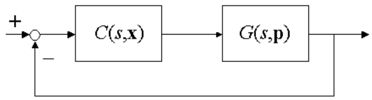

Fig. 1. A standard feedback system

fixed-order. The performance of the control system can often be characterized by a vector $\mathbf{a} = [a_1, \ldots, a_l]$ which are functions of the plant and controller parameters $\mathbf{p}$ and $\mathbf{x}$: $a_i = a_i(\mathbf{x}, \mathbf{p})$, $i = 1, \ldots, l$. Here we take the a_i to represent the coefficients of a closed-loop performance function. Suppose that the target value of the closed-loop performance vector is $\Delta_T = [\delta_1^T, \ldots, \delta_l^T] \in \mathbb{R}^l$. We denote the nominal value of the plant parameter by $\mathbf{p} = \mathbf{p}^0$ and deal with uncertainty in the plant by letting $\mathbf{p}$ lie in a box Π given by

$$\Pi = \{\mathbf{P} \mid p_i^- \leq p_i \leq p_i^+,\ i = 1, 2, \ldots, s\} .$$

The control system design is ideally to choose the controller parameter vector $\mathbf{x}$ which satisfies the set of equations $a_i(\mathbf{p}^0, \mathbf{x}) = \delta_i^T$, $i = 1, \ldots, l$. However, this is in general not attainable. Moreover, under plant perturbations the ideal performance deteriorates. Hence the robust performance problem requires ensuring that the controller design vector $\mathbf{x}$ can be chosen so that the performance aggravation, which could occur as $\mathbf{p}$ ranges over the uncertainty set Π, remains within acceptable limits. Therefore, we suppose that target performance Δ_T is given as the interval set

$$\mathbf{\Delta}_T = \{\Delta_T \mid \delta_i^{T-} \leq \delta_i^T \leq \delta_i^{T+},\ i = 1, 2, \ldots, l\} ,$$

which is the relaxation of the desired target performance δ_T. Then the goal of controller synthesis problem is to find a controller parameter vector $\mathbf{x}$ which satisfies the following set of inequalities

$$\delta_i^{T-} \leq a_i(\mathbf{p}, \mathbf{x}) \leq \delta_i^{T+}, \quad i = 1, 2, \ldots, l \tag{3}$$

for all $\mathbf{p} \in \Pi$. Assume that the parameter $\mathbf{p}$ appears linearly or multilinearly in $\mathbf{a}(\mathbf{p}, \mathbf{x})$ whereas $\mathbf{x}$ appears linearly (this is valid in many control problems). Then the constraints (3) obviously have a standard LP form. To estimate possible relaxation of the given specification, we regard endpoints of the target box as parameters in (3) and then compute the possible range of the endpoints so that a feasible controller exists by applying QE to a parametric version of (3). A typical example is the fixed-order robust pole assignment problem (see §4).

4 Computational Examples

In this section we show some computational examples to illustrate how SyNRAC works and its application to concrete control problems:[4] We load the packages:

[4] All computations were executed on a Pentium III 1 GHz processor.

```
> read "synrac";   with(combinat);
```

First we solve the QE problem $\forall x > 0,\ a_2x^2 + a_1x + a_0 > 0$:

```
> qe_sdc(a2*x^2+a1*x+a0,x);

                   -a0 < 0 and  a1 < 0 and -4*a0+a1^2 < 0 or
                   -a0 < 0 and -a1 < 0 and -4*a0+a1^2 < 0 or
                   -a0 < 0 and -a1 < 0 and  4*a0-a1^2 < 0

  time = 0.02, bytes = 123614
```

Next we solve the existential linear QE problem $\exists x \exists y (y > 2x+3 \wedge x > 0 \wedge y < s)$:

```
> qe_lin([x,y], y>2*x+3 and x>0 and y<s);

                                -1/2*s < -3/2

  time = 0.03, bytes = 144686
```

Finally we show the examples of decision problems for both commands:

```
> qe_sdc(x^5-x^2+3*x-9,x);
                                   false
  time = 1.11, bytes = 8774262

> qe_lin([x,y], y<2*x+2 and y<=-3*x+12 and y>(1/3)*x+5);
                                   true
                    A sample point:  [x, y], [52/25, 144/25]

  time = 0.03, bytes = 155078
```

Example 1 (H_∞ norm constraints for sensitivity function). Consider the feedback system shown in Fig.1 with $G(s) = \frac{1}{s-1}$, $C(s, \mathbf{x}) = x_1 + \frac{x_2}{s}$. We want to find feasible regions of the controller parameter $\mathbf{x}$ so that the system satisfies a finite frequency H_∞ norm constraint of a complementary sensitivity function $T(s)$:

$$||T(s)||_{[\omega_t,\infty]} \equiv \max_{\omega_t \le \omega \le \infty} ||T(\mathbf{i}\omega)|| < \gamma_t \ , \tag{4}$$

where $T(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$ and ω_t and γ_t are given real values. We can see from a simple symbolic computation that (4) is reduced to the following SDC:

$$x > 0,\ f_t(x) \equiv x^2 + a_1x + a_0 > 0 \ ,$$

where $a_1 = 2\omega_t^2 - 2x_2 + (1-x_1)^2 - x_1^2/\gamma_t^2$, $a_0 = \omega_t^4 - (2x_2 - (1-x_1)^2 + x_1^2/\gamma_t^2)\omega_t + m^2(1 - 1/\gamma_t^2)$. Performing qe_sdc in SyNRAC to $f_t(x)$ instantly gives us the following equivalent formula:

$$(-a_0 < 0 \wedge a_1 < 0 \wedge -4a_0 + a_1^2 < 0) \vee (-a_0 < 0 \wedge -a_1 < 0 \wedge -4a_0 + a_1^2 < 0) \vee$$
$$(-a_0 < 0 \wedge -a_1 < 0 \wedge 4a_0 - a_1^2 < 0)$$

After manual simplification we have the only one equivalent condition:

$$(a_0 > 0 \wedge a_1 > 0 \wedge 4a_0 - a_1^2 < 0) \ .$$

Moreover we can finally simplify the condition to obtain $(a_0 > 0 \wedge a_1 > 0)$ because $4a_0 - a_1^2 < 0$ is true due to the structure of a_0, a_1. Then if we specify the values of ω_t, γ_t, we immediately have possible regions of controller parameters x_1, x_2 which satisfy the given complementary sensitivity constraint. For example, the possible region for $\omega_t = 20$, $\gamma_t = -0.1$ with stability condition $(x_1 > 1 \wedge x_2 > 0)$ obtained from the Hurwitz criterion, is shown as the shaded region in Fig 2.

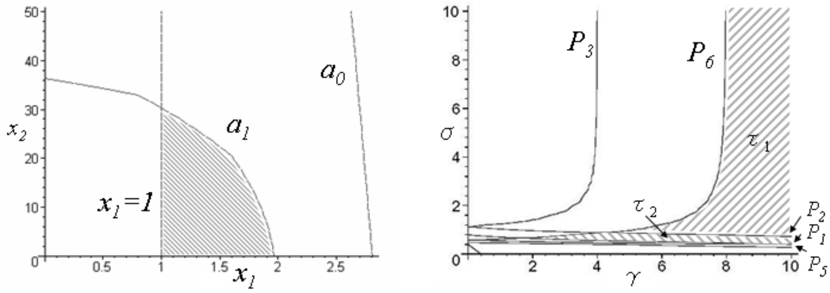

Fig. 2. The possible regions of Example 1 (left) and 2 (right).

Example 2 (Possible relaxation of robust pole assignment specification). We consider a PI control system with $C(s) = x_1 + \frac{x_2}{s}$ for the plant $G(s) = \frac{1}{(d_2s^2+d_1s+d_0)}$. The closed-loop characteristic polynomial is

$$\delta(s) = d_2s^3 + d_1s^2 + (x_1 + c_0)s + x_2 \ .$$

Then pole assignment problem is to locate the roots of $\delta(s)$ at(within) desired place(region). The target pole location is given as roots of a given target polynomial. Now we want to estimate how much we can relax the given infeasible specification. The target(relaxed) characteristic polynomial is given by

$$\delta_T(s) = \delta_3^T s^3 + \delta_2^T s^2 + \delta_1^T s + \delta_0^T \ ,$$

where $\delta_i^{T-} \leq \delta_i^T \leq \delta_i^{T+}$. Assume the endpoints have the following structure: $\delta_i^{T-} = \sigma^i(\delta_i^0 - e_i\gamma)$, $\delta_i^{T+} = \sigma^i(\delta_i^0 + e_i\gamma)$ for all i where δ_i^0, e_i are given constants and σ and γ are parameters which stand for changes of the time-scale (or frequency range) and a magnitude of perturbations, respectively. Then we find the possible region of δ_i^{T-}, δ_i^{T+} (i.e., γ, σ) so that there exists a controller parameter $\mathbf{x}$ satisfying that all the roots of $\delta(s)$ are within the root space of $\delta_T(s)$. Based on the argument in §3.2 we have the following formulas corresponding to (3):

$$\begin{aligned}\varphi \equiv ((&\delta_3^{T-} \leq d_2 \leq \delta_3^{T+}) \wedge (\delta_2^{T-} \leq d_1 \leq \delta_2^{T+}) \wedge (\delta_1^{T-} \leq x_1 + d_0 \leq \delta_1^{T+}) \wedge \\ &(\delta_0^{T-} \leq x_2 \leq \delta_0^{T+}) \wedge (-1 \leq d_0 \leq 1) \wedge (1 \leq d_1 \leq 3/2) \wedge (-1/2 \leq d_2 \leq 3/2)) \ .\end{aligned}$$

Here we take $\delta_0^0 = 4, \delta_1^0 = 6, \delta_2^0 = 4, \delta_3^0 = 1$, $e_0 = 1, e_1 = 3/4, e_2 = 1/2, e_3 = 1/4$. We execute `qe_lin` in SyNRAC to the first-order formula

$$\exists x_1 \exists x_2 \exists d_0 \exists d_1 \exists d_2 \ \varphi(x_1, x_2, d_0, d_1, d_2, \sigma, \gamma)$$

to obtain instantly an equivalent quantifier-free formula of the form $\bigvee_{i=1}^{16} \tau_i(\gamma, \sigma)$, where τ_i is the conjunction of atomic formulas. After additional simplification we finally get the following quantifier free formula

$$\psi(\gamma, \sigma) = \tau_1(\gamma, \sigma) \vee \tau_2(\gamma, \sigma) \ ,$$

where

$$\tau_1 = (\ P_2 \geq 0 \ \wedge \ P_3 \geq 0 \ \wedge \ P_5 \geq 0 \wedge P_6 \geq 0 \ \wedge \ P_7 \geq 0 \ \wedge \ P_8 \geq 0\ ),$$
$$\tau_2 = (\ P_1 \geq 0 \ \wedge \ P_2 \leq 0 \ \wedge \ P_5 \geq 0 \wedge P_6 \geq 0 \ \wedge \ P_7 \geq 0 \ \wedge \ P_8 \geq 0\ ),$$

$$\begin{array}{lll} P_1 = \gamma\sigma^3 + 4\sigma^3 - 2, & P_2 = \gamma\sigma^3 + 4\sigma^3 - 6, & P_3 = \gamma\sigma^3 - 4\sigma^3 + 6, \\ P_4 = \gamma\sigma^2 + 8\sigma^2 - 2, & P_5 = \gamma\sigma^2 + 8\sigma^2 - 3, & P_6 = \gamma\sigma^2 - 8\sigma^2 + 3, \\ P_7 = \gamma\sigma, & P_8 = \gamma. & \end{array}$$

The possible region of γ, σ given by ψ is illustrated as the shaded region in Fig.2. Since we finally have all the possible region as semialgebraic sets in γ-σ space, we can easily obtain the minimum relaxation.

5 Conclusions

We have presented maple-package SyNRAC for solving real algebraic constraints. Although our project is currently under development and there is still a considerable way to go until the state-of-the-art techniques in real quantifier elimination are implemented in SyNRAC, we think our package has now reached a stage of development that justifies publication. We are continually improving the efficiency of implemented algorithms and are going to implement other algorithms (including symbolic-numeric algorithms) for solving real algebraic constraints into SyNRAC. We also plan to develop some toolboxes tailored for specific applications (e.g., parametric robust control toolbox) based on SyNRAC. In order to make our system applicable to those who are interested in but not familiar with symbolic computation and maple software, we are going to incorporate SyNRAC into MATLAB and implement interfaces to modeling formulas and sophisticated visualization facility of feasible parameter regions in a parameter space.

Acknowledgements. The authors would like to thank Shinji Hara and Kazuhiro Yokoyama for their invaluable advice.

References

1. Anai, H., Hara, S.: Fixed-structure robust controller synthesis based on sign definite condition by a special quantifier elimination. In: Proceedings of American Control Conference 2000. (2000) 1312–1316

2. Anai, H., Hara, S.: Linear programming approach to robust controller design by a quantifier elimination. In: Proceedings of SICE Annual Conference 2002 (Osaka, Japan). (2002) 863–869
3. Anai, H., Hara, S.: A parameter space approach for fixed-order robust controller synthesis by symbolic computation. In: Proceedings of IFAC World Congress on Automatic Control b'02. (2002)
4. González-Vega, L.: A combinatorial algorithm solving some quantifier elimination problems. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and monographs in symbolic computation. Springer-Verlag (1998) 365–375
5. Dolzmann, A., Sturm, T.: Simplification of quantifier-free formulae over ordered fields. Journal of Symbolic Computation **24** (1997) 209–231
6. Collins, G.E.: Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In Brakhage, H., ed.: Automata Theory and Formal Languages. 2nd GI Conference. Volume 33 of Lecture Notes in Computer Science., Gesellschaft für Informatik, Springer-Verlag, Berlin, Heidelberg, New York (1975) 134–183
7. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. Journal of Symbolic Computation **12** (1991) 299–328
8. González-Vega, L., Lombardi, H., Recio, T., Roy:, M.F.: Sturm-habicht sequence. In: Proceedings of ISSAC"89, Portland, ACM Press (1989) 136–146
9. González-Vega, L., Recio, T., Lombardi, H., Roy, M.F.: Sturm-habicht sequences determinants and real roots of univariate polynomials. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 300–316
10. Weispfenning, V.: The complexity of linear problems in fields. Journal of Symbolic Computation **5** (1988) 3–27
11. Loos, R., Weispfenning, V.: Applying linear quantifier elimination. The Computer Journal **36** (1993) 450–462 Special issue on computational quantifier elimination.
12. Abdallah, C., Dorato, P., Yang, W., Liska, R., Steinberg, S.: Application of quantifier elimination theory to control system design. In: Proceedings of 4th IEEE Mediteranean Symposium on Control and Automation. Maleme, Crete. (1996) 340–345
13. Dorato, P., W.Yang, C.Abdallah: Robust multi-objective feedback design by quantifier elimination. J. Symb. Comp. **24** (1997) 153–159
14. Jirstrand, M.: Constructive Methods for Inequality Constraints in Control. PhD thesis, Linköping University, Sweden (1998)
15. Anai, H.: On solving semidefinite programming by quantifier elimination. In Proc. of American Control Conference, Philadelphia (1998) 2814–2818
16. Hara, S., Kimura, T., Kondo, R.: H $_\infty$ control system design by a parameter space approach. In: Proceedings of MTNS-91, Kobe, Japan (1991) 287–292
17. Kimura, T., Hara, S.: A robust control system design by a parameter space approach based on sign definition condition. In: Proceedings of KACC-91, Soul, Korea (1991) 1533–1538
18. Kondo, R., Hara, S., Kaneko, T.: Parameter space design for H_∞ control (in japanese). Trans. of SICE **27** (1991) 714–716
19. Šiljak, D.D.: New algebraic criterion for positive realness. Journal of the Franklin Institute **291** (1971) 109–120
20. Keel, L., Bhattacharyya, S.: Robust control under parametric uncertainty. part II: design. In: SYMBOLIC METHODS in control system analysis and design. Number 56 in IEE Control Engineering Series, IEE (1999) 203–226