Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Robust Stability of Polynomic Interval Polynomials

P. Husek

Dept. of Control Engineering, Faculty of Electrical Engineering,
Czech Technical University in Prague,.
Technická 2, 166 27 Praha 6, Czech Republic
(husek@fel.cvut.cz)

Abstract: In this paper an algorithm for robust stability analysis of linear systems with parametric uncertainty is presented. The algorithm is based on multivariate interpolation-evaluation methods and fast Fourier transform used for computing determinant of a multivariate polynomial matrix. Positivity of a multivariate polynomial on a hyperrectangle is tested by Bernstein algorithm. The high efficiency of the proposed algorithm is demonstrated on automatic steering control of Daimler Benz city bus – an 8-th order closed-loop system with two uncertain parameters.

1. INTRODUCTION

Robust stability of polynomials with parametric uncertainty is intensively studied since the celebrated Kharitonov theorem (Kharitonov, 1978) has been published. Kharitonov solves the robust stability problem for interval continuous-time polynomials. The same problem for polynomials with parametric uncertainty of affine structure is solved by the Edge theorem (Bartlett *et al.,* 1988) that states that it is sufficient to check stability of polynomials lying on exposed edges. Mapping theorem (Zadeh and Desoer, 1963) shows that the non-convex value set of multilinear interval polynomials is contained in the convex polygon given by its vertices.

To date there are only few results solving the problem of robust stability of polynomials with polynomic structure of coefficients (polynomic interval polynomials). There are two basic approaches – algebraic and geometric. The first one is based on utilization of criteria commonly used for stability analysis of fixed polynomials – Hurwitz or Routh criterion – and their generalization for uncertain polynomials. The second one transforms the multidimensional problem in twodimensional test of frequency plot of the closed-loop polynomial using zero exclusion principle. Such algorithm is used in (Zettler and Garloff, 1998). In this paper the attention will be focused on algebraic approach and its improvement.

To be able to perform Hurwitz stability test we have to compute determinant of Hurwitz matrix and test its positivity. For systems with polynomic dependency of the coefficients of characteristic polynomials on system parameters Hurwitz matrix turns into multivariate polynomial matrix. Computation of its determinant is usually performed by generalization of the procedure used for constant matrices – triangularization and subsequent multiplication of the elements on the main diagonal or by symbolic computations. The drawback of both methods consists in their low efficiency. Even for systems of moderate order or number of parameters the computational time goes to the tenths of seconds. The presented algorithm reduces the computational time dramatically.

2. PROBLEM STATEMENT

A family of continuous-time polynomials is considered, where the coefficients are polynomic functions of an interval parameter $\mathbf{q} \in Q \subset \Re^l$, i.e. the corresponding family P can be expressed in the following form:

$$P = \{ p(s,\mathbf{q}) : \mathbf{q} \in Q \},\ p(s,\mathbf{q}) = a_n(\mathbf{q})s^n + \cdots + a_1(\mathbf{q})s + a_0(\mathbf{q}) \quad (1)$$

where

$$\mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_l]^T,\quad q_i \in [q_i^-, q_i^+],\quad i = 1,\ldots,l \quad (1)$$

is a multidimensional interval parameter.

Each coefficient $a_j(\mathbf{q})$, j=0,...,n is supposed to be a polynomic function of $\mathbf{q}$. Such a family is referred to as a polynomic interval polynomial. Our task is to determine if the polynomic interval polynomial (1) is robustly stable, i.e. whether each member of the family is asymptotically stable.

3. HURWITZ STABILITY CRITERION

The well-known Hurwitz criterion for stability test of a fixed polynomial is also applicable in robust stability analysis.

For an uncertain polynomial (1) of degree n the $n \times n$ array

$$\mathbf{H}_n(\mathbf{q}) = \begin{bmatrix} a_{n-1}(\mathbf{q}) & a_{n-3}(\mathbf{q}) & a_{n-5}(\mathbf{q}) & \cdots \\ a_n(\mathbf{q}) & a_{n-2}(\mathbf{q}) & a_{n-4}(\mathbf{q}) & \cdots \\ 0 & a_{n-1}(\mathbf{q}) & a_{n-3}(\mathbf{q}) & \cdots \\ 0 & a_n(\mathbf{q}) & a_{n-2}(\mathbf{q}) & \cdots \\ \vdots & & & \\ 0 & \cdots & 0 & a_{n-1}(\mathbf{q}) \\ 0 & \cdots & 0 & a_n(\mathbf{q}) \end{bmatrix}$$

$$\begin{bmatrix} a_1(\mathbf{q}) & 0 & 0 & \cdots & 0 \\ a_2(\mathbf{q}) & a_0(\mathbf{q}) & 0 & \cdots & 0 \\ a_3(\mathbf{q}) & a_1(\mathbf{q}) & 0 & \cdots & 0 \\ a_4(\mathbf{q}) & a_2(\mathbf{q}) & a_0(\mathbf{q}) & \cdots & 0 \\ & \ddots & & & \vdots \\ a_{n-3}(\mathbf{q}) & a_{n-5}(\mathbf{q}) & \cdots & a_1(\mathbf{q}) & 0 \\ a_{n-2}(\mathbf{q}) & a_{n-4}(\mathbf{q}) & \cdots & a_2(\mathbf{q}) & a_0(\mathbf{q}) \end{bmatrix} \quad (3)$$

is called the Hurwitz matrix associated with $p(s,\mathbf{q})$.

Then the family of polynomials (1) is stable if and only if

a) there exists a stable polynomial $p(s,\mathbf{q})\in P$,

b) $\det \mathbf{H}_n(\mathbf{q}) \neq 0$ for all $\mathbf{q}\in Q$.

Since all the coefficients $a_j(\mathbf{q})$, $j=0,\ldots,n$ of polynomial (1) are continuous functions the condition b) is equivalent to testing positivity (or negativity) of the determinant of $\mathbf{H}_n(\mathbf{q})$ on the set Q.

4. DETERMINANT OF MULTIVARIATE POLYNOMIAL MATRIX

In order to be able to use Hurwitz stability criterion the determinant of multivariate polynomial matrix has to be determined. For small or moderate number of parameters this can be performed using symbolic computations but for more extensive problems with higher number of parameters and/or higher order of uncertain polynomial this method is not effective. In this section new algorithm for computing determinant of multivariate polynomic matrix is presented. The procedure is based on interpolation techniques (Bini and Pan, 1994).

First of all it will be shown that an l-variate polynomial can be uniquely represented by its values in appropriately chosen N interpolation points (l-tuples) and all its coefficients can be recovered from these two sets.

Let p be an l-variate polynomial

$$p = p(q_1,\ldots,q_l) = \sum_{i_1=0}^{d_1}\sum_{i_2=0}^{d_2}\cdots\sum_{i_l=0}^{d_l} c_{i_1 i_2 \ldots i_l} q_1^{i_1} q_2^{i_2} \ldots q_l^{i_l} \quad (4)$$

where d_i, i=1,...,l are degrees of p in variable q_i. The polynomial (4) can be expressed as

$$\begin{aligned} p(q_1,\ldots,q_l) &= p_0(q_1,\ldots,q_{l-1}) + q_l p_1(q_1,\ldots,q_{l-1}) + \cdots \\ &+ q_l^{d_l} p_{d_l}(q_1,\ldots,q_{l-1}) \end{aligned} \quad (5)$$

where $p_{i_l}(q_1,\ldots,q_{l-1})$, i_l=0,...,d_l are some (l-1)-variate polynomials. Let us suppose that ω_l is a primitive (d_l+1)-th root of 1 and the values of $p(q_1,...,q_l)$ are known for some fixed (l-1)-tuple $(q_1,...,q_{l-1}) = (s_1,...,s_{l-1})$ where s_i, i=1,...,l-1 are complex scalars, and for all $q_l = \omega_l^h$, h=0,...,d_l. Then using the inverse fast Fourier transform (FFT) algorithm all the values of the (l-1)-variate polynomials $p_{i_l}(q_1,\ldots,q_{l-1})$, i_l=0,...,d_l in the point $(s_1,...,s_{l-1})$ can be recovered. This step can be repeated for any point ((l-1)-tuple) $(s_1,...,s_{l-1})$.

Any of the (l-1)-variate polynomials $p_{i_l}(q_1,\ldots,q_{l-1})$, i_l=0,...,d_l can be written as

$$\begin{aligned} p_{i_l}(q_1,\ldots,q_{l-1}) &= p_{0i_l}(q_1,\ldots,q_{l-2}) + q_{l-1}p_{1i_l}(q_1,\ldots,q_{l-2}) \\ &+ \cdots + q_{l-1}^{d_{l-1}} p_{d_{l-1}i_l}(q_1,\ldots,q_{l-2}) \\ i_l &= 0,\ldots,d_l \end{aligned} \quad (6)$$

where $p_{i_{l-1}i_l}(q_1,\ldots,q_{l-2})$, i_{l-1}=0,...,d_{l-1}, i_l=0,...,d_l are some (l-2)-variate polynomials. Let us again suppose that ω_{l-1} is a primitive (d_{l-1}+1)-th root of 1 and the values of $p_{i_l}(q_1,\ldots,q_{l-1})$, i_l=0,...,d_l are known for some fixed (l-2)-tuple $(q_1,...,q_{l-2}) = (s_1,...,s_{l-2})$ where s_i, i=1,...,l-2 are complex scalars, and for all $q_{l-1} = \omega_{l-1}^h$, h=0,...,d_{l-1}. These values can be determined using previous step of this algorithm. Then again using the inverse FFT algorithm all the values of the (l-2)-variate polynomials $p_{i_{l-1}i_l}(q_1,\ldots,q_{l-2})$, i_{l-1}=0,...,d_{l-1}, i_l=0,...,d_l in the point $(s_1,...,s_{l-2})$ can be recovered. This step can be repeated for any point ((l-2)-tuple) $(s_1,...,s_{l-2})$.
Repeating this procedure one arrives in the last but one step. Any of the bivariate polynomials $p_{i_3 i_4 \ldots i_l}(q_1,q_2)$, i_j=0,...,d_j, j=3,...,l can be written as

$$\begin{aligned} p_{i_3 i_4 \ldots i_l}(q_1,q_2) &= p_{0 i_3 i_4 \ldots i_l}(q_1) + q_2 p_{1 i_3 i_4 \ldots i_l}(q_1) + \cdots + \\ &+ q_2^{d_2} p_{d_2 i_3 i_4 \ldots i_l}(q_1) \\ i_j &= 0,\ldots,d_j,\ j=3,\ldots,l \end{aligned} \quad (7)$$

where $p_{i_2 i_4 \ldots i_l}(q_1)$, i_j=0,...,d_j, j=2,...,l are some univariate polynomials. Let us again suppose that ω_2 is a primitive (d_2+1)-th root of 1 and the values of $p_{i_3 i_4 \ldots i_l}(q_1,q_2)$, i_j=0,...,d_j, j=2,...,l are known for some fixed point (scalar) $q_1=s_1$ and for all $q_2 = \omega_2^h$, h=0,...,d_2. These values can be determined using previous steps of this algorithm. Then using the inverse FFT algorithm all the values of the univariate polynomials $p_{i_2 i_3 \ldots i_l}(q_1)$, i_j=0,...,d_j, j=2,...,l in the point s_1 can be recovered. This procedure can be repeated for any point s_1.

Finally, the last step can be performed. Each of the univariate polynomials $p_{i_2 i_3 \ldots i_l}(q_1)$, i_j=0,...,d_j, j=2,...,l can be expressed as

$$\begin{aligned} p_{i_2 i_3 \ldots i_l}(q_1) &= c_{0 i_2 i_3 \ldots i_l} + q_1 c_{1 i_2 i_3 \ldots i_l} + \cdots + q_1^{d_1} c_{d_1 i_2 i_3 \ldots i_l} \\ i_j &= 0,\ldots,d_j,\ j=2,\ldots,l \end{aligned} \quad (8)$$

where $c_{i_1 \ldots i_l}$, i_j=0,...,d_j, j=1,...,l are the coefficients of the original l-variate polynomial (4). Suppose that ω_1 is a primitive (d_1+1)-th root of 1 and the values of $p_{i_2 i_3 \ldots i_l}(q_1)$, i_j=0,...,d_j, j=2,...,l are known for all $q_1 = \omega_1^h$, h=0,...,d_1. These values can be determined using previous steps of this algorithm. Then using the inverse FFT algorithm all the coefficients $c_{i_1 \ldots i_l}$, i_j=0,...,d_j, j=1,...,l can be determined.

The described procedure proves that an l-variate polynomial can be uniquely determined by its values given in $N = \prod_{j=1}^{l}(d_j+1)$ distinct interpolation points (l-tuples). Moreover, if the interpolation l-tuples (s_1,...,s_l) are chosen as

$$(s_1,\ldots,s_l):\; s_j = \omega_j^h,\; j=1,\ldots,l,\; h=0,\ldots,d_j \tag{9}$$

where ω_j, j=1,...,l are primitive (d_j+1)-th roots of 1, then the coefficients $c_{i_1 \ldots i_l}$, i_j=0,...,d_j, j=1,...,l of (4) can be determined by ($\sum_{j=1}^{l} d_j + l$)-multiple using of inverse FFT algorithm.

In order to evaluate an l-variate polynomial (4) in the prescribed set of interpolation points (9) the forward FFT algorithm is used. One can run the above mentioned procedure in the reverse order. At first the univariate polynomials

$$\begin{gathered} p_{i_2 i_3 \ldots i_l}(q_1) = c_{0 i_2 \ldots i_l} + q_1 c_{1 i_2 i_3 \ldots i_l} + \cdots + q_1^{d_1} c_{d_1 i_2 i_3 \ldots i_l} \\ i_j = 0,\ldots,d_j,\; j=2,\ldots,l \end{gathered} \tag{10}$$

are evaluated at d_1 scalar Fourier points $q_1 = \omega_1^h$ (ω_1 is a primitive (d_1+1)-th root of 1), h=0,...,d_1, using forward FFT algorithm. Then the bivariate polynomials

$$\begin{gathered} p_{i_3 i_4 \ldots i_l}(q_1,q_2) = p_{0 i_3 i_4 \ldots i_l}(q_1) + q_2 p_{1 i_3 i_4 \ldots i_l}(q_1) + \cdots + \\ + q_2^{d_2} p_{d_2 i_3 i_4 \ldots i_l}(q_1) \\ i_j = 0,\ldots,d_j,\; j=3,\ldots,l \end{gathered} \tag{11}$$

are evaluated at (d_1+1)*(d_2+1) Fourier points (pairs) (q_1,q_2), $q_1 = \omega_1^{h_1}$ and $q_2 = \omega_2^{h_2}$ (ω_2 is a primitive (d_2+1)-th root of 1), h_1=0,...,d_1, h_2=0,...,d_2 using forward FFT algorithm and the values computed in the previous step. The last but one step of the procedure consists in evaluation of (l-1)-variate polynomials

$$\begin{gathered} p_{i_m}(q_1,\ldots,q_{l-1}) = p_{0 i_l}(q_1,\ldots,q_{l-2}) + q_{l-1} p_{1 i_l}(q_1,\ldots,q_{l-2}) \\ + \cdots + q_{l-1}^{d_{l-1}} p_{d_{l-1} i_l}(q_1,\ldots,q_{l-2}) \\ i_l = 0,\ldots,d_l \end{gathered} \tag{12}$$

at $\prod_{j=1}^{l-1}(d_j+1)$ Fourier points ((l-1)-tuples) (q_1,...,q_{l-1}), $q_j = \omega_j^{h_j}$ (ω_{l-1} is a primitive (d_{l-1}+1)-th root of 1), h_j=0,...,d_j, j=1,...,l-1 using forward FFT algorithm and all the values computed in all the previous steps. Finally, the original polynomial (4) is evaluated at $\prod_{j=1}^{l}(d_j+1)$ Fourier points (l-tuples) (q_1,...,q_l), $q_j = \omega_j^{h_j}$ (ω_l is a primitive (d_l+1)-th root of 1), h_j=0,...,d_j, j=1,...,l using FFT algorithm and all the values computed in all the previous steps.

Now the algorithm computing the determinant of multivariate polynomial matrix using evaluation-interpolation techniques can be described.

Let $\mathbf{A}(\mathbf{q})$ be an $(n \times n)$ multivariate polynomial matrix, $\mathbf{q} = (q_1,\ldots,q_l)$. The task is to compute the coefficients of its determinant

$$c(q_1,\ldots,q_l) = \det(\mathbf{A}(\mathbf{q})) = \sum_{i_1=0}^{\delta_1}\sum_{i_2=0}^{\delta_2}\cdots\sum_{i_l=0}^{\delta_l} c_{i_1 i_2 \ldots i_l} q_1^{i_1} q_2^{i_2} \ldots q_l^{i_l} \,. \tag{13}$$

At first the highest degrees δ_i, i = 1,...l of all variables of the resulted determinant has to be estimated (or at least their lowest possible values). One can for example take

$$\delta_i = \sum_{j=1}^{n} col \deg_j^{q_i}(\mathbf{A}(\mathbf{q})), i=1,\ldots,l \tag{14}$$

i.e., δ_i, i = 1,...l, are taken as the sum of column degrees of corresponding variables (the highest degree of each variable appearing in the column).

Let ω_i denote a primitive (δ_i+1)-th root of 1, i = 1,...,l. Then the whole procedure consists of three steps:

1. evaluate the matrix $\mathbf{A}(\mathbf{q})$ in the $N = \prod_{i=1}^{l}(\delta_i+1)$ Fourier points (l-tuples) $(\omega_1^{k_1},\ldots,\omega_l^{k_l})$, $k_i = 0,\ldots,\delta_i$, $i=1,\ldots,l$ (by means of multiple application of forward FFT algorithm),
2. compute the values of determinant of $\mathbf{A}(\mathbf{q})$ in those N complex interpolation points (l-tuples): $c(\omega_1^{k_1},\ldots,\omega_l^{k_l}) = \det \mathbf{A}(\omega_1^{k_1},\ldots,\omega_l^{k_l})$, for $k_i = 0,\ldots,\delta_i$, $i=1,\ldots,l$,
3. recover all the coefficients of the determinant $c(\mathbf{q})$ of $\mathbf{A}(\mathbf{q})$ from the set of its values $c(\omega_1^{k_1},\ldots,\omega_l^{k_l})$ by applying the multiple inverse FFT algorithm on the $N = \prod_{i=1}^{l}(\delta_i+1)$ Fourier l-tuples $(\omega_1^{k_1},\ldots,\omega_l^{k_l})$, $k_i = 0,\ldots,\delta_i$, $i=1,\ldots,l$.

This procedure is applied to compute the Hurwitz determinant (3). The complexity bound of the presented procedure is $O(N\log_2 N\log_2\log_2 m)$, $N = m^l$. This is a considerable improvement to the straightforward algorithm with numerical complexity $O(N^2)$.

In order to test positivity of determinant of Hurwitz matrix determined by the algorithm mentioned above on the uncertainty set the algorithm of Bernstein expansion is used. The Bernstein form of a polynomial is well-known for a long time but its generalization to multivariate case is quite recent The first application to the range of bivariate polynomials was given by Garloff in (Garloff, 1985). The generalization to multivariate case is proposed in (Garloff, 1993). Some improvements of the original algorithm are in (Garloff *et al.*, 1997) and (Zettler and Garloff, 1998) where also more details can be found. Using iterative sweep procedure the algorithm is able to determine lower and upper bound of a multivariate polynomial on a hyperrectangle. Here the basic algorithm of Bernstein expansion will be described.

5. BERNSTEIN EXPANSION

Define a multi-index $\mathbf{I}$ as an ordered l-tuple of nonnegative integers $(i_1,\ldots,i_l)$ and for $\mathbf{q}=(q_1,\ldots,q_l)\in\Re^l$ set $\mathbf{q}^{\mathbf{I}} = q_1^{i_1}\cdot q_2^{i_2}\cdot\cdots\cdot q_l^{i_l}$. Write $\mathbf{I}\le\mathbf{N}$ if $\mathbf{N}=(n_1,\ldots,n_l)$ and if $0\le i_k \le n_k$, $k = 1,\ldots,l$. Further, let $S = \{\mathbf{I} : \mathbf{I} \le \mathbf{N}\}$.

Consider a polynomial

$$w(\mathbf{q}) = \sum_{I\in S} w_I \mathbf{q}^{\mathbf{I}}, \quad \mathbf{q}\in\Re^l. \tag{15}$$

The univariate i-th Bernstein polynomial of degree n is defined as

$$b_{n,i} = \binom{n}{i} q^i (1-q)^{n-i}, \quad 0 < i < n \tag{16}$$

for an arbitrary $q\in\Re$. In the multivariate case, the $\mathbf{I}$-th Bernstein polynomial of degree $\mathbf{N}$ is defined by

$$\begin{aligned} B_{\mathbf{N},\mathbf{I}}(\mathbf{q}) &= b_{n_1,i_1}(q_1)\cdot\cdots\cdot b_{n_l,i_l}(q_l), \\ \mathbf{q} &= (q_1,\ldots,q_l)\in\Re^l \end{aligned}. \tag{17}$$

Without loss of generality consider the unit box $U=[0,1]^l$, since any nonempty box of $\Re^l$ can be mapped affinely onto this box.

The transformation of a polynomial into its Bernstein form results in

$$w(\mathbf{q}) = \sum_{\mathbf{I}\in S} b_{\mathbf{I}}(U) B_{\mathbf{N},\mathbf{I}}(\mathbf{q}) \tag{18}$$

where the Bernstein coefficicents $b_{\mathbf{I}}(U)$ of w over U are given by

$$b_{\mathbf{I}}(U) = \sum_{\mathbf{J}\le\mathbf{I}} \frac{\binom{\mathbf{I}}{\mathbf{J}}}{\binom{\mathbf{N}}{\mathbf{J}}} w_{\mathbf{J}}, \quad \mathbf{I}\in S. \tag{19}$$

Denote by S_0 a special subset of the index set S comprising those indices which correspond to the indices of the vertices of the array $b_{\mathbf{I}}(U)$, i.e.

$$S_0 = \{0, n_1\}\times\cdots\times\{0, n_l\}. \tag{20}$$

The corresponding Bernstein coefficients $b_{\mathbf{I}}(U)$, $\mathbf{I}\in S_0$ are called sharp ones.

Let us list some useful properties of the Bernstein coefficients:

$$B_{\mathbf{N},\mathbf{I}}(\mathbf{q}) \ge 0 \;\; \forall\mathbf{q}\in U, \mathbf{I}\in S \tag{21}$$

$$\sum_{\mathbf{I}\in S} B_{\mathbf{N},\mathbf{I}}(\mathbf{q}) = 1 \;\; \forall\mathbf{q}\in\Re^l \tag{22}$$

$$b_{\mathbf{I}}(U) = w(\mathbf{I}/\mathbf{N}) \;\; \forall\mathbf{I}\in S_0 \tag{23}$$

$$Conv\{(\mathbf{q}, w(\mathbf{q})); \mathbf{q}\in U\} \subseteq Conv\{(\mathbf{I}/\mathbf{N}, b_{\mathbf{I}}(U)); \mathbf{I}\in S\}. \tag{24}$$

The following statements are the consequence of (21)-(24). If the minimum of the Bernstein coefficients $b_{\mathbf{I}}(U)$, $\mathbf{I}\in S$ is positive, the $w(\mathbf{q})$ is positive on U. If the maximum of the Bernstein coefficients $b_{\mathbf{I}}(U)$, $\mathbf{I}\in S$ is negative, the $w(\mathbf{q})$ is negative on U. If there exists a nonpositive sharp Bernstein coefficient $b_{\mathbf{I}_0}(U)$, $\mathbf{I}_0\in S_0$, the polynomial $w(\mathbf{q})$ is not positive on U, if there exists a nonnegative sharp Bernstein coefficient $b_{\mathbf{I}_0}(U)$, $\mathbf{I}_0\in S_0$, the polynomial $w(\mathbf{q})$ is not negative on U. In other case it is necessary to test positivity of $w(\mathbf{q})$ computing Bernstein coefficients on new regions by using the Sweep procedure.

5.1 Sweep procedure

Let D be any subbox of U generated by sweep operations (at the beginning $D=U$, then subsequently D is obtained by successively dividing). Define a sweep in the r-th direction ($1\le r\le l$) as recursively applied linear interpolation. Starting with $B^{(0)}(D)=B(D)$ set for $k=1,\ldots,n_r$ the following equation:

$$b^{(k)}_{i_1,\ldots,i_r,\ldots i_l}(D) = \begin{cases} b^{(k-1)}_{i_1,\ldots,i_r,\ldots i_l}(D), \; i_r = 0,\ldots,k-1 \\ (1-\lambda) b^{(k-1)}_{i_1,\ldots,i_{r-1},\ldots i_l}(D) + \\ + \lambda b^{(k-1)}_{i_1,\ldots,i_r,\ldots i_l}(D), \; i_r = k,\ldots,n_r \end{cases} \tag{25}$$

for $i_j = 0,\dots,n_j$ and $j = 1,\dots,r-1,r+1,\dots,l$.

Then the Bernstein coefficients on D_0, where the subbox D_0 is given by

$$\begin{aligned} D_0 = &\left[\underline{d}_1, \overline{d}_1\right] \times \cdots \times \left[\underline{d}_r, \underline{d}_r + \lambda\left(\overline{d}_r - \underline{d}_r\right)\right] \times \\ &\cdots \times \left[\underline{d}_l, \overline{d}_l\right] \end{aligned} \tag{26}$$

are obtained as $B(D_0) = B^{(n_r)}(D)$. The Bernstein coefficients $B(D_1)$ on the subbox D_1 given by

$$\begin{aligned} D_1 = &\left[\underline{d}_1, \overline{d}_1\right] \times \cdots \times \left[\underline{d}_r + \lambda\left(\overline{d}_r - \underline{d}_r\right), \underline{d}_r\right] \times \\ &\cdots \times \left[\underline{d}_l, \overline{d}_l\right] \end{aligned} \tag{27}$$

are obtained as

$$b_{i_1,\dots,n_r-k,\dots i_l}(D_1) = b^{(k)}_{i_1,\dots,n_r,\dots i_l}(D) \tag{28}$$

The sweep procedure is repeated as long as positivity, negativity, nonpositivity or nonnegativity of $w(\mathbf{q})$ is proved for all the subregions of U or such two subregions are found that on one of them the tested polynomial is positive and on another of them is negative.

The presented algorithm was used to verify robust stability of automatic steering control of the Daimler Benz city bus.

6. EXAMPLE

The Daimler-Benz city bus is an example of automatic car steering system (Ackermann, 1993). A guiding wire in the street may play the role of the planned path. The magnetic field from the guiding wire is measured by a sensor at the front end of the vehicle in order to determine the lateral deviation from the guiding wire. The reference trajectory may also be calculated from the data of a TV camera. The deviation is kept small by feedback control via the steering motors.

In the example two-wheel car steering (2WS) is considered. It means that the reference trajectory is kept small only by front wheel steering, the rear wheel angle remains constant. The pair of wheels is steered by the same angle. It is supposed that the front wheel steering angle is generated by an integrating actuator.

The transfer function of the uncontrolled bus from the front wheel angle Δ_f [rad] to the measured displacement from the guiding wire y [m] obtained by linearization yields

$$G(s,m,v) = \frac{Y(s)}{\Delta_f(s)} = \frac{b(s,m,v)}{a(s,m,v)}$$

$$\begin{aligned} b(s,m,v) &= 6.076 \cdot 10^5 mv^2 s^2 + 3.886 \cdot 10^{11} vs \\ &\quad + 4.803 \cdot 10^{10} v^2 \\ a(s,m,v) &= s^3 (m^2 v^2 s^2 + 1.075 \cdot 10^6 mvs \\ &\quad + 1.663 \cdot 10^4 mv^2 + 2.690 \cdot 10^{11}) \end{aligned} \tag{29}$$

where m [kg] is the mass of the bus and v [m/s] is its velocity. The mass of the bus and the velocity are considered as uncertain parameters of the system with the values lying inside the intervals $m \in [9950,32000]$ and $v \in [1,20]$.

In (Muench, 1996) the constant controller with the transfer function $R(s)$ was designed:

$$R(s) = \frac{2.348 \cdot 10^3 s^2 + 1.094 \cdot 10^4 s + 9.375 \cdot 10^3}{s^3 + 50 s^2 + 1.25 \cdot 10^3 s + 1.563 \cdot 10^4}. \tag{30}$$

The controller (7) with the uncontrolled bus (6) leads to the closed-loop polynomial of the 8-th order

$$p(s,m,v) = \sum_{i=0}^{8} a_i(m,v) s^i \tag{31}$$

with both uncertain parameters entering quadratically to the coefficients that are given as

$$\begin{aligned} a_0(m,v) &= 4.503 \cdot 10^{14} v^2 \\ a_1(m,v) &= 5.253 \cdot 10^{14} v^2 + 3.625 \cdot 10^{15} v \\ a_2(m,v) &= 5.699 \cdot 10^9 mv^2 + 1.128 \cdot 10^{14} v^2 + 4.299 \cdot 10^{15} v \\ a_3(m,v) &= 6.908 \cdot 10^9 mv^2 + 9.062 \cdot 10^{14} v + 4.203 \cdot 10^{15} \\ a_4(m,v) &= 1.448 \cdot 10^9 mv^2 + 1.680 \cdot 10^{10} mv + 3.363 \cdot 10^{14} \\ a_5(m,v) &= 1.563 \cdot 10^4 m^2 v^2 + 8.315 \cdot 10^5 mv^2 \\ &+ 1.344 \cdot 10^9 mv + 1.345 \cdot 10^{13} \\ a_6(m,v) &= 1.25 \cdot 10^3 m^2 v^2 + 1.663 \cdot 10^4 mv^2 \\ &+ 5.376 \cdot 10^7 mv + 2.690 \cdot 10^{11} \\ a_7(m,v) &= 50 m^2 v^2 + 1.075 \cdot 10^6 mv \\ a_8(m,v) &= m^2 v^2. \end{aligned} \tag{32}$$

The question is whether the controller (30) stabilizes the bus for all admissible values of the mass m and velocity v, i.e., whether the closed-loop polynomial (31) is robustly Hurwitz stable.

The determinant of the 8-th order Hurwitz matrix was computed using the presented algorithm based on evaluation-interpolation techniques in less than 0.01s. The determinant

$$c(m,v) = \det\left(\mathbf{H}_8(m,v)\right) = \sum_{i_1=0}^{16} \sum_{i_2=0}^{16} c_{i_1 i_2} m^{i_1} v^{i_2} \tag{33}$$

contains 225 nonzero coefficients, about one third of them is negative.

The Bernstein algorithm reports after 1 sweep in less than 0.01s that the determinant $c(m, v)$ is positive on the given uncertainty set. The obtained result corresponds with result reported in (Ackermann, 1993) achieved in 11s using symbolic packages for determination of the Hurwitz determinant. All the computations were performed on a Pentium 4 CPU 3GHz 504MB RAM.

7. CONCLUSIONS

An improved algorithm for robust stability analysis of systems with polynomic parameter uncertainty is presented in this paper. The proposed method is based on Hurwitz stability criterion. The main step consisting in computing determinant of multivariate polynomial matrix was performed by very efficient and numerically reliable algorithm based on interpolation methods. Positivity of the determinant of Hurwitz matrix is tested by Bernstein algorithm. The proposed algorithm was used to verify robust stability of automatic steering control of Daimler Benz city bus.

ACKNOWLEDGEMENTS

This work has been supported by the project INGO 1P2007LA297, Research Program MSM6840770038 (sponsored by the Ministry of Education of the Czech Republic), the project 1H-PK/22 (sponsored by Ministry of Industry and Trade of the Czech Republic) and the project GACR 102/05/0903 (sponsored by the Grant Agency of the Czech Republic).

REFERENCES

Ackermann, J. (1993). *Robust control, systems with uncertain physical parameters*, Springer, London.

Bartlett A.C., C.V. Hollot and H. Lin (1988). Root Location of an Entire Polytope of Polynomials: It suffices to check the edges. In: *Mathematics of Controls, Signals and System*., **Vol. 1**, pp. 61-71.

Bini D. and V. Pan (1994). Polynomial and Matrix Computations. In: *Fundamental Algorithms*, Birkhäuser, Boston.

Garloff J. (1985). Convergent Bounds for the Range of Multivariate Polynomials. *Interval Mathematics, Lecture Notes in Computer Science*, **Vol. 212**, (K. Nickel, Ed.), pp. 37-56. Springer Verlag Berlin.

Garloff J. (1993). The Bernstein algorithm. In: *Interval Computations*, **Vol. 2**, pp. 154-168.

Garloff J., B. Graf and M. Zettler (1997). Speeding Up an Algorithm for Checking Robust Stability of Polynomials. Fachhochschule Konstanz, Internal Report.

Muench R. (1986). Reglerauslegung und Robustheitsanalyse fuer einen spurgefuehrten Bus. Rep. IB 515-86-14, DLR, Oberpfaffenhofen, Germany.

Харитонов В.Л. (1978). Об Асимптотичесной Устойчивости Положения Равновесия Семейства Систем Линейных Дифференциальных Уравнений. In: *Дифференциальные Уравнения,* **Vol. 14**, pp. 2086-2088.

Zadeh L.A. and C. A. Desoer (1963). *Linear Systems Theory*. McGraw Hill Book Co., New York, NY.

Zettler M. and J. Garloff (1998). Robustness Analysis of Polynomials with Polynomial Parameter Dependency Using Bernstein Expansion. In: *IEEE Transactions on Automatic Control*, **Vol. 43**, pp. 425-431.