Purpose

Value set of parametric polynomial
(A tool for robust stability testing via Zero Exclusion Condition).

Syntax

```
V = vset(q1,...,qm,ExpressionString,p0,p1,…,pn[,omega][,qType])

vsetplot(V[,PlotType][,'new'])
```

Description

The command

```
V = vset(q1,...,qm,ExpressionString,p0,p1,…,pn[,omega[,qType]])
```

computes values at generalized frequencies given by the vector ω of a family of polynomials depending on m independent parameters. The selected parameter values are given by the vectors $q_1,\ldots,q_m$ and the results are stored in a matrix V of complex numbers with values at the particular frequencies organized column wise. The arguments $p_0,\ldots,p_n$ are given fixed polynomials that define the family. Its uncertainty structure is described by the string variable `ExpressionString.` This string is a Matlab-syntax expression for $a_0(q_1,...,q_m)p_0+\cdots+a_n(q_1,...,q_m)p_n$ that is composed of the parameter names and of the fixed polynomials' names. The "coefficients" $a_i(q_1,...,q_m)$ is any Matlab-syntax expression consisting of the parameter names acting here as scalar symbols. Beware that the input arguments representing both the parameters and the fixed polynomials must already exists in the current workspace and, moreover, must be written using their names (rather than values) in the function call. The command use is further explained in the examples below.

Once the value matrix V is ready, one can plot it typing

```
vsetplot(V[,PlotType][,'new'])
```

The plot consists of the sets $V(\omega_i)$ of values for particular generalized frequencies. Depending on the optional argument `PlotType, they` can be composed of lines (default or `PlotType='lines'`) or points (`PlotType='points'`). With the input string argument `'new'`, the plot is displayed in a new window.

By default or when the string argument `qType='r'`, the grid consists of combinations of entries in the vectors $q_1,\ldots,q_m$. When `qType='e'`, then the grid consists l particular points defined by their coordinates in m-dimensional space, all the $q_1,\ldots,q_m$ must be of the same length l..

vset, vsetplot

Scope

This couple of macros is aimed to test robust stability of the polynomial family via Zero Exclusion Condition [1]. If the family contains a stable member and if the value set for all generalized frequencies on the stability region boundary excludes the point 0, then the family is concluded to be robustly stable (stable for all parameters ranging given intervals). For more details, see [1] or another robust control textbook.

To perform the robust stability test, we first find a stable member in the family. Typically, the nominal value is stable or we proceed by trial and error. Once a stable member is found, we substitute into the family several generalized frequencies from the particular stability boundary and plot the corresponding value sets. It is important to use frequencies leading to value set close to the point 0. If none of the sets contains or touches the critical point, the robust stability is verified.

To plot value sets for special uncertainty structures such as polytopic or even interval uncertainty, more efficient macros are available, namely `ptopplot` and `khplot`, respectively.

Examples

To familiarize the command use, go through the following simple examples.

Example 1: Continuous-time case

Consider an uncertain polynomial

$$p(s,q_1,q_2) = p_0(s) + q_1 p_1(s) + q_2 p_2(s) + q_1 q_2 p_{12}(s)$$

composed of four fixed polynomials

$$p_0 = 1.853 + 3.164s + 2.871s^2 + 2.56s^3 + s^4$$

$$p_1 = 3.773 + 4.841s + 2.06s^2 + s^3$$

$$p_2 = 1.985 + 1.561s + 1.561s^2 + s^3$$

$$p_{12} = 4.032 + 1.06s + s^2$$

and check its robust stability for $q_1 \in [0,1]$ and $q_2 \in [0,3]$. To this end, first enter the data

```
p0=pol([1.853 3.164 2.871 2.56 1],4);
p1=pol([3.773 4.841 2.06 1],3);
p2=pol([1.985 1.561 1.561 1],3);
p12=pol([4.032 1.06 1],2);
```

describe the uncertainty structure

```
expr='p0+q1*p1+q2*p2+q1*q2*p12'
```

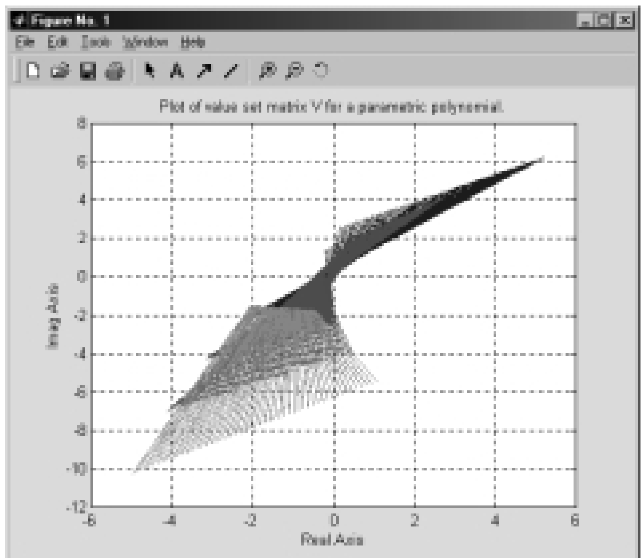
and set a reasonable gridding for the parameter intervals

```
q1=0:1/50:1;q2=0:3/50:3;
```

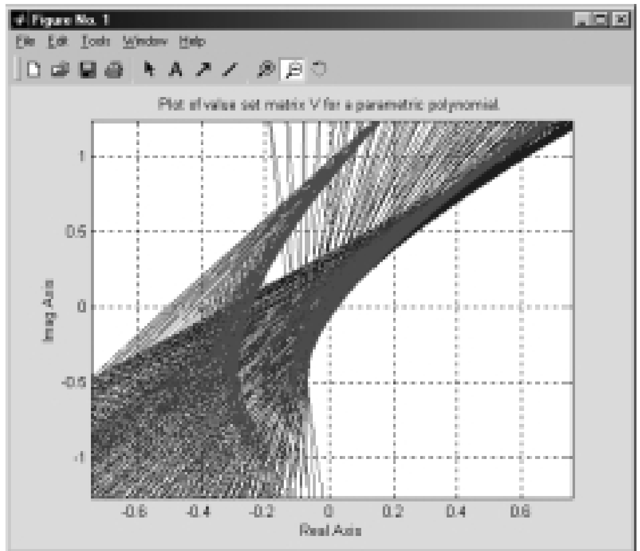
As the polynomials are of continuous-time nature, it is necessary to plot value sets for several critical frequencies on imaginary axis. So pick $\omega_i = 1.3, 1.4, 1.5, 1.6$ and type

```
V=vset(q1,q2,expr,p0,p1,p2,p12,j*[1.3:.1:1.6]);
vsetplot(V,'points')
```

to get



Note that the value sets are not convex. This typically happens whenever the uncertainty structure is multilinear or more complex. As one of the value sets (namely for $\omega_i = 1.4$) seems to include the critical point 0, you better zoom the figure in to see more details:



Now it is evident that $0 \in V(1.4)$ and hence the family is *not* robustly stable.

Example 2: Discrete-time case

Now consider a family of discrete-time polynomials with quite complicated uncertainty

$$p(z^{-1}, k, l, m) = e(z^{-1}) + \sin(k) f(z^{-1}) - \cos(k) k g(z^{-1}) + l^2 h(z^{-1})$$

where

$$e(z^{-1}) = (z^{-1} - 1.5)(z^{-1} + 2)(z^{-1} - 2)$$

$$f(z^{-1}) = 1$$

$$g(z^{-1}) = z^{-1}$$

$$h(z^{-1}) = z^{-2}$$

and $k, l, m \in [-1, 1]$. Here the data to be entered are

```
e=(zi-1.5)*(zi+2)*(zi-2);f=1; g=zi; h=zi^2;
uncrty='e+sin(k)*f-cos(m)*k*g+(l^2)*h';
```

and, say,

```
k=-1:.1:1;l=k;m=k;
```

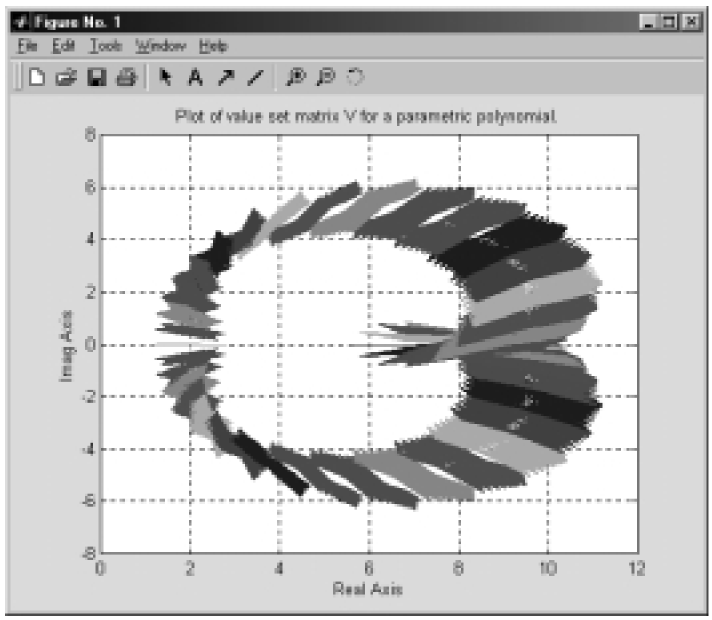
Before using the Zero Exclusion Condition to test robust stability, you must check that the family contains at least one stable member. Indeed, here the nominal polynomial $p(z^{-1},0,0,0)=e(z^{-1})$ is stable:

```
isstable(e)
ans =
     1
```

Now you evaluate and plot value sets at 40 generalized frequencies evenly spread around unit circle

```
V=vset(k,l,m,uncrty,e,f,g,h,exp(j*(0:2*pi/40:2*pi)));
vsetplot(V)
```

and get the following picture



As all the sets are far enough to the right from the critical point, the robust stability is verified.

Example 3: Incorrect calls

The user must not forget about calling the function with named variable arguments.

Even if the parameters

```
q0=1:5;
```

already exist in the workspace it must be represented by its name. The following call is definitely incorrect

```
vset(1:5,'q0*p',p,j)
??? Error using ==> vset
Undefined function or variable 'q0'.
```

Algorithm

The method is quite easy: Overall picture is composed of value sets for particular generalized frequencies. Each of them is achieved by substituting into the uncertainty formula of the frequency and all parameter values achieved by gridding of the parameter set.

Diagnostics

The macro `vset` displays an error messages if

- The set of generalized frequencies is not a non-empty vector.
- There are not enough input arguments.
- The expression string cannot be correctly evaluated. Here the error message is returned by lasterr and hence its text may vary according to the particular inconsistency encountered.

The macro `vsetplot` displays an error messages if

- The value set matrix is not a non-empty 2-dimensional double.
- An inappropriate input string argument is used.

See also

`khplot`	Value set for an interval polynomial.
`ptopplot`	Value set for a polytope of polynomials.

References

R. Barmish: *New Tools for Robustness of Linear Systems*. Macmillan Publishing Company. New York, 1994.