Purpose

Use Tsypkin-Polyak function to determine the l_2 robustness margin for a continuous spherical polynomial family with interval uncertainty structure.

Syntax

```
R = tsyp2(p0,omega,weight)
R = tsyp2(p0,omega)
R = tsyp2(p0)
R = tsyp2(p0,[],weight)
[R,OMEGA] = tsyp2(p0)
tsyp2(p0,omega,weight)
tsyp2(p0,omega)
tsyp2(p0)
tsyp2(p0,[],weight)
```

Description

Given the nominal polynomial p0, the macro finds a robustnes margin R such that the resulting interval polynomial described by

$$p_R(s,\mathbf{q}) = p_0(s) + \sum_{i=0}^{n} q_i s^i$$

is robustly stable iff the weighted Euclidian norm of the uncertainty vector satisfies the following bound

$$\|\mathbf{q}\|_{2,\mathbf{W}} \le R$$

`R = tsyp2(p0,omega,weight)` computes the robustness margin for an interval polynomial `p0` over the frequencies given by the vector `omega` and with weighting matrix diagonal entries given by the vector `weight`.

`R = tsyp2(p0,omega)`assumes that the weighting matrix **W = I**.

`R = tsyp2(p0)`assumes that the weighting matrix **W = I** and it supplies its own vector of frequencies.

tsyp2

`R = tsyp2(p0,[],weight)` requires the weighting vector but computes its own frequency vector

`[R,OMEGA] = tsyp2(p0)` and

`[R,OMEGA] = tsyp2(p0,[],weight)` return the computed vector of frequencies as the second output for possible use with `spherplot` function.

If no output is specified, the graphical output of Tsypkin-Polyak function and the robustness margin disc is generated.

Examples

Example 1

Consider the nominal polynomial [2], pp.147:

```
p0 = 433.5 + 667.25*s + 502.25*s^2 + 251.25*s^3 + ...
80.25*s^4 + 14*s^5 + s^6;
```

The diagonal entries of the weighting matrix are given

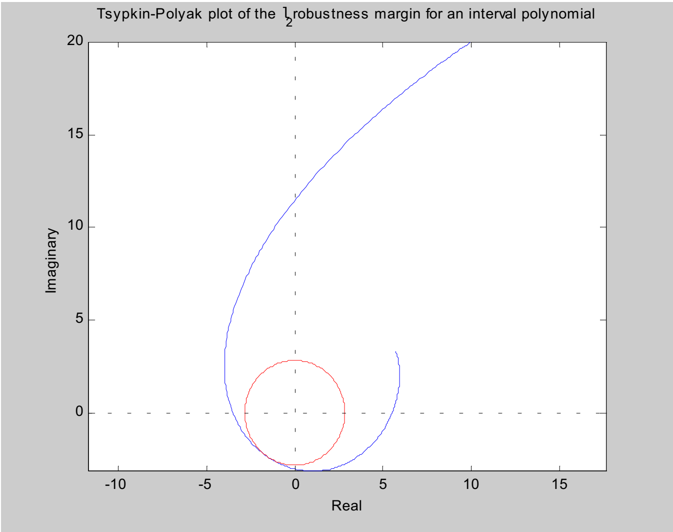
```
weight  = [43.5, 33.36, 25.137, 15.075, 5.6175, 1.4,...0.1].^2;
```

The vector of frequencies is chosen as

```
omega = 0:0.01:5;
```

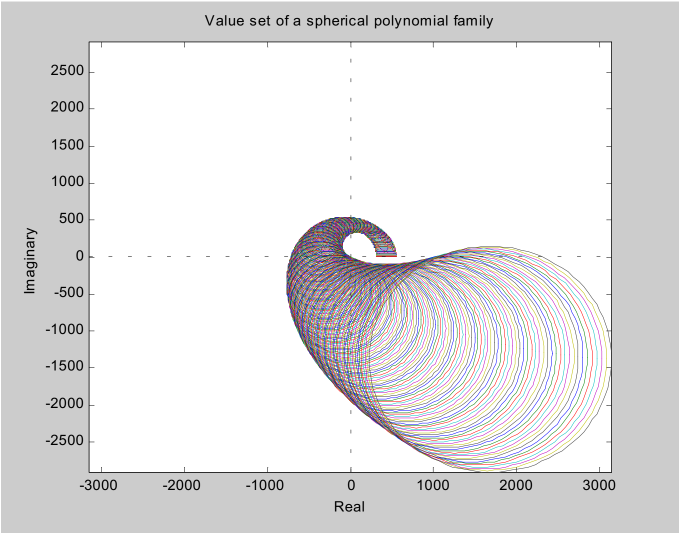
In order to compute the bound on the norm on the uncertainty vector that still preserves stability just type

```
tsyp2(p0,omega,weight)
ans =
    2.8312
```



Now it is possible to check the result by invoking the graphical routine SPHERPLOT and visually check the Zero Exclusion Condition.

```
spherplot(p0,0:0.01:3,2.8312,weight)
```



tsyp2

Algorithm

Based on the l_2 version of complex-valued Tsypkin-Polyak function $G_{TP}(\omega)$ described in [2], pp.146. It uses standard Matlab FMINBND minimization routine to find the minimum of absolute value of such function. Actually, this result is also known as Soh-Berger-Dabke theorem, see [1], pp.267.

Diagnostics

Since the quality of the result of the minimization routine depends considerably on quality of the initial guess, the proper choice of the frequency range is important. The program by no means automatically validates its result, and therefore it is necessary to examine the graphical output and/or use the SPHERLOT routine to test visually for Zero Exclusion Condition violation. If problems are encountered, it is advisable to run the routine again with extended frequency range and/or denser gridding.

See also

`tsyp`	compute the l_∞ robustnes margin for a continuous interval polynomial family.
spherplot	plot the value set for a spherical polynomial family for a range of frequencies.

References

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

[2] Bhattacharyya, S.P., Chapellat, H., Keel, L.H. *Robust Control: The Parametric Approach*, Prentice Hall, 1995.