tsyp

Purpose Determine l_∞ robustness margin for a continuous interval polynomial using Tsypkin-Polyak function.

Syntax

```
R = tsyp(p0,w,epsilon)
R = tsyp(p0,w)
R = tsyp(p0)
R = tsyp(p0,[],epsilon)
[R,W] = tsyp(p0)

tsyp(p0,w,epsilon)
tsyp(p0,w)
tsyp(p0)
tsyp(p0,[],epsilon)
```

Description Given the nominal polynomial $p_0(s)$ the macro finds a robustnes margin R such that the resulting interval polynomial

$$p_R(s,q) = p_0(s) + R\sum_{i=0}^{n}[-\varepsilon_i,\varepsilon_i]s^i$$

is robustly stable.

`R = tsyp(p0,w,epsilon)` returns the robustness margin for an interval polynomial w given by `p0` with scale factors specified by the vector `epsilon` . The computation is done at frequencies given by the vector `w` .

`R = tsyp(p0,w)`uses implicitly the coefficients of the nominal polynomial `p0` as scale factors.

`R = tsyp(p0)`uses implicitly the coefficients of the nominal polynomial `p0` as scale factors and supplies its own vector of frequencies.

`R = tsyp(p0,[],epsilon)` uses the supplied scale factors but computes its own frequency vector

`[R,W] = tsyp(p0) and [R,W] = tsyp(p0,[],epsilon)` return the computed vector of frequencies as the second output for possible use with function `khplot`.

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

tsyp

Examples

Example 1

For interval polynomial family P_r ([1], pp.99) with nominal polynomial

$$p_0(s) = 676 + 1365s + 1019s^2 + 420s^3 + 104s^4 + 15s^5 + s^6$$

and scaling factors

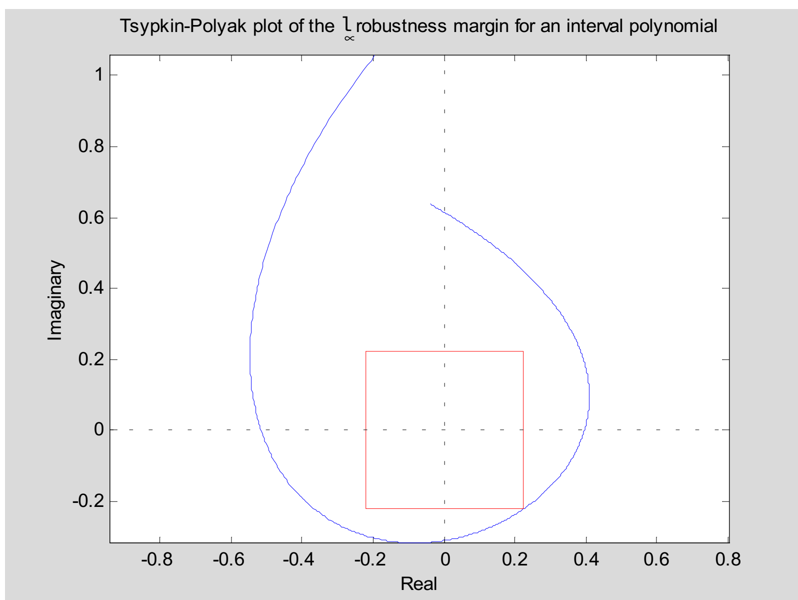
$$\varepsilon_0 = 676,\ \varepsilon_1 = 682.5, \varepsilon_2 = 509.5, \varepsilon_3 = 210, \varepsilon_4 = 52, \varepsilon_5 = 15, \varepsilon_6 = 1$$

we can find a robustness margin R such that the resulting interval polynomial is robustly stable. We enter the data

```
p0 = pol([676 1365 1019 420 104 15 1],6);
w = 1:0.01:10;
epsilon = [676 682.5 509.5 210 52 15 1];
```

and then simply type

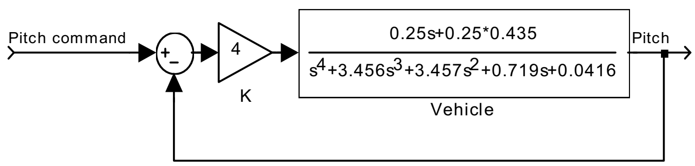
```
tsyp(p0,w,epsilon)
ans =
    0.2218
```



Thus we got the robustness margin R = 0.2218 that can be viewed as the largest possible radius of a box inscribed into the plot of the Tsypkin-Polyak function.

Example 2 - simple feedback

The nominal pitch control system ([1], pp.101) is described in the following figure. Find the robustness margin for K=4.



With the data

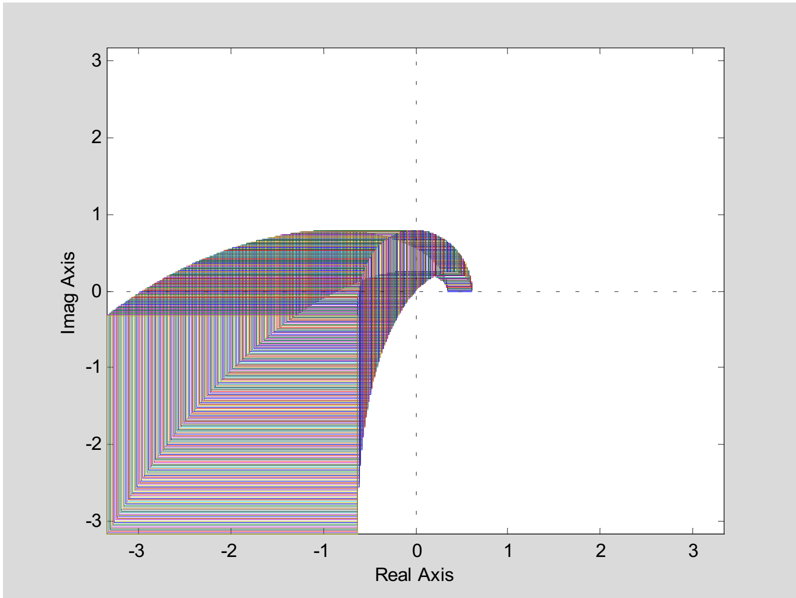
```
K = 4;
num = pol([0.25*0.435 0.25],1);
den = pol([.0416 .719 3.457 3.456 1],4);
p0 = den + K*num;
```

we compute the margin

```
[R,W] = tsyp(p0); R
R =
    0.2746
```

The margin is easy to verify by the function `khplot` :

```
pminus = p0 - R*p0;
pplus = p0 + R*p0;
khplot(pminus, pplus, W/3)
```



Thus we have found the robustness margin R and now it is easy to find the uncertainty bounds on the coefficients of the polynomial:

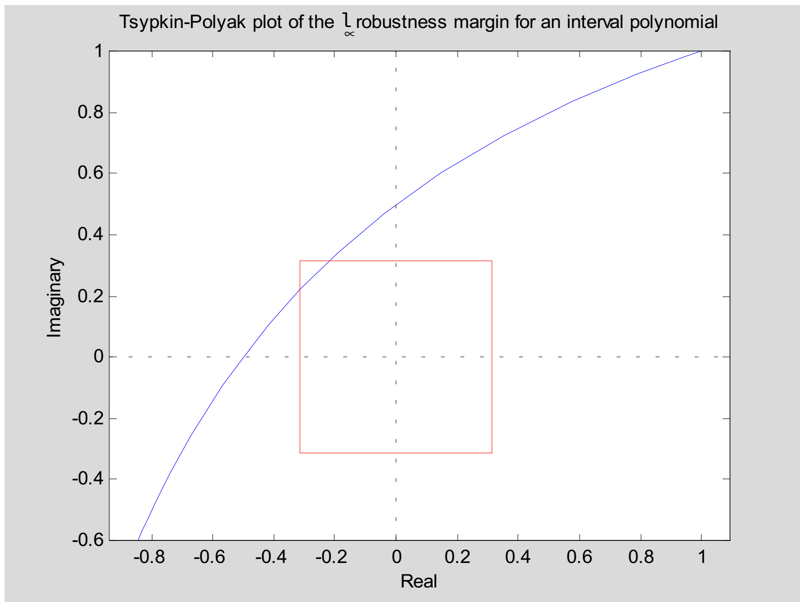
```
Qbounds = [pminus{:}' pplus{:}']

Qbounds =
    0.3457    0.6075
    1.2469    2.1911
    2.5077    4.4063
    2.5069    4.4051
    0.7254    1.2746
```

If the coefficients remain within these intervals, the polynomial is guaranteed to be stable.

Example 3 - Improper choice of the frequency vector
Sometimes, the frequence vector is not well chosen to gurantee robust stability. So for

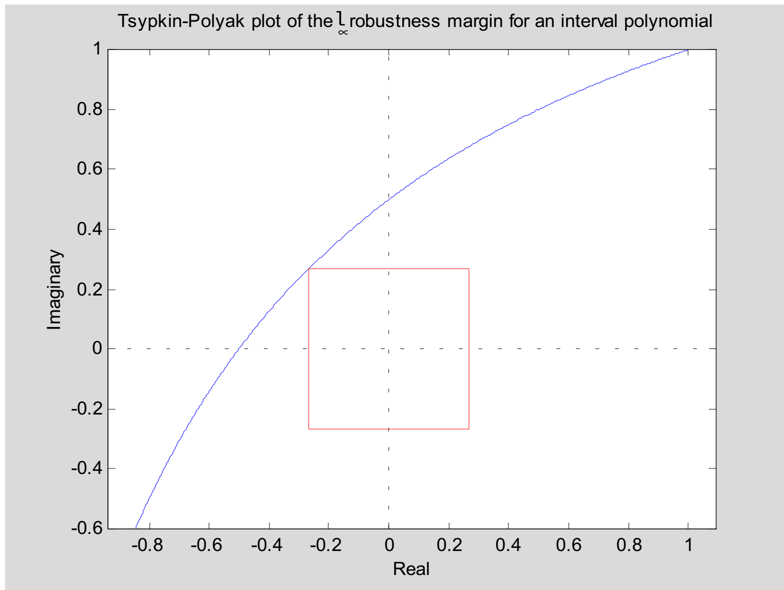
```
p0 = 1 + s + 3*s^2 +s^3;
tsyp(p0,0:.1:2)
```

```
Warning: Resulting margin does not guarantee robust stability of the
interval polynomial. Run again with extended frequency range and/or
denser gridding.
ans =
    0.3141
```



The problem was detected by the macro and a warning was issued. To remedy this situtaion, we can rerun the function with a denser ferquency grid. This will help the minimization routine make better initial guess:

```
tsyp(p0,0:.001:2)

ans =
    0.2672
```



The resulting $R = 0.2672$ is more conservative but guarantees robust stability.

Algorithm

The macro is based on Tsypkin-Polyak function $G_{TP}(\omega)$ described in [1], pp. 97. It finds a robust margin R such that the condition $\|G_{TP}(\omega)\|_\infty > R$ is satisfied for all frequencies (where $\|z\|_\infty = \max\{|\mathrm{Re}\; z|, |\mathrm{Im}\; z|\},\; z \in C$). It uses standard Matlab FMINBND minimization routine.

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 automatically validates its result by testing robust stability of the four Kharitonov polynomials. If these are not robustly stable, a Warning appears: `"Resulting margin does not guarantee robust stability of the interval polynomial. Run again with extended frequency range and/or denser gridding."` One can also use the graphical output to assess the acceptability of the result.

See also

`kharit` return Kharitonov polynomials

`khplot` plot Kharitonov rectangles

References

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