Zur Vorlesung Signale+Systeme (MFB320) - Praktikum 5
Lösen von LZI Dgls mittels Laplace Transf.

von Prof. Dr. O.Wallrapp, FK06 HM - 30.11.2009 - erstellt mit Maple 12
update: 18.12.2009/Prof. Wallrapp

Gegeben ist die Dgl. 3. Ordnung

2 y''' + 4 y'' + 14 y' + 20 y = cos(omega*t) mit Anfangsbedg. y(0) = 0.5, y'(0) = 5, y''(0) = 1
und omega = 0.4 rad/s.
(aus Praktikum Prof. Dr. Steger 22.10.2007)

1. Bestimmen die Lösung der Dgl. im Bildbereich
2. Diskutieren die Lösung im Bildbereich: Pole, Stabilität, Anfangswert, Endwert.
3. Transformieren die Lösung in den Zeitbereich und plotten diese (bis 40 s)
4. Plotte den Frequenzgang und diskutiere diese mit der Zeitlösung.

Verwende hierzu die Hinweise in *DGL-Bildbereich-Anhang.pdf.*

1. Lösung der Dgl. im Bildbereich

```
> restart:
  with(inttrans):          # Bib Integraltransform.
  with(plots):with(plottools):with(StringTools): # bib for
  graphics
```

```
> P := 2*s^3 + 4*s^2 + 14*s + 20;     # linke Seite Y-Teil
```

$$P := 2\,s^3 + 4\,s^2 + 14\,s + 20$$

```
> Q := simplify(laplace(cos(0.4*t),t,s));  # rechte Seite
```

$$Q := \frac{25.\,s}{25.\,s^2 + 4.}$$

```
> Arow := Vector[row]([2, 4, 14]);    #a3,a2,a1
  Smat     := Matrix ([[s^2, s, 1  ],
                      [s  , 1, 0  ],
                      [1  , 0, 0  ]]);
  Ivec     := Vector  ([0.5, 5, 1 ]);    #y0,y0',y0''
  P0       := Arow.Smat.Ivec;
```

$$Arow := \begin{bmatrix} 2 & 4 & 14 \end{bmatrix}$$

$$Smat := \begin{bmatrix} s^2 & s & 1 \\ s & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$Ivec := \begin{bmatrix} 0.5 \\ 5 \\ 1 \end{bmatrix}$$

$$P0 := 29.0 + 1.0\,s^2 + 12.0\,s$$

```
> Y := (Q+P0) / P;
  Y := normal(Y);                    # Lösung im Bildbereich
```

$$Y := \frac{\frac{25.\,s}{25.\,s^2 + 4.} + 29.0 + 1.0\,s^2 + 12.0\,s}{2\,s^3 + 4\,s^2 + 14\,s + 20}$$

$$Y := \frac{1}{2}\,\frac{73.\,s + 729.\,s^2 + 116. + 25.\,s^4 + 300.\,s^3}{(s^3 + 2.\,s^2 + 7.\,s + 10.)\,(25.\,s^2 + 4.)}$$

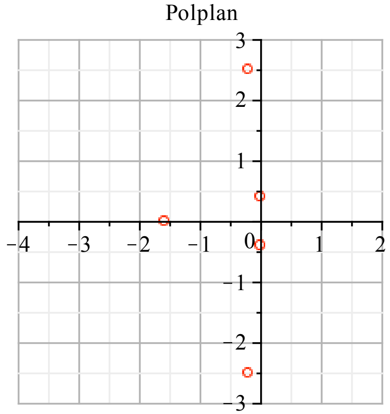
Nun können wir die Lösung diskutieren

2. Pole, Stabilität, Anfangswert, Endwert

```
> pole:=solve(denom(Y),s);
```

pole := −0.2107049309 + 2.508061000 I, −1.578590138, −0.2107049309 − 2.508061000 I, 0.4000000000 I, −0.4000000000 I

```
> complexplot([pole],style=point,          # Polplan
    symbol=circle,symbolsize=15,thickness=2,
    scaling=constrained,view=[-4..2,-3..3], title="Polplan",
  gridlines=true);
```



Das gesamte System ist stabil, da alle Pole links der Im -Achse. Die Anregung liefert das Paar +- 0.4 I, die linke Seite die anderen Werte.

Test der linken Seite mit Hurwitz-Kriterum:

```
> a3 := 2: a2 := 4: a1 := 14:  a0 := 20:
```

```
> if a2>0 and a1>0 and a0>0 and a1*a2-a3*a0>0 then stabil end
  if;
```

$$stabil$$

Anfangs- und Endwert:

```
> y0:=limit(s*Y,s=infinity);      # y für t -> 0
```

$$y0 := 0.5000000000$$

```
> yoo:=limit(s*Y,s=0);             # y für t -> oo
```

$$yoo := 0.$$

Hinweis: Null ist nicht korrekt, wie wir später sehen werden.!

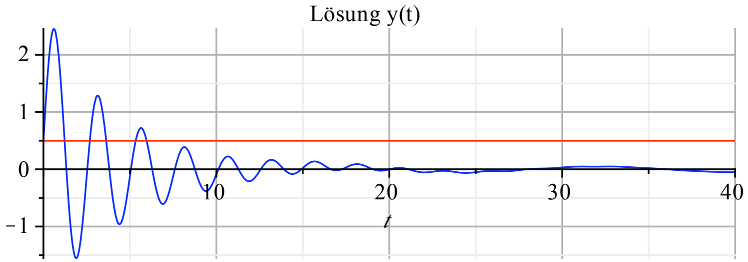
3. Lösung im Zeitbereich

Zurück in den Zeitbereich: => die Lösung der Dgl. im Zeitbereich.

```
> y := invlaplace(Y,s,t);
```

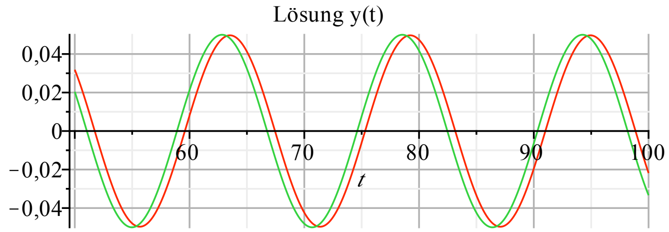
$$y := 0.7323186603\, e^{-1.578590138\, t} + (-0.1400751868 + 1.214403119\, I)\, e^{(-0.2107049309 - 2.508061000\, I)\, t} + (-0.1400751868 - 1.214403119\, I)\, e^{(-0.2107049309 + 2.508061000\, I)\, t} + 0.04783171333 \cos(0.4000000000\, t) + 0.01351937682 \sin(0.4000000000\, t)$$

```
> plot([y,y0],t=0..40,color=[blue,red], title="Lösung y(t)
  ",numpoints=500, gridlines=true);
```



Man erkennt in der Lösung die gedämpfte Schwingung mit omega = 2.5 rad/s (vgl. Polplot) v0n 0 .. 20 sec und die ungedämpfe langsame Schwingung omega = 0.4 rad/s, vgl. Polplot) im stationärem Zustand infolge der rechten Seite (Anregung). Bei 0.5 ist der Startwert y0.
Der Endzustand ist also die cos -Schwingung mit +1, 0, -1 und nicht null, wie y00 uns sagen wollte.

```
> plot([y,0.05*cos(0.4*t)],t=50..100,title="Lösung y(t)
  ",numpoints=500, gridlines=true);
```

Im eingeschwungenem Zustand haben wir eine Amplitude von ca. 0.05 und eine Phasenverschiebung.

4. Frequenzgang

Sinusantwort = Frequenzgang

u(t) = harmonische Schwingung $A\, e^{j\omega t}$, Amplitude A und Kreisfrequenz ω(rad/s).

=> $y(t=\infty) = G(j\omega) \cdot A\, e^{j\omega t}\big|_{t=\infty} = R(\omega) \cdot A \cdot e^{j(\omega t + \varphi(\omega))}$,

mit der Amplitudenverstärkung R(ω) = Amplitudengang

und der Phasenverschiebung φ(ω) = Phasengang

Anstelle der Anregung cos(0.4*t) nehmen wir A e^jωt und bertrachten t => oo

```
> NG := s-> 2*s^3 + 4*s^2 + 14*s + 20;
```

$$NG := s \rightarrow 2\, s^3 + 4\, s^2 + 14\, s + 20$$

```
> ZG := s-> 1;
```

$$ZG := s \rightarrow 1$$

```
> G:= ZG/NG;
```

$$G := \frac{ZG}{NG}$$

```
> G(I*omega);
```

$$\frac{1}{-2\, I\, \omega^3 - 4\, \omega^2 + 14\, I\, \omega + 20}$$

Amplitudengang:

```
> amp := evalc(abs(G(I*omega)));     # Amplitudenverstärkung
```

$$amp := \frac{1}{2\sqrt{-10\,\omega^4 + 9\,\omega^2 + 100 + \omega^6}}$$

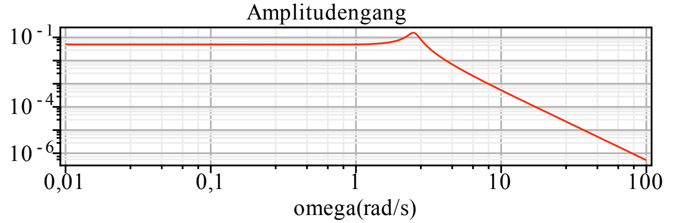
```
> amp_par := subs(amp);     # par sind bereits alle gesetzt,
  deshalb leer.
```

$$amp\_par := \frac{1}{2\sqrt{-10\,\omega^4 + 9\,\omega^2 + 100 + \omega^6}}$$

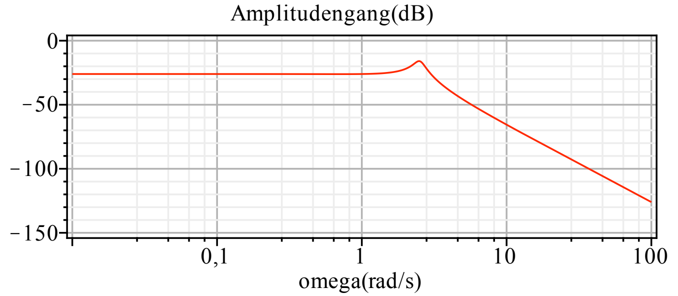
```
> amp_dB := simplify(20*log10(amp_par));
```

$$amp\_dB := -\frac{10\left(2\ln(2)+\ln\left(-10\,\omega^4+9\,\omega^2+100+\omega^6\right)\right)}{\ln(2)+\ln(5)}$$

```
> loglogplot([amp_par], omega=0.01..100, axes=boxed, title=
"Amplitudengang",labels=["omega(rad/s)",""], numpoints=400,
gridlines=true);
```




```
> semilogplot([amp_dB], omega=0.01..100, axes=boxed, title=
"Amplitudengang(dB)",labels=["omega(rad/s)",""], numpoints=
400, view=[0.01..100, -150..0], gridlines=true);
```



Finde das omega, wo amp ein Maximum hat, also das System eine Eigenresonanz hat.
Lösung: amp ableiten und erg. zu null setzen!

```
> omega0 := fsolve(diff(amp_par,omega),omega,1..4);
```

$$\omega 0 := 2.486228988$$

```
> R_omega0  := evalf(subs(omega=omega0, amp_par));
```

$$R\_omega0 := 0.1603337034$$

Vergl. Polplot: wir hatten das gedämpfte Pol-paar mit omega bei ca. 2.5 rad/s

```
> R0_dB := evalf(limit(amp_dB,omega=0));
```

$$R0\_dB := -26.02059991$$

```
> R0    := evalf(limit(amp_par,omega=0));
```

$$R0 := 0.05000000000$$

```
> R_w04 := eval(amp_par, omega=0.4);
```

$$R\_w04 := 0.04970559678$$

```
> R_w04_dB := evalf(subs(omega=0.4, amp_dB));
```

$$R\_w04\_dB := -26.07189415$$

Phasengang

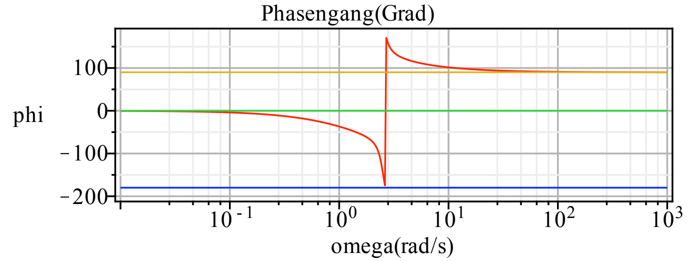

```
> pha := evalc(argument(G(I*omega)));     # Phasenverschiebung
```

$$pha := \arctan\left(2\,\omega^3-14\,\omega,\,-4\,\omega^2+20\right)$$

```
> pha_par := subs(pha);
```

$$pha\_par := \arctan\left(2\,\omega^3-14\,\omega,\,-4\,\omega^2+20\right)$$

```
> semilogplot([pha_par*180/Pi, 0, 90, -180], omega=0.01..1000,
axes=boxed, numpoints=400, title="Phasengang(Grad)",labels=
["omega(rad/s)","phi"], view=[0.01..1000, -200..180],
gridlines=true);
```



weitere Phasenwerte:

```
> pha_w04 := evalf(eval(pha_par*180/Pi, omega=0.4));
```

$$phi\_04 := -15.78265980$$

```
> pha_omega0 := evalf(eval(pha_par*180/Pi, omega=omega0));
```

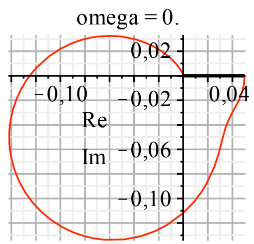
$$pha\_omega0 := -139.2557942$$

```
> pha_omega0 := evalf(eval(pha_par*180/Pi, omega=omega0+0.2));
```

$$pha\_omega0 := 172.5467667$$

Ortskurve mit Animation des Ortsvektors im Re-Im Ebene.

```
> bg := complexplot(subs(G(I*omega)), omega=0..10,
     scaling=constrained, labels=["Re","Im"], title="Ortskurve
",gridlines=true):   # Ortskurve
> G_vec := proc(omega)
        local x,y;
        x := Re(subs(G(I*omega))):
        y := Im(subs(G(I*omega))):
        display(line([0,0],[x,y],thickness=2),scaling=
constrained);
   end:
> animate(G_vec,[omega],omega=0..5,frames=60, background=bg);
```



vgl. die Zeitlösung für große Werte von t: bei omega = 0.4 ist R = 0.05 und

Ende Prakt 5