Starrkörperdynamik Lösungsblatt 5.1

Aufgabe 1:

Die Standardform des Anfangswertproblems ist

$$\begin{aligned}
\dot{\omega}_1 &= \frac{J_2 - J_3}{J_1}\omega_2\omega_3 &= f_1(\omega_1, \omega_2, \omega_3) \\
\dot{\omega}_2 &= \frac{J_3 - J_1}{J_2}\omega_3\omega_1 &= f_2(\omega_1, \omega_2, \omega_3) \\
\dot{\omega}_3 &= \frac{J_1 - J_2}{J_3}\omega_1\omega_2 &= f_3(\omega_1, \omega_2, \omega_3)
\end{aligned}$$

In Octave steht mit der Funktion `lsode` ein leistungsfähiges implizites Verfahren zur numerischen Integration von Differenzialgleichungssystemen zur Verfügung. Zur effizienten Lösung des dabei auftretenden Gleichungssystems wird neben der Funktion

$$[\boldsymbol{f}(\omega_1, \omega_2, \omega_3)] = \begin{bmatrix} f_1(\omega_1, \omega_2, \omega_3) \\ f_2(\omega_1, \omega_2, \omega_3) \\ f_3(\omega_1, \omega_2, \omega_3) \end{bmatrix}$$

auch die Jacobi-Matrix

$$[\boldsymbol{J}(\omega_1, \omega_2, \omega_3)] = \begin{bmatrix} \frac{\partial f_1}{\partial \omega_1} & \frac{\partial f_1}{\partial \omega_2} & \frac{\partial f_1}{\partial \omega_3} \\ \frac{\partial f_2}{\partial \omega_1} & \frac{\partial f_2}{\partial \omega_2} & \frac{\partial f_2}{\partial \omega_3} \\ \frac{\partial f_3}{\partial \omega_1} & \frac{\partial f_3}{\partial \omega_2} & \frac{\partial f_3}{\partial \omega_3} \end{bmatrix}$$

benötigt.

Die Elemente der Jacobi-Matrix berechnen sich wie folgt:

$$\frac{\partial f_1}{\partial \omega_1} = 0, \quad \frac{\partial f_1}{\partial \omega_2} = \frac{J_2 - J_3}{J_1}\omega_3, \quad \frac{\partial f_1}{\partial \omega_3} = \frac{J_2 - J_3}{J_1}\omega_2$$

$$\frac{\partial f_2}{\partial \omega_1} = \frac{J_3 - J_1}{J_2}\omega_3, \quad \frac{\partial f_2}{\partial \omega_2} = 0, \quad \frac{\partial f_2}{\partial \omega_3} = \frac{J_3 - J_1}{J_2}\omega_1$$

$$\frac{\partial f_3}{\partial \omega_1} = \frac{J_1 - J_2}{J_3}\omega_2, \quad \frac{\partial f_3}{\partial \omega_2} = \frac{J_1 - J_2}{J_3}\omega_1, \quad \frac{\partial f_3}{\partial \omega_3} = 0$$

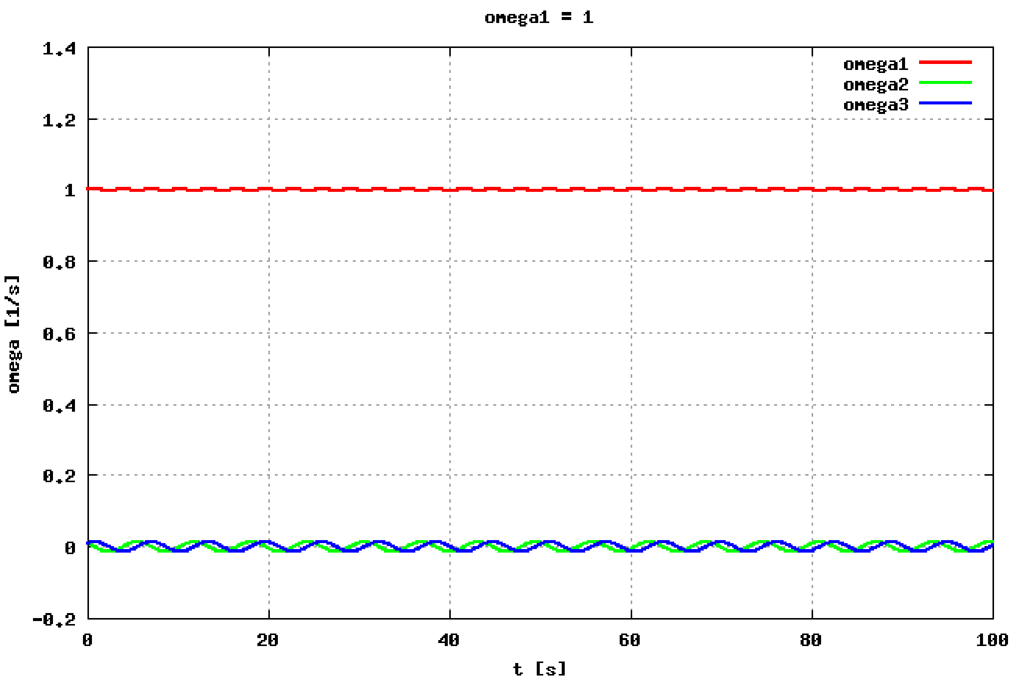
Programmcode für Octave:

```
# Uebungsblatt 5.1, Aufgabe 1: Kreiselgleichungen
#
# ------------------------------------------------------------

  global J;

# Massentraegheitsmomente

  J = [ 1500, 1000, 500];

# Zeitraum

  t0 =    0; % Anfang
  tE =  100; % Ende
  N  = 1000; % Anzahl der Zeitschritte

  t = linspace(t0, tE, N + 1);

# Kreiselgleichungen

  function y = f(x, t)
     global J;
     y(1) = (J(2) - J(3)) * x(2) * x(3) / J(1);
     y(2) = (J(3) - J(1)) * x(3) * x(1) / J(2);
     y(3) = (J(1) - J(2)) * x(1) * x(2) / J(3);
  endfunction

# Jacobi-Matrix

  function detJ = df(x, t)
     global J;
     J23 = (J(2) - J(3)) / J(1);
     J31 = (J(3) - J(1)) / J(2);
     J12 = (J(1) - J(2)) / J(3);
     detJ(1, 1) = 0;
     detJ(1, 2) = J23 * x(3);
     detJ(1, 3) = J23 * x(2);
     detJ(2, 1) = J31 * x(3);
     detJ(2, 2) = 0;
     detJ(2, 3) = J31 * x(1);
     detJ(3, 1) = J31 * x(2);
     detJ(3, 2) = J31 * x(1);
     detJ(3, 3) = 0;
  endfunction

  fcn = {"f"; "df"};

  set(0, "defaultlinelinewidth", 2);


# Fall 1: omega01 = 1
```

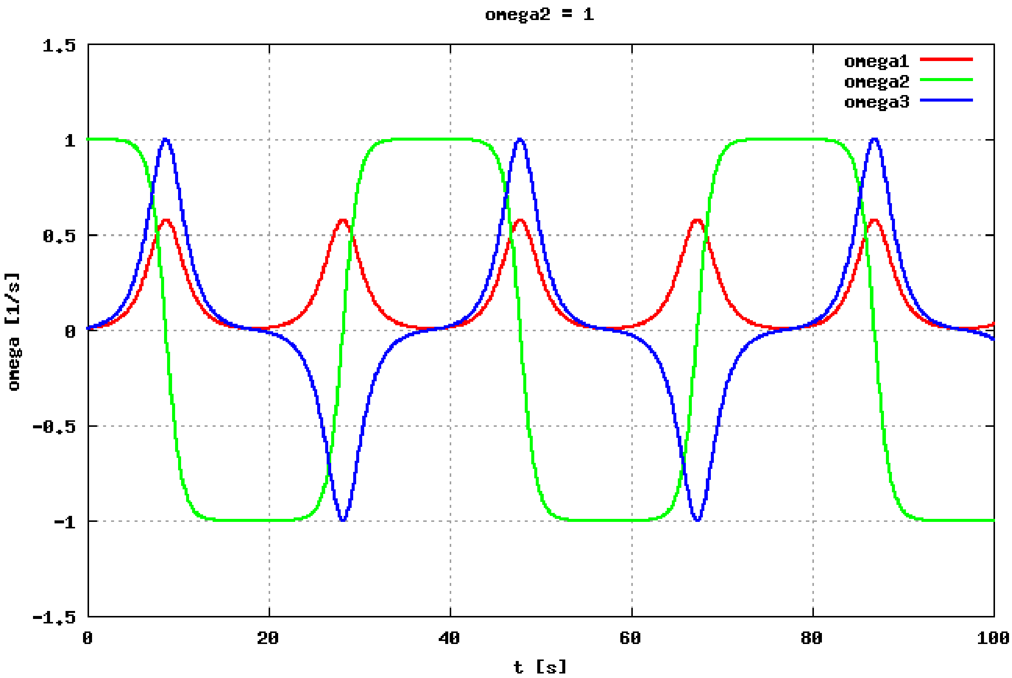
```
  x0 = [ 1.00, 0.01, 0.01];

  x = zeros(N + 1, 3);
  x = lsode(fcn, x0, t);

  figure(1, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 1), "color", "red", ...
       t, x(:, 2), "color", "green", ...
       t, x(:, 3), "color", "blue");

  legend("omega1", "omega2", "omega3");
  grid("on");
  xlabel("t [s]");
  ylabel("omega [1/s]");
  title("omega1 = 1");

  print("u5_1_1a.png", "-dpng");


# Fall 2: omega02 = 1

  x0 = [ 0.01, 1.00, 0.01];

  x = zeros(N + 1, 3);
  x = lsode(fcn, x0, t);

  figure(2, "position", [1100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 1), "color", "red", ...
       t, x(:, 2), "color", "green", ...
       t, x(:, 3), "color", "blue");

  legend("omega1", "omega2", "omega3");
  grid("on");
  xlabel("t [s]");
  ylabel("omega [1/s]");
  title("omega2 = 1");

  print("u5_1_1b.png", "-dpng");

# Fall 3: omega03 = 1

  x0 = [ 0.01, 0.01, 1.00];

  x = zeros(N + 1, 3);
  x = lsode(fcn, x0, t);

  figure(3, "position", [100, 100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);
```

```
plot(t, x(:, 1), "color", "red", ...
     t, x(:, 2), "color", "green", ...
     t, x(:, 3), "color", "blue");

legend("omega1", "omega2", "omega3");
grid("on");
xlabel("t [s]");
ylabel("omega [1/s]");
title("omega3 = 1");

print("u5_1_1c.png", "-dpng");
```

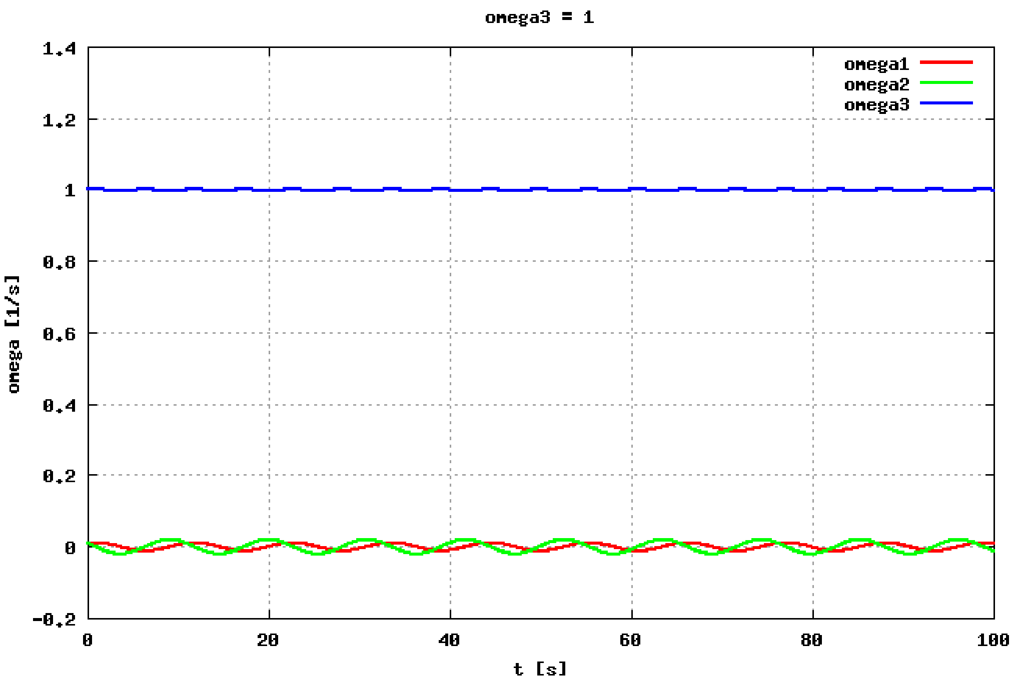
a) Ergebnisse für Drehung um die 1. Hauptachse:



b) Ergebnisse für Drehung um die 2. Hauptachse:



c) Ergebnisse für Drehung um die 3. Hauptachse:



Aufgabe 2:

Winkelgeschwindigkeiten und Euler-Winkel ergeben sich aus der numerischen Lösung des Differenzialgleichungssystems

$$\begin{bmatrix}\dot{\omega}_1\\ \dot{\omega}_2\\ \dot{\omega}_3\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}f_1(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\\ f_2(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\\ f_3(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\\ f_4(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\\ f_5(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\\ f_6(\omega_1,\omega_2,\omega_3,\phi,\theta,\psi)\end{bmatrix}=\begin{bmatrix}\dfrac{J_2-J_3}{J_1}\omega_2\omega_3\\ \dfrac{J_3-J_1}{J_2}\omega_3\omega_1\\ \dfrac{J_1-J_2}{J_3}\omega_1\omega_2\\ \omega_1+\sin(\phi)\tan(\theta)\omega_2+\cos(\phi)\tan(\theta)\omega_3\\ \cos(\phi)\omega_2-\sin(\phi)\omega_3\\ \dfrac{1}{\cos(\theta)}\left(\sin(\phi)\omega_2+\cos(\phi)\omega_3\right)\end{bmatrix}$$

Anschließend können die Komponenten des Drallvektors berechnet werden. Im körperfesten Koordinatensystem gilt

$$[\boldsymbol{L}]_B=\begin{bmatrix}L_1\\ L_2\\ L_3\end{bmatrix}=\begin{bmatrix}J_1\omega_1\\ J_2\omega_2\\ J_3\omega_3\end{bmatrix} \quad .$$

Die Umrechnung auf das ortsfeste Koordinatensystem erfolgt mit der Transformationsmatrix

$$[\boldsymbol{T}]_{OB}=\begin{bmatrix}\cos(\theta)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi)-\cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi)+\sin(\phi)\sin(\psi)\\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi)+\cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi)-\sin(\phi)\cos(\psi)\\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta)\end{bmatrix} ,$$

die zu jedem Zeitpunkt aus den Euler-Winkeln berechnet werden kann. Die Transformation lautet

$$[\boldsymbol{L}]_O=[\boldsymbol{T}]_{OB}[\boldsymbol{L}]_B \quad .$$

Ebenso lassen sich die Koordinaten der Punkte P_1, P_2 und P_3 im ortsfesten Koordinatensystem ermitteln. Für die gegebenen Koordinaten entsprechen die Koordinaten der Punkte gerade der ersten, zweiten und dritten Spalte der Transformationsmatrix.

<u>Programmcode für Octave:</u>

```
# Uebungsblatt 5.1, Aufgabe 2: Kreiselgleichungen
#
# Es werden auch die Euler-Winkel berechnet:
#   x(1) - x(3)  Winkelgeschwindigkeiten
#   x(4) - x(6)  Euler-Winkel phi, theta, psi
#
# -----------------------------------------------------------
```

```
  global J;

# Massentraegheitsmomente

  J = [ 1500, 1000, 500];

# Zeitraum

  t0 =    0; % Anfang
  tE =   50; % Ende
  N  = 1000; % Anzahl der Zeitschritte

  t = linspace(t0, tE, N + 1);

# Kreiselgleichungen und Kinematik

  function y = f(x, t)
     global J;
     y(1) = (J(2) - J(3)) * x(2) * x(3) / J(1);
     y(2) = (J(3) - J(1)) * x(3) * x(1) / J(2);
     y(3) = (J(1) - J(2)) * x(1) * x(2) / J(3);
     sphi = sin(x(4));
     cphi = cos(x(4));
     cth  = cos(x(5));
     tth  = tan(x(5));
     y(4) = x(1) + (sphi * x(2) + cphi * x(3)) * tth;
     y(5) = cphi * x(2) - sphi * x(3);
     y(6) = (sphi * x(2) + cphi * x(3)) / cth;
  endfunction

  fcn = "f";

  set(0, "defaultlinelinewidth", 2);


# Fall 1: omega01 = 1
# -------------------

  x0 = [ 1.00, 0.01, 0.01, 0, 0, 0];

  x = zeros(N + 1, 6);
  x = lsode(fcn, x0, t);
  x(:, 4:6) = rem(x(:, 4:6), 2 * pi);

  figure(1, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 1), "color", "red", ...
       t, x(:, 2), "color", "green", ...
       t, x(:, 3), "color", "blue");

  legend("omega1", "omega2", "omega3");
```

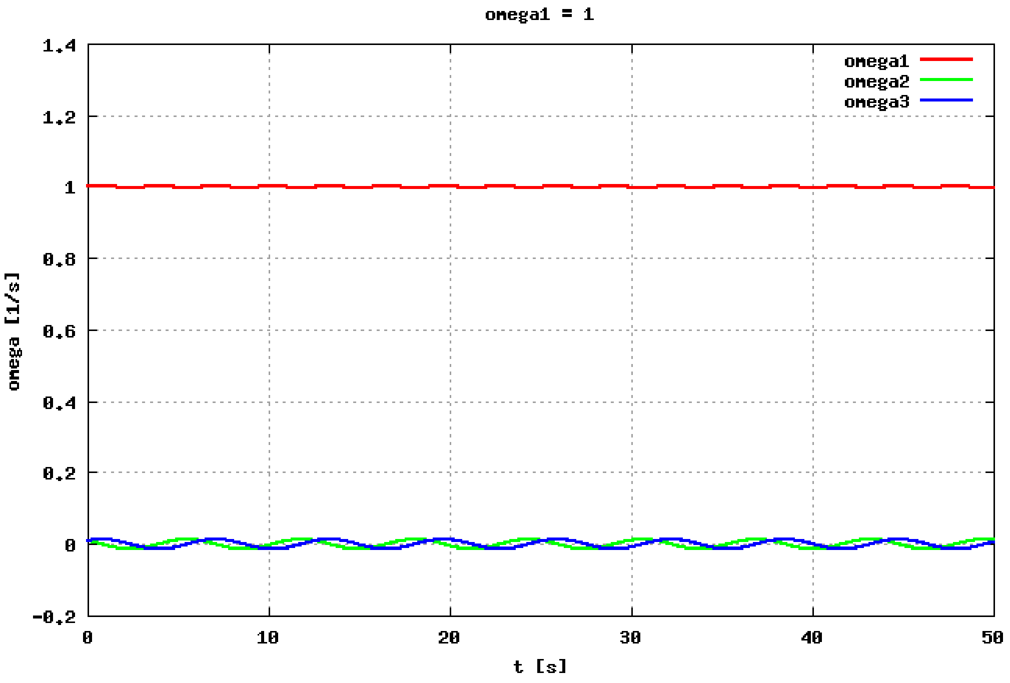
```
  grid("on");
  xlabel("t [s]");
  ylabel("omega [1/s]");
  title("omega1 = 1");

  print("u5_1_2a1.png", "-dpng");

  figure(2, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 4), "color", "red", ...
       t, x(:, 5), "color", "green", ...
       t, x(:, 6), "color", "blue");

  legend("phi", "theta", "psi");
  grid("on");
  xlabel("t [s]");
  ylabel("Winkel [rad]");
  title("omega1 = 1");

  print("u5_1_2a2.png", "-dpng");

# Drall im koerperfesten System

  LB = diag(J) * x(:, 1:3)';

# Drall im raumfesten System und Bahnen der Punkte

  for k = 1 : N + 1;
      TEB = eulmat(x(k, 4), x(k, 5), x(k, 6));
      LO(:, k)   = TEB * LB(:, k);
      P1(1:3, k) = TEB(:, 1);
      P2(1:3, k) = TEB(:, 2);
      P3(1:3, k) = TEB(:, 3);
  endfor

  figure(3, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, LO(1, :), "color", "red", ...
       t, LO(2, :), "color", "green", ...
       t, LO(3, :), "color", "blue");

  legend("LOx", "LOy", "LOz");
  grid("on");
  xlabel("t [s]");
  ylabel("LO [kg m^2 / s]");
  title("omega1 = 1");

  print("u5_1_2a3.png", "-dpng");

  figure(4, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 4.5, 3]);
```

```
  plot3(P1(1, :), P1(2, :), P1(3, :), "color", "red", ...
        P2(1, :), P2(2, :), P2(3, :), "color", "green", ...
        P3(1, :), P3(2, :), P3(3, :), "color", "blue");

  legend("P1", "P2", "P3");
  view(40, 30);
  grid("on");
  xlabel("x");
  ylabel("y");
  zlabel("z");
  title("omega1 = 1");

  print("u5_1_2a4.png", "-dpng");


# Fall 2: omega02 = 1
# -------------------

  x0 = [ 0.01, 1.00, 0.01, 0, 0, 0];

  x = zeros(N + 1, 6);
  x = lsode(fcn, x0, t);
  x(:, 4:6) = rem(x(:, 4:6), 2 * pi);

  figure(5, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 1), "color", "red", ...
       t, x(:, 2), "color", "green", ...
       t, x(:, 3), "color", "blue");

  legend("omega1", "omega2", "omega3");
  grid("on");
  xlabel("t [s]");
  ylabel("omega [1/s]");
  title("omega2 = 1");

  print("u5_1_2b1.png", "-dpng");

  figure(6, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, x(:, 4), "color", "red", ...
       t, x(:, 5), "color", "green", ...
       t, x(:, 6), "color", "blue");

  legend("phi", "theta", "psi");
  grid("on");
  xlabel("t [s]");
  ylabel("Winkel [rad]");
  title("omega2 = 1");
```

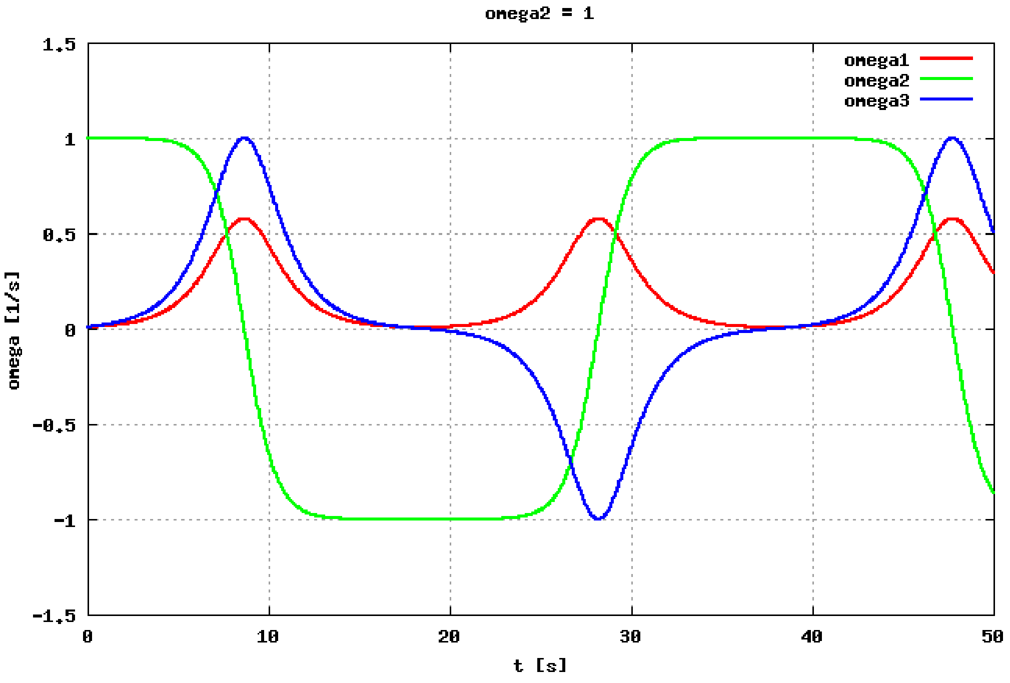
```
  print("u5_1_2b2.png", "-dpng");

# Drall im koerperfesten System

  LB = diag(J) * x(:, 1:3)';

# Drall im raumfesten System und Bahnen der Punkte

  for k = 1 : N + 1;
      TEB = eulmat(x(k, 4), x(k, 5), x(k, 6));
      LO(:, k)   = TEB * LB(:, k);
      P1(1:3, k) = TEB(:, 1);
      P2(1:3, k) = TEB(:, 2);
      P3(1:3, k) = TEB(:, 3);
  endfor

  figure(7, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 5, 3]);

  plot(t, LO(1, :), "color", "red", ...
       t, LO(2, :), "color", "green", ...
       t, LO(3, :), "color", "blue");

  legend("LOx", "LOy", "LOz");
  grid("on");
  xlabel("t [s]");
  ylabel("LO [kg m^2 / s]");
  title("omega2 = 1");

  print("u5_1_2b3.png", "-dpng");

  figure(8, "position", [100, 1100, 1000, 700], ...
            "paperposition", [0, 0, 4.5, 3]);

  plot3(P1(1, :), P1(2, :), P1(3, :), "color", "red", ...
        P2(1, :), P2(2, :), P2(3, :), "color", "green", ...
        P3(1, :), P3(2, :), P3(3, :), "color", "blue");

  legend("P1", "P2", "P3");
  grid("on");
  view(40, 30);
  xlabel("x");
  ylabel("y");
  zlabel("z");
  title("omega2 = 1");

  print("u5_1_2b4.png", "-dpng");
```

a) <u>Winkelgeschwindigkeiten:</u>

Fall 1:

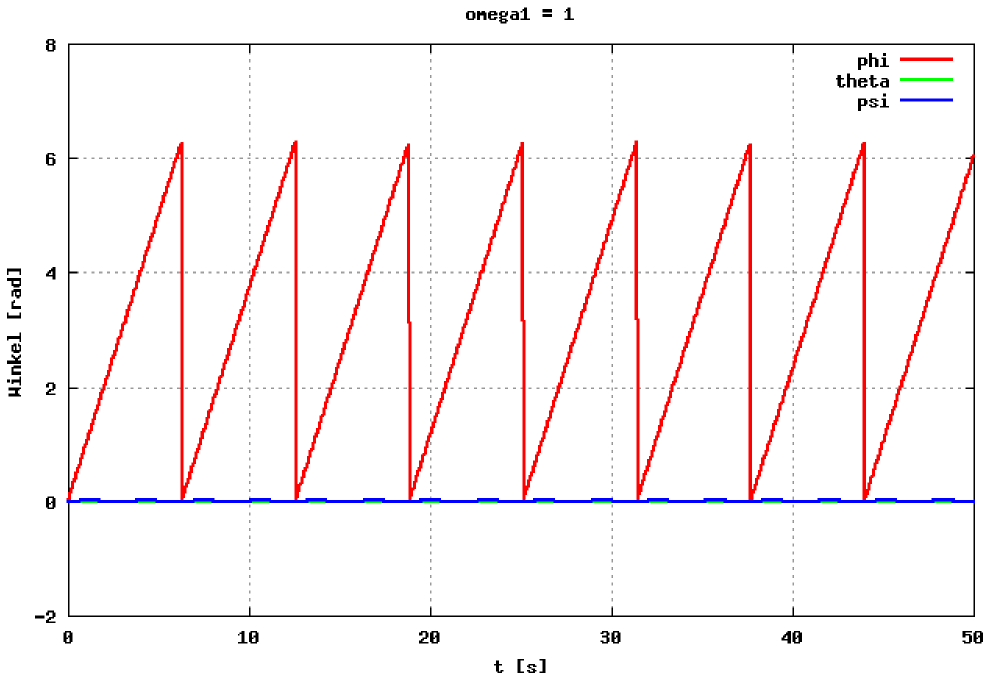


Fall 2:

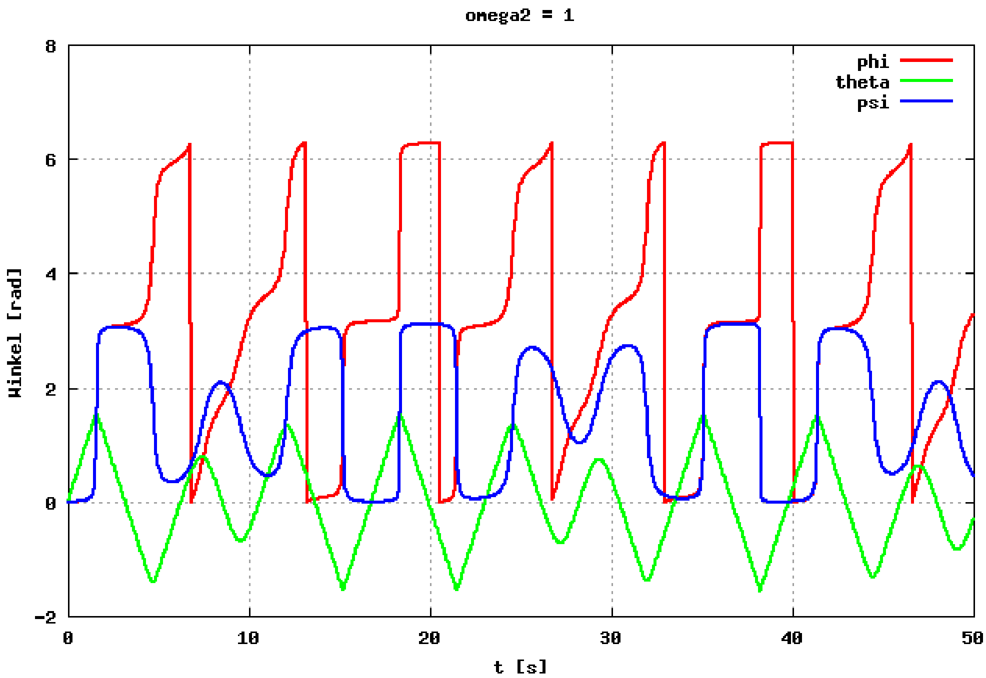


b) <u>Euler-Winkel:</u>

Fall 1:

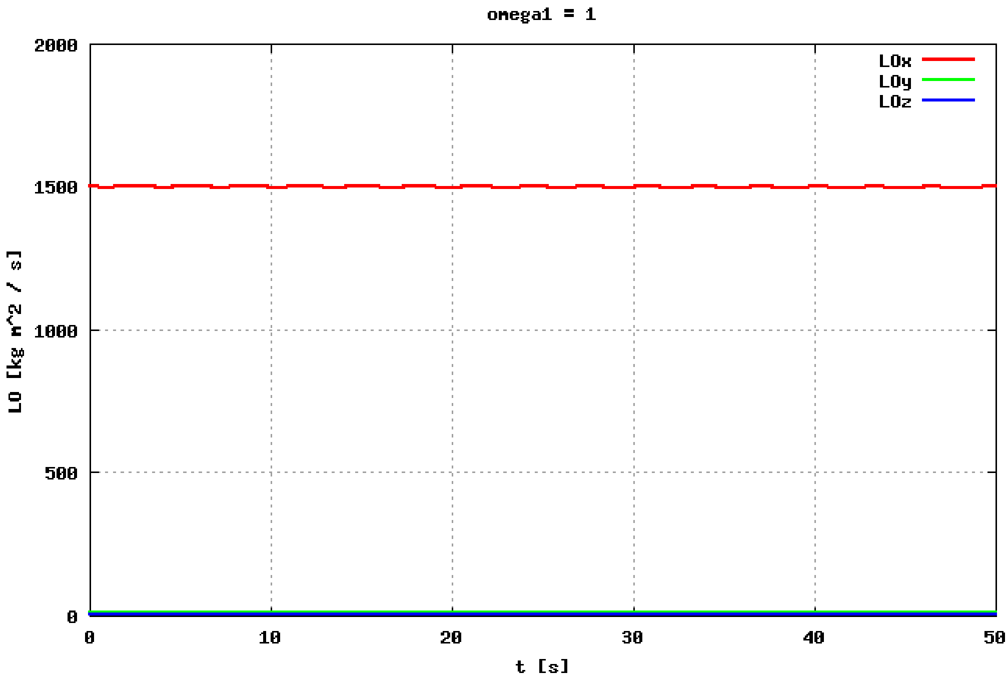


Fall 2:

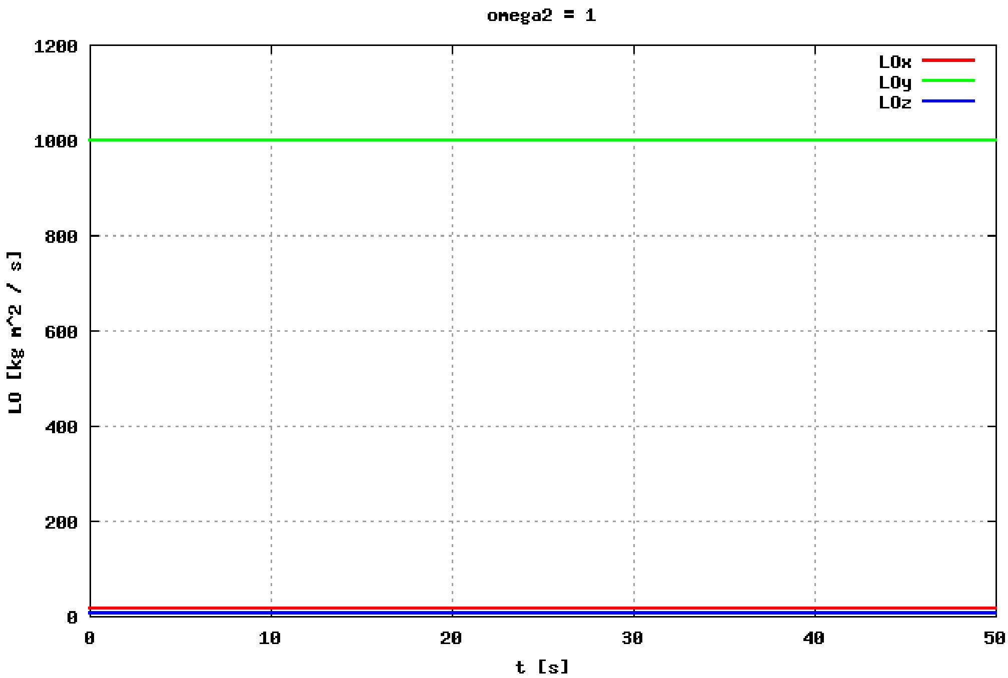


c) <u>Komponenten des Drallvektors:</u>

Fall 1:



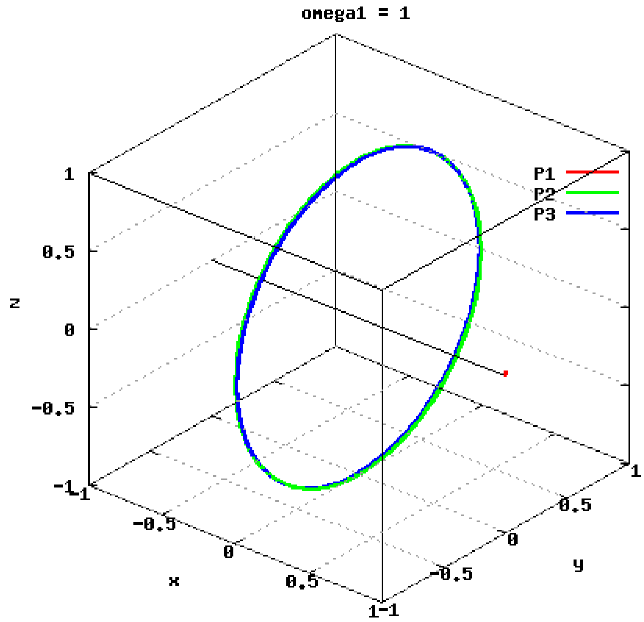
Fall 2:



In beiden Fällen sind die Komponenten des Drallvektors im ortsfesten Koordinatensystem zeitlich konstant, wie es der Drallsatz erfordert.

d) <u>Bahnen der Punkte:</u>

Fall 1:



Fall 2: