

```

In[120]:= Manipulate[
  (*by Nasser M. Abbasi, 6/28/14
  060615 add h vector display, fixed bug in spin, simplified,
  061015, fixed space cone for Ix<Iz*)
  tick;

Module[{mass = m * h * Pi r^2, lam, angle, Ix, Iy, Iz, g, debug = False, lengthOfw,
  hVector, theta, gamma, acc = 0.5, scale = 1, w1RPM, w2RPM, w3RPM},
  Iz = (1/2) mass r^2;
  Iy = (1/4) mass r^2 + 1/12 mass h^2;
  Ix = Iy;
  lam = w30 * (Iz - Ix) / Ix;

  w1 = w10 * Cos[lam * t] + acc w10 / lam * Sin[lam * t];
  w2 = w10 * Sin[lam * t] - acc w10 / lam * Cos[lam * t];
  w1RPM = w1 / (2 * Pi) * 60;
  w2RPM = w2 / (2 * Pi) * 60;
  w3RPM = w3 / (2 * Pi) * 60;

  currentThetax =
    Mod[(w10 * (acc - acc * Cos[lam * t] + lam * Sin[lam * t])) / lam^2, 2 Pi];
  currentThetay = Mod[(w10 * (lam - lam * Cos[lam * t] - acc * Sin[lam * t])) / lam^2,
    2 Pi];
  currentThetaz = Mod[w30 * t, 2 Pi];
  t = t + delT; (*time tick*)

  lengthOfw = Norm[{w1, w2, w3}];
  hVector = {Ix w1, Iy w2, Iz w3}; (*in body frame*)
  gamma = ArcCos[Dot[{0, 0, 1} / Norm[{0, 0, 1}], {w1, w2, w3} / lengthOfw]];
  theta = ArcCos[Dot[{0, 0, 1} / Norm[{0, 0, 1}], hVector / Norm@hVector]];

  g = Grid[{
    {Grid[{
      {"time (sec)", "Ix", "Iy", "Iz", "|w|", "γ°", "θ°", "|H→|"},
      {padIt2[currentTime, {5, 2}]},
      {padIt2[Ix, {4, 3}]},
      {padIt2[Iy, {4, 3}]},
      {padIt2[Iz, {4, 3}]},
      {padIt2[N@lengthOfw, {3, 2}]},
      {padIt2[gamma * 180. / Pi, {3, 1}]},
      {padIt2[theta * 180. / Pi, {3, 1}]},
      {padIt2[Norm@hVector, {5, 3}]}
```

```

    }
  }, Alignment → Center, Frame → All, Spacings → {.5, .7}]
}
,
{Grid[{
  {" $\theta_x^\circ$ ", " $\theta_y^\circ$ ", " $\theta_z^\circ$ ", " $\omega_x$  (rpm)", " $\omega_y$  (rpm)", " $\omega_z$  (rpm)", " $|\vec{\omega}|$ "},
  {
    padIt2[N@Mod[currentThetax * 180 / Pi, 360], {4, 1}],
    padIt2[N@Mod[currentThetay * 180 / Pi, 360], {4, 1}],
    padIt2[N@Mod[currentThetaz * 180 / Pi, 360], {4, 1}],
    padIt1[N[w1RPM], {4, 2}], (*RPM*)
    padIt1[N[w2RPM], {4, 2}],
    padIt1[N[w3RPM], {4, 2}],
    padIt2[Sqrt[w1RPM^2 + w2RPM^2 + w3RPM^2], {5, 3}]
    (*,
    Row[{padIt1[hVector[[1]], {4, 3}], "i",
      padIt1[hVector[[2]], {4, 3}], "j", padIt1[hVector[[3]], {4, 3}], "k"}] *)
  }
}, Alignment → Center, Frame → All, Spacings → {.5, .7}]
}
,
{Framed@
Graphics3D[
  Rotate[Rotate[Rotate[
    {
      If[showh,
        {Arrow[{{0, 0, 0}, 1.1 * h * hVector / Norm[hVector]}]}
      ]
    },
    ,
    If[showSpaceCone,
      If[Iz ≥ Ix,
        {EdgeForm[Red], FaceForm[Blue], Opacity[op],
          Cone[{scale * lengthOfw * Cos[gamma - theta] * hVector / Norm[hVector],
            {0, 0, 0}}, scale * lengthOfw * Sin[gamma - theta]
          ]},
        {EdgeForm[Red], FaceForm[Blue], Opacity[op],
          Cone[{scale * lengthOfw * Cos[theta - gamma] * hVector / Norm[hVector],
            {0, 0, 0}}, scale * lengthOfw * Sin[theta - gamma]
          ]}
      ]
    ]
  ],
  If[showCone, (*body cone*)
    {EdgeForm[Red], FaceForm[Gray], Opacity[op], Cone[{scale * lengthOfw *

```

```

        Cos[gamma] {0, 0, 1}, {0, 0, 0}}, scale * Sin[gamma] * lengthOfw
    ]]
],
If[showW,
  {Blue, Arrow[{{0, 0, 0}, scale * {w1, w2, w3} / lengthOfw]}]
],
',
If[showAxes,
  {
    {Red, Arrowheads[Medium], Arrow[{{0, 0, 0}, {.7, 0, 0}]},

    Inset[Graphics[Text[Style["x", Red, 14]]], {0.75, 0, 0}],
    {Red, Arrowheads[Medium], Arrow[{{0, 0, 0}, {0, .7, 0}]},

    Inset[Graphics[Text[Style["y", Red, 14]]], {0, 0.75, 0}],

    {Red, Arrowheads[Medium], Arrow[{{0, 0, 0}, {0, 0, .7}]},
    Inset[Graphics[Text[Style["z", Red, 14]]], {0, 0, 0.75}]
  }
],

If[showSection,
  {EdgeForm[Black], FaceForm[Glow@LightGray], Black,
    Polygon[{{0, 0, 0}, {w1, w2, 0}, {w1, w2, w3}, {0, 0, w3}, {0, 0, 0}]}]
],

{Opacity[op], Cylinder[{{0, 0, -h/2}, {0, 0, h/2}}, r]},
Sphere[{0, 0, 0}, .02]
(*Rotate[Rotate[Rotate[Polygon[{{0,0,0},{w1,w2,w3}], currentThetaz,
  {0,0,1}], currentThetax, {1,0,0}], currentThetay, {0,1,0}]*
  ), currentThetaz, {0, 0, 1}],
currentThetax, {1, 0, 0}], currentThetay, {0, 1, 0}
],

Axes → False, AxesLabel → {"x", "y", "z"},
PlotRange → {{-range, range}, {-range, range}, {-range, range}},
SphericalRegion → True, Boxed → False, ImagePadding → 2, ImageSize → 400
]
}
}

```

```

];

Which[state == "RUN",
  currentTime += delT;
  If[currentTime ≥ 1000, currentTime = 0];
  tick = Not[tick]
];

If[Abs[w1] > 10 || Abs[w2] > 10 || Abs[w3] > 10,
  state = "STOP"
];

g
],

Grid[{
  {
    Grid[{
      {"radius", Manipulator[Dynamic[r, {r = #, tick = Not[tick]} &],
        {.1, .5, .05}, ImageSize → Small], Dynamic[padIt2[r, {2, 2}]]},
      {"height", Manipulator[Dynamic[h, {h = #, tick = Not[tick]} &],
        {.1, 1, .1}, ImageSize → Small], Dynamic[padIt2[h, {2, 1}]]},
      {"density", Manipulator[Dynamic[m, {m = #, tick = Not[tick]} &],
        {.1, 50, .1}, ImageSize → Small], Dynamic[padIt2[m, {3, 1}]]}
    ], Frame → True, FrameStyle → Gray
  ]
},
(*
{
  Row[{"Initial angular positions"}]
},
{
  Grid[{
    {"θx(0)", Manipulator[Dynamic[θx, {θx = #;
      currentThetax = θx * Pi / 180;
      tick = Not[tick]} &], {-15, 15, 1}, ImageSize → Small], Dynamic[padIt1[θx, 2]]},
    {"θy(0)", Manipulator[Dynamic[θy, {θy = #;
      currentThetay = θy * Pi / 180;
      tick = Not[tick]} &], {-15, 15, 1}, ImageSize → Small], Dynamic[padIt1[θy, 2]]},
    {"θz(0)", Manipulator[Dynamic[θz, {θz = #;
      currentThetaz = θz * Pi / 180;
      tick = Not[tick]} &], {-15, 15, 1}, ImageSize → Small], Dynamic[padIt1[θz, 2]]}
  ], Frame → True]
}

```

```

},
*)
{
  Grid[{
    {Style["Initial Z axes spin rate (RPM)", 10], SpanFromLeft},
    {Manipulator[Dynamic[initialSpinRate, (initialSpinRate = #;
      w30 = w3 = initialSpinRate * 2 * Pi / 60;
      w20 = w2 = w1 = w10 = 0;
      tick = Not[tick]) &], {.1, 10, .1}, ImageSize -> Small
    ]},
    Dynamic[padIt1[N@initialSpinRate, {4, 2}]]
  ]}
}, Frame -> True, FrameStyle -> Gray
]
},
{
  Grid[{
    {Button[Text@Style["run", 12], {state = "RUN";
      tick = Not[tick]}, ImageSize -> {40, 40}],
    Button[Text@Style["step", 12], {state = "STEP";
      tick = Not[tick]}, ImageSize -> {40, 40}],
    Button[Text@Style["stop", 12], {state = "STOP";
      tick = Not[tick]}, ImageSize -> {40, 40}],
    Button[Text@Style["reset", 12], {state = "RESET";
      currentThetax = 0;
      currentThetay = 0;
      currentThetaz = 0;
      w10 = w1 = 0;
      w20 = w2 = 0;
      w30 = w3 = 10 * 2 * Pi / 60;
      t = 0;
      op = 1;
      h = .8;
      r = .3;
      m = 1;
      spin = "Z-axes";
      initialSpinRate = 10;
      (*delT=0.1; *) (*due to bug in Manipulate, long story*)
      currentTime = 0;
      tick = Not[tick]}, ImageSize -> {40, 40}] (*fix*)
  ]}
}, Spacings -> {.2, 0}, Frame -> True, FrameStyle -> Gray
]

```

```

},
{
  Grid[{
    {
      Button[Text@Style["perturbe", 12], {
        w10 += (0.01 * w30);
        tick = Not[tick]}, ImageSize -> {120, 40}]
    }
  ]
}
,
{Grid[{
  {Row[{"opacity", Spacer[3], Manipulator[Dynamic[op, {op = #;
    tick = Not[tick]} &], {.1, 1, .1}, ImageSize -> Small],
    Spacer[3], Dynamic[padIt1[op, {1, 1}]]]}]},
  {Row[{"slow", Spacer[3], Manipulator[Dynamic[delT,
    {delT = #; (*Print["changed delT to ", delT]; *) tick = Not[tick]} &],
    {0.01, 0.2, .01}, ImageSize -> Small], Spacer[3], "fast"]]},
  {Row[{"zoom in", Spacer[3], Manipulator[Dynamic[range, {range = #;
    tick = Not[tick]} &],
    {0.1, 2, .01}, ImageSize -> Small], Spacer[3], "zoom out"]]},
  {Row[{"show body axes", Spacer[3], Checkbox[Dynamic[showAxes, {showAxes = #;
    tick = Not[tick]} &]]]}]},
  {Row[{"show angular velocity direction",
    Spacer[2], Checkbox[Dynamic[showW, {showW = #;
    tick = Not[tick]} &]]]}, SpanFromLeft},
  {Row[{"show body cone", Spacer[3], Checkbox[Dynamic[showCone, {showCone = #;
    tick = Not[tick]} &]]]}]},
  {Row[{"show space cone", Spacer[3],
    Checkbox[Dynamic[showSpaceCone, {showSpaceCone = #;
    tick = Not[tick]} &]]]}]},
  {Row[{"show sweep section",
    Spacer[3], Checkbox[Dynamic[showSection, {showSection = #;
    tick = Not[tick]} &]]]}]},
  {Row[{"show h vector (angular momentum",
    Spacer[3], Checkbox[Dynamic[showh, {showh = #;
    tick = Not[tick]} &]]]}]},
  {Button[Text@Style["clear all", 12], {op = 1;
    showW = False;
    showAxes = False;
    showCone = False;
    showSpaceCone = False;
    showSection = False;

```

```

        showh = False;
        tick = Not[tick]], ImageSize → {80, 40}], SpanFromLeft}
    ], Frame → True, Alignment → Left, FrameStyle → Gray
  ]
}
], Frame → False, Alignment → Center, FrameStyle → Gray
],

```

```

{{t, 0}, None}, (*current time*)
{{w1, 0}, None},
{{w10, 0}, None},
{{w2, 0}, None},
{{w20, 0}, None},
{{w3, 10 * 2 * Pi / 60}, None},
{{w30, 10 * 2 * Pi / 60}, None},
{{range, 1.2}, None},

```

```

{{tick, False}, None},
{{state, "RESET"}, None},
{{currentTime, 0}, None},
{{delT, 0.1}, None},
{{op, .6}, None},
{{spin, "X-axes"}, None},
{{initialSpinRate, 10}, None},
{{showAxes, True}, None},
{{showCone, True}, None},
{{showSection, False}, None},
{{showSpaceCone, True}, None},
(*{{θx, 0}, None},
{{θy, 0}, None},
{{θz, 0}, None},
*)

```

```

{{r, .3}, None},
{{h, .8}, None},
{{m, 1}, None},
{{showW, True}, None},
{{showh, False}, None},

```

```

{{currentThetax, 0}, None},
{{currentThetay, 0}, None},
{{currentThetaz, 0}, None},

```

```

TrackedSymbols => {tick},
SynchronousUpdating -> False,
ControlPlacement -> Left, Alignment -> Center, ImageMargins -> 0, FrameMargins -> 0,
Initialization ->
(
  integerStrictPositive = (IntegerQ[#] && # > 0 &);
  integerPositive = (IntegerQ[#] && # ≥ 0 &);
  numericStrictPositive = (Element[#, Reals] && # > 0 &);
  numericPositive = (Element[#, Reals] && # ≥ 0 &);
  numericStrictNegative = (Element[#, Reals] && # < 0 &);
  numericNegative = (Element[#, Reals] && # ≤ 0 &);
  bool = (Element[#, Booleans] &);
  numeric = (Element[#, Reals] &);
  integer = (Element[#, Integers] &);
  (*-----*)
  padIt1[v_?numeric, f_List] := AccountingForm[v,
    f, NumberSigns -> {"-", "+"}, NumberPadding -> {"0", "0"}, SignPadding -> True];
  (*-----*)
  padIt1[v_?numeric, f_Integer] := AccountingForm[Chop[v],
    f, NumberSigns -> {"-", "+"}, NumberPadding -> {"0", "0"}, SignPadding -> True];
  (*-----*)
  padIt2[v_?numeric, f_List] := AccountingForm[v,
    f, NumberSigns -> {"", ""}, NumberPadding -> {"0", "0"}, SignPadding -> True];
  (*-----*)
  padIt2[v_?numeric, f_Integer] := AccountingForm[Chop[v],
    f, NumberSigns -> {"", ""}, NumberPadding -> {"0", "0"}, SignPadding -> True];
  (*-----*)
)
]

```