In[62]:=
```
ClearAll["Global`*"]
```

In[63]:=

(*Data*)

In[64]:=
```
(*Total mass of the structure+transfer line of SIGRUM*)
Mass1 = 32 000;
(*Position of the centre of mass of SIGRUM*)
Rg = 2.44;
(*Moment of inertia of SIGRUM (structure + transfer line) with respect to the rotation axis*)
Inertia1 = 340 000;
(*Moment of inertia of SCSG and FTG
  (structure + transfer line) with respect to the rotation axis*)
Inertia2 = 700 000;
(*Moment of inertia of BSIGRUM and SSG
  (structure + transfer line) with respect to the rotation axis*)
Inertia3 = 500 000;
```

In[69]:=

(*Equation of motion of the gantry in nominal condition*)

In[70]:=

(*Parameters for a trapezoidal equation of motion: acceleration, constant speed, deceleration*)

In[71]:=
```
Δsmax = π;
(*time to cover half a rotation Δsmax=π*)
Tmax = 30;
(*assumed maximum time of acceleration or deceleration phase*)
Tstop = 6;
(*proportional time calculation,
with the hypothesis of linearity between rotation time and angle step*)
T[δs_] := Tmax * δs / Δsmax
```

In[75]:=
```
(*division point between rotations for which the proportional
  time is smaller than the maximum time of acceleration/deceleration*)
sol = Solve[Tstop / T[x] == 1 / 2, x]
divpoint = x /. sol[[1]]
```

Out[75]= $\left\{\left\{x \to \frac{2\pi}{5}\right\}\right\}$

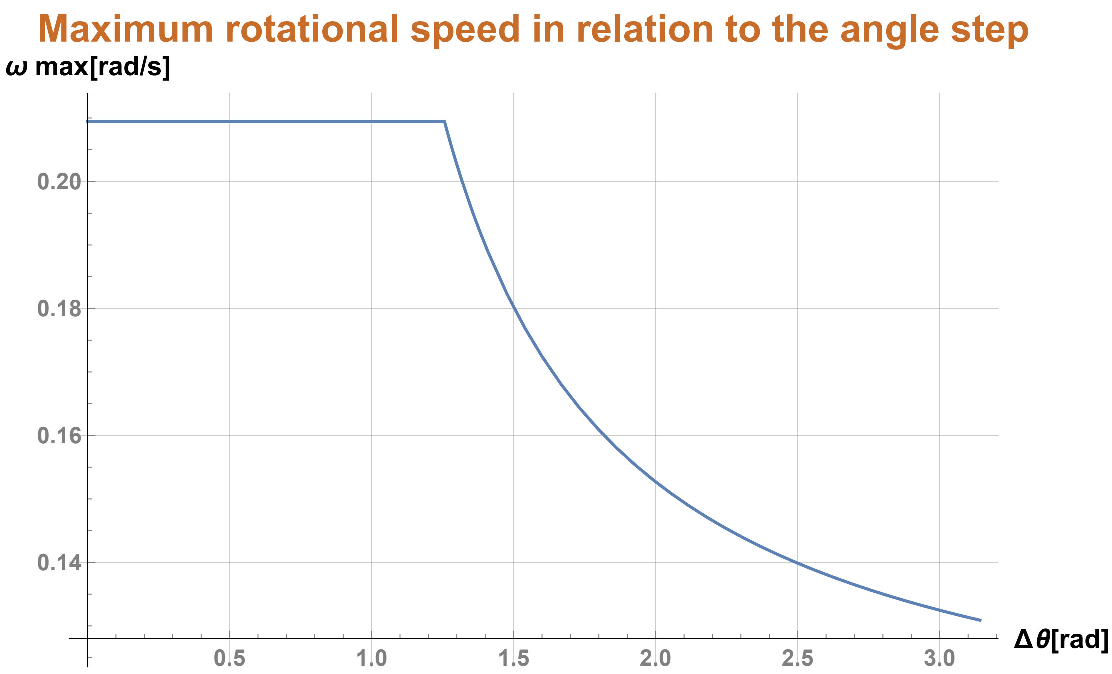
Out[76]= $\frac{2\pi}{5}$

In[77]:=
```
(*ratio between acc/dec time and proportional time*)
λ[δs_] := Piecewise[{{Tstop / T[δs], δs ≥ divpoint}, {1 / 2, δs < divpoint}}]
(*ratio between constant speed phase time and proportional time*)
λ2[δs_] := 1 - 2 * λ[δs]
```

In[79]:=
```
(*maximum nominal rotational speed*)
ωmax[δs_] := δs / T[δs] * 2 / (2 - λ[δs] - λ[δs]);
(*maximum rotational acceleration*)
A[δs_] := δs / T[δs] ^2 * 2 / (λ[δs] * (2 - λ[δs] - λ[δs]));
```

In[81]:=
```
Plot[{ωmax[x]}, {x, 0, π}, AxesLabel → {"Δθ[rad]", "ω max[rad/s]"},
 LabelStyle → Directive[Black, Bold, FontSize → 14],
 TicksStyle → Directive[Gray, Medium], GridLines → Automatic,
 ImageSize → Large, PlotRange → All, Exclusions → None, PlotLabel →
  Style[StringForm["Maximum rotational speed in relation to the angle step"],
   20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
```

Out[81]=



In[82]:=
```
(*Trapezoidal equation of motion of the gantry*)
```

In[83]:=
```
θ2[start_, δs_, t_] := Piecewise[{{start - 1 / 2 * A[δs] * t^2, t <= λ[δs] * T[δs]},
   {start - (1 / 2 * A[δs] * (λ[δs] * T[δs]) ^2 + ωmax[δs] * (t - λ[δs] * T[δs])),
    λ[δs] * T[δs] ≤ t < (λ[δs] + λ2[δs]) * T[δs]},
   {start - (1/2 T[δs] (A[δs] T[δs] λ[δs]^2 + 2 λ2[δs] ωmax[δs]) +
       ωmax[δs] * (t - T[δs] * (λ[δs] + λ2[δs])) -
       1 / 2 * A[δs] * (t - (λ2[δs] + λ[δs]) * T[δs]) ^2), t < T[δs]}}, Indeterminate]
```

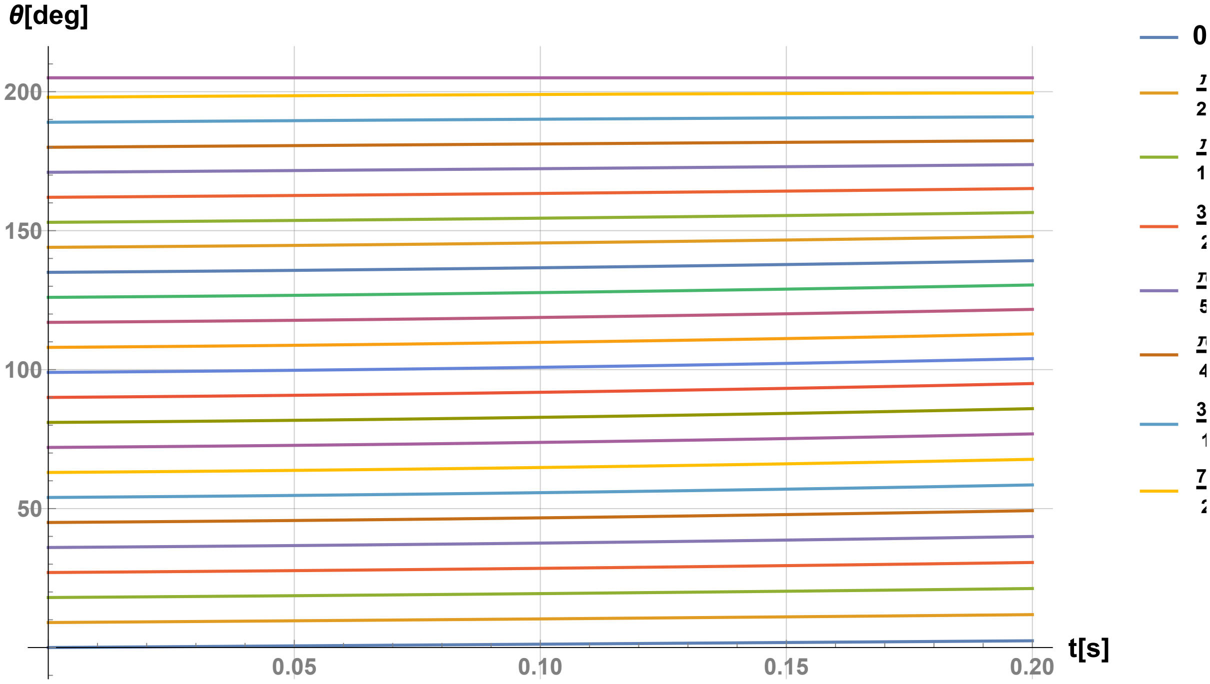
In[84]:=
```
(*Solution of the physical
  pendulum equation for the SIGRUM
  (unbalanced structure in case of failure)*)
```

In[85]:=
```
(*assumed reaction time before starting to emergency brake the system*)
ReactionTime = 0.2;
(*parameter to divide half a rotation in equal angle steps
  and analyse the worst case scenario (kinetic energy point of view),
which is the one in which the gantry is at maximum speed,
a failure occurs and the machine accelerate more due to gravity. This depends
  clearly on the starting position, thus is necessary to evaluate the worst case

    Note that "prec" must be an even number
    to pick up the worst case that is for a starting position = π/2
*)
prec = 20;
(*worst case starting speed*)
startingv = ωmax[divpoint];
(*solution of the physical pendulum equation,
changing starting position "start" and setting initial speed =  maximum nominal speed*)
s = Table[NDSolve[{x''[t] == Mass1 * 9.81 * Rg * Sin[x[t]] / Inertia1,
      x[0] == start, x'[0] == startingv}, x,
     {t, 0, ReactionTime}], {start, 0, 20 / 18 * π, π / prec}];
Plot[{Evaluate[(x[t] /. s) * 180 / π], 205}, {t, 0, ReactionTime},
 PlotRange → All, PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
 AxesLabel → {"t[s]", "θ[deg]"},
 LabelStyle → Directive[Black, Bold, FontSize → 14],
 TicksStyle → Directive[Gray, Medium], GridLines → Automatic,
 ImageSize → Large, PlotLabel → Style[StringForm[
    "Angular position during a reaction time of `` s\n in relation to the
      starting position",
    ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
Plot[Evaluate[x'[t] /. s], {t, 0, ReactionTime}, PlotRange → All,
 PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
 AxesLabel → {"t[s]", "ω[rad/s]"},
 LabelStyle → Directive[Black, Bold, FontSize → 14],
 TicksStyle → Directive[Gray, Medium], GridLines → Automatic,
 ImageSize → Large, PlotLabel → Style[StringForm[
    "Angular speed during a reaction time of `` s\n in relation to the
      starting position",
    ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
Plot[Evaluate[x''[t] /. s], {t, 0, ReactionTime}, PlotRange → All,
 PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}], GridLines → Automatic,
 ImageSize → Large, AxesLabel → {"t[s]", "θ''[rad/s^2]"},
 LabelStyle → Directive[Black, Bold, FontSize → 14],
 TicksStyle → Directive[Gray, Medium], PlotLabel → Style[StringForm[
    "Angular acceleration during a reaction time of `` s\n in relation to the
      starting position", ReactionTime],
   20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
```

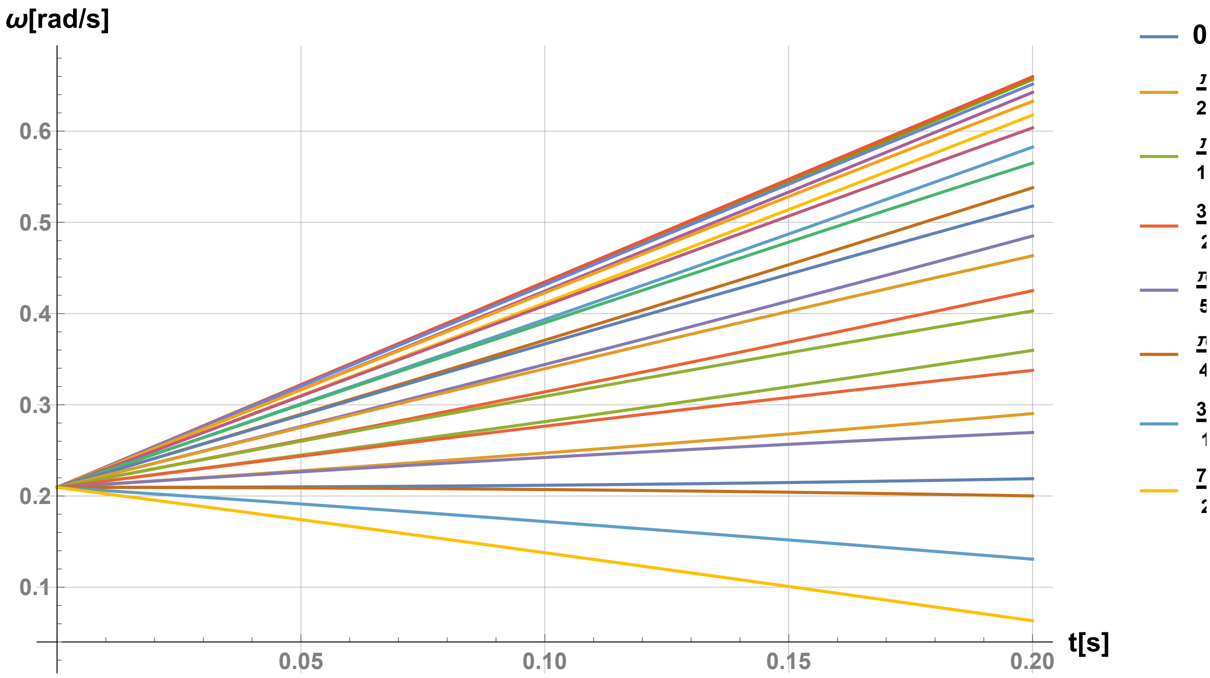
Angular position during a reaction time of 0.2 s in relation to the starting position

Out[89]=

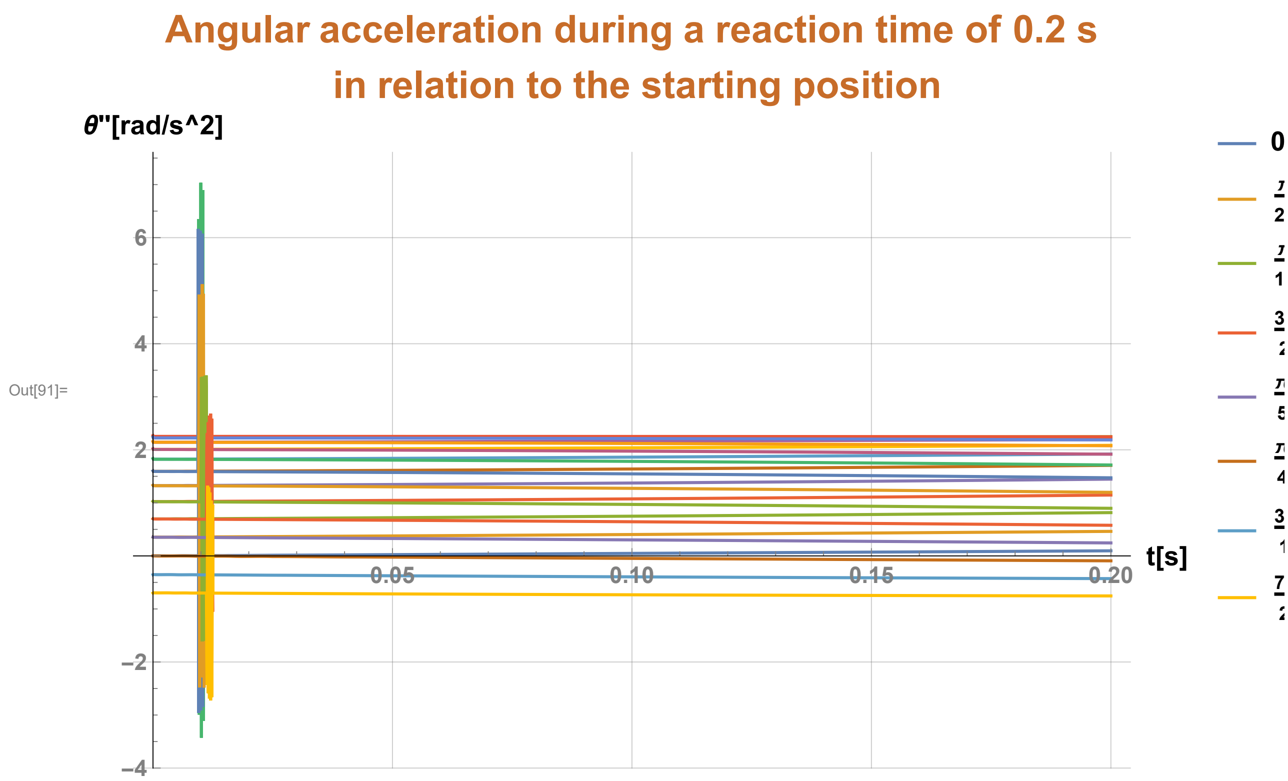


Angular speed during a reaction time of 0.2 s in relation to the starting position

Out[90]=



Out[91]=



In[92]:=

(*Solution of the equation of motion of balanced structures in case of failure*)

In[93]:=

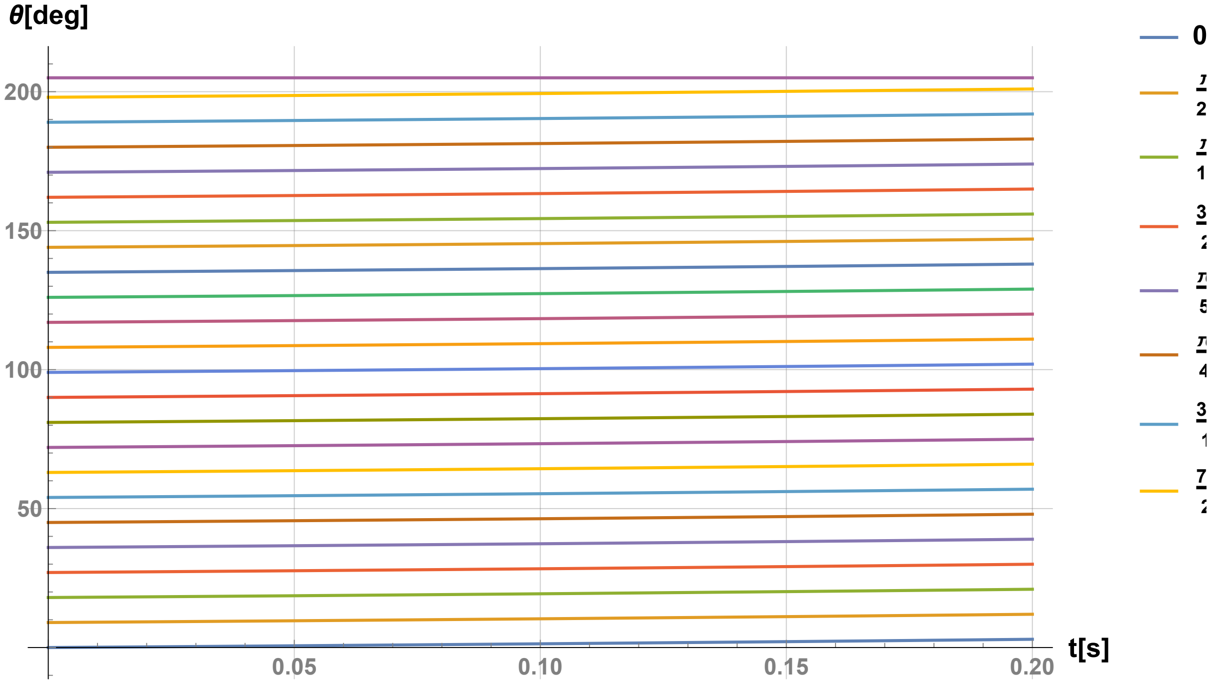
(*solution of the equation of motion with the hypothesis that a failure adds torque. Specifically, it has been assumed that the added torque is twice the nominal torque*)

```
s2[torque_, inertia_] := Table[
  NDSolve[{y''[t] == 2 * torque * 1000 / inertia, y[0] == start, y'[0] == startingv},
   y, {t, 0, ReactionTime}], {start, 0, 20 / 18 * π, π / prec}]
```

In[94]:=
```
Plot[{Evaluate[(y[t] /. s2[125.6, Inertia3]) * 180 / π], 205}, {t, 0, ReactionTime},
  PlotRange → All, PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
  AxesLabel → {"t[s]", "θ[deg]"},
  LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium],
  GridLines → Automatic, ImageSize → Large, PlotLabel →
   Style[StringForm["B-SIGRUM angular position during a reaction time
        of `` s\n in relation to the starting position",
     ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
Plot[{Evaluate[(y[t] /. s2[5.1 * (8 / 0.25), Inertia3]) * 180 / π], 205},
  {t, 0, ReactionTime}, PlotRange → All,
  PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
  AxesLabel → {"t[s]", "θ[deg]"},
  LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium], GridLines → Automatic, ImageSize → Large,
  PlotLabel → Style[StringForm["SSG angular position during a reaction
        time of `` s\n in relation to the starting position",
     ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
Plot[{Evaluate[(y[t] /. s2[14.3 * (3 / 0.25), Inertia2]) * 180 / π], 205},
  {t, 0, ReactionTime}, PlotRange → All,
  PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
  AxesLabel → {"t[s]", "θ[deg]"},
  LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium], GridLines → Automatic, ImageSize → Large,
  PlotLabel → Style[StringForm["FTG angular position during a reaction
        time of `` s\n in relation to the starting position",
     ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
Plot[{Evaluate[(y[t] /. s2[11 * (4 / 0.25), Inertia2]) * 180 / π], 205},
  {t, 0, ReactionTime}, PlotRange → All,
  PlotLegends → Table[i, {i, 0, 20 / 18 * π, π / prec}],
  AxesLabel → {"t[s]", "θ[deg]"},
  LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium], GridLines → Automatic, ImageSize → Large,
  PlotLabel → Style[StringForm["SCSG angular position during a reaction
        time of `` s\n in relation to the starting position",
     ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
```

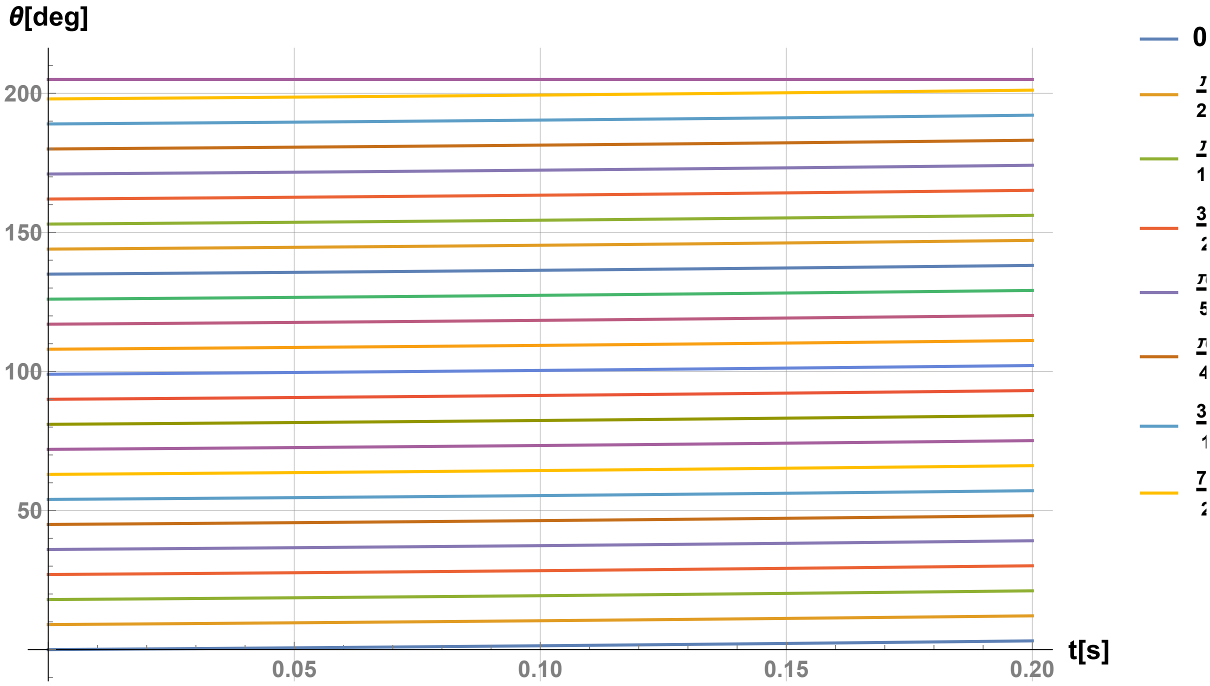
B–SIGRUM angular position during a reaction time of 0.2 s in relation to the starting position

Out[94]=



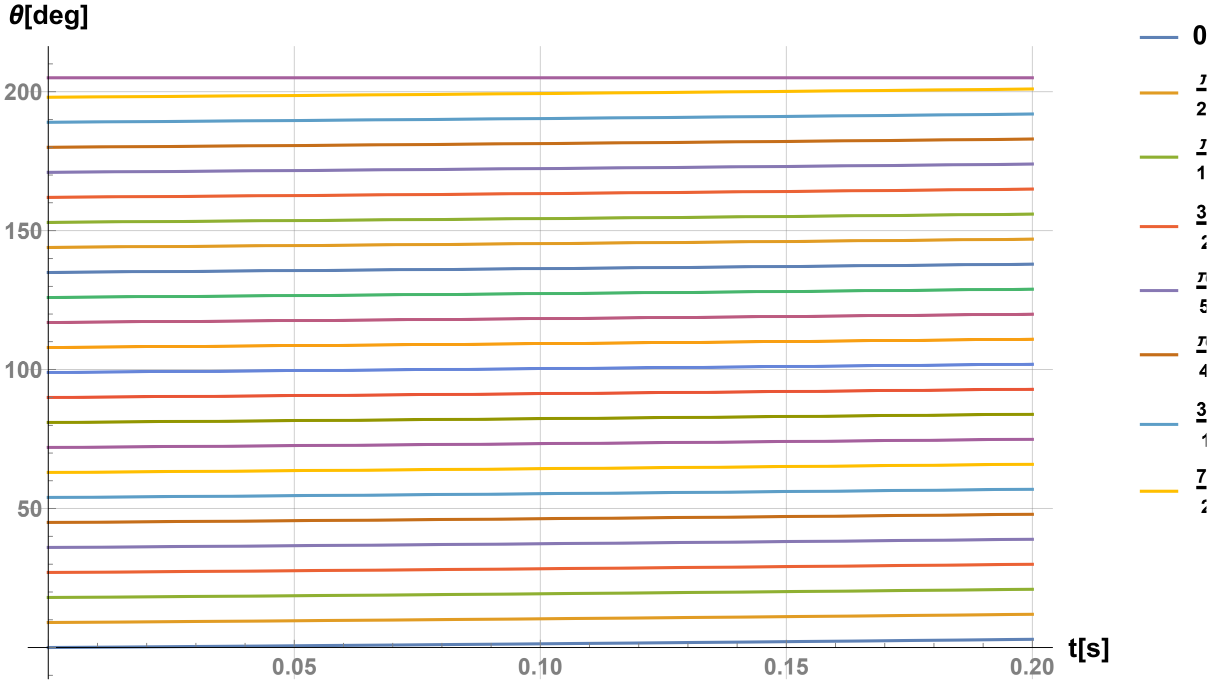
SSG angular position during a reaction time of 0.2 s in relation to the starting position

Out[95]=



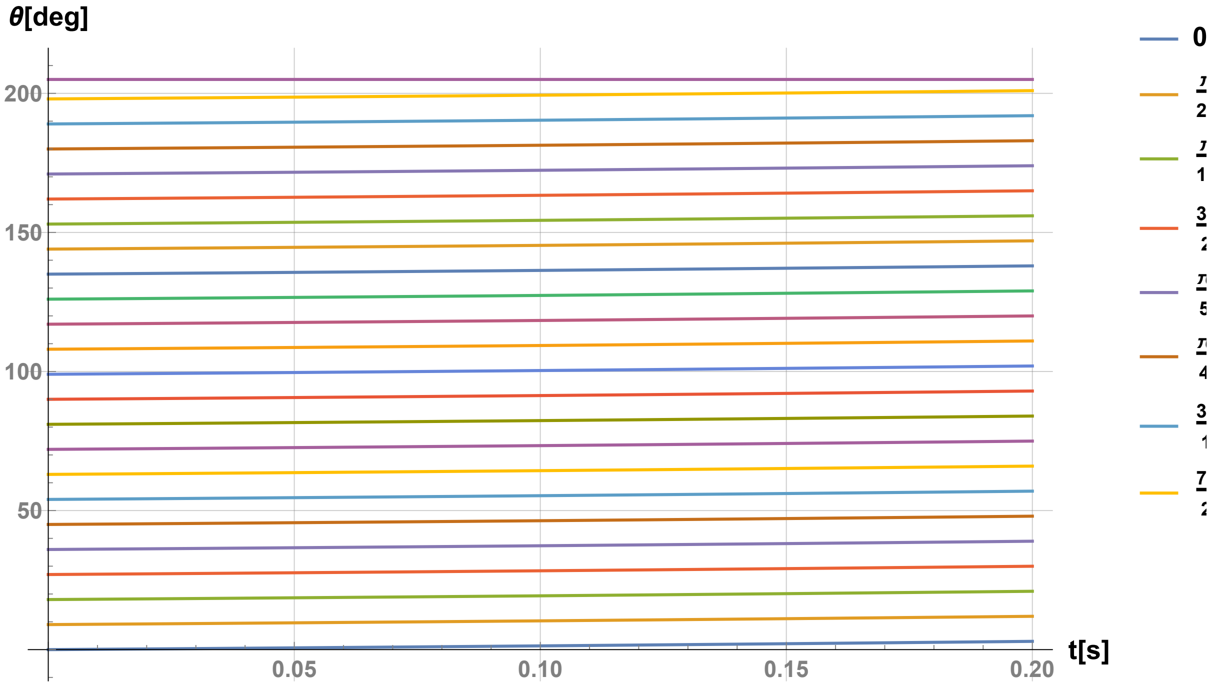
Out[96]=

FTG angular position during a reaction time of 0.2 s
in relation to the starting position



Out[97]=

SCSG angular position during a reaction time of 0.2 s
in relation to the starting position

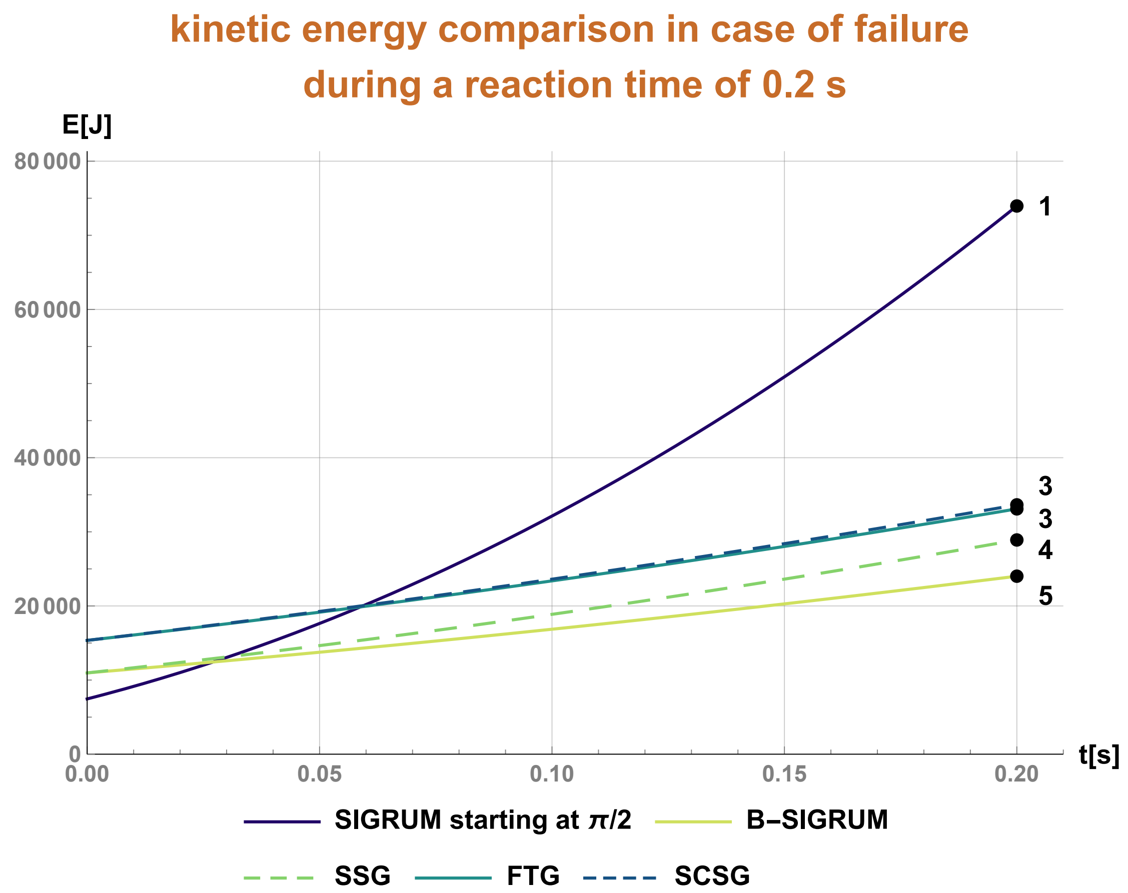


In[98]:= (*Kinetic energy comparison between SIGRUM and balanced structures*)

```
In[99]:= (*points on the graph*)
pt = {
    {t, Evaluate[(x'[t] /. s[[prec / 2 + 1]]) ^2 * Inertia1 * 1 / 2][[1]]},
    {t, 1 / 2 * Inertia3 * (2 * 125.6 * 1000 * t / Inertia3 + startingv) ^2},
    {t,
     1 / 2 * Inertia3 * (2 * 5.1 * (8 / 0.25) * 1000 * t / Inertia3 + startingv) ^2},
    {t, 1 / 2 * Inertia2 * (2 * 14.3 * (3 / 0.25) * 1000 * t / Inertia2 + startingv) ^2},
    {t,
     1 / 2 * Inertia2 * (2 * 11 * (4 / 0.25) * 1000 * t / Inertia2 + startingv) ^2}} /.
   t → ReactionTime;

In[100]:= (*plot*)
g1 = Plot[{
    Evaluate[(x'[t] /. s[[prec / 2 + 1]]) ^2 * Inertia1 * 1 / 2],
    1 / 2 * Inertia3 * (2 * 125.6 * 1000 * t / Inertia3 + startingv) ^2,
    1 / 2 * Inertia3 * (2 * 5.1 * (8 / 0.25) * 1000 * t / Inertia3 + startingv) ^2,
    1 / 2 * Inertia2 * (2 * 14.3 * (3 / 0.25) * 1000 * t / Inertia2 + startingv) ^2,
    1 / 2 * Inertia2 * (2 * 11 * (4 / 0.25) * 1000 * t / Inertia2 + startingv) ^2
   },
   {t, 0, ReactionTime},
   Epilog → {Black, PointSize[Large], Point[pt], {
      Text[Style["1", Bold, FontSize → 14], Offset[{15, 0}, pt[[1]]]],
      Text[Style["5", Bold, FontSize → 14], Offset[{15, -10}, pt[[2]]]],
      Text[Style["4", Bold, FontSize → 14], Offset[{15, -5}, pt[[3]]]],
      Text[Style["3", Bold, FontSize → 14], Offset[{15, -5}, pt[[4]]]],
      Text[Style["3", Bold, FontSize → 14], Offset[{15, 10}, pt[[5]]]]
     }},
   PlotStyle → {
     Directive[ColorData["BlueGreenYellow"][0]],
     Directive[ColorData["BlueGreenYellow"][0.95]],
     Directive[ColorData["BlueGreenYellow"][0.80], Dashing[0.020]],
     Directive[ColorData["BlueGreenYellow"][0.40]],
     Directive[ColorData["BlueGreenYellow"][0.20], Dashing[0.015]]
    },
   PlotRange → {{0, 1.05 * ReactionTime}, {0, 1.1 * pt[[1]][[2]]}},
   PlotLegends →
    Placed[{"SIGRUM starting at π/2", "B-SIGRUM", "SSG", "FTG", "SCSG"}, Below],
   AxesLabel → {"t[s]", "E[J]"},
   LabelStyle → Directive[Black, Bold, FontSize → 14],
   TicksStyle → Directive[Gray, Medium],
   GridLines → Automatic,
   ImageSize → Large,
   PlotLabel → Style[StringForm["kinetic energy comparison in
        case of failure \n during a reaction time of `` s ",
      ReactionTime], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
```

Out[100]=



In[101]:=
```
(*uncomment and set "PATH" to export graph*)
(*SetDirectory["PATH"]

  cm=72/2.54 (*centimetre*)

  Export["figure.jpg",Show[g1,ImageSize→20 cm],ImageResolution→300]*)
```

In[102]:=

(*Calculation of the work of the motor during a given gantry rotation i.e 180°*)

In[103]:=

(*work done by the motor of SIGRUM (unbalanced structure):
is the sum of the potential gravitational energy of the mass
concentrated ad the centre of mass + the rotational kinetic energy
work calculated as a function of the starting position, angle step, and time*)

```
worksimplified[start_, δs_, tend_] :=
 Mass1 * 9.81 * Rg * (Cos[θ2[start, δs, tend]] - Cos[start]) +
   1 / 2 * Inertia1 * D[θ2[start, δs, t], {t, 1}]^2 /. t → tend
```

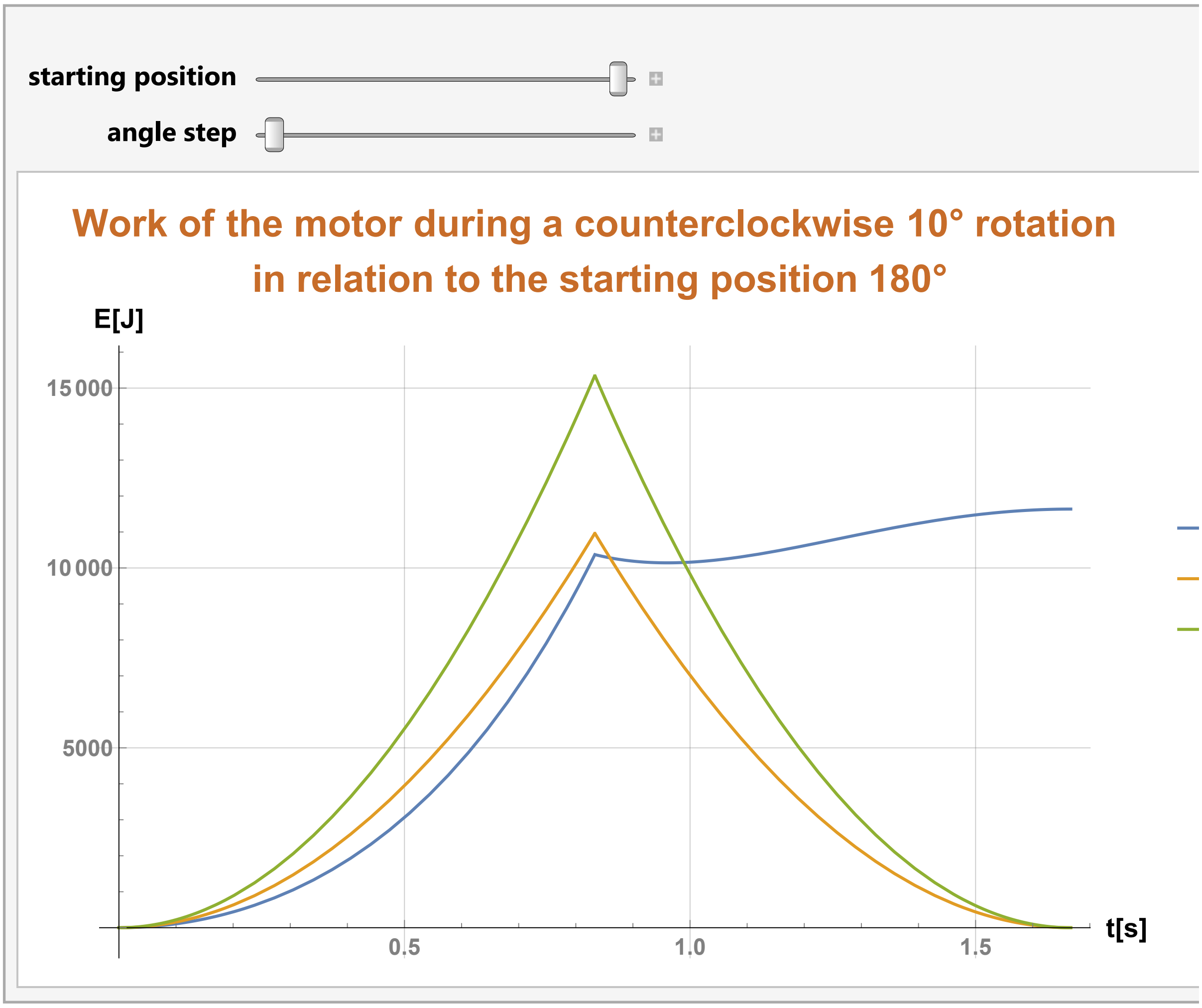
(*work done by the motor in the case of balanced
structures: the only contribution is the rotational kinetic energy
work calculated as a function of the starting position, angle step, and time*)

```
work1simplified[start_, δs_, tend_, inertia_] :=
 1 / 2 * inertia * (D[θ2[start, δs, t], {t, 1}])^2 /. t → tend
```

```
In[105]:= (*work data of SIGRUM for plotting: "res" is the resolution*)
datahalfturnwork[res_] :=
 Table[{i, worksimplified[π, π, 0.9999 * Tmax]}, {i, 0, 180, 180 / res}]
(*work done by the motor in the case of balanced structures:
   since the work doesn't depend on the starting position. the total work for half turn is the
     work done in a angle step multiplied by the number of equal angle steps in half a turn*)
halfturnwork1[δs_, inertia_] :=
 work1simplified[0, δs, T[δs] / 2, inertia] * (π / δs) * 2
(*work data of balanced structure for plotting*)
datahalfturnwork1[res_, inertia_] :=
 Table[{i, halfturnwork1[i * π / 180, inertia]}, {i, 0, 180, 180 / res}]
```

In[108]:=
```
(*manipulate work in function of starting position and angle step*)
Manipulate[Plot[{worksimplified[a * π / 180, q * π / 180, x],
    work1simplified[a * π / 180, q * π / 180, x, Inertia3],
    work1simplified[a * π / 180, q * π / 180, x, Inertia2]}, {x, 0, T[q * π / 180]},
  PlotRange → Full, AxesLabel → {"t[s]", "E[J]"}, GridLines → Automatic,
  PlotLegends → {"SIGRUM", "B-SIGRUM and SSG", "SCSG and FTG"},
  ImageSize → Large, LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium],
  PlotLabel → Style[StringForm["Work of the motor during a counterclockwise
       ``° rotation\n in relation to the starting position ``°",
     q, a], 20, RGBColor[199 / 255, 108 / 255, 41 / 255]]],
 {{a, 180, "starting position"}, 0, 180},
 {{q, 10, "angle step"},
  10, a},
 LabelStyle → Directive[Black, Bold, FontSize → 14]]
```

Out[108]=



In[109]:=
```
(*data manipulation to plot points*)
pt2[δs_] := {
  {δs, Log[halfturnwork1[δs * π / 180, Inertia3]] },
  {δs, Log[halfturnwork1[δs * π / 180, Inertia2]] }
 }

txt[a_, b_, ofsettx_, ofsetty_] := Text[Style[StringForm["``%",
    Round[E^pt2[a][[b]][[2]] / worksimplified[π, π, 0.9999 * Tmax] * 100]],
   Bold, FontSize → 14], Offset[{ofsettx, ofsetty}, pt2[a][[b]]]]
```

In[111]:=
```
(*grid definition*)
xgrid = Table[i, {i, 10, 60, 5}];
ygrid = Flatten[Table[i * 10^j, {i, 1, 10}, {j, 1, 6}]];

g2 = ListLogPlot[{datahalfturnwork[100],
   datahalfturnwork1[100, Inertia3], datahalfturnwork1[100, Inertia3],
   datahalfturnwork1[100, Inertia2], datahalfturnwork1[100, Inertia2]},
  PlotRange → {{8, 50}, {50 000, 2 000 000}}, AxesOrigin → {{10, 50 000}},
  Joined → True, GridLines → {xgrid, ygrid}, ImageSize → Large,
  PlotStyle → {
    Directive[ColorData["BlueGreenYellow"][0]],
    Directive[ColorData["BlueGreenYellow"][0.95]],
    Directive[ColorData["BlueGreenYellow"][0.80], Dashing[0.020]],
    Directive[ColorData["BlueGreenYellow"][0.40]],
    Directive[ColorData["BlueGreenYellow"][0.20], Dashing[0.015]]
   },
  Epilog →
   {PointSize[Large], Black, Point[pt2[10]], txt[10, 1, 25, 0], txt[10, 2, 25, 0],
    Black, Point[pt2[45]], txt[45, 1, 15, 10], txt[45, 2, 15, 10]},
  PlotLegends → Placed[{"SIGRUM", "B-SIGRUM", "SSG", "FTCG", "Mixed"}, Below],
  AxesLabel → {"angle step δ[deg]", "E[J]"},
  LabelStyle → Directive[Black, Bold, FontSize → 14],
  TicksStyle → Directive[Gray, Medium],
  PlotLabel → Style[StringForm["Work done by the motor during half
       turn\n in relation to the angle step subdivision"],
    20, RGBColor[199 / 255, 108 / 255, 41 / 255]]]
```

Power::infy : Infinite expression $\frac{1}{0}$ encountered. ≫

Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. ≫

Power::infy : Infinite expression $\frac{1}{0}$ encountered. ≫

Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. ≫

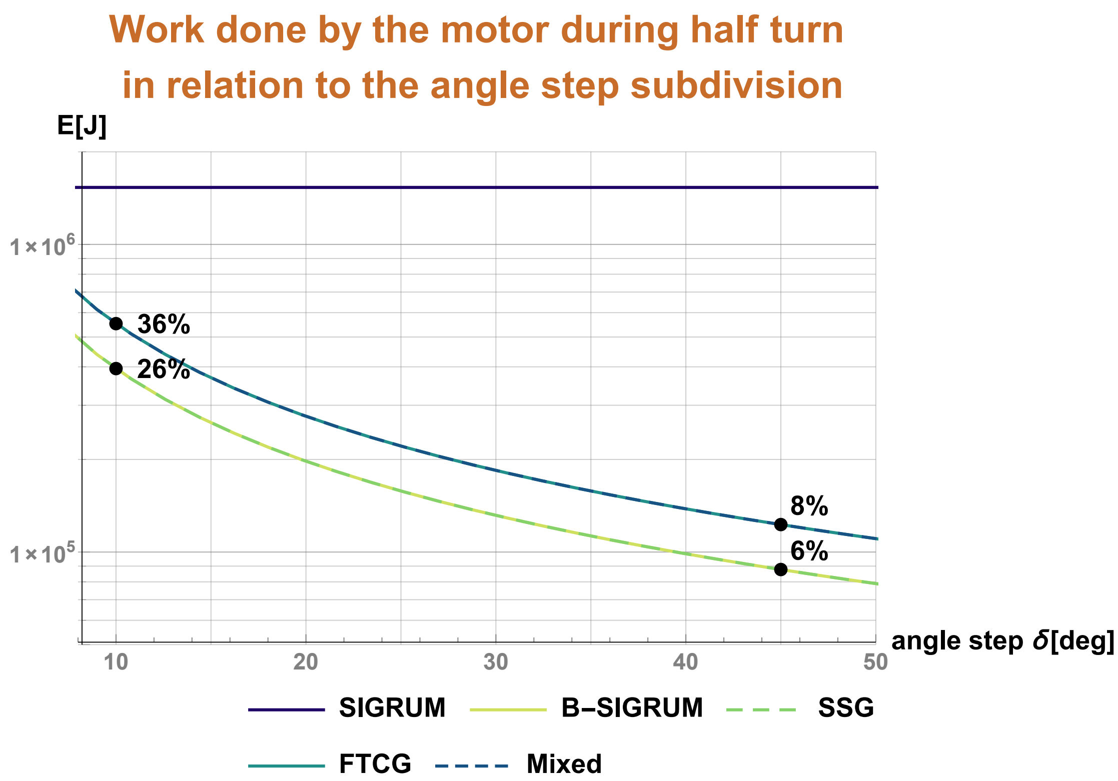
Power::infy : Infinite expression $\frac{1}{0}$ encountered. ≫

General::stop : Further output of Power::infy will be suppressed during this calculation. ≫

Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. ≫

General::stop : Further output of Infinity::indet will be suppressed during this calculation. ≫

Out[113]=



In[114]:=

```
(*Export["figure2.jpg",Show[g2,ImageSize→20 cm],ImageResolution→300]*)
```