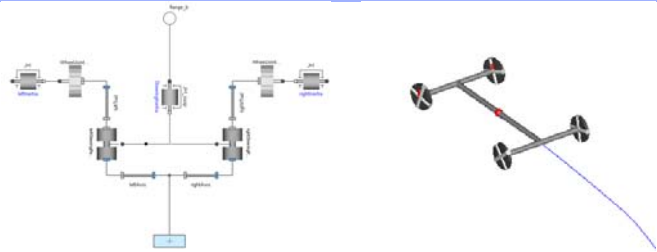


Virtual Physics Equation-Based Modeling

TUM, December 09, 2014

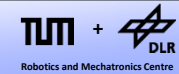
Two-Track Car Model



Dr. Dirk Zimmer

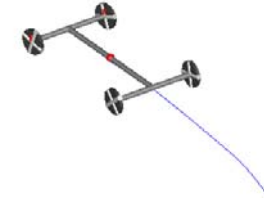
German Aerospace Center (DLR), Robotics and Mechatronics Centre

Two-Track Model



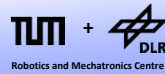
In this lecture, let us look at the modeling of a two-track car model:

- This is still a planar mechanical model
- It will be later enhanced by a few 3D-elements.



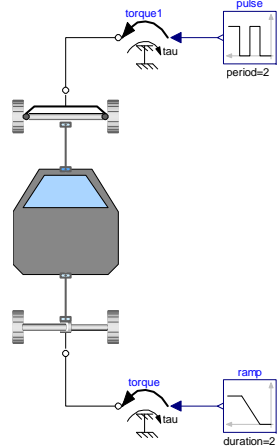
© Dirk Zimmer, December 2014, Slide 2

Two Track Model



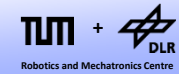
In this lecture, let us look at the modeling of a two-track car model:

- It essentially consists in three parts:
 - The rear axis
 - The front axis
 - The chassis

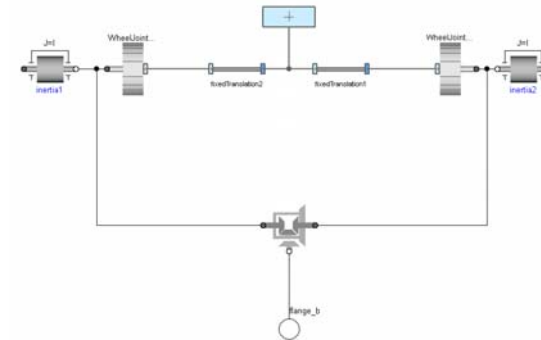


© Dirk Zimmer, December 2014, Slide 3

Rear Axis

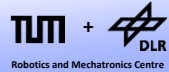


- The rear axis connects the rotation of the two wheels by a differential.
- The wheels are dry-friction based.

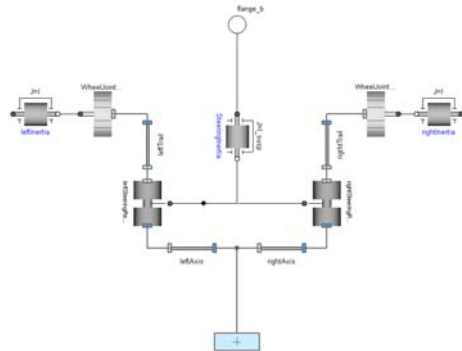


© Dirk Zimmer, December 2014, Slide 4

Front Axis

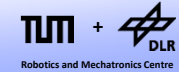


- In the front axis, the steering revolute are rigidly connected.
- The whole steering mechanism shares one common inertia.

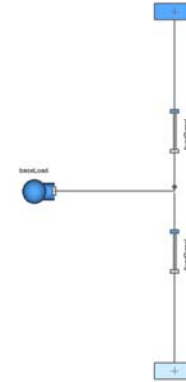


© Dirk Zimmer, December 2014, Slide 5

Chassis

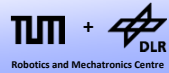


- Modeling the chassis represents a triviality.
- It simply models the "geometry" and puts a mass with inertia at the center.

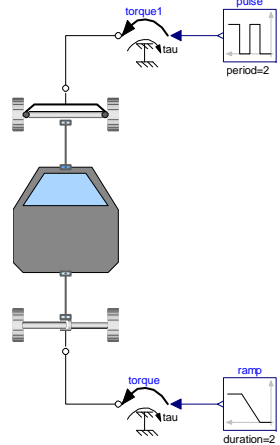


© Dirk Zimmer, December 2014, Slide 6

Experiment

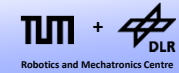


- The experiment on the right ramps up the driving torque
- A pulse wise torque acts on the steering and leads to sudden deflections.

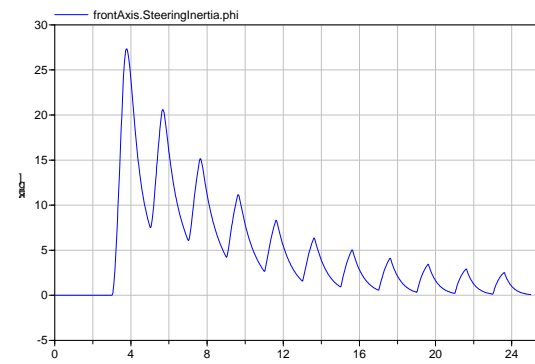


© Dirk Zimmer, December 2014, Slide 7

Experiment Results



- The car is accelerating and the pulsed torque leads to smaller steering angles.



© Dirk Zimmer, December 2014, Slide 8

3D Chassis

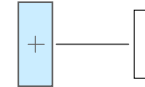
On top of the classis two-track model, we want to build a 3D-chassis that can tilt in two directions (roll and pitch)

- We need a conversion interface from 2D to 3D mechanics.
- We need to model the balancing behavior.

© Dirk Zimmer, December 2014, Slide 9

From 2D to 3D

Let us look at the interface from 2D to 3D:



- From 3D perspective this element is like a joint with 3 degrees of freedom.
- Or alternatively, the “joint” establishes 3 holonomic constraints by restricting the motion to a single plane.

```
model PlanarToMultiBody
  Frame_a frame_a;
  MB.Interfaces.Frame_b frame_b;

protected
  SI.Force fz "Normal Force";
  SI.Force f0[3] "Force vector";

equation
  frame_a.x = frame_b.r_0[1];
  frame_a.y = frame_b.r_0[2];
  0 = frame_b.r_0[3];
  frame_b.R =
    MB.Frames.planarRotation({0,0,1},
      -frame_a.phi, -der(frame_a.phi));

  f0 = {frame_a.fx, frame_a.fy, fz};
  f0*frame_b.R.T + frame_b.f = zeros(3);
  -frame_a.t + frame_b.t[3] = 0;

Connections.root(frame_b.R);
end PlanarToMultiBody
```

© Dirk Zimmer, December 2014, Slide 10

From 2D to 3D

Let us look at the interface from 2D to 3D:



- We prescribe the position...
- ...and the orientation

```
model PlanarToMultiBody
  Frame_a frame_a;
  MB.Interfaces.Frame_b frame_b;

protected
  SI.Force fz "Normal Force";
  SI.Force f0[3] "Force vector";

equation
  frame_a.x = frame_b.r_0[1];
  frame_a.y = frame_b.r_0[2];
  0 = frame_b.r_0[3];
  frame_b.R =
    MB.Frames.planarRotation({0,0,1},
      -frame_a.phi, -der(frame_a.phi));

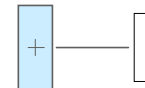
  f0 = {frame_a.fx, frame_a.fy, fz};
  f0*frame_b.R.T + frame_b.f = zeros(3);
  -frame_a.t + frame_b.t[3] = 0;

Connections.root(frame_b.R);
end PlanarToMultiBody
```

© Dirk Zimmer, December 2014, Slide 11

From 2D to 3D

Let us look at the interface from 2D to 3D:



- The force vector is composed and resolved w.r.t. to the body system in order to formulate the balance equation
- There is no need to transform the torque since the torque-vector points in direction of the rotation axis.

```
model PlanarToMultiBody
  Frame_a frame_a;
  MB.Interfaces.Frame_b frame_b;

protected
  SI.Force fz "Normal Force";
  SI.Force f0[3] "Force vector";

equation
  frame_a.x = frame_b.r_0[1];
  frame_a.y = frame_b.r_0[2];
  0 = frame_b.r_0[3];
  frame_b.R =
    MB.Frames.planarRotation({0,0,1},
      -frame_a.phi, -der(frame_a.phi));

  f0 = {frame_a.fx, frame_a.fy, fz};
  frame_b.R.T*f0 + frame_b.f = zeros(3);
  -frame_a.t + frame_b.t[3] = 0;

Connections.root(frame_b.R);
end PlanarToMultiBody
```

© Dirk Zimmer, December 2014, Slide 12

From 2D to 3D

Let us look at the interface from 2D to 3D:



- There remains a strange statement: "Connections.root(...)" It is needed for the handling of kinematic loops.
- We tell Dymola that this component is a potential source of overdetermination.

```

model PlanarToMultiBody
  Frame_a frame_a;
  MB.Interfaces.Frame_b frame_b;

  protected
  SI.Force fz "Normal Force";
  SI.Force f0[3] "Force vector";

  equation
  frame_a.x = frame_b.r_0[1];
  frame_a.y = frame_b.r_0[2];
  0 = frame_b.r_0[3];
  frame_b.R =
    MB.Frames.planarRotation({0,0,1},
      -frame_a.phi, -der(frame_a.phi));

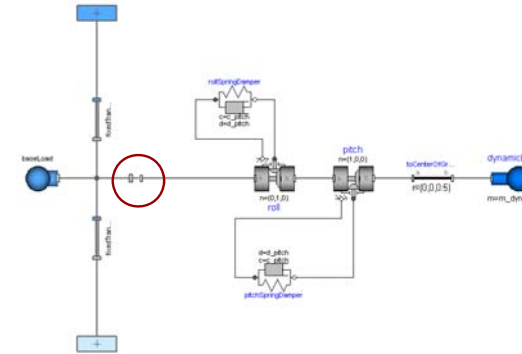
  f0 = {frame_a.fx, frame_a.fy, fz};
  f0*frame_b.R.T + frame_b.f = zeros(3);
  -frame_a.t + frame_b.t[3] = 0;

  Connections.root(frame_b.R);
end PlanarToMultiBody
    
```

© Dirk Zimmer, December 2014, Slide 13

3D Chassis

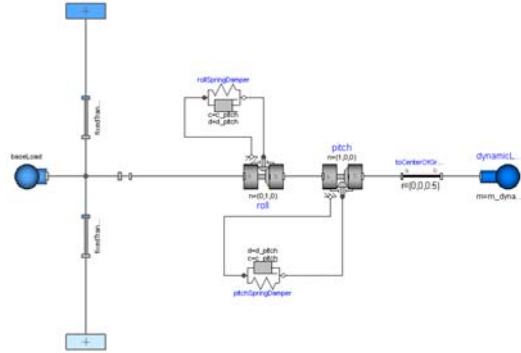
Now, we apply this interface model to build a 3D chassis.



© Dirk Zimmer, December 2014, Slide 14

3D Chassis

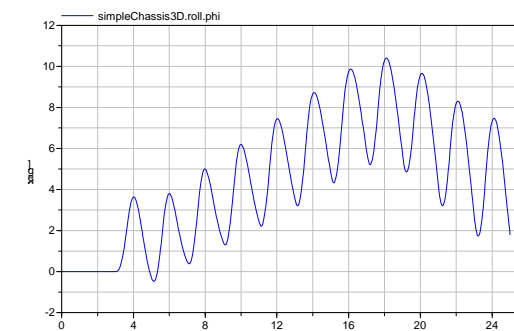
The dynamic load is represented by two revolute joints a mass and a rigid rod. Spring-Damper elements represent the flexibility of the suspension.



© Dirk Zimmer, December 2014, Slide 15

3D Chassis: Experiment

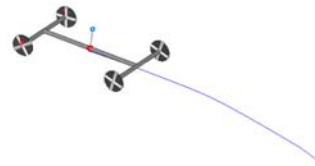
If we repeat the same experiment with the 3D-chassis we can observe the roll angle:



© Dirk Zimmer, December 2014, Slide 16

Summary

- The 3D chassis models the tilt of the chassis but not the dynamic influence on the normal load of the individual wheels.
- This will be your task in the last practical modeling exercise.



© Dirk Zimmer, December 2014, Slide 17

Questions ?