PROJECT CHRONO

Chrono::Vehicle Hands-on

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

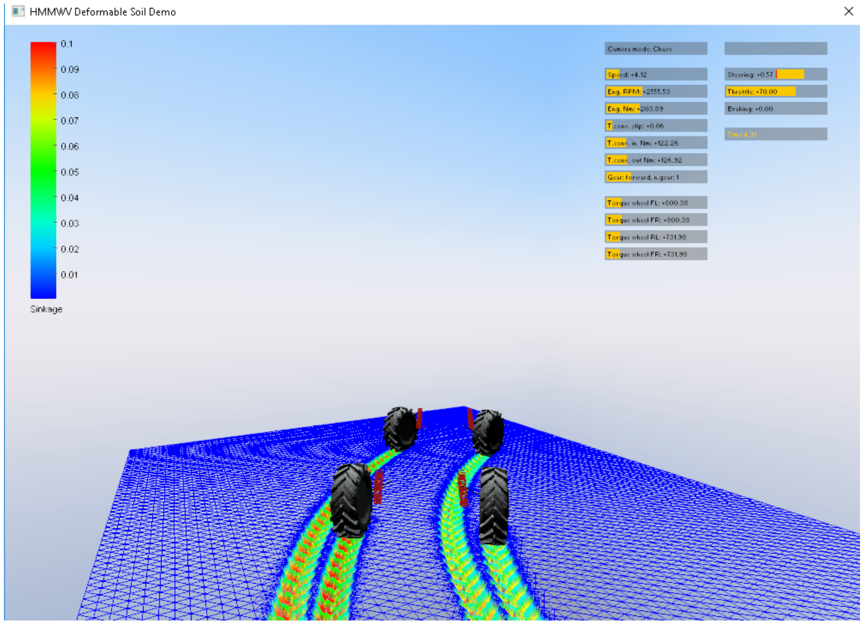
UNIVERSITÀ DEGLI STUDI DI PARMA

Part I

Demos

Wheeled vehicles

- **demo_VEH_HMMWV**
  - HMMWV vehicle model running over a rigid terrain
  - Vehicle defined via Chrono classes (HMMWV_Full)
  - Driver, powertrain, driveline, and tire can be selected by user
  - Terrain can be flat, height map, or mesh
  - Inputs to the vehicle through keyboard/obstacle
  - Online visualization with Irrlicht
  - Optional output for off-line visualization in Pov-Ray
- Other flavors:
  - **demo_VEH_HMMWV_DefSoil**: HMMWV on SCM terrain
  - **demo_VEH_HMMWV9**: Simplified 9-body HMMWV model

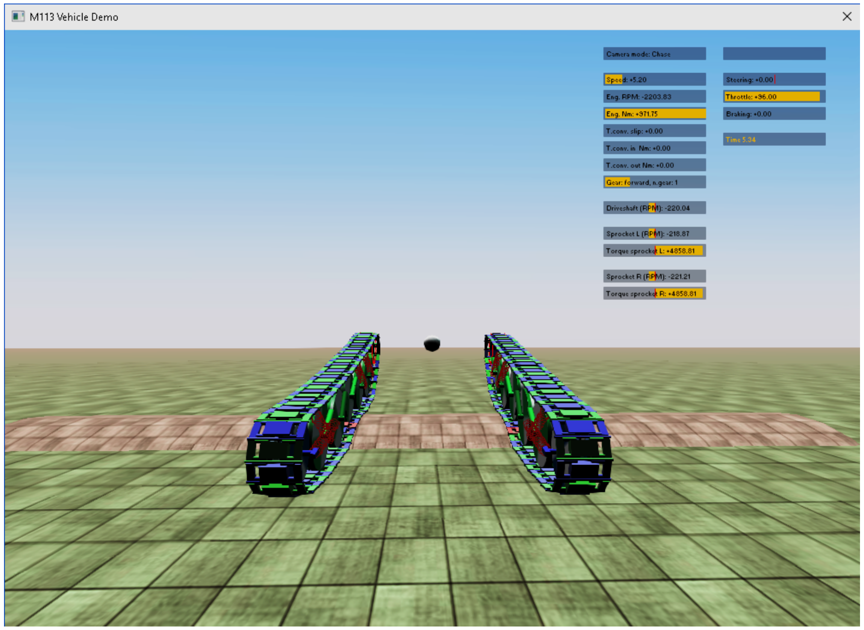


Camera mode: Chase
Speed: +4.12
Eng. RPM: +2555.53
Eng. Nm: +203.89
T.conv. slip: +0.06
T.conv. in Nm: +122.26
T.conv. out Nm: +126.92
Gear: forward, n.gear: 1
Torque wheel FL: +800.38
Torque wheel FR: +800.38
Torque wheel RL: +731.98
Torque wheel FR: +731.98

Steering: +0.57
Throttle: +70.00
Braking: +0.00
Time 4.31

Tracked vehicles

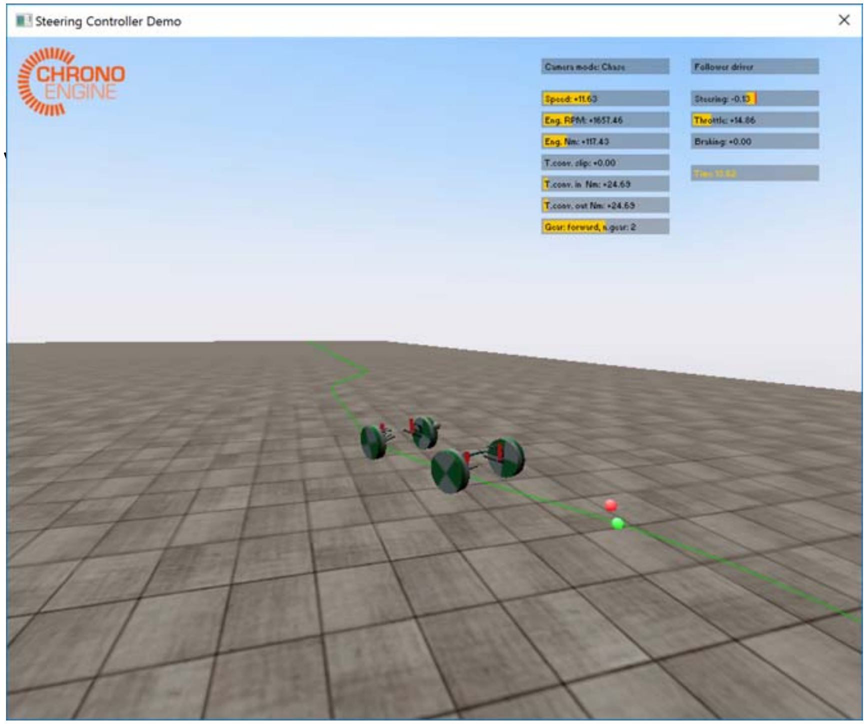
- **demo_VEH_M113**
  - Single-pin M113 tracked vehicle model running over a rigid obstacle
  - Vehicle defined via Chrono classes (M113_Vehicle)
  - Contact can be solved with DEM-Penalty or DEM-Complementarity
  - Inputs to the vehicle through keyboard/obstacle
  - Online visualization with Irrlicht
  - Optional output for off-line visualization in Pov-Ray
- Other flavors:
  - **demo_VEH_M113_DefSoil**: M113 on SCM (def. terrain)
  - **demo_VEH_M113_Parallel**: System simulated in Chrono::Parallel



```
Camera mode: Chase
Speed: +5.20                      Steering: +0.00
Eng. RPM: -2203.83                Throttle: +96.00
Eng. Nm: +971.75                  Braking: +0.00
T.conv. slip: +0.00
T.conv. in  Nm: +0.00             Time 5.34
T.conv. out Nm: +0.00
Gear: forward, n.gear: 1

Driveshaft (RPM): -220.04

Sprocket L (RPM): -218.87
Torque sprocket L: +4858.81

Sprocket R (RPM): -221.21
Torque sprocket R: +4858.81
```

Vehicle with path-follower controller

- **demo_VEH_SteeringController**
  - HMMWV vehicle model running over a rigid terrain
  - Vehicle defined via Chrono classes (HMMWV_Full)
  - Flat, rigid terrain
  - PID steering controller (path-follower)
    - path represents an ISO double lane change maneu
  - PID speed controller (constant speed)
  - Online visualization with Irrlicht
  - Optional output for off-line visualization in Pov-Ray



Camera mode: Chase

Speed: +11.63

Eng. RPM: +1657.46

Eng. Nm: +117.43

T.conv. slip: +0.00

T.conv. in Nm: +24.69

T.conv. out Nm: +24.69

Gear: forward, n.gear: 2

Follower driver

Steering: -0.13

Throttle: +14.86

Braking: +0.00

Time: [illegible]

Vehicles specified through JSON files

- **demo_VEH_WheeledJSON**
  - HMMWV vehicle model running over a rigid terrain
  - Vehicle specified via JSON files (generic/vehicle/Vehicle_DoubleWishbones.json)
  - Driver, powertrain, driveline, and tire are also specified via JSON files

- **demo_VEH_WheeledJSON**
  - Tracked vehicle model running over a rigid terrain
  - Vehicle specified via JSON files (M113/vehicle/M113_Vehicle.json)
  - Driver and simple powertrain are specified via JSON files

Part II

Create a simplified suspension model through a JSON file

Wheeled vehicle suspension example

Goal: Create your own (rear) suspension for a Chrono wheeled vehicle and include it in the default vehicle model

- By default, tutorial's wheeled vehicle uses full wishbone suspensions
- Simplified wheeled vehicle suspensions can be created by the user through JSON files
- Example: DoubleWishboneReduced.json

Example: Reduced double-wishbone suspension

```
{
  "Name":                        "Double Wishbone Reduced",
  "Type":                        "Suspension",
  "Template":                    "DoubleWishboneReduced",

  "Spindle" :
  {
    "Mass":                      15.91,
    "COM":                       [0.036, 0.910, -0.026],
    "Inertia":                   [2, 4, 2],
    "Radius":                    0.15,
    "Width":                     0.06
  },

  "Upright":
  {
    "Mass":                      27.27,
    "COM":                       [0.036, 0.751, -0.026],
    "Inertia":                   [5, 5, 5],
    "Radius":                    0.025
  },

  "Upper Control Arm":
  {
    "Location Chassis Front":   [0.349, 0.462, 0.228],
    "Location Chassis Back":    [0.077, 0.462, 0.224],
    "Location Upright":         [0.036, 0.716, 0.216]
  },
```

Make sure to use the proper template name

Example: Reduced double-wishbone suspension

```
 "Lower Control Arm":
  {
    "Location Chassis Front":   [0.223, 0.307, 0.000],
    "Location Chassis Back":    [-0.223, 0.307, 0.000],
    "Location Upright":         [0.036, 0.787, -0.118]
  },

  "Tierod":
  {
    "Location Chassis":         [0.223, 0.416, 0.059],
    "Location Upright":         [0.170, 0.821, -0.009]
  },

  "Shock":
  {
    "Location Chassis":         [-0.104, 0.498, 0.323],
    "Location Upright":         [-0.097, 0.544, -0.038],
    "Free Length":              0.382
    "Spring Coefficient":       59149.000,
    "Damping Coefficient":      23921.000
  },

  "Axle":
  {
    "Inertia":                  0.4
  }
}
```

Use these attachment points for the spring-damper element

Note that a DoubleWishboneReduced JSON file can only specify linear spring-damper

Hands-on exercise

Steps:

- Locate the input file for the rear suspension of the default wheeled vehicle in the tutorial
- Take a look at the geometric and mechanical parameters defined therein
- Create a simpler suspension model which resembles the JSON on the previous slide
- Note that you will need to change the suspension parameters from the previous slide so that the simplified suspension model resembles the dynamic behavior of the more comprehensive model initially used in the tutorial's wheeled vehicle model