Autonomous Vehicle Simulation Support in Chrono
Connected Autonomous Vehicle Emulator

- Connected Autonomous Vehicle Emulator (CAVE)
  - Connected – simulated connectivity, V2V
  - Autonomous – Chrono sensors
  - Vehicle – Chrono vehicle support
  - Emulator – virtual world support

- Chrono::CAVE
Server and Client

- Distributed Simulation
  - Server in Madison
  - Clients anywhere in world

- Server does not handle any physics

- Server passes agent and world data to Clients

- Clients pass agent data to Server
Server and Client

• Heartbeat
  • Agents must be able to reach next “real-world time” marker within a $\Delta T$ amount of computational time
    • “real-world time” markers are $\delta t$ apart
    • $\Delta T$ called heartbeat
  • Fast agents sleep

• Interactive time for human agents
  • Soft real time

• Agents to play in Server
  • Autonomous vehicles
  • Avatar vehicles
  • Avatar pedestrians
  • Bicyclists
Simulating Connectivity in Chrono

- Simulated Connectivity
  - Vehicles send data directly to nearby agents
  - V2V communication

- Draws on a *Dedicated Short Range Communication* (DSRC) protocol
Sensor Support in Chrono

• Need to be able to simulate sensing

  • LiDAR
    • Sensor implemented without noise
    • Uses collision detection to determine ray length
  • GPS
    • Barebones sensor implemented
  • IMU
    • Barebones sensor implemented
  • Camera
    • Not currently supported, but next in line
    • Dependent on render engine
/In simulation setup
std::shared_ptr<ChRaySensor> lidar = std::make_shared<ChRaySensor>(
  //parent body, update rate, visualize
  my_hmmwv.GetChassis()->GetBody(), 30, true);

lidar->Initialize(chrono::ChCoordsys<double>(
  //offset position
  chrono::ChVector<double>({2.3, 0, 0}),
  //offset orientation
  chrono::ChQuaternion<double>(Q_from_NasaAngles({0, 0, 0}))),
  //samples about y, samples about z, y min/max angle,
  //z min/max angle, min dist, max dist
  1, 100, 0, 0, -1.5, 1.5, .2, 25);

//During simulation loop
lidar->Update();

//To Get Data
lidar->Ranges(); //returns vector containing distance for each ray

Sensor Construction (LiDAR)
Virtual World

• Madison mesh in Chrono from Infraworks/Open Street Maps

• Future Virtual World
  • Based on physical world
    • Buildings, trees, terrain, signs, etc.
  • Environmental effects
    • Rain, snow, ice, fog, etc.
CAVE Demonstration
Future Work

• Server
  • Heartbeat to mandate consistent simulation progression
  • Scaling to allow multi-agent connectivity

• Sensors
  • Expanded sensor capabilities as a module for feedback in Chrono
    • Camera
    • Physically realistic noise models

• Virtual World
  • Physically realistic virtual world
  • Chunk loading management in Chrono
  • Environmental effects