



INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

MATERIALS MANAGEMENT DIVISION

Powai, Mumbai - 400076

Purchase Requisition No. 1000015610 (SRM / RFx No. 6100000417)

Technical Specifications for Truck simulator

Hardware specifications:

The Simulator should consist of the following hardware components:

- Integrated motion-based platform with 3 DOF (degree of freedom).
- Fully instrumented motorcycle frame providing maximum possible realism and accuracy with respect to the locations of controls, driver visibility and feel of driving.
- 180 degree curved screen projection for visual display.
- A high-fidelity force feedback bike handle/braking system that can emulate realistic force cues depending on steering angle and vehicle speed. A synchronized audio feedback system should also be enabled with this simulator.
- This system should have very low inertia, zero cogging, and tight coupling with the vehicle dynamics allowing for good steering feel to provide increased vehicle control and transfer of training.
- An integrated laptop based operator system for scenario control.
- All the instructions and controls of the simulator should be in English language.

Software specifications:

- The software should enable the user to design driving scenarios with different road conditions and terrains.
- The core software of the vehicle simulator should have graphical modeling environment with standard library of components for basic mathematical operations and signals.
- Should include a GUI (Graphical User Interface) based customizable driving conditions, i.e. the simulator software should have GUI supported tile based database and scenario authoring tool to create various rural, suburban and city driving scenarios.
- Scenario control should include ambient traffic simulation, scriptable events, relational behaviors and environmental controls.
- The simulator software should have a GUI based easy to edit vehicle data files and easy to create new vehicle types interface for user defined vehicles for Indian road conditions.
- The software should be equipped with user-defined events such as pedestrian crossing, and enable the user to change the driving environment (such as different intensities of daylight, rain, fog, mist and snow)

- The software should be compatible with latest version of MS windows based operating system.
- Should include all type of vehicles matching Indian road transport scenarios.
- The system should have data collection/performance measurement functions with capability of providing output in multiple formats (such as video, standard CSV/Excel file, text file etc.) for the following parameters:

Parameters	Value Range
Time stamp	From start to end of the simulation
Vehicle position (x,y,z) coordinate	All vehicle (surrounding and driven vehicle) positions at regular interval (e.g. every 0.5 sec. or less)
Vehicle Roll, pitch, yaw, speed, lateral acceleration, longitudinal acceleration data	All data at regular interval (e.g. every 0.5 sec. or less) for the driven vehicle in SI unit.
Accelerator, Brake position, Brake pedal force, Steering angle, Gear position, Engine RPM, Turn signal	All data at regular interval (e.g. every 0.5 sec. or less) for the driven vehicle.
Headway distance, Headway time, Tailway distance, Tailway time, Time to collision, Lane number, Lane offset, Road offset	All data at regular interval (e.g. every 0.5 sec. or less) for the driven vehicle.
Collision Information	All vital vehicle parameters of surrounding and driven vehicle at the time of collision

- The software should be enabled to perform the following functions related to vehicle dynamics as shown in the below table:

Here, ‘entity’ refers to a dynamic entity (scenario vehicle, subject vehicle) in the simulation.

VARIABLES	FUNCTIONS
Set Fixed Velocity	A value representing the "FIXED" velocity state of the entity (e.g., giving a fixed value to the scenario vehicle).
SetDesiredVelocity	Sets the desired velocity for this entity.
SetVelocity	Sets the velocity and overrides any acceleration/deceleration commands
SetVelocityOnce	Instantaneously sets the velocity to a specific value
GetVelocity	Gets the current velocity for the entity in M/S.
SetMaxAcceleration	Sets the desired acceleration for the entity.
SetAcceleration	Sets the acceleration for the entity and will override any velocity commands
GetAcceleration	Gets the desired acceleration for the entity.
SetMinDeceleration	Sets the desired deceleration for the entity.
GetDeceleration	Gets the desired deceleration for the entity.
SetDesiredHeadwayTime	Sets the desired headway time for the entity.
GetHeadwavTime	Gets the desired headway time for the entity.
SetDesiredHeadwayDistance	Sets the desired headway distance for the entity.
GetHeadwayDistance	Gets the desired headway distance for the entity
SetDesiredTailwayTime	Sets the desired Tailway time for the entity.
GetTailwayTime	Gets the desired Tailway time for the entity.
SetDesiredTailwayDistance	Sets the desired Tailway distance for the entity.
GetTailwayDistance	Gets the desired Tailway distance for the entity.
SetLaneoffset	Sets the desired lane offset distance tor the entity.
GetLaneoffset	Gets the actual lane offset distance for the entity.
SetRoadoffset	Sets the road Offset for an entity

GetRoadoffset	Gets the offset of the entity relative to the center line +ve is the offset to positive lanes (meters)
GetHeadingError	Gets the heading error in the lane to the right (degrees)
GetRoadDistance	Gets the distance down the road from the zero point (meters)
Random Lane	A value used to select a "RANDOM" lane
Left most lane	A value used to select the leftmost lane on the roadway, in the current direction of travel.
Right most lane	A value used to select the rightmost lane on the roadway, in the current direction of travel.
Left turn lane	A value used to select the left turn lane on the roadway, in the current direction of travel.
Right turn lane	A value used to select the right lane on the roadway, in the current direction of travel.
SetDesiredLane	Specifies a lane that this entity should switch to as soon as possible.
GetLane	Gets the lane that the vehicle is in and is relative to the direction of travel.
Turn left	A value to indicate the vehicle should turn left at the next intersection.
Turn right	A value used to indicate the vehicle should turn right at the next intersection.
Continue straight	A value to indicate the vehicle should go straight at the next intersection.
Set Desired turn	Specifies a direction the vehicle should turn at the next opportunity.
Set light state	Sets the light state of the entity, eq. brake, turn signal, hazards
Steering	Gets the steering value for this entity.
Throttle	Gets the throttle value for this entity.
Brake	Gets the Brake value for this entity.
SpatialDatum	Gets the SpatialDatum of this entity.
Gear	Gets the Gear value for this entity.
Accessory	Sets the accessory to be displayed for this entity
Posture	Sets the entities posture.
ExecuteAction	Executes a specified action animation for the entity.
Stop traverse	A value indicating that the entity should come to a complete stop when it completes the traversal.
Join traverse	A value indicating that the entity should attempt to join the roadway network at the end of the traversal.
Traverse	Attaches the entity to the Given object, causing the entity approach and traverse the object.
CurrentTraversal	Retrieves the current traversal state of this entity.
Join	Join's the entity to the road if a road exists
All behavior	value representing all behavior parameters that can be set.
Behavior headway	Value representing the HEADWAY distance behavior.
Behavior tailway	value representing the TAILWAY distance behavior.
Speed limit	value representing the SPEEDLIMIT behavior.
Acceleration	Value representing the ACCELERATION behavior.
Deceleration	Value representing the DECELERATION behavior.

Lane behavior	Value representing the LANE behavior.
Lane offset	Value representing the LANE_OFFSET behavior
Headway time	value representing the HEADWAY_TIME behavior.
Tailway time	Value representing the TAILWAY TIME behavior.
ResetBehavior	Resets the internal behavior parameters of the entity control model based on the flags supplied.
Subject	Checks if this entity is a/the proxy for the "subject" entity.
GetDirection	Gets the IntersectionController direction that the entity is approaching on the intersection.
GetIntersectionController	Gets the IntersectionController that the entity is approaching.
GetPhase	Gets the traffic light phase that affects the entity.
SetPhase	Sets the traffic light phase for the intersection and direction that will affect the entity.
GetCollision	Gets the collision object For an entity. Currently only implemented on the subject vehicle.
SetTrafficResponse	Implemented for StaticObjects. Changes where an object is considered a traffic participant
SetVisible	Causes the entity to become visible or invisible.
GetVisible	Returns if an entity is visible
GetZOffset	Returns the height above terrain for the StaticObject
SetZOffset	Sets the height above the terrain for the StaticObject

Additional requirements:

- The simulator should be compatible with respect to the connected network (for future connection with other simulators where the drivers can drive and interact with one another in a connected environment).The connected environment is a dynamic feedback system between the routing behavior of drivers and system performance.
- The system should have option to seamlessly integrate with PTV VISSIM, SUMO and other traffic simulation software and programming platforms.
- Should support exporting/importing simulation models to MATLAB, UNITY 3D,C/C++ code and libraries for use in external code compiler.