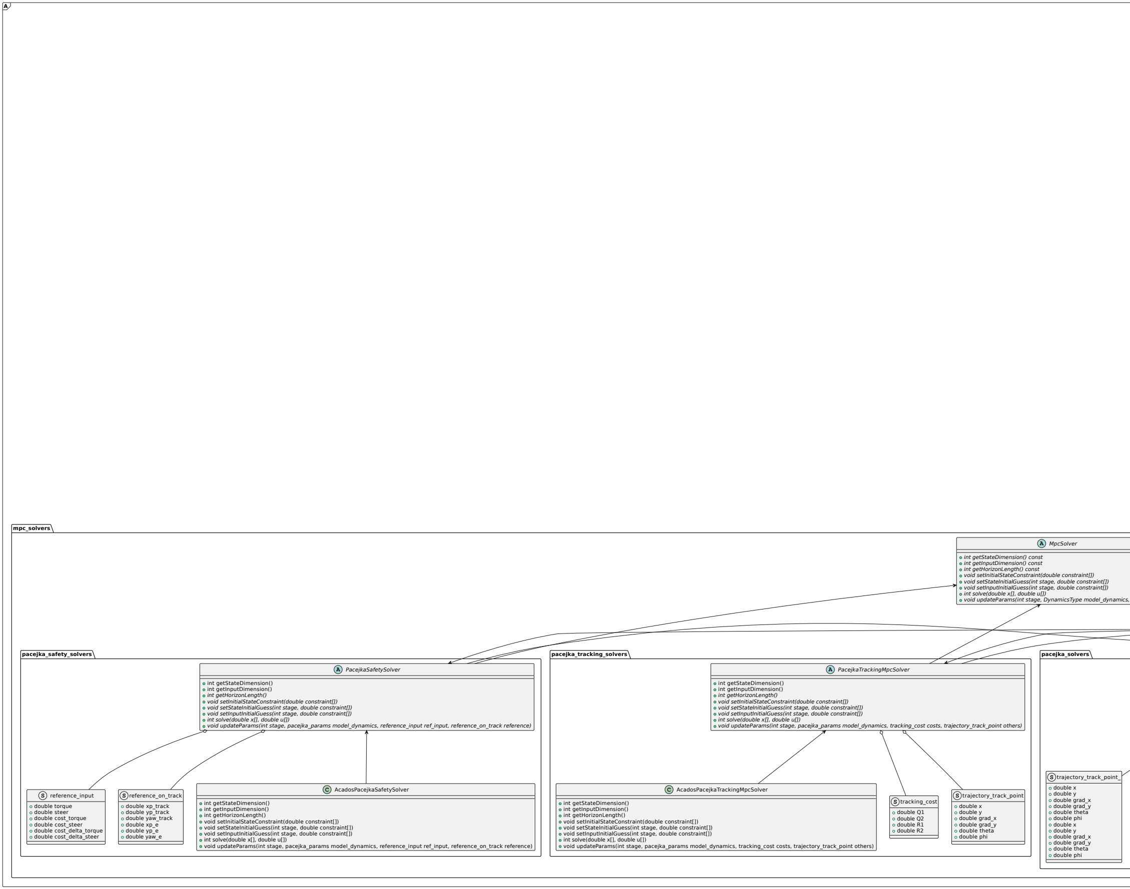
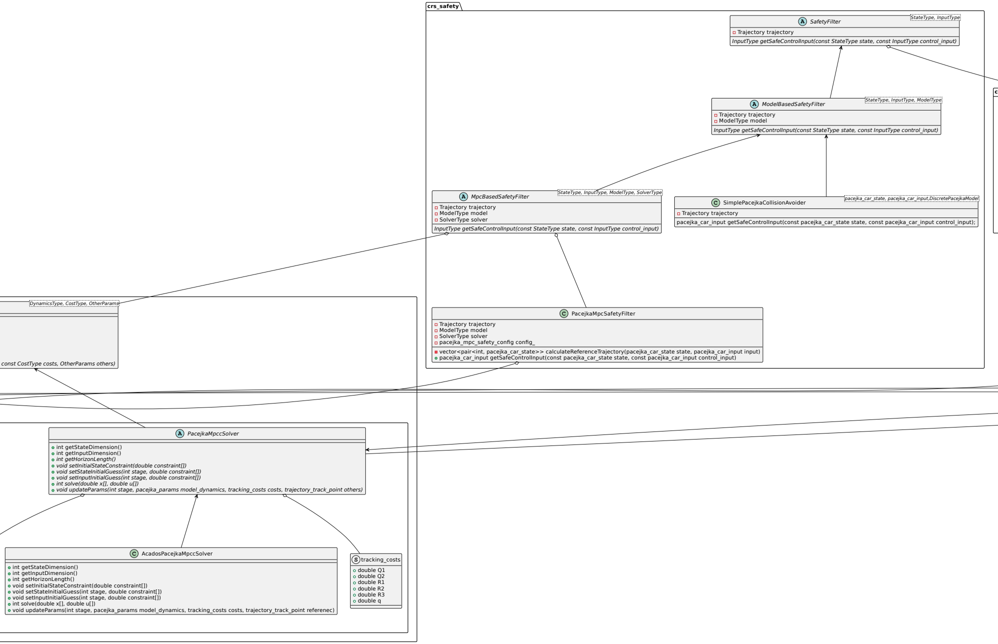
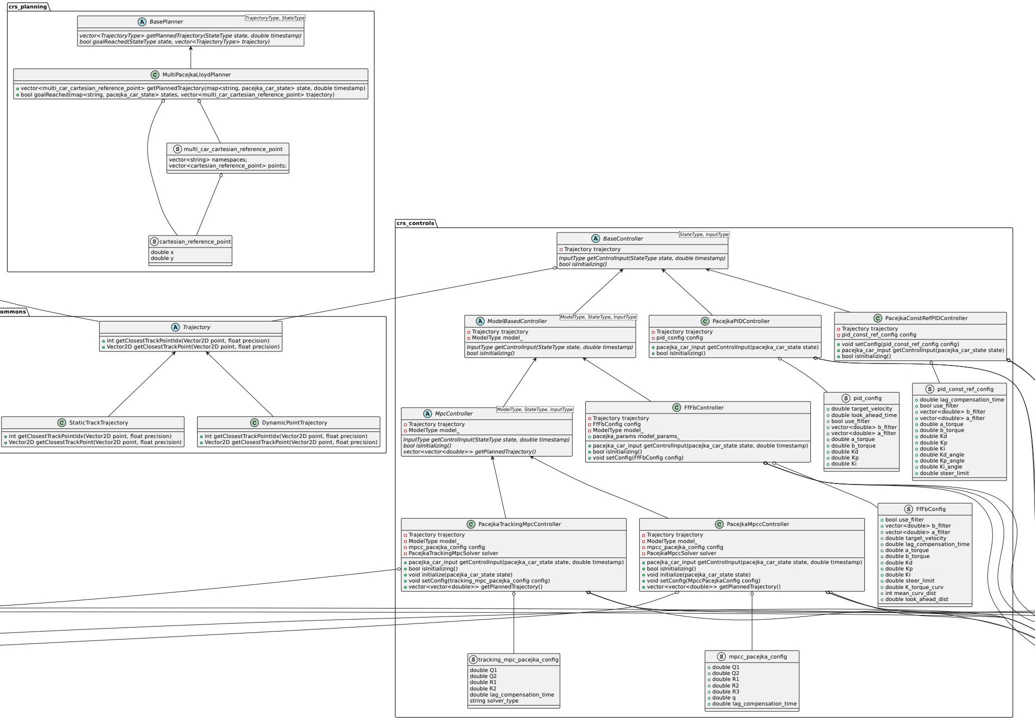
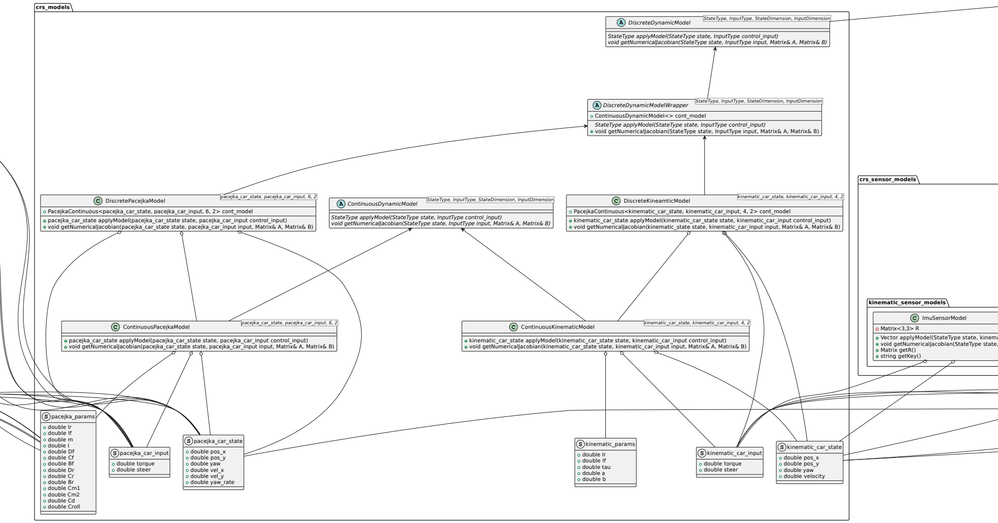
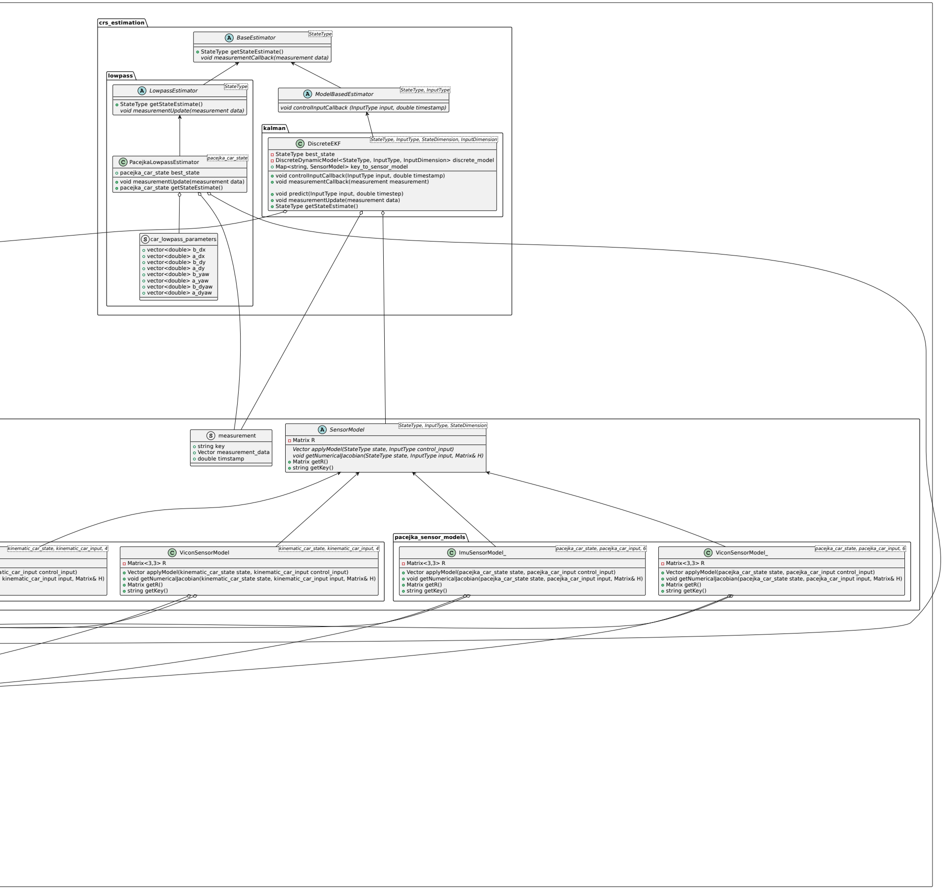

A
mpc_solvers
MpcSolver
+ int getStateDimension() const
+ int getInputDimension() const
+ int getHorizonLength() const
+ void setInitialStateConstraint(double constraint[])
+ void setStateInitialGuess(int stage, double constraint[])
+ void setInputInitialGuess(int stage, double constraint[])
+ int solve(double x[], double u[])
+ void updateParams(int stage, DynamicsType model_dynamics,
pacejka_safety_solvers
PacejkaSafetySolver
+ int getStateDimension()
+ int getInputDimension()
+ int getHorizonLength()
+ void setInitialStateConstraint(double constraint[])
+ void setStateInitialGuess(int stage, double constraint[])
+ void setInputInitialGuess(int stage, double constraint[])
+ int solve(double x[], double u[])
+ void updateParams(int stage, pacejka_params model_dynamics, reference_input ref_input, reference_on_track reference)
reference_input
+ double torque
+ double steer
+ double cost_torque
+ double cost_steer
+ double cost_delta_torque
+ double cost_delta_steer
reference_on_track
+ double xp_track
+ double yp_track
+ double yaw_track
+ double xp_e
+ double yp_e
+ double yaw_e
AcadosPacejkaSafetySolver
+ int getStateDimension()
+ int getInputDimension()
+ int getHorizonLength()
+ void setInitialStateConstraint(double constraint[])
+ void setStateInitialGuess(int stage, double constraint[])
+ void setInputInitialGuess(int stage, double constraint[])
+ int solve(double x[], double u[])
+ void updateParams(int stage, pacejka_params model_dynamics, reference_input ref_input, reference_on_track reference)
pacejka_tracking_solvers
PacejkaTrackingMpcSolver
+ int getStateDimension()
+ int getInputDimension()
+ int getHorizonLength()
+ void setInitialStateConstraint(double constraint[])
+ void setStateInitialGuess(int stage, double constraint[])
+ void setInputInitialGuess(int stage, double constraint[])
+ int solve(double x[], double u[])
+ void updateParams(int stage, pacejka_params model_dynamics, tracking_cost costs, trajectory_track_point others)
AcadosPacejkaTrackingMpcSolver
+ int getStateDimension()
+ int getInputDimension()
+ int getHorizonLength()
+ void setInitialStateConstraint(double constraint[])
+ void setStateInitialGuess(int stage, double constraint[])
+ void setInputInitialGuess(int stage, double constraint[])
+ int solve(double x[], double u[])
+ void updateParams(int stage, pacejka_params model_dynamics, tracking_cost costs, trajectory_track_point others)
tracking_cost
+ double Q1
+ double Q2
+ double R1
+ double R2
trajectory_track_point
+ double x
+ double y
+ double grad_x
+ double grad_y
+ double theta
+ double phi
pacejka_solvers
trajectory_track_point_
+ double x
+ double y
+ double grad_x
+ double grad_y
+ double theta
+ double phi
+ double x
+ double y
+ double grad_x
+ double grad_y
+ double theta
+ double phi


crs_safety
SafetyFilter
StateType, InputType
Trajectory trajectory
InputType getSafeControlInput(const StateType state, const InputType control_input)
ModelBasedSafetyFilter
StateType, InputType, ModelType
Trajectory trajectory
ModelType model
InputType getSafeControlInput(const StateType state, const InputType control_input)
MpcBasedSafetyFilter
StateType, InputType, ModelType, SolverType
Trajectory trajectory
ModelType model
SolverType solver
InputType getSafeControlInput(const StateType state, const InputType control_input)
SimplePacejkaCollisionAvoider
pacejka_car_state, pacejka_car_input,DiscretePacejkaModel
Trajectory trajectory
pacejka_car_input getSafeControlInput(const pacejka_car_state state, const pacejka_car_input control_input);
PacejkaMpcSafetyFilter
Trajectory trajectory
ModelType model
SolverType solver
pacejka_mpc_safety_config config_
vector<pair<int, pacejka_car_state>> calculateReferenceTrajectory(pacejka_car_state state, pacejka_car_input input)
pacejka_car_input getSafeControlInput(const pacejka_car_state state, const pacejka_car_input control_input)
DynamicsType, CostType, OtherParams
const CostType costs, OtherParams others)
PacejkaMpccSolver
int getStateDimension()
int getInputDimension()
int getHorizonLength()
void setInitialStateConstraint(double constraint[])
void setStateInitialGuess(int stage, double constraint[])
void setInputInitialGuess(int stage, double constraint[])
int solve(double x[], double u[])
void updateParams(int stage, pacejka_params model_dynamics, tracking_costs costs, trajectory_track_point others)
AcadosPacejkaMpccSolver
int getStateDimension()
int getInputDimension()
int getHorizonLength()
void setInitialStateConstraint(double constraint[])
void setStateInitialGuess(int stage, double constraint[])
void setInputInitialGuess(int stage, double constraint[])
int solve(double x[], double u[])
void updateParams(int stage, pacejka_params model_dynamics, tracking_costs costs, trajectory_track_point referenec)
tracking_costs
double Q1
double Q2
double R1
double R2
double R3
double q


crs_planning
A BasePlanner
TrajectoryType, StateType
vector<TrajectoryType> getPlannedTrajectory(StateType state, double timestamp)
bool goalReached(StateType state, vector<TrajectoryType> trajectory)
C MultiPacejkaLloydPlanner
vector<multi_car_cartesian_reference_point> getPlannedTrajectory(map<string, pacejka_car_state> state, double timestamp)
bool goalReached(map<string, pacejka_car_state> states, vector<multi_car_cartesian_reference_point> trajectory)
S multi_car_cartesian_reference_point
vector<string> namespaces;
vector<cartesian_reference_point> points;
S cartesian_reference_point
double x
double y
ommons
A Trajectory
int getClosestTrackPointIdx(Vector2D point, float precision)
Vector2D getClosestTrackPoint(Vector2D point, float precision)
C StaticTrackTrajectory
int getClosestTrackPointIdx(Vector2D point, float precision)
Vector2D getClosestTrackPoint(Vector2D point, float precision)
C DynamicPointTrajectory
int getClosestTrackPointIdx(Vector2D point, float precision)
Vector2D getClosestTrackPoint(Vector2D point, float precision)
crs_controls
A BaseController
StateType, InputType
Trajectory trajectory
InputType getControlInput(StateType state, double timestamp)
bool isInitializing()
A ModelBasedController
ModelType, StateType, InputType
Trajectory trajectory
ModelType model_
InputType getControlInput(StateType state, double timestamp)
bool isInitializing()
C PacejkaPIDController
Trajectory trajectory
pid_config config
pacejka_car_input getControlInput(pacejka_car_state state)
bool isInitializing()
C PacejkaConstRefPIDController
Trajectory trajectory
pid_const_ref_config config
void setConfig(pid_const_ref_config config)
pacejka_car_input getControlInput(pacejka_car_state state)
bool isInitializing()
S pid_config
double target_velocity
double look_ahead_time
bool use_filter
vector<double> b_filter
vector<double> a_filter
double a_torque
double b_torque
double Kd
double Kp
double Ki
S pid_const_ref_config
double lag_compensation_time
bool use_filter
vector<double> b_filter
vector<double> a_filter
double a_torque
double b_torque
double Kd
double Kp
double Ki
double Kd_angle
double Kp_angle
double Ki_angle
double steer_limit
A MpcController
ModelType, StateType, InputType
Trajectory trajectory
ModelType model_
InputType getControlInput(StateType state, double timestamp)
bool isInitializing()
vector<vector<double>> getPlannedTrajectory()
C FfFbController
Trajectory trajectory
FfFbConfig config
ModelType model_
pacejka_params model_params_
pacejka_car_input getControlInput(pacejka_car_state state, double timestamp)
bool isInitializing()
void setConfig(FfFbConfig config)
S FfFbConfig
bool use_filter
vector<double> b_filter
vector<double> a_filter
double target_velocity
double lag_compensation_time
double a_torque
double b_torque
double Kd
double Kp
double Ki
double steer_limit
double K_torque_curv
int mean_curv_dist
double look_ahead_dist
C PacejkaTrackingMpcController
Trajectory trajectory
ModelType model_
mpcc_pacejka_config config
PacejkaTrackingMpcSolver solver
pacejka_car_input getControlInput(pacejka_car_state state, double timestamp)
bool isInitializing()
void initialize(pacejka_car_state state)
void setConfig(tracking_mpc_pacejka_config config)
vector<vector<double>> getPlannedTrajectory()
C PacejkaMpccController
Trajectory trajectory
ModelType model_
mpcc_pacejka_config config
PacejkaMpccSolver solver
pacejka_car_input getControlInput(pacejka_car_state state, double timestamp)
bool isInitializing()
void initialize(pacejka_car_state state)
void setConfig(MpccPacejkaConfig config)
vector<vector<double>> getPlannedTrajectory()
S tracking_mpc_pacejka_config
double Q1
double Q2
double R1
double R2
double lag_compensation_time
string solver_type
S mpcc_pacejka_config
double Q1
double Q2
double R1
double R2
double R3
double q
double lag_compensation_time


crs_models
DiscreteDynamicModel
StateType, InputType, StateDimension, InputDimension
StateType applyModel(StateType state, InputType control_input)
void getNumericalJacobian(StateType state, InputType input, Matrix& A, Matrix& B)
DiscreteDynamicModelWrapper
StateType, InputType, StateDimension, InputDimension
ContinuousDynamicModel<> cont_model
StateType applyModel(StateType state, InputType control_input)
void getNumericalJacobian(StateType state, InputType input, Matrix& A, Matrix& B)
DiscretePacejkaModel
pacejka_car_state, pacejka_car_input, 6, 2
PacejkaContinuous<pacejka_car_state, pacejka_car_input, 6, 2> cont_model
pacejka_car_state applyModel(pacejka_car_state state, pacejka_car_input control_input)
void getNumericalJacobian(pacejka_car_state state, pacejka_car_input input, Matrix& A, Matrix& B)
ContinuousDynamicModel
StateType, InputType, StateDimension, InputDimension
StateType applyModel(StateType state, InputType control_input)
void getNumericalJacobian(StateType state, InputType input, Matrix& A, Matrix& B)
DiscreteKineamticModel
kinematic_car_state, kinematic_car_input, 4, 2
PacejkaContinuous<kinematic_car_state, kinematic_car_input, 4, 2> cont_model
kinematic_car_state applyModel(kinematic_car_state state, kinematic_car_input control_input)
void getNumericalJacobian(kinematic_state state, kinematic_car_input input, Matrix& A, Matrix& B)
ContinuousPacejkaModel
pacejka_car_state, pacejka_car_input, 6, 2
pacejka_car_state applyModel(pacejka_car_state state, pacejka_car_input control_input)
void getNumericalJacobian(pacejka_car_state state, pacejka_car_input input, Matrix& A, Matrix& B)
ContinuousKinematicModel
kinematic_car_state, kinematic_car_input, 4, 2
kinematic_car_state applyModel(kinematic_car_state state, kinematic_car_input control_input)
void getNumericalJacobian(kinematic_car_state state, kinematic_car_input input, Matrix& A, Matrix& B)
pacejka_params
double lr
double lf
double m
double I
double Df
double Cf
double Bf
double Dr
double Cr
double Br
double Cm1
double Cm2
double Cd
double Croll
pacejka_car_input
double torque
double steer
pacejka_car_state
double pos_x
double pos_y
double yaw
double vel_x
double vel_y
double yaw_rate
kinematic_params
double lr
double lf
double tau
double a
double b
kinematic_car_input
double torque
double steer
kinematic_car_state
double pos_x
double pos_y
double yaw
double velocity
crs_sensor_models
kinematic_sensor_models
ImuSensorModel
Matrix<3,3> R
Vector applyModel(StateType state, kinema
void getNumericalJacobian(StateType state,
Matrix getR()
string getKey()


crs_estimation
BaseEstimator
StateType
StateType getStateEstimate()
void measurementCallback(measurement data)
lowpass
LowpassEstimator
StateType
StateType getStateEstimate()
void measurementUpdate(measurement data)
PacejkaLowpassEstimator
pacejka_car_state
pacejka_car_state best_state
void measurementUpdate(measurement data)
pacejka_car_state getStateEstimate()
car_lowpass_parameters
vector<double> b_dx
vector<double> a_dx
vector<double> b_dy
vector<double> a_dy
vector<double> b_yaw
vector<double> a_yaw
vector<double> b_dyaw
vector<double> a_dyaw
ModelBasedEstimator
StateType, InputType
void controlInputCallback (InputType input, double timestamp)
kalman
DiscreteEKF
StateType, InputType, StateDimension, InputDimension
StateType best_state
DiscreteDynamicModel<StateType, InputType, InputDimension> discrete_model
Map<string, SensorModel> key_to_sensor_model
void controlInputCallback(InputType input, double timestamp)
void measurementCallback(measurement measurement)
void predict(InputType input, double timestep)
void measurementUpdate(measurement data)
StateType getStateEstimate()
measurement
string key
Vector measurement_data
double timstamp
SensorModel
StateType, InputType, StateDimension
Matrix R
Vector applyModel(StateType state, InputType control_input)
void getNumericalJacobian(StateType state, InputType input, Matrix& H)
Matrix getR()
string getKey()
kinematic_car_state, kinematic_car_input, 4
atic_car_input control_input)
kinematic_car_input input, Matrix& H)
ViconSensorModel
kinematic_car_state, kinematic_car_input, 4
Matrix<3,3> R
Vector applyModel(kinematic_car_state state, kinematic_car_input control_input)
void getNumericalJacobian(kinematic_car_state state, kinematic_car_input input, Matrix& H)
Matrix getR()
string getKey()
pacejka_sensor_models
ImuSensorModel_
pacejka_car_state, pacejka_car_input, 6
Matrix<3,3> R
Vector applyModel(pacejka_car_state state, pacejka_car_input control_input)
void getNumericalJacobian(pacejka_car_state state, pacejka_car_input input, Matrix& H)
Matrix getR()
string getKey()
ViconSensorModel_
pacejka_car_state, pacejka_car_input, 6
Matrix<3,3> R
Vector applyModel(pacejka_car_state state, pacejka_car_input control_input)
void getNumericalJacobian(pacejka_car_state state, pacejka_car_input input, Matrix& H)
Matrix getR()
string getKey()