ikf

IKalmanFilter

- HistBelief : TTimeHorizonBuffer<ptr_belief>
- HistMeas : TTimeHorizonBuffer<MeasData, TMultiHistoryBuffer<MeasData>>
- max_time_horizon_sec :double
- m_handle_delayed_meas: bool

+ IKalmanFilter(...)
+ IKalmanFilter(...)
+ reset() : void
+ initialize(ptr_belief): void
+ set_horizon(double const) : void
+ redo_updates_after_t(Timestamp const& t) : bool
+ current_t() : Timestamp
+ current_belief() : ptr_belief
+ exist_belief_at_t(Timestamp const& t) : bool
+ get_belief_at_t(Timestamp const& t) : ptr_belief
+ get_belief_at_t(Timestamp const& t, ptr_belief& bel) : bool
+ set_belief_at_t(ptr_belief const& bel, Timestamp const&t) : void
+ get_belief_before_t(Timestamp const&t, ptr_belief& bel, Timestamp &t_before) : bool
# progapation_measurement(MeasData const& m) : ProcessMeasResult_t = 0
# local_private_measurement(MeasData const& m) :ProcessMeasResult_t = 0
# correct_belief_at_t(Eigen::VectorXd const& mean_corr, Eigen::MatrixXd const& Sigma_apos, Timestamp const&t) : bool
# remove_beliefs_after_t(Timestamp const& t) : void
# check_horizon() : void
# reprocess_measurement(MeasData const& m) : ProcessMeasResult_t
# propagate_from_to(const Timestamp &t_a, const Timestamp &t_b) : bool
# apply_propagation(const Eigen::MatrixXd &Phi_II_ab, const Eigen::MatrixXd &Q_II_ab, const Timestamp &t_a, const Timestamp &t_b) : bool
# apply_private_observation(const Eigen::MatrixXd &H_II, const Eigen::MatrixXd &R, const Eigen::VectorXd &z, const Timestamp &t) : bool

typedef std::unordered_map<size_t, TTimeHorizonBuffer<Eigen::MatrixXd>> HistDict_t
typedef std::unordered_map<size_t, std::shared_ptr<IIsolatedKalmanFilter>> DictID_t
typedef std::shared_ptr<IBelief> ptr_belief;
typedef std::shared_ptr<IsolatedKalmanFilterHandler> ptr_handler,
typedef std::shared_ptr<IIsolatedKalmanFilter> ptr_IKF;

IBelief

- m_mean: Eigen::VectorXd
- m_Sigma: : Eigen::MatrixXd
- m_timestamp : ikf::Timestamp

+ clone() : IBelief
+ correct(Eigen::VectorXd const&)

*

MeasData

+ t_m/t_p : Timestmap
+ id_sensor : size_t
+ meas_tpye/meta_info : string
+ obs_type : eObservationType
+ z : Eigen::VectorXd
+ R : Eigen::MatrixXd

*

eObservationType < enum class

+ UNKNOWN = 0,
+ PROPAGATION = 1,
+ PRIVATE_OBSERVATION = 2,
+ JOINT_OBSERVATION = 3,

IIsolatedKalmanFilter

- HistCrossCovFactors: HistDict_t
- ptr_Handler : std::shared_ptr<IsolatedKalmanFilterHandler>

+ IIsolatedKalmanFilter(ptr_handler, size_t const ID, ...)
+ ID(): size_t
+ reset() : void
+ initialize(ptr_belief): void
+ set_horizon(double const) : void
+ process_measurement(MeasData const&) : ProcessMeasResult_t
+ get_measurements_after/from_t(Timestamp const&) : TMultiHistoryBuffer<MeasData>>
+ get_measurements_at_t(Timestamp const&t) : std::vector<MeasData>
+ reprocess_measurement(MeasData const&) : ProcessMeasResult_t
+ get_CrossCovFact_at_t(Timestamp const&, size_t ID_J) : Eigen::MatrixXd
+ set_CrossCovFact_at_t(Timestamp const&, size_t unique_ID, Eigen::MatrixXd const& ccf) : void
+ apply_correction_at_t(Timestamp const&t, Eigen::MatrixXd const& Sigma_apri, Eigen::MatrixXd const Sigma_apos) : bool
+ remove_after/from_t(Timestamp const& t) : void
# redo_updates_after_t(Timestamp const& t) : bool
# local_joint_measurement(MeasData const& m) : ProcessMeasResult_t = 0
# get_CrossCovFact_at/before_t(Timestamp const& t, size_t ID_J, Eigen::MatrixXd &) : bool
# propagate_CrossCovFact(Timestamp const& t_a, Timestamp const& t_b, Eigen::MatrixXd const& M_a_b) : void
# check_horizon()
# add_correction_at_t(const Timestamp &t_a,Timestamp const& t_b, Eigen::MatrixXd const& Phi_a_b) : bool
# apply_correction_at_t(Timestamp const&t, Eigen::MatrixXd const& Factor) : bool
# apply_propagation() : bool
# apply_private_observation(...) : bool
# apply_joint_observation(const size_t ID_I, const size_t ID_J, const Eigen::MatrixXd &H_II, const Eigen::MatrixXd &H_JJ, const Eigen::MatrixXd &R, const Eigen::VectorXd &z, const Timestamp &t) : bool
# stack_Sigma(...) : Eigen::MatrixXd
# split_Sigma(...) : void
# stack_apri_covariance(...) : Eigen::MatrixXd
# get_Sigma_IJ_at_t(...) : Eigen::MatrixXd
# set_Sigma_IJ_at_t(...) : void

*

1

IsolatedKalmanFilterHandler

- id_dict : DictID_t
- HistMeas : TTimeHorizonBuffer<MeasData, TMultiHistoryBuffer<MeasData>>
- max_time_horizon_sec :double
- m_handle_delayed_meas : bool

+ add(ptr_IKF p_IKF) : bool
+ get(const size_t ID) :ptr_IKF
+ remove(const size_t ID) : bool
+ exists(const size_t ID) : bool
+ get_instance_ids() : std::vector<size_t>
+ sort_measurements_from_t(Timestamp const& t) : void
+ process_measurement(MeasData const& m) : ProcessMeasResult_t
+ get_measurements_after/from_t(Timestamp const&) : TMultiHistoryBuffer<MeasData>>
+ reprocess_measurement(MeasData const&) : ProcessMeasResult_t
+ remove_beliefs_after_t(Timestamp const& t) : void
+ remove_beliefs_from_t(Timestamp const& t) : void
# redo_updates_from_t(const Timestamp &t) : bool
# redo_updates_after_t(const Timestamp &t) : bool

Mass_Spring_Damper_1D