Public Member Functions
//퍼블릭 멤버함수

KalmanFilter ()
/* 생성자 */

KalmanFilter (int dynamParams, int measureParams, int controlParams=0, int type=CV_32F)
/* 파라미터 : dynamParams Dimensionality of the state.
measureParams Dimensionality of the measurement.
controlParams Dimensionality of the control vector.
type Type of the created matrices that should be CV_32F or CV_64F. */

const Mat & correct (const Mat &measurement)
/* Updates the predicted state from the measurement. 측정에서 예측상태를 업데이트
파라미터 : measurement The measured system parameters */

void init (int dynamParams, int measureParams, int controlParams=0, int type=CV_32F)
/* Re-initializes Kalman filter. The previous content is destroyed. 칼만필터 초기화. 이전 콘텐츠는 삭제됨
파라미터 : dynamParams Dimensionality of the state.
measureParams Dimensionality of the measurement.
controlParams Dimensionality of the control vector.
type Type of the created matrices that should be CV_32F or CV_64F. */

const Mat & predict (const Mat &control=Mat())
/* Computes a predicted state. 예측된 상태를 계산한다.
파라미터 : control The optional input control */

==========

Public Attributes

Mat controlMatrix
/* control matrix (B) (not used if there is no control)
매트릭스 컨트롤(B???) (컨트롤 없는 경우 사용되지 않음) */

Mat errorCovPost
/* posteriori error estimate covariance matrix
사후 오류 추정 공분산 행렬
(P(k)): P(k)=(I-K(k)*H)*P'(k) */

Mat errorCovPre
/* priori error estimate covariance matrix
선형적 오차 추정 공분산 행렬
(P'(k)): P'(k)=A*P(k-1)*At + Q)*/ (<-- " */ " 이 붙어있다.. 왜지?)
*/

Mat gain
/* Kalman gain matrix
칼만 게인 행렬
(K(k)): K(k)=P'(k)*Ht*inv(H*P'(k)*Ht+R) */

Mat measurementMatrix
/* measurement matrix (H)
측정 매트릭스 (H) */

Mat measurementNoiseCov
/* measurement noise covariance matrix (R)
측정 잡음 공분산 행렬 (R) */

Mat processNoiseCov
/* process noise covariance matrix (Q)
프로세스 잡음 공분산 행렬 (Q) */

Mat statePost
/* corrected state 수정된 상태

(x(k)): x(k)=x'(k)+K(k)*(z(k)-H*x'(k)) */

Mat statePre
/* predicted state 예측 상태
(x'(k)): x(k)=A*x(k-1)+B*u(k) */

Mat temp1

Mat temp2

Mat temp3

Mat temp4

Mat temp5

Mat transitionMatrix
/* state transition matrix (A) 상태 전이 행렬(A) */

==========

칼만필터는 재귀적으로 동작한다.
칼만필터는 바로 이전 시간에 추정한 값을 토대로 해서 현재의 값을 추정하며, 또한 바로 이전시간 외의 측정값이나 추정값은 사용되지 않는다.
각 추정 계산은 두 단계로 이루어지며, 먼저 이전 시간에 추정된 상태에 대해, 그 상태에서 사용자 입력을 가했을 때 예상되는 상태를 계산한다. 이 단계는 예측(prediction) 단계라고 부른다. 그 다음, 앞서 계산된 예측 상태와 실제로 측정된 상태를 토대로 정확한 상태를 계산한다. 이 단계는 보정(update) 단계라고 부른다.

제어 이론에서 칼만 필터는 가장 유용한 것으로 linear quadratic estimation (LQE)으로 알려져 있다.

칼만 필터의 다양성은 계속 지금까지 개발되고 있는데 칼만의 원래 공식은 이제 단순 칼만 필터라고 부르고, 쉬미츠의 확장 필터, 그외 Bierman, Thornton에 의한 information 필터와 square-root필터가 있다. 아마도 가장 흔하게 사용되는 칼만 필터

형태는 phase-locked loop으로서 현재 유비쿼터스의 라디오, 컴퓨터 그리고 대부분의 통신 및 비디오 장비에 사용된다.

https://ko.wikipedia.org/wiki/%EC%B9%BC%EB%A7%8C_%ED%95%84%ED%84%B0