

# ***Derivation of the Kalman-Bucy Filter Using Parameter Optimization***

**MICHAEL ATHANS**

MIT & ISR/IST

Last Revised: October 30, 2001

Ref. No. KF #8A

# Theme

- We will provide a **direct proof** of the continuous-time Kalman-Bucy filter equations **without making any gaussian assumptions**
- We shall **constrain** the structure of the state-estimator to be governed by linear, time-varying, vector differential equations with “undetermined” coefficients
- We shall evaluate the “undetermined” estimator coefficients by formulating and solving a **(matrix) optimization problem** requiring
  - **unbiased** state-estimates
  - **minimum mean-sum-square-error** (MSSE) of the state variable estimates
  - and with the help of **gradient matrices**
- Similar techniques can be used to prove the discrete-time Kalman filter, again without making any gaussian assumptions on the probability density functions

# Summary Of Kalman-Bucy Filter

- Plant - sensor dynamics:  $x(t_0), \xi(t), \theta(t)$  all independent,

$$(1) \quad \frac{dx(t)}{dt} = A(t)x(t) + B(t)u(t) + L(t)\xi(t); \quad x(t_0) \sim N(\bar{x}_0, \Sigma_0); \quad E\{\xi(t)\} = 0$$

$$(2) \quad z(t) = C(t)x(t) + \theta(t); \quad E\{\theta(t)\} = 0$$

$$E(\xi(t)\xi'(\tau)) = \Xi(t)\delta(t - \tau); \quad E(\theta(t)\theta'(\tau)) = \Theta(t)\delta(t - \tau)$$

- KBF state - estimator

$$(3) \quad \frac{d\hat{x}(t)}{dt} = A(t)\hat{x}(t) + B(t)u(t) + H(t)[z(t) - C(t)\hat{x}(t)]; \quad \hat{x}(t_0) = \bar{x}_0$$

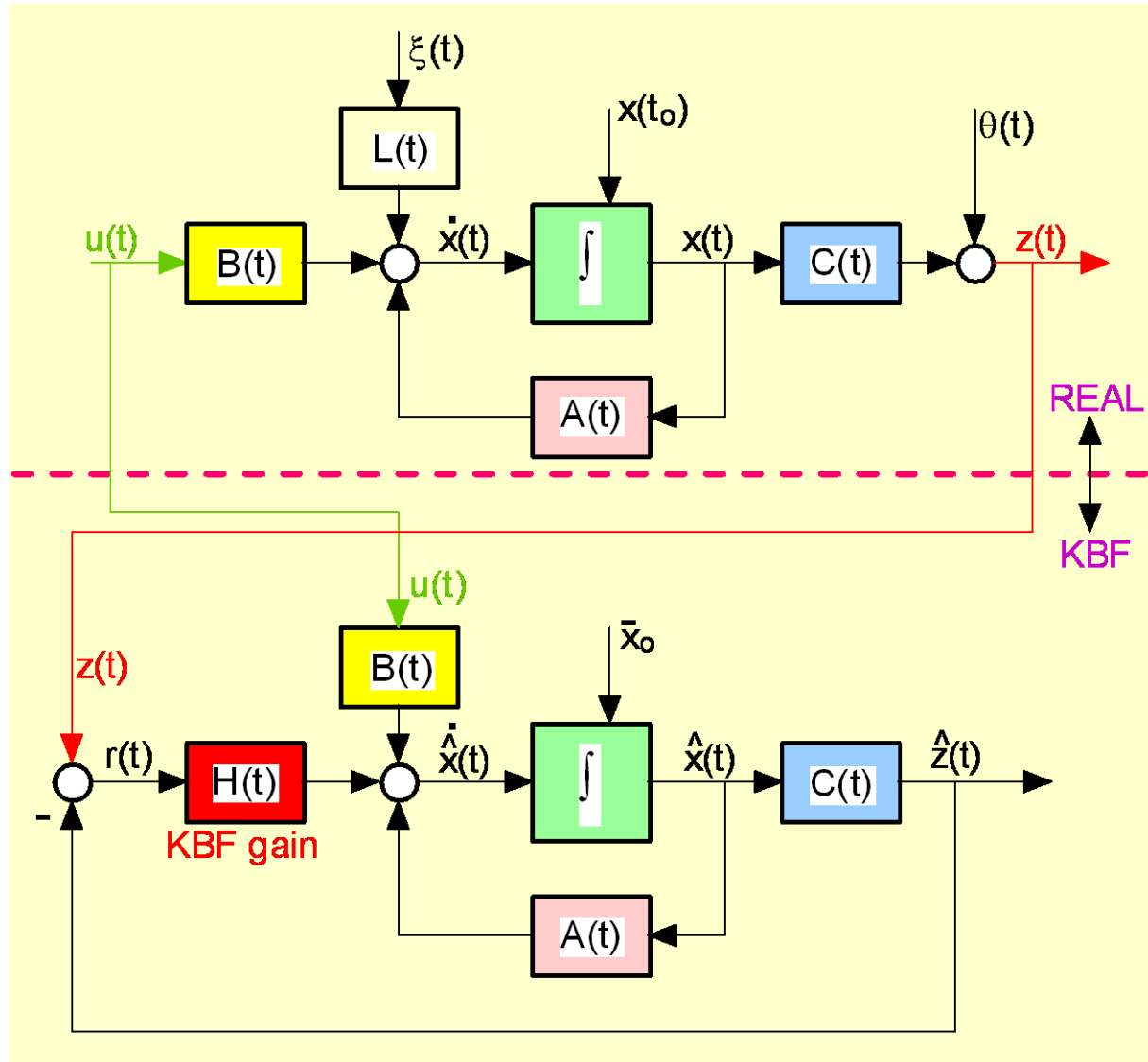
- KBF gain matrix  $H(t)$

$$(4) \quad H(t) = \Sigma(t)C'(t)\Theta^{-1}(t)$$

- KBF covariance matrix,  $\Sigma(t) = \Sigma'(t) \geq 0$ , satisfies the matrix Riccati differential equation

$$(5) \quad \frac{d\Sigma(t)}{dt} = A(t)\Sigma(t) + \Sigma(t)A'(t) + L(t)\Xi(t)L'(t) \\ - \Sigma(t)C'(t)\Theta^{-1}(t)C(t)\Sigma(t); \quad \Sigma(t_0) = \Sigma_0$$

# Visualization of the KBF



# *Alternate Derivation of the KBF: Key Idea*

- We do **not** make the gaussian assumption
- All we need are the means and covariances of the initial state,  $x(t_o)$ , the plant white noise,  $\xi(t)$ , and the sensor white noise,  $\theta(t)$
- Since the state is an  $n$ -dimensional vector, described by LTV differential equations, let us try to **generate also an  $n$ -dimensional state-estimate**, also governed by LTV differential equations **driven by the available signals**, the control vector  $u(t)$  and the sensor measurement vector  $z(t)$
- These estimator LTV differential equations contain **time-varying matrix and vector parameters** that must be determined
- The estimator parameters will be chosen so that the **state-estimation error vector is “small”** in some precise sense (that must be defined)

# General Assumed Filter Structure

- Plant and sensor equations:  $x(t) \in R^n$ ,  $z(t) \in R^m$

$$(6) \quad \frac{dx(t)}{dt} = A(t)x(t) + B(t)u(t) + L(t)\xi(t)$$

$$(7) \quad z(t) = C(t)x(t) + \theta(t)$$

- Postulated general state - estimator (filter) structure,

where  $\hat{x}(t) \in R^n$  is state - estimate of  $x(t)$ ,

$$(8) \quad \frac{d\hat{x}(t)}{dt} = F(t)\hat{x}(t) + G(t)u(t) + M(t)z(t) + v(t)$$

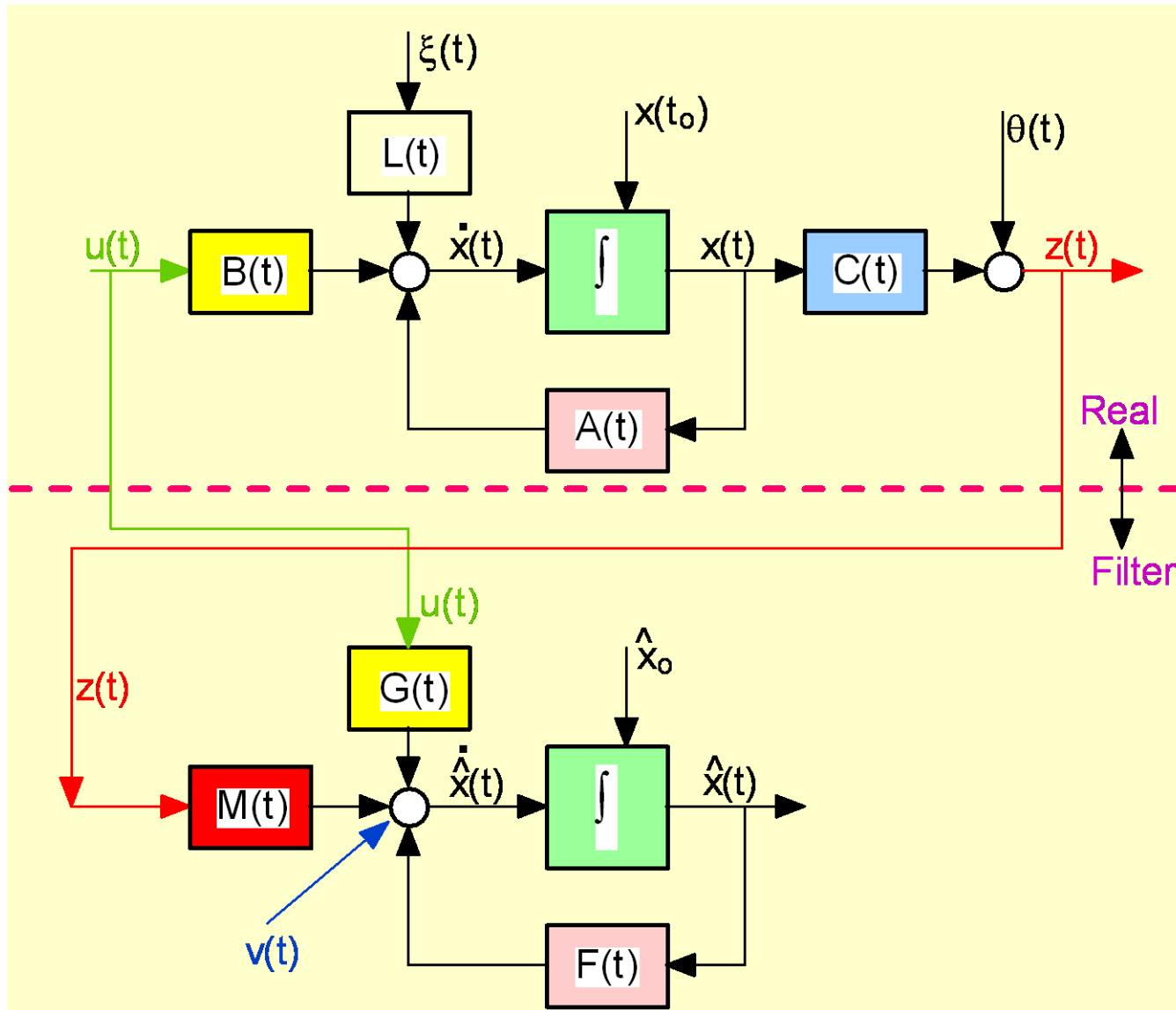
- Need to find "optimal" values for the matrices  $F(t)$ ,  $G(t)$ ,  $M(t)$ , and the vector  $v(t)$ , as well as the initial state - estimate  $\hat{x}(t_0)$
- Define "state - estimate error" vector,  $\tilde{x}(t)$

$$(9) \quad \tilde{x}(t) \equiv x(t) - \hat{x}(t) \Rightarrow \frac{d\tilde{x}(t)}{dt} = \frac{dx(t)}{dt} - \frac{d\hat{x}(t)}{dt}$$

- Estimation error dynamics, from eqs. (6) to (9), satisfy the stochastic differential equation

$$(10) \quad \frac{d\tilde{x}(t)}{dt} = F(t)\tilde{x}(t) + [A(t) - F(t) - M(t)C(t)]x(t) + [B(t) - G(t)]u(t) \\ - v(t) + L(t)\xi(t) - M(t)\theta(t); \quad \tilde{x}(t_0) = x(t_0) - \hat{x}(t_0)$$

# Visualization of General Filter



# Criteria for Optimality

- Estimation error dynamics, from eq. (10),

$$(11) \quad \frac{d\tilde{x}(t)}{dt} = F(t)\tilde{x}(t) + [A(t) - F(t) - M(t)C(t)]x(t) + [B(t) - G(t)]u(t) - v(t) + L(t)\xi(t) - M(t)\theta(t)$$

- Desired attributes of state-estimator: determine  $F(t)$ ,  $G(t)$ ,  $M(t)$ ,  $v(t)$ ,  $\tilde{x}(t_0)$  such that

(a): the estimation error has zero mean for all time

(b): the Mean-Sum-Squared-Error (MSSE) is minimum at each time

- Zero-mean estimation error implies

$$(12) \quad E\{\tilde{x}(t)\} = 0, \Rightarrow \frac{d}{dt}E\{\tilde{x}(t)\} = 0 \Rightarrow E\left\{\frac{d\tilde{x}(t)}{dt}\right\} = 0 \quad \forall t$$

- The MSSE is defined as the cost  $J$  (to be minimized)

$$(13) \quad J \equiv E\left\{\sum_{i=1}^n \tilde{x}_i^2(t)\right\} = E\{\tilde{x}'(t)\tilde{x}(t)\} = E\{tr[\tilde{x}(t)\tilde{x}'(t)]\} = \\ = tr[E\{\tilde{x}(t)\tilde{x}'(t)\}] = tr[\Sigma_e(t)]$$

where  $\Sigma_e(t) \equiv E\{\tilde{x}(t)\tilde{x}'(t)\}$  is the covariance matrix of the estimation error vector  $\tilde{x}(t)$ , for any given choice of the free parameters in (11)



# The Class of Unbiased Estimators

- Take the expected value of both sides of eq. (11), and since both noises  $\xi(t)$  and  $\theta(t)$  have zero mean, we find that

$$(13) \quad E\left\{\frac{d\tilde{x}(t)}{dt}\right\} = F(t)E\{\tilde{x}(t)\} + [A(t) - F(t) - M(t)C(t)]E\{x(t)\} + [B(t) - G(t)]u(t) - v(t)$$

- Use the "unbiasdness" requirement of eq. (12) in eq. (13):

$$(14) \quad 0 = [A(t) - F(t) - M(t)C(t)]E\{x(t)\} + [B(t) - G(t)]u(t) - v(t)$$

- In general,  $E\{x(t)\} \neq 0$ ,  $u(t) \neq 0$ , so to satisfy eq. (14) select

$$(15) \quad A(t) - F(t) - M(t)C(t) = 0 \Rightarrow F(t) = A(t) - M(t)C(t)$$

$$(16) \quad B(t) - G(t) = 0 \Rightarrow G(t) = B(t)$$

$$(17) \quad v(t) = 0$$

$$(18) \quad \tilde{x}(t_0) = 0 \Rightarrow \hat{x}(t_0) = \bar{x}_0$$

- From eqs. (15) to (18) we see that the state-estimator (8) reduces to

$$(19) \quad \frac{d\hat{x}(t)}{dt} = [A(t) - M(t)C(t)]\hat{x}(t) + B(t)u(t) + M(t)z(t); \hat{x}(t_0) = \bar{x}_0$$

# ***Unbiased Estimate and Error Dynamics***

- An alternate way to write eq. (19) is

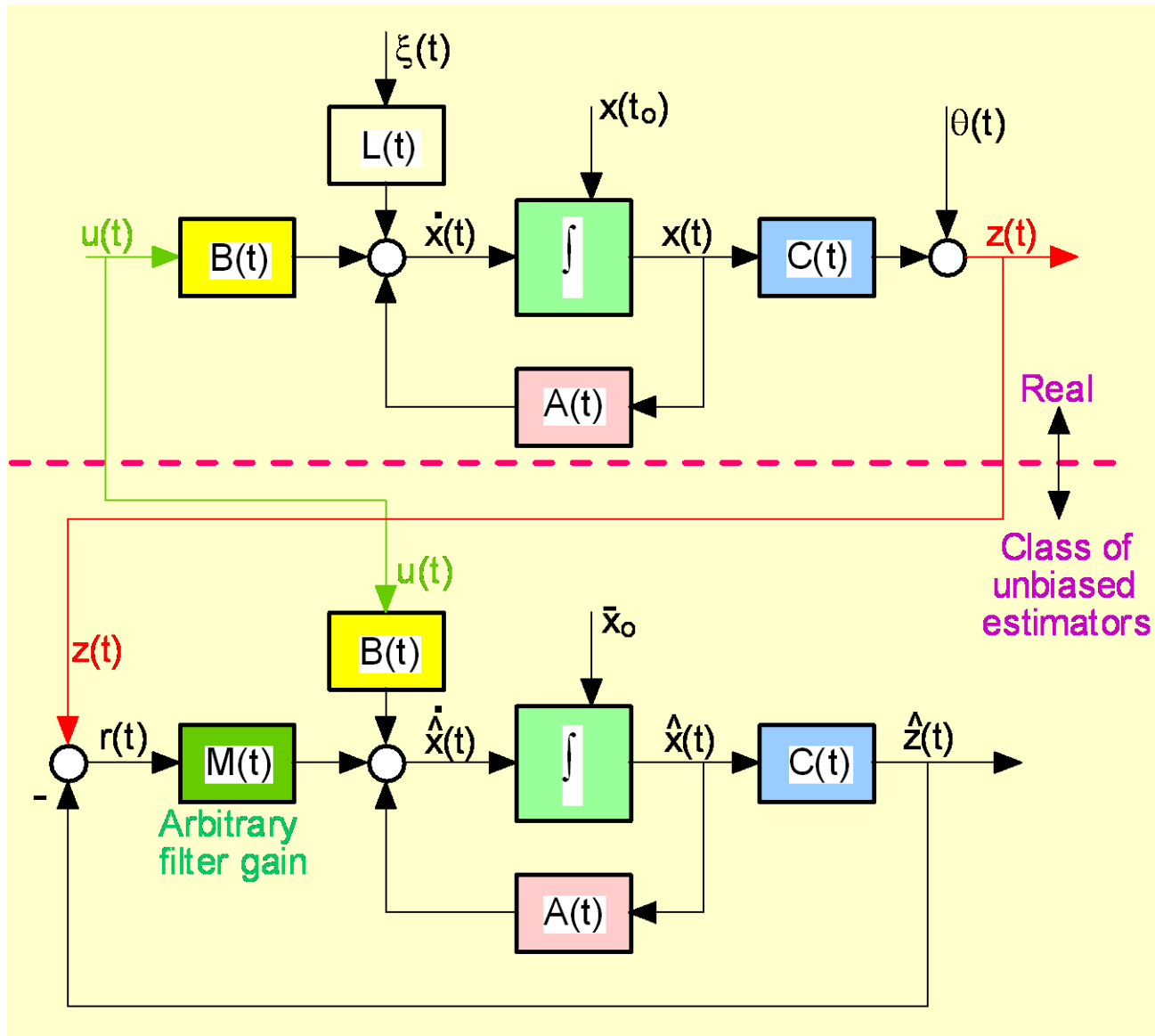
$$(20) \quad \frac{d\hat{x}(t)}{dt} = A(t)\hat{x}(t) + B(t)u(t) + M(t)[z(t) - C(t)\hat{x}(t)]; \quad \hat{x}(t_0) = \bar{x}_0$$

- Note that the structure of the "unbiased estimator" (20) is the same as the structure of the KBF, except that an arbitrary gain matrix  $M(t)$  multiplies the "residual" vector  $r(t) \equiv [z(t) - C(t)\hat{x}(t)]$
- The corresponding (unbiased) state - estimation error now satisfies (substitute eqs. (15) to (18) into eq. (11))

$$(21) \quad \frac{d\tilde{x}(t)}{dt} = [A(t) - M(t)C(t)]\tilde{x}(t) + L(t)\xi(t) - M(t)\theta(t); \quad \tilde{x}(t_0) = 0$$

- Clearly, the error covariance matrix  $\Sigma_e(t) \equiv E\{\tilde{x}(t)\tilde{x}'(t)\}$  will depend on the choice of the gain matrix  $M(t)$

# Structure of Unbiased Estimator



# Discussion

- Notice that both the Kalman-Bucy filter (KBF) and the class of unbiased estimators involve
  - building an **exact model of the plant and sensors**
  - **replacing** all random quantities with their **mean**
  - **updating** the derivative of the state-estimate vector by first forming the **residual vector** and then **multiplying it with a gain matrix**,  $M(t)$  or  $H(t)$ , resulting in a multivariable feedback system
- The only difference up to now is that we have a “**precise recipe**” for calculating the KBF gain matrix,  $H(t)$ , while in the unbiased estimator class this gain matrix,  $M(t)$ , is still **arbitrary**
- Next, we form and solve an optimization problem that determines the “optimal” value of  $M(t)$ , in the sense that it minimizes the Mean-Sum-Square-Error (MSSE)
  - the “**optimal**” **value of  $M(t)$**  turns out to be **identical to the KBF gain  $H(t)$** !

# ***The Error Covariance Depends on $M(t)$***

- The estimation error stochastic dynamics are repeated below:

$$(21) \quad \frac{d\tilde{x}(t)}{dt} = [A(t) - M(t)C(t)]\tilde{x}(t) + L(t)\xi(t) - M(t)\theta(t); \quad \tilde{x}(t_0) = 0$$

- Since eq. (21) is a linear system, driven by two independent white noises  $\xi(t)$  and  $\theta(t)$ , and since  $E\{\tilde{x}(t)\} = 0$ , we readily deduce the matrix Lyapunov differential equation of  $\Sigma_e(t) = E\{\tilde{x}(t)\tilde{x}'(t)\}$

$$(22) \quad \frac{d\Sigma_e(t)}{dt} = [A(t) - M(t)C(t)]\Sigma_e(t) + \Sigma_e(t)[A(t) - M(t)C(t)]' + \\ + L(t)\Xi(t)L'(t) + M(t)\Theta(t)M'(t); \quad \Sigma_e(t_0) = \Sigma_0$$

- To minimize the MSSE, we must make  $tr[\Sigma_e(t)]$  as small as possible, for each  $t$ , in view of eq. (13). From eq. (22) we have

$$(23) \quad tr\left[\frac{d\Sigma_e(t)}{dt}\right] = tr[[A(t) - M(t)C(t)]\Sigma_e(t)] + tr[\Sigma_e(t)[A(t) - M(t)C(t)]'] + \\ + tr[L(t)\Xi(t)L'(t)] + tr[M(t)\Theta(t)M'(t)]$$

# Minimizing the MSSE

- The derivative and the trace operators are linear and they commute

$$(24) \quad \text{tr} \left[ \frac{d}{dt} \Sigma_e(t) \right] = \frac{d}{dt} \text{tr} [\Sigma_e(t)]$$

so, from eq. (23), we obtain

$$(25) \quad \begin{aligned} \frac{d}{dt} \text{tr} [\Sigma_e(t)] = & \text{tr} [A(t) \Sigma_e(t)] - \text{tr} [M(t) C(t) \Sigma_e(t)] + \text{tr} [\Sigma_e(t) A'(t)] \\ & - \text{tr} [\Sigma_e(t) C'(t) M'(t)] + \text{tr} [L(t) \Xi(t) L'(t)] + \text{tr} [M(t) \Theta(t) M'(t)] \end{aligned}$$

- Since the covariance  $\Sigma_e(t)$  is positive semidefinite, all its diagonal elements are nonnegative, and so  $\text{tr} [\Sigma_e(t)] \geq 0$ . To make  $\text{tr} [\Sigma_e(t)] = \text{MSSE}$  as small as possible, we should select  $M(t)$  to minimize  $\frac{d}{dt} \text{tr} [\Sigma_e(t)]$  i.e. make it as negative as possible, or, equivalently, minimize the RHS of eq. (25)

# The Optimization Problem

- The formal optimization problem is:

$$(26) \quad \min_{M(t)} \frac{d}{dt} \text{tr}[\Sigma_e(t)] \text{ or, equivalently,}$$

$$(27) \quad \min_{M(t)} \left\{ \text{tr}[A(t)\Sigma_e(t)] - \text{tr}[M(t)C(t)\Sigma_e(t)] + \text{tr}[\Sigma_e(t)A'(t)] \right. \\ \left. - \text{tr}[\Sigma_e(t)C'(t)M'(t)] + \text{tr}[L(t)\Xi(t)L'(t)] + \text{tr}[M(t)\Theta(t)M'(t)] \right\}$$

- Since the trace is a scalar-valued function of a matrix, and since we seek to minimize with respect to a matrix, it would be nice if we could solve the minimization problem (27) by setting

$$\frac{\partial}{\partial M(t)} \{.\} = 0, \text{ and then solve the resultant equation.}$$

- TECHNICAL PROBLEM: How do we compute the derivative of a scalar-valued function with respect to a matrix? **BY GRADIENT MATRICES!**

# ***Digression: Gradient Vectors***

- Recall that if we are given a scalar-valued function of  $n$  variables, then its gradient vector is the  $n$ -dimensional column vector of its partial derivatives with respect to each variable

- Review: Scalar - valued function of a vector  $f(x)$ ,  $x \in R^n$

$$f(x) = f(x_1, x_2, \dots, x_n)$$

- Gradient vector:  $\frac{\partial f(x)}{\partial x} \in R^n$

$$\frac{\partial f(x)}{\partial x} \equiv \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} \end{bmatrix}$$



# Digression: Gradient Matrices

- Given an  $n \times m$  matrix, we can define a **scalar-valued** function of the  $n \cdot m$  matrix elements. Then the **gradient matrix** is an  $n \times m$  matrix whose elements are the partial derivatives with respect to the associated matrix element. See [3], also [1], p. 22, and [2].

- Given an  $n \times m$  matrix  $X$  with elements  $x_{ij}$ ,  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, m$
- Let  $f(X) = f(x_{11}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{n1}, \dots, x_{nm})$  be a scalar-valued function of the elements  $x_{ij}$  of  $X$
- The gradient matrix of  $f(\cdot)$  with respect to the matrix  $X$  is the  $n \times m$  matrix, denoted by  $\frac{\partial f(X)}{\partial X}$ , and defined by

$$\frac{\partial f(X)}{\partial X} \equiv \begin{bmatrix} \frac{\partial f(\cdot)}{\partial x_{11}} & \frac{\partial f(\cdot)}{\partial x_{12}} & \cdots & \frac{\partial f(\cdot)}{\partial x_{1m}} \\ \frac{\partial f(\cdot)}{\partial x_{21}} & \frac{\partial f(\cdot)}{\partial x_{22}} & \cdots & \frac{\partial f(\cdot)}{\partial x_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(\cdot)}{\partial x_{n1}} & \frac{\partial f(\cdot)}{\partial x_{n2}} & \cdots & \frac{\partial f(\cdot)}{\partial x_{nm}} \end{bmatrix}$$

# Useful Gradient Matrices

- We summarize certain useful gradient matrices that we shall need in the derivation of the Kalman-Bucy filter
- See Ref. [3] for detailed derivations
- In all cases,  $tr[A]$ , denotes the sum of the diagonal elements of a **square** matrix  $A$
- Recall,  $tr[A+B]=tr[A]+tr[B]$  and  $tr[A]=tr[A']$
- In trace of matrix products, say  $tr[ABC]$ , only  $ABC$  must be square, not  $A$ ,  $B$ , or  $C$

$$\frac{\partial}{\partial A} tr[A] = I; \quad A: n \times n$$

$$\frac{\partial}{\partial A} tr[BA] = \frac{\partial}{\partial A} tr[A'B'] = B'$$

$$\frac{\partial}{\partial A} tr[B'A] = \frac{\partial}{\partial A} tr[A'B] = B$$

$$\frac{\partial}{\partial A} tr[AC] = \frac{\partial}{\partial A} tr[C'A'] = C'$$

$$\frac{\partial}{\partial A} tr[AC'] = C$$

$$\frac{\partial}{\partial A} tr[BAC] = B'C'$$

$$\frac{\partial}{\partial A} tr[ABA'] = 2AB$$

# The Optimal Filter Gain, $M^*(t)$

- Recall our optimization problem

$$(27) \quad \min_{M(t)} \left\{ \text{tr}[A(t)\Sigma_e(t)] - \text{tr}[M(t)C(t)\Sigma_e(t)] + \text{tr}[\Sigma_e(t)A'(t)] \right. \\ \left. - \text{tr}[\Sigma_e(t)C'(t)M'(t)] + \text{tr}[L(t)\Xi(t)L'(t)] + \text{tr}[M(t)\Theta(t)M'(t)] \right\}$$

- Now we compute the gradient matrix of  $\{ \}$  with respect to  $M(t)$  and set it equal to zero. Use the notation  $M^*(t)$  to denote the "optimal"

$$(28) \quad 0 = -\frac{\partial}{\partial M(t)} \text{tr}[M(t)C(t)\Sigma_e(t)] - \frac{\partial}{\partial M(t)} \text{tr}[\Sigma_e(t)C'(t)M'(t)] + \\ + \frac{\partial}{\partial M(t)} \text{tr}[M(t)\Theta(t)M'(t)] \Rightarrow$$

$$(29) \Rightarrow 0 = -\Sigma_e(t)C'(t) - \Sigma_e(t)C'(t) + 2M^*(t)\Theta(t)$$

- Therefore, the optimal gain matrix  $M^*(t)$  is given by

$$(30) \quad M^*(t) = \Sigma_e(t)C'(t)\Theta^{-1}(t)$$

# The Optimal Covariance, $\Sigma_e^*(t)$

- Recall, eq. (22), the matrix Lyapunov differential equation that describes the error covariance,  $\Sigma_e(t) = E\{\tilde{x}(t)\tilde{x}'(t)\}$  for any  $M(t)$

$$(31) \quad \frac{d\Sigma_e(t)}{dt} = [A(t) - M(t)C(t)]\Sigma_e(t) + \Sigma_e(t)[A(t) - M(t)C(t)]' + L(t)\Xi(t)L'(t) + M(t)\Theta(t)M'(t); \quad \Sigma_e(t_0) = \Sigma_0$$

- Substitute the optimal gain  $M^*(t)$  from eq. (30) into eq. (31), and use  $\Sigma_e^*(t)$  to denote the resulting "optimal" error covariance

$$(32) \quad \frac{d\Sigma_e^*(t)}{dt} = A(t)\Sigma_e^*(t) + \Sigma_e^*(t)A'(t) + L(t)\Xi(t)L'(t) - \Sigma_e^*(t)C'(t)\Theta^{-1}(t)C(t)\Sigma_e^*(t) - \Sigma_e^*(t)C'(t)\Theta^{-1}(t)C(t)\Sigma_e^*(t) + \Sigma_e^*(t)C'(t)\Theta^{-1}(t) \cdot \Theta(t) \cdot \Theta^{-1}(t)C(t)\Sigma_e^*(t) \Rightarrow$$

$$(33) \quad \frac{d\Sigma_e^*(t)}{dt} = A(t)\Sigma_e^*(t) + \Sigma_e^*(t)A'(t) + L(t)\Xi(t)L'(t) - \Sigma_e^*(t)C'(t)\Theta^{-1}(t)C(t)\Sigma_e^*(t)$$

- Clearly the covariance equation (33) and the filter gain equation (30) are identical to those of the KBF (see eqs. (4) and (5)), with

$$(34) \quad H(t) = M^*(t), \quad \Sigma(t) = \Sigma_e^*(t) \quad \text{QED}$$

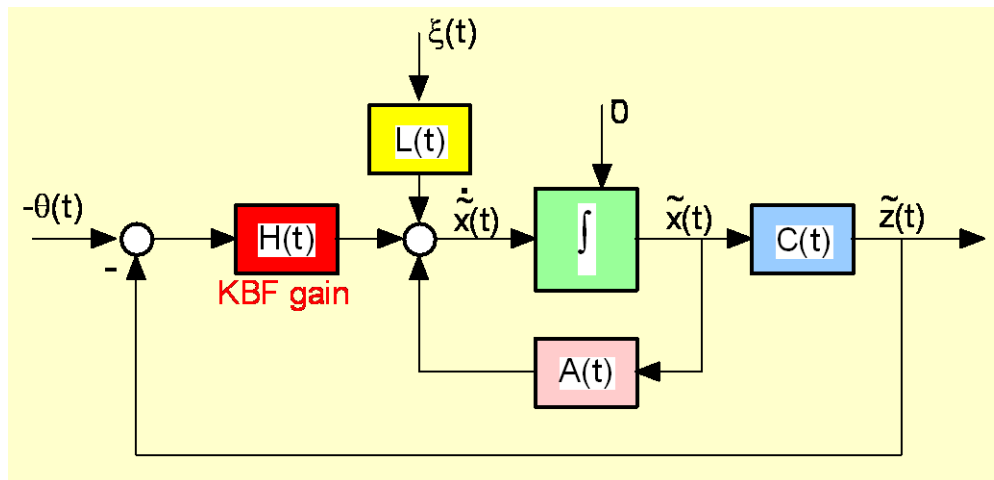
# The Optimal Estimation Error Dynamics

- From eq. (21) the "optimal" state - estimate error dynamics, with  $M(t) = H(t)$ , the optimum KBF gain matrix, are

$$(35) \quad \frac{d\tilde{x}(t)}{dt} = [A(t) - H(t)C(t)]\tilde{x}(t) + L(t)\xi(t) - H(t)\theta(t); \quad \tilde{x}(t_0) = 0, \text{ or}$$

$$(36) \quad \frac{d\tilde{x}(t)}{dt} = A(t)\tilde{x}(t) + L(t)\xi(t) - H(t)[C(t)\tilde{x}(t) + \theta(t)]; \quad \tilde{x}(t_0) = 0$$

- Hence, the error dynamics are a replica of the KBF loop driven by the two white noises  $\xi(t)$  and  $\theta(t)$



# Discussion

- The method of proof presented **does not require any gaussian assumptions**
  - It demonstrates that **the KBF is the optimal linear estimator in a “least-squares” sense**, i.e. it minimizes the MSSE cost
  - Of course, in the nongaussian case, we are not generating the true conditional pdf of the state, nor its true conditional mean and covariance (this requires nonlinear estimators)
- Actually, one can prove that **the linear KBF is optimal if we minimize a cost different from the MSSE**
- Similar techniques can be used to prove the **optimality of the discrete-time Kalman filter**

# Other Criteria

- In the previous derivation, we have minimized the MSEE cost

$$J = E\{\tilde{x}'(t)\tilde{x}(t)\} = \text{tr}[\Sigma_e(t)]$$

- The KBF remains optimal, [2], if we minimize the so - called " weighted least - squares" cost

$$J_w = E\{\tilde{x}'(t)Q(t)\tilde{x}(t)\} = \text{tr}[Q(t)\Sigma_e(t)]; \quad Q(t) = Q'(t) > 0$$

- The KBF also remains optimal, [2], if we minimize the determinant of the error covariance matrix, i.e.

$$J_d = \det[\Sigma_e(t)]$$

- The fact that linear Kalman filters are optimal with respect to several criteria is the reason for their popularity and for the fact that there are several ways of proving the same result

# *Concluding Remarks*

- Provided a **direct proof** of the continuous-time Kalman-Bucy filter **without making any gaussian assumptions**
- We **constrained** the structure of the state-estimator to be governed by linear, time-varying, vector differential equations with “undetermined” coefficients (matrices and vectors)
- We evaluated the “undetermined” coefficients by formulating and **solving a (matrix) optimization problem** requiring
  - **unbiased** state-estimates
  - **minimum mean-sum-square-error** (MSSE) of the state variable estimates
  - and with the help of **gradient matrices**
- BOTTOM LINE: The Kalman-Bucy filter is the **optimal linear state-estimator**



# References

- [1]. A. Gelb, *Applied Optimal Estimation*, MIT Press, 1974
- [2]. E. Tse and M. Athans, "A Direct Derivation of the Optimal Linear Filter Using the Maximum Principle," *IEEE Trans. Auto. Control*, Vol. AC-12, Dec. 1967, pp. 690-698
- [3]. M. Athans and F.C. Schweppe, *Gradient Matrices and Matrix Calculations*, Technical Note 1965-63, MIT Lincoln Lab., Lexington, Mass., November 1965