Курсовая работа на тему:

Синтез и анализ алгоритмов нелинейной фильтрации координат цели и управления на основе расширенного дискретного фильтра Калмана

В мире существуют вещи важнее самых прекрасных открытий – это знание методов, которыми они были сделаны.

Готфрид Лейбниц 17 в.

1. Аннотация

Рассматривается конкретная задача синтеза системы наведения ЛА на цель. В качестве алгоритмического обеспечения в работе используются уравнения расширенного (нелинейного) дискретного фильтра Калмана, так как модели состояния и наблюдения являются нелинейными.

2. Математическая постановка задачи. Модель состояния системы

В основе решения задачи синтеза системы наведения лежат кинематические соотношения, связывающие координаты цели с параметрами движения ЛА (объекта) и цели, которые могут быть описаны системой стохастических дифференциальных уравнений.

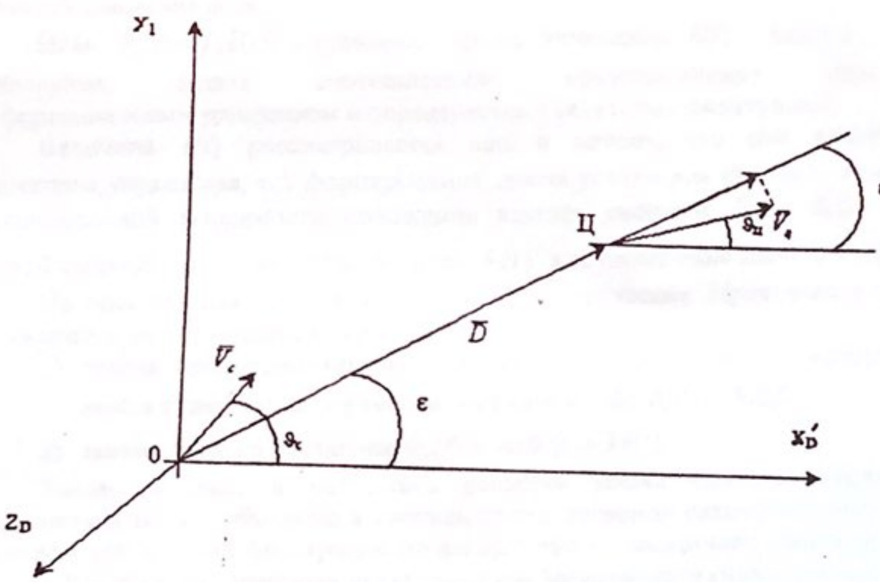


Рис.1. Кинематическая схема системы наведения ЛА на цель в продольной плоскости

$\overline{D}$ - вектор дальности;
ε - угол визирования цели относительно ЛА;
$\overline{V}_c$ - вектор скорости ЛА;
$\overline{V}_ц$ - вектор скорости цели;
ϑ_c - угловое положение вектора скорости самолёта;
$\vartheta_ц$ - угловое положение вектора скорости цели.

Уравнение состояния представляет собой проекции скоростей на линию, перпендикулярную линии дальности и на линию дальности:

$$\dot{\varepsilon}(t)D(t) = V_c \sin(\varepsilon(t) - \vartheta_c(t)) - V_ц \sin(\varepsilon(t) - \vartheta_ц(t)).$$

Разделив на дальность, введём угловую скорость:

$$\dot{\varepsilon}(t) = \frac{1}{D(t)}[V_c \sin(\varepsilon(t) - \vartheta_c(t)) - V_ц \sin(\varepsilon(t) - \vartheta_ц(t))] + W_\varepsilon(t). \quad (1)$$

$$\dot{D}(t) = -V_c \cos(\varepsilon(t) - \vartheta_c(t)) - V_ц \cos(\varepsilon(t) - \vartheta_ц(t)) + W_D(t). \quad (2)$$

$$\dot{\beta}(t) = 0. \quad (3)$$

Данная система уравнений является незамкнутой, так как отсутствует управление:

$$\dot{\vartheta}_c(t) = f(\hat{\varepsilon}(t), \hat{D}(t), \hat{V}_ц(t), \hat{\vartheta}_ц(t)).$$

Будем считать, что $\overline{D}(t)$ измеряется с помощью бортового радиолокатора, и оценка дальности $\hat{D}(t)$ получается в результате фильтрации. Скорость цели будем считать известной по данным доплеровского измерителя скоростей. $\hat{\vartheta}_c(t)$ может определяться по данным датчика скоростей. $\hat{\vartheta}_ц(t)$ будем считать известной по априорным сведениям в соответствии с предполагаемой гипотезой движения цели.

Итак, $\dot{\vartheta}_c(t) = f(\hat{\varepsilon}(t))$ - уравнение закона управления. $\hat{\varepsilon}(t)$ – оценка угла визирования, задана соотношением, представленным первым дифференциальным уравнением и определяется в результате фильтрации.

Величина $\varepsilon(t)$ рассматривается ещё и потому, что она является параметром управления, т.е. формирование закона управления строится в виде функциональной зависимости положения вектора скорости ЛА $\vartheta_c(t)$ от угловой скорости линии визирования цели $\dot{\varepsilon}(t)$ в её оценочном значении $\dot{\hat{\varepsilon}}(t)$.

На практике известно несколько законов управления. Практически все они являются модификациями двух основных законов:

1) закона пропорционального наведения, в соответствии с которым линия визирования перемещается || самой себе $\dot{\vartheta}_c(t) = N\dot{\hat{\varepsilon}}(t)$;
2) закона прямого наведения $\dot{\vartheta}_c(t) = -\alpha\vartheta_c(t) + k\hat{\varepsilon}(t)$.

Таким образом, в результате решения задачи синтеза углового координатора цели необходимо, в соответствии с теоремой разделения, решить на первом этапе задачу фильтрации по алгоритмам расширенного дискретного фильтра Калмана, рассматривая управление как детерминированный сигнал; на втором этапе решить задачу субоптимального управления, приняв оценки вектора состояния системы за точные его значения.

3. Модель наблюдения

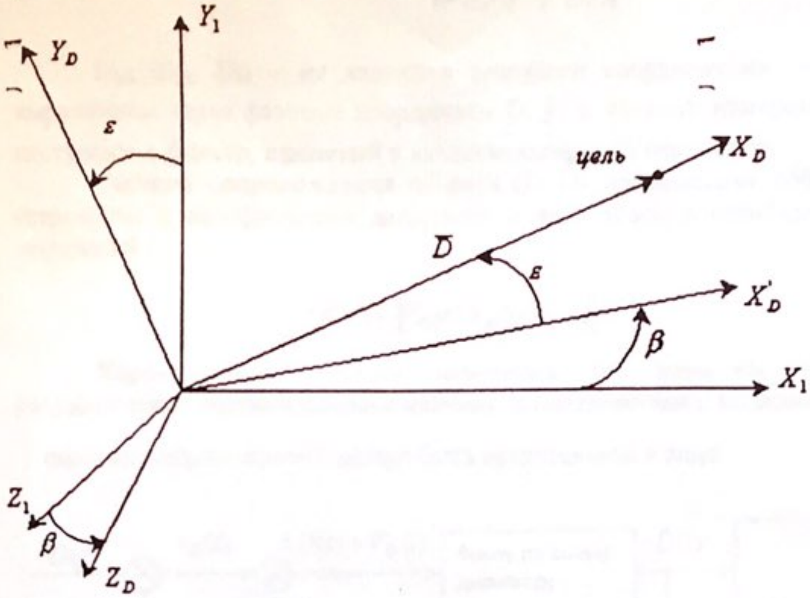
Для воздушного объекта в качестве системы сопровождения используется визирное устройство, представляющее собой систему измерителей дальности и двух углов $\overline{D}(D, \beta, \varepsilon)$.

Уравнения наблюдения имеют вид:

$$r_D(t) = D(t) + V_D(t);$$
$$r_\varepsilon(t) = \varepsilon(t) + V_\varepsilon(t);$$
$$r_\beta(t) = \beta(t) + V_\beta(t).$$

$V_D(t), V_\beta(t), V_\varepsilon(t)$ - ошибки измерения дальности.

Инерциальная система координат "g" $\equiv OX_gY_gZ_g \xrightarrow{A_g^1(\psi,\vartheta,\gamma)}$ с.к., связанная с самолётом "1" $\equiv OX_1Y_1Z_1 \xrightarrow{A_1^D(\beta,\varepsilon)}$ с.к., связанная с визирным устройством "D" $\equiv OX_DY_DZ_D$.



Система сопровождения объекта (цели) находится на ЛА и в данной задаче интерес представляет вектор дальности, определяющий положение объекта относительно ЛА. Тогда вектор дальности в с.к. "g":

$$\left\|\overline{D}\right\|_g = A_g^1(\psi,\vartheta,\gamma)A_1^D(\beta,\varepsilon)\left\|\overline{D}\right\|_D$$

Будем считать, что ψ, θ, γ измеряются с помощью гироплатформы инерциальной навигационной системы или вычисляются (в отсутствии гироплатформы) и являются точно известными.

Положение $\overline{D}$ в самолётной с.к. будет определяться:

$$\left\|\overline{D}\right\|_1 = A_1^D(\beta,\varepsilon)\left\|\overline{D}\right\|_D = A_1^D(\beta,\varepsilon)\begin{Vmatrix} D \\ 0 \\ 0 \end{Vmatrix};$$

$$A_1^D(\beta,\varepsilon) = \begin{Vmatrix} \cos\beta\cos\varepsilon & -\cos\beta\sin\varepsilon & \sin\beta \\ \sin\varepsilon & \cos\varepsilon & 0 \\ -\sin\beta\cos\varepsilon & \sin\beta\sin\varepsilon & \cos\beta \end{Vmatrix}.$$

Модель измерителя в с.к. "1":

$$\left\|\overline{r}(t)\right\|_1 = \left\|\overline{D}(\varepsilon)\right\|_1 + \left\|\overline{V}(t)\right\|_1.$$

Таким образом, измеряется как бы не дальность с углами β и ε, а проекции вектора дальности на оси связанной с ЛА с.к. в присутствии шумов.

Модель измерителя $\|\,\|$:

$$\|r(t)\|_1 = \begin{Vmatrix} D_{x1}(t) \\ D_{y1}(t) \\ D_{z1}(t) \end{Vmatrix} + \begin{Vmatrix} V_{x1}(t) \\ V_{y1}(t) \\ V_{z1}(t) \end{Vmatrix}$$

D_{x1}, D_{y1}, D_{z1} – не являются фазовыми координатами, но нелинейно выражаются через фазовые координаты D, β, ε. Именно измерения модели II поступают в фильтр, принятый в качестве алгоритма обработки.

Система сопровождения объекта (ССО) представляет собой визирное устройство с измерителями дальности и двух углов с ошибками (шумами) измерений

$$V(t) = \|V_D(t)\ V_\beta(t)\ V_\varepsilon(t)\|^T \qquad (*)$$

Характеристики шумов измерения (*) известны (по данным разработчиков системы сопровождения). В соответствии с моделью измерителя I система сопровождения может быть представлена в виде:

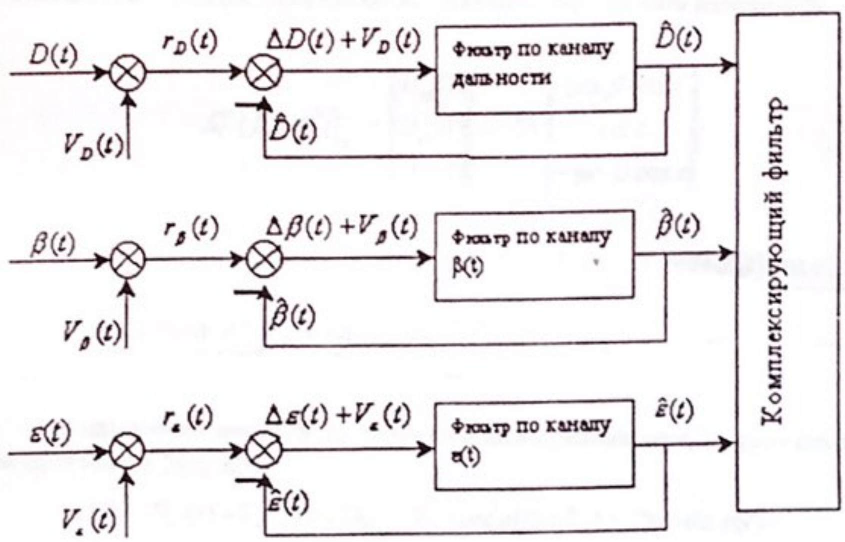


Рис.2. Схема дальномера (следящая система с отрицательной обратной связью)

В расширенный дискретный фильтр Калмана поступают сигналы измерителей в соответствии с моделью II - проекции вектора дальности $\overline{D}(t)$ на оси ЛА. Модель II - нелинейная, поэтому для перехода от модели измерений I к модели II проведём линеаризацию нелинейных функций относительно ошибок измерения.

$$\|\overline{D}(t)\|_1 = A_1^D(\beta(t), \varepsilon(t)) \begin{Vmatrix} D \\ 0 \\ 0 \end{Vmatrix};$$

$$\left\|\overline{D}(t)+\overline{V}(t)\right\|_1 = A_1^D(\beta(t)+V_\beta,\varepsilon(t)+V_\varepsilon)\begin{Vmatrix} D(t)+V_D \\ 0 \\ 0 \end{Vmatrix} =$$
$$F_{S\to V}(D(t)+V_D,\beta(t)+V_\beta,\varepsilon(t)+V_\varepsilon).$$

Разложим нелинейные функции в ряд Тейлора:

$$f(x+\Delta x, y+\Delta y, z+\Delta z) = f(x,y,z)+\frac{\partial f}{\partial x}\Delta x+\frac{\partial f}{\partial y}\Delta y+\frac{\partial f}{\partial z}\Delta z.$$

Для канала по D(t) линеаризованная модель измерителя имеет вид:

$$r_1(t) = D_{x1}(t)+V_{x1}(t) = f_1(D(t)+V_D,\beta(t)+V_\beta,\varepsilon(t)+V_\varepsilon) = f_1(D,\beta,\varepsilon)+$$
$$+\frac{\partial f_1}{\partial D}V_D+\frac{\partial f_1}{\partial \beta}V_\beta+\frac{\partial f_1}{\partial \varepsilon}V_\varepsilon;$$

(V_D, V_β, V_ε задаются разработчиком).

Так как координаты $D_{x1}(t)$, $D_{y1}(t)$, $D_{z1}(t)$ не являются фазовыми, но нелинейно выражаются через фазовые координаты (D, β, ε), получим эти зависимости с учётом линеаризации относительно шумов измерений:

$$A_1^D(\beta,\varepsilon)\left\|\overline{D}\right\|_D = \begin{Vmatrix} D_{x1}(t) \\ D_{y1}(t) \\ D_{z1}(t) \end{Vmatrix} = D\begin{Vmatrix} \cos\beta\cos\varepsilon \\ \sin\varepsilon \\ -\sin\beta\cos\varepsilon \end{Vmatrix};$$

$$r_1(t) = (D(t)+V_D)\cos(\beta(t)+V_\beta)\cos(\varepsilon(t)+V_\varepsilon) = \underbrace{D(t)\cos\beta(t)\cos\varepsilon(t)}_{D_{x1}(t)}+$$
$$+\underbrace{\cos\beta(t)\cos\varepsilon(t)}_{a_{11}}V_D+\underbrace{D(t)(-\sin\beta(t))\cos\varepsilon(t)}_{a_{12}}V_\beta+\underbrace{D(t)\cos\beta(t)(-\sin\varepsilon(t))}_{a_{13}}V_\varepsilon.$$

Аналогично могут быть получены линеаризованные модели измерителей для каналов по β(t), ε(t):

$$r_2(t) = D_{y1}(t)+V_{y1}(t) = (D(t)+V_D)\sin(\varepsilon(t)+V_\varepsilon) = D(t)\sin\varepsilon(t)+$$
$$+\sin\varepsilon(t)V_D+0+D(t)\cos\varepsilon(t)V_\varepsilon;$$
$$r_3(t) = D_{z1}(t)+V_{z1}(t) = (D(t)+V_D)(-\sin(\beta(t)+V_\beta))\cos(\varepsilon(t)+V_\varepsilon) =$$
$$= -D(t)\sin\beta(t)\cos\varepsilon(t)-\sin\beta(t)\cos\varepsilon(t)V_D-D(t)\cos\beta(t)\cos\varepsilon(t)V_\beta+$$
$$+D(t)\sin\beta(t)\sin\varepsilon(t)V_\varepsilon.$$

Вектор шумов измерений для линеаризованной модели $\|\overline{\ }\|$ имеет вид:

$$\begin{Vmatrix} V_{x1}(t) \\ V_{y1}(t) \\ V_{z1}(t) \end{Vmatrix} = \begin{Vmatrix} a_{11}V_D & a_{12}V_\beta & a_{13}V_\varepsilon \\ a_{21}V_D & a_{22}V_\beta & a_{23}V_\varepsilon \\ a_{31}V_D & a_{32}V_\beta & a_{33}V_\varepsilon \end{Vmatrix} = \begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix}\begin{Vmatrix} V_D \\ V_\beta \\ V_\varepsilon \end{Vmatrix},$$

где $\begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix}$ - матрица коэффициентов при векторе шумов измерений;

$\begin{Vmatrix} V_D \\ V_\beta \\ V_\varepsilon \end{Vmatrix}$ - случайный вектор шумов (ошибок) измерений системы сопровождения объекта.

Матрица интенсивностей непрерывных белых шумов (ошибок) измерений для приведенной модели будет иметь вид:

$$R = \begin{Vmatrix} \sigma_{Vx1}^2(t) & R_{Vx1Vy1}(t) & R_{Vx1Vz1}(t) \\ R_{Vy1Vx1}(t) & \sigma_{Vy1}^2(t) & R_{Vy1Vz1}(t) \\ R_{Vz1Vx1}(t) & R_{Vz1Vy1}(t) & \sigma_{Vz1}^2(t) \end{Vmatrix}.$$

Вне диагонали стоят взаимные интенсивности шумов приведенной модели измерений.

$$\sigma_{Vx1}^2(t) = a_{11}^2\sigma_{V_D}^2(t) + a_{12}^2\sigma_{V\beta}^2(t) + a_{13}^2\sigma_{V\varepsilon}^2(t);$$

$$R_{Vx1Vy1}(t) = R_{Vy1Vx1}(t) = a_{11}a_{21}\sigma_{V_D}^2(t) + a_{12}a_{22}\sigma_{V\beta}^2(t) + a_{13}a_{23}\sigma_{V\varepsilon}^2(t);$$

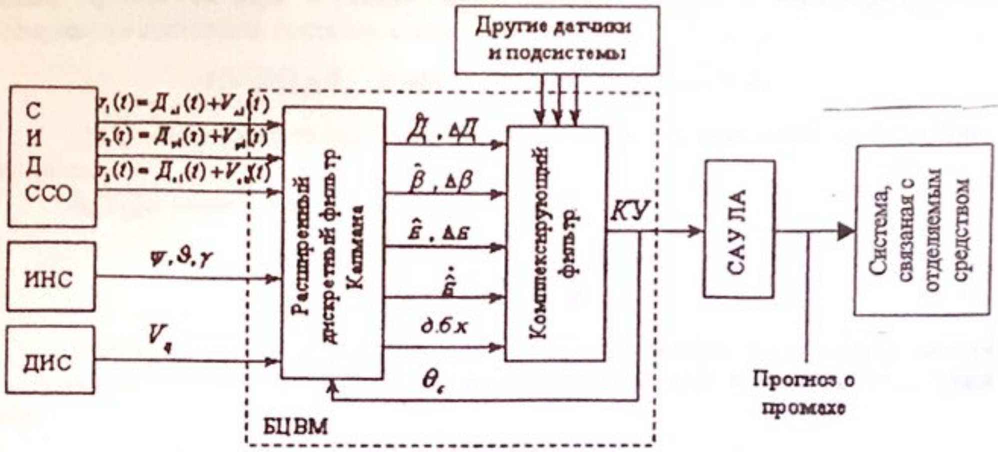
$$R_{Vx1Vz1}(t) = R_{Vz1Vx1}(t) = a_{11}a_{31}\sigma_{V_D}^2(t) + a_{12}a_{32}\sigma_{V\beta}^2(t) + a_{13}a_{33}\sigma_{V\varepsilon}^2(t);$$

$$\sigma_{Vy1}^2(t) = a_{21}^2\sigma_{V_D}^2(t) + a_{22}^2\sigma_{V\beta}^2(t) + a_{23}^2\sigma_{V\varepsilon}^2(t);$$

$$R_{Vy1Vz1}(t) = R_{Vz1Vy1}(t) = a_{21}a_{31}\sigma_{V_D}^2(t) + a_{22}a_{32}\sigma_{V\beta}^2(t) + a_{23}a_{33}\sigma_{V\varepsilon}^2(t);$$

$$\sigma_{Vy1}^2(t) = a_{31}^2\sigma_{V_D}^2(t) + a_{32}^2\sigma_{V\beta}^2(t) + a_{33}^2\sigma_{V\varepsilon}^2(t).$$

Структура ПНК с субоптимальным алгоритмическим обеспечением



Q_c – закон управления.

КУ – команды управления.

∂.б.к. – данные блока ковариаций.

В результате оценивания фазовых координат $\left(\hat{Д}, \hat{\beta}, \hat{\varepsilon}\right)$, получаемых на выходе РДФК можно сформулировать прогноз о промахе на основе информации, заключенной в статистических характеристиках линейного промаха, приняв за оценку точности наведения величину математического ожидания линейного промаха, приближенно вычисляемую по формуле

$$M\left[\Delta h(t)\right] = \frac{\left(\hat{Д}^{*}(t)\right)^2}{\left|\dot{\hat{Д}}(t)\right|} \dot{\hat{\varepsilon}}^{*}(t),$$

где $\hat{Д}(t)$ $\left(\hat{Д}^{*}(t)\right)$ – оценка дальности между ЛА и целью в текущий момент времени (на момент срыва наведения).

В настоящее время в ПНК широкое распространение получили субоптимальные алгоритмы обработки информации, базирующиеся на непрерывных дифференциальных и разностных математических моделях.

Ведущая роль в алгоритмическом обеспечении ПНК принадлежит теории Калмановской фильтрации. Ввиду нелинейности поставленной задачи в данной работе в качестве алгоритмического обеспечения используется расширенный дискретный фильтр Калмана.

Алгоритм расширенного дискретного фильтра Калмана

Математическая модель динамики состояний системы

$$x(k+1) = \Phi[x(k),k] + \Gamma[x(k),k]\,W(k), \quad x(k_0) - н.у., \tag{I}$$

где x(к) – недоступный непосредственному наблюдению фазовый вектор-столбец переменных состояний системы размерности [n×1];

x(к$_0$) – начальное состояние системы – есть случайный вектор с математическим ожиданием $M[x(k_0)] = \bar{x}_0$, и заданной ковариационной матрицей

$$P_{x0} = P_x(k_0) = \operatorname{cov}[x(k_0), x(k_0)] = M[(x(k_0) - \bar{x}_0)(x(k_0) - \bar{x}_0)^T].$$

Замечание: P_{x0} – есть симметрическая матрица размерности [n×n], в общем случае она является диагональной, элементы которой, стоящие по диагонали представляют собой дисперсии вектора-столбца переменных состояний системы.

W(k) – случайный векторный процесс, представляющий собой дискретный белый гауссовский шум, в общем случае размерности [n×1], характеризующий возмущения состояний системы, с характеристиками:

$$M[W(k)] = 0\,, \quad \mathrm{cov}[W(k), W(j)] = Q_1(k)\delta_k(k-j)\,.$$

$Q_1(k)$ – неотрицательно определенная ковариационная матрица размерности [n×n].

$\delta_k(k\text{-}j)$ – символ Кронекера.

$$\delta_k(k-j) = \begin{cases} 1 & , k = j\,; \\ 0 & , k \neq j\,. \end{cases}$$

Г[x(k),k] – матрица размерности [n×n], отражающая зависимость между фазовым вектором-столбцом переменных состояний x(к) и шумом состояния W(k).

Модель наблюдения системы имеет вид

$$r(k) = S[x(k), k] + V(k)\,, \tag{II}$$

где r(k) – вектор измерений [m×1].

S[x(k),k] – нелинейная векторная функция [m×1] переменных состояния системы представляющая собой аналитическое выражение сигналов различной физической природы.

Замечание: нелинейная зависимость полезного сигнала измерителя от вектора фазовых координат является характерной особенностью прицельно-навигационных задач.

V(k) – векторный процесс типа белого гауссовского шума, характеризующий шумы (ошибки) измерений с характеристиками

$$M[V(k)] = 0\,, \quad \mathrm{cov}[V(k), V(j)] = R_1(k)\delta_k(k-j)\,,$$

где $R_1(k)$ – положительно определенная ковариационная матрица интенсивности шумов измерений системы, вне диагонали которой стоят взаимные интенсивности шумов или ошибок измерений.

Полагаем, что вектор начального состояния $x(k_0)$, вектор ошибок измерений V(k) и вектор шумов возмущения состояний системы W(k) взаимно не коррелированны.

$$\mathrm{cov}[x(k_0), V(k)] = \mathrm{cov}[x(k_0), W(k)] = \mathrm{cov}[V(k), W(j)] = 0\,, \quad \text{для}\ \forall\, k, j > k_0\,.$$

Алгоритм фильтрации (оценивания)

Замечание: будем обозначать

$$\hat{x}(k+1, k+1) = \hat{x}(k+1)\,,$$
$$\hat{x}(k, k) = \hat{x}(k)\,.$$

$$\hat{x}(k+1) = \hat{x}(k+1, k) + K(k+1)[r(k+1) - S[\hat{x}(k+1, k), k+1]\,. \tag{1}$$

Одношаговое предсказание или прогноз оценки на (к+1) шаг

$$\hat{x}(k+1, k) = \Phi[\hat{x}(k), k]\,, \quad \text{н.у.}\ \hat{x}(k_0) = \hat{x}_0\,. \tag{2}$$

Алгоритм прогнозной (априорной) ковариационной матрицы ошибок оценивания

$$P(k+1,k) = \frac{\partial\Phi[\hat{x}(k),k]}{\partial\hat{x}(k)} P(k,k) \frac{\partial\Phi^T[\hat{x}(k),k]}{\partial\hat{x}(k)} + \Gamma[\hat{x}(k),k]\ Q_1(k)\ \Gamma^T[\hat{x}(k),k], \qquad (3)$$

$н.у.\ P_1(k_0) = P(0,0)$.

Алгоритм апостериорной ковариационной матрицы ошибок оценивания

$$P(k+1,k+1) = P(k+1,k) - P(k+1,k)\frac{\partial S^T[\hat{x}(k+1,k),k+1]}{\partial\hat{x}(k+1,k)}\left[\frac{\partial S[\hat{x}(k+1,k),k+1]}{\partial\hat{x}(k+1,k)}\cdot\right.$$

$$\left.\cdot P(k+1,k)\frac{\partial S^T[\hat{x}(k+1,k),k+1]}{\partial\hat{x}(k+1,k)} + R_1(k+1)\right]^{-1}\frac{\partial S[\hat{x}(k+1,k),k+1]}{\partial\hat{x}(k+1,k)}P(k+1,k). \qquad (4)$$

Коэффициент усиления в матричной форме (в зависимости от ковариационной матрицы ошибок оценивания)

$$K(k+1) = P(k+1,k+1)\cdot\frac{\partial S^T[\hat{x}(k+1,k),k+1]}{\partial\hat{x}(k+1,k)}R_1^{-1}(k+1). \qquad (5)$$

Ошибка оценивания на (к+1) шаге

$$\Delta x(k+1) = x(k+1) - \hat{x}(k+1). \qquad (6)$$

Начальные условия для расчета по алгоритму РДФК соответствуют априорным значениям математического ожидания $\hat{x}(k_0) = \bar{x}_0$ вектора состояния и ковариационной матрицы в начальный момент времени $P_x(k_0) = P(0,0)$.

Структурная схема алгоритма обработки по уравнениям РДФК (1)-(6) при постановке задачи (I)-(II) представлена на рисунке (стр.11).

Субоптимальной оценкой, выдаваемой на выходе с учетом наблюдений $r(k)$, поступающих на вход фильтра, является условное математическое ожидание на интервале наблюдения (k_0,k)

$$\hat{x}(k) = M\big[x(k)/r(k)\big]$$

Нелинейный фильтр Калмана формирует оценку лишь приближенно, более того само уравнение ковариаций для нелинейных моделей состояний системы и наблюдений является приближенным и справедливым лишь при достаточной малости ошибок оценивания.

Уравнение ковариационной матрицы ошибок оценивания зависит от производных нелинейных функций уравнений состояния системы и производных нелинейных функций полезных сигналов наблюдений от одношагового предсказания по соответствующим оценкам.

Функция $S\big[\hat{x}(k+1,k),k+1\big]$ является нелинейной функцией прогноза оценки на (к+1) шаг.

Производные $\dfrac{\partial\Phi\big[\hat{x}(k),k\big]}{\partial\hat{x}(k)}$ *и* $\dfrac{\partial S\big[\hat{x}(k+1,k),k+1\big]}{\partial\hat{x}(k+1,k)}$

представляют собой матрицы Якоби.

Функция $S(x) = \big[S_1(x)\cdots S_m(x)\big]$ – m-мерная векторная функция векторного аргумента $x = \|x_1\cdots x_n\|^T$ и матрицы Якоби будут иметь размерности [m×n] и представляют собой в каждой строке градиент функции S_i, $i=\overline{1,m}$ по компонентам вектора состояния x_j, $j=\overline{1,n}$, т.е. в каждой строке градиент поля.

Замечание: дискретный и непрерывный алгоритмы Калмана являются точными только в том случае, если модели состояния и наблюдения линейны. Данный алгоритм выдает наилучшую оценку вектора состояния (в силу линеаризации), минимизирующую ковариационную матрицу ошибок оценивания.

Для получения РДФК был выбран метод линеаризации в силу следующих причин: раскладывая функцию Ф в ряд, получали оценку $\hat{x}(k-1)$, затем при поступлении измерений $r(k)$ на основании невязок $r(k)-S[x(k,k-1),k]$ получали оценку $\hat{x}(k)$; допускали, что оценка $\hat{x}(k)$ "вероятно" лучше, чем прогноз оценки $\hat{x}(k,k-1)$, поэтому осуществляли линеаризацию функции Ф в оценочной точке $\hat{x}(k)$. Таким образом был получен итерационный дискретный фильтр, пересчитывающий оценку вектора состояния в итерационном процессе любое желаемое число раз с учетом линеаризации функций Ф и S в окрестностях точек $\hat{x}(k)$ и $\hat{x}(k+1,k)$, соответственно.

Наилучшая оценка $\hat{x}(k)$ на интервале наблюдения зависит от выбранного критерия качества. В расширенном фильтре Калмана наилучшая оценка $\hat{x}(k)$ на основе измерений $r(k)$ на интервале наблюдения (k_0,k) является оценкой максимизирующей апостериорную плотность вероятности $\omega(x(k)/r(k))$ по функции x.

Максимизация апостериорной плотности эквивалентна задаче минимизации функционала следующего вида

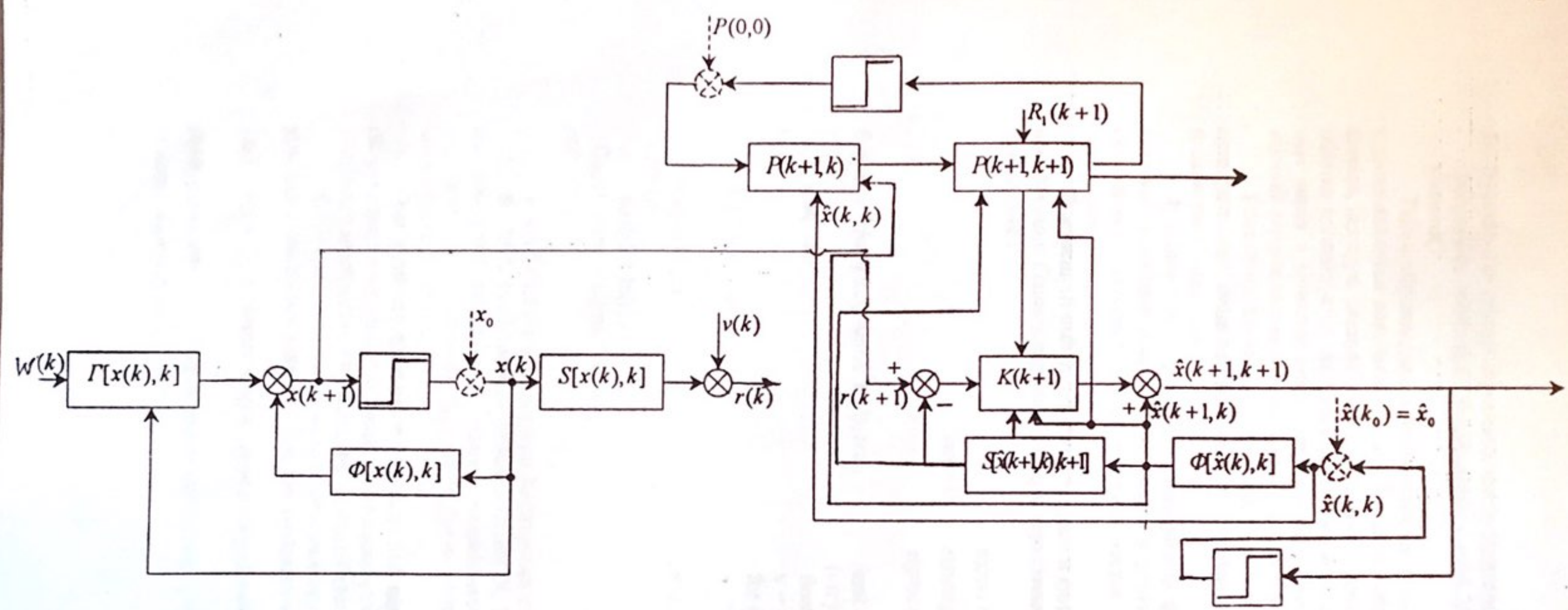
$$I=\frac{1}{2}\|\hat{x}(k_0)-\bar{x}_0\|^2 P_x(k_0)+\frac{1}{2}\sum_{K=K_0+1}^{K_f}\|r(k)-S[\hat{x}(k,k-1),k]\|^2 R_1^{-1}(k)+\sum_{K=K_0}^{K_f}\|W(k)\|^2 Q_1^{-1}(k)$$

Фильтр, реализующий максимум апостериорной плотности вероятности (МАВ), является, в общем случае, дискретным фильтром, т.к. сама апостериорная плотность вероятности, на основе которой получены алгоритмы фильтрации, не существует при неограниченном устремлении к нулю шага дискретизации.

Моделирование случайных процессов

Функция-генератор Random генерирует вещественные случайные числа, распределенные с равномерной плотностью вероятности на интервале, и имеет следующие особенности:

1) В том случае, если вызов функции Random осуществляется без аргумента, то интервал распределения, возвращаемых случайной функцией значений, будет $0 \le r < 1$. Если функция имеет параметр N (Random(N)), то интервал распределения случайных чисел – $0 \le r < N$.



Структурная схема РДФК

2) Функция-генератор случайных чисел Random при последовательных запусках программы генерирует последовательность с одними и теми же случайными числами.

Таким образом, данный генератор не является оптимальным, поскольку результат является псевдослучайным. В связи с этим используется процедура Randomize, которую следует включать в начале программы. В процедуре Randomize обычно используется так называемый встроенный таймер для получения случайных чисел в качестве новой затравки при генерации последовательности, сбивающей эту последовательность на новое случайное число.

Генератор случайных чисел Random с равномерным распределением используется при создании генераторов случайных последовательностей с другими видами распределения.

Функция Гаусса (Gauss), использующая функцию-генератор Random, возвращает случайное вещественное число с нормальным законом распределения, имеющим характеристики: mo (математическое ожидание) и sigma (среднеквадратическое отклонение).

В курсовой работе ошибки измерений формируются с помощью стандартной функции Gauss с заданными характеристиками для каждого канала системы измерителей.

$$mo = 0 \quad , \quad \begin{matrix} sigma\,D \\ sigma\,\beta \\ sigma\,\varepsilon \end{matrix}$$

```
Function Gauss(mo, sigma: real): real;
var
  a, b, r, sq: real;
begin
  repeat
    a := 2 * Random - 1;
    b := 2 * Random - 1;
    r := sqr(a) + sqr(b);
  until (r<1);
  sq := sqrt(-2 * ln(r) / r);
  Gauss := mo + sigma * a * sq;
end;
```

```
float Gauss(float sig)
{int j;
 float v;
 v = 0;
 for (j = 0; j < 12; j++)
 v += ((double) rand( ) / 32768 - 0.5);
 v *= sig;
return v; }
```

В VAX / UNIX используется библиотека стандартных подпрограмм LSSP.

В UNIX используется функция Gauss(IX, S, AM, V), где IX – целое нечетное девятизначное число, S – среднеквадратическое отклонение, AM – желаемое математическое ожидание, V – генерируемое случайное число с нормальным законом распределения.

Эта функция использует комбинацию случайных чисел, распределенных по равномерному закону с целью получения нормального распределения в соответствии с центральной предельной теоремой (теоремой Ляпунова).

Если имеется случайная последовательность чисел с одним и тем же законом распределения, одним и тем же математическим ожиданием и дисперсией $\{x_k\}$, то $y = \sum_{k=1}^{n} x_k$ будет иметь закон распределения близкий к нормальному с параметрами: nm_x – математическое ожидание, $\sqrt{m_x} \cdot \sigma_x$ – дисперсия, и тем ближе (к нормальному), чем больше n.

2) Функция-генератор случайных чисел Random при последовательных запусках программы генерирует последовательность с одними и теми же случайными числами.

Таким образом, данный генератор не является оптимальным, поскольку результат является псевдослучайным. В связи с этим используется процедура Randomize, которую следует включать в начале программы. В процедуре Randomize обычно используется так называемый встроенный таймер для получения случайных чисел в качестве новой затравки при генерации последовательности, сбивающей эту последовательность на новое случайное число.

Генератор случайных чисел Random с равномерным распределением используется при создании генераторов случайных последовательностей с другими видами распределения.

Функция Гаусса (Gauss), использующая функцию-генератор Random, возвращает случайное вещественное число с нормальным законом распределения, имеющим характеристики: mo (математическое ожидание) и sigma (среднеквадратическое отклонение).

В курсовой работе ошибки измерений формируются с помощью стандартной функции Gauss с заданными характеристиками для каждого канала системы измерителей.

$$mo = 0 \quad , \quad \begin{array}{l} sigma\, D \\ sigma\, \beta \\ sigma\, \varepsilon \end{array}$$

```
Function Gauss(mo, sigma: real): real;
var
  a, b, r, sq: real;
begin
  repeat
    a := 2 * Random - 1;
    b := 2 * Random - 1;
    r := sqr(a) + sqr(b);
  until (r<1);
  sq := sqrt(-2 * ln(r) / r);
  Gauss := mo + sigma * a * sq;
end;
```

```
float Gauss(float sig)
{int j;
  float v;
  v = 0;
  for (j = 0; j < 12; j++)
  v += ((double) rand( ) / 32768 - 0.5);
  v *= sig;
return v; }
```

В VAX / UNIX используется библиотека стандартных подпрограмм LSSP.

В UNIX используется функция Gauss(IX, S, AM, V), где IX – целое нечетное девятизначное число, S – среднеквадратическое отклонение, AM – желаемое математическое ожидание, V – генерируемое случайное число с нормальным законом распределения.

Эта функция использует комбинацию случайных чисел, распределенных по равномерному закону с целью получения нормального распределения в соответствии с центральной предельной теоремой (теоремой Ляпунова):

Если имеется случайная последовательность чисел с одним и тем же законом распределения, одним и тем же математическим ожиданием и дисперсией $\{x_k\}$, то $y = \sum_{k=1}^{n} x_k$ будет иметь закон распределения близкий к нормальному с параметрами: nm_x – математическое ожидание, $\sqrt{m_x} \cdot \sigma_x$ – дисперсия, и тем ближе (к нормальному), чем больше n.


```
float Gauss(float mo,float sigma) {
 float a,b,r,Sq;
 float v;
  do {
   a=2*rand()-1;
   b=2*rand()-1;
   r=pow(a,2)+pow(b,2);
  } while(r<1);
 Sq=sqrt(2*logl(r)/r);
 v=mo+sigma*a*Sq;
 return v;
}
```