

Библиотеки алгоритмов навигации и определения препятствий

СК Роботикс

17 апреля 2023 г.

1 Библиотека алгоритмов навигации - теория

В состав библиотеки алгоритмов навигации входят следующие алгоритмы ориентации и навигации:

- Алгоритм гироскопирования (Степан);
- Алгоритм курсоверткали, оценивающий ориентацию робота (углы курса, крена и тангажа) относительно навигационной системы координат (СК). В основе алгоритма лежит комплексирование информации от векторных датчика угловой скорости, магнитометра и акселерометра с использованием расширенного фильтра Калмана;
- Алгоритм навигации, вырабатывающий оценки координат, скоростей и азимутального угла ориентации робота относительно навигационной системы координат. Представлены две разновидности алгоритма, отличающиеся источником внешней информации, применяемой для корректирования оценок: с использованием измерений дальностей до локально установленных базовых станций радионавигационной системы UWB (Ultra-wide band, сверхширокополосная система малой мощности); с использованием измерений координат и скоростей, доставляемых глобальной спутниковой навигационной системой. Комплексирование в этих двух разновидностях комплексной системы навигации осуществляется с помощью расширенного фильтра Калмана;
- Алгоритм навигационной системы, комплексировающей измерения от локальной радионавигационной системы UWB с управляющими воздействиями, поступающими от системы траекторного управления колесного робота. Алгоритм обеспечивает оценку координат робота и его азимутального угла разворота относительно навигационной системы координат. Комплексирование осуществляется с использованием нелинейного Байесовского фильтра - фильтра частиц PF (particle filter).

1.1 Алгоритм гироскопирования

Степан

1.2 Алгоритм курсоверткали

Задачей алгоритма курсоверткали является определение углов ориентации колесного робота, в частности - азимутального угла разворота робота относительно навигационной СК.

Курсовертикаль включает три микромеханических датчика угловой скорости и три микромеханических акселерометра со взаимно ортогональными осями чувствительности. Алгоритмически курсовертикаль построена на принципе коррекции ориентации, получаемой путем интегрирования показаний датчиков угловой скорости. Эта коррекция осуществляется по информации от векторных акселерометра и магнитометра.

Комплексная обработка информации, используемая в алгоритме, основана на применении расширенного фильтра Калмана, оценивающего соответствующий вектор ошибок $\delta \mathbf{x}_M$. Определим состав этого вектора. Уравнение ошибок $\delta \dot{\Psi}^n$ ориентации, используемое в алгоритме курсовертикали:

$$\delta \dot{\Psi}^n = -C_b^n \delta \omega^b. \quad (1)$$

Уравнение измерений и модель ошибок измерений сформируем на примере определения вектора ускорения силы тяжести \mathbf{g}^n . В случае равномерно и прямолинейно движущегося, либо неподвижного робота вектор \mathbf{g}^n может быть записан так:

$$\mathbf{g}^n = C_b^n \mathbf{f}^b, \quad (2)$$

где \mathbf{f}^b - вектор кажущегося ускорения, измеренный векторным акселерометром. Для получения модели ошибок определения вектора ускорения силы тяжести проварьируем (2): Возмущенное значение $\tilde{\mathbf{g}}^n$ таково:

$$\tilde{\mathbf{g}}^n = \tilde{C}_b^n \tilde{\mathbf{f}}^b; \quad (3)$$

$$\mathbf{g}^n + \delta \mathbf{g}^n = (I - E^n) C_b^n (\mathbf{f}^b + \delta \mathbf{f}^b); \quad (4)$$

$$\mathbf{g}^n + \delta \mathbf{g}^n = C_b^n \mathbf{f}^b + C_b^n \delta \mathbf{f}^b - E^n C_b^n \mathbf{f}^b - E^n C_b^n \delta \mathbf{f}^b. \quad (5)$$

Удерживая в (5) только члены первого порядка малости, представим модель ошибок определения вектора ускорения силы тяжести так:

$$\delta \mathbf{g}^n = C_b^n \delta \mathbf{f}^b + \left([C_b^n \mathbf{f}^b] \times \right) \delta \Psi^n. \quad (6)$$

Вектор измерения \mathbf{z}_f в канале коррекции курсовертикали с использованием акселерометра сформируем как разность оценки вектора ускорения силы тяжести $\hat{\mathbf{g}}^n$ (полученного с использованием соотношения типа (2)) и его эталонного значения \mathbf{g}^n :

$$\mathbf{z}_f = \hat{\mathbf{g}}^n - \mathbf{g}^n = \delta \mathbf{g}^n. \quad (7)$$

С другой стороны, имеет место соотношение:

$$\mathbf{z}_f = H_f \delta \mathbf{x}_M + \boldsymbol{\nu}_f, \quad (8)$$

где H_f - матрица измерений в канале коррекции курсовертикали с использованием акселерометров; $\boldsymbol{\nu}_f$ - вектор шумов измерений. Вектор ошибок курсовертикали $\delta \mathbf{x}_M$ определим следующим образом:

$$\delta \mathbf{x}_M = [\delta \Psi^n \quad \delta \omega^b \quad \delta \mathbf{f}^b]^T. \quad (9)$$

Из (6), (9) следует:

$$H_f = \left[[C_b^n \mathbf{f}^b] \times \quad \mathbf{0} \quad C_b^n \right]. \quad (10)$$

Отметим, что в общем случае для обеспечения коррекции вычисленной курсовертикалью ориентации платформы недостаточно выполнить измерение направления одного вектора, в частности вектора ускорения силы тяжести \mathbf{g}^n , так как ориентация робота в плоскости,

перпендикулярной этому вектору не определена, а ошибки ориентации по этой угловой координате - ненаблюдаемы. В качестве дополнительного вектора используем вектор магнитного поля Земли \mathbf{m}^b , измеряемый векторным магнитометром в связанной с роботом СК. Для этого вектора, по аналогии с вектором ускорения силы тяжести, соотношения для модели измерения таковы:

$$\mathbf{z}_m = \hat{\mathbf{n}}^n - \mathbf{m}^n; \quad (11)$$

$$\hat{\mathbf{m}}^n = \hat{C}_b^n \hat{\mathbf{m}}^b; \quad (12)$$

$$\mathbf{z}_m = H_m \delta \mathbf{x}_M + \boldsymbol{\nu}_m; \quad (13)$$

$$H_m = [[C_b^n \mathbf{m}^b] \times \quad \mathbf{0} \quad \mathbf{0}], \quad (14)$$

где \mathbf{z}_m , H_m , $\boldsymbol{\nu}_m$ - соответственно вектор измерений, матрица измерений и вектор шумов измерений в канале коррекции курсовертикали с использованием магнитометра. Полагая, что динамика ошибок измерений $\delta \dot{\mathbf{f}}^b$ и $\delta \dot{\boldsymbol{\omega}}^b$ определяется соотношениями $\delta \dot{\mathbf{f}}^b = \mathbf{n}_f$ и $\delta \dot{\boldsymbol{\omega}}^b = \mathbf{n}_\omega$, запишем уравнения ошибок курсовертикали в матричной форме так:

$$\begin{bmatrix} \delta \dot{\Psi}^n \\ \delta \dot{\boldsymbol{\omega}}^b \\ \delta \dot{\mathbf{f}}^b \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -C_b^n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \Psi^n \\ \delta \boldsymbol{\omega}^b \\ \delta \mathbf{f}^b \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \mathbf{n}_\omega \\ \mathbf{n}_f \end{bmatrix} \quad (15)$$

где \mathbf{n}_ω и \mathbf{n}_f - белые шумы гироскопов и акселерометров соответственно ($M[\mathbf{n}_\omega \mathbf{n}_\omega^T] = Q_\omega(t) \delta(t - \tau)$, $M[\mathbf{n}_f \mathbf{n}_f^T] = Q_f(t) \delta(t - \tau)$), Q_ω и Q_f - матрицы интенсивностей белых шумов, $\delta(\cdot)$ - дельта-функция).

Соотношения, приведенные выше, достаточны для формирования дискретного расширенного фильтра Калмана. Результатом работы алгоритма курсовертикали являются оценки ориентации робота, сдвигов нулей датчиков угловой скорости и векторного акселерометра.

1.2.1 Моделирование работы алгоритма курсовертикали

Для оценки точности работы алгоритма курсовертикали проведено моделирование в рамках которого задается опорное движение твердого тела с курсовертикалью с помощью алгоритма SPIN-CONE, синтезирующего кинематические параметры тела, вращающегося с постоянной угловой скоростью вокруг некоторой оси вращения, которая в свою очередь описывает конус вокруг оси прецессии с постоянной угловой скоростью прецессии. На Рис. 1 - 3 приведены результаты моделирования работы алгоритма курсовертикали демонстрирующие сходящийся процесс оценивания ошибок углов ориентации и состоятельные оценки сдвигов нулей гироскопов.

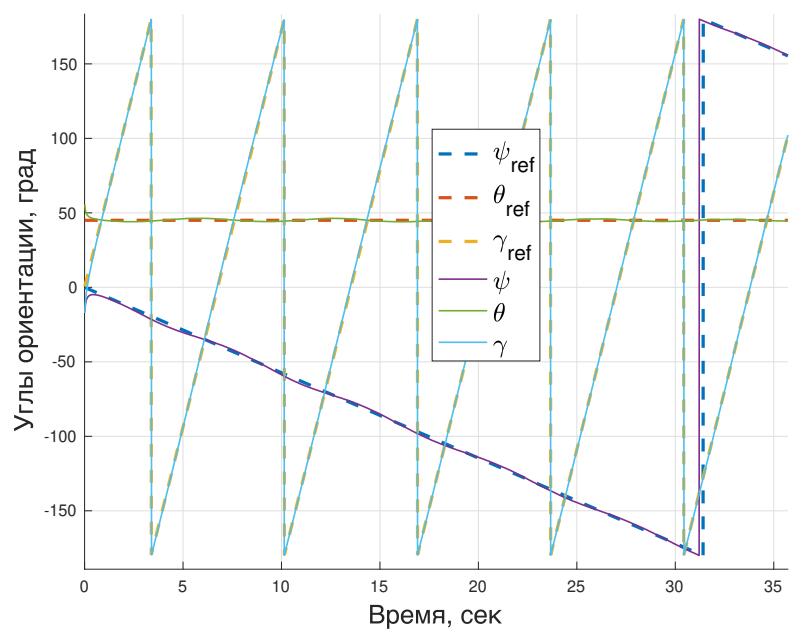


Рис. 1: Опорные углы ориентации и их оценки алгоритмом курсовертикали

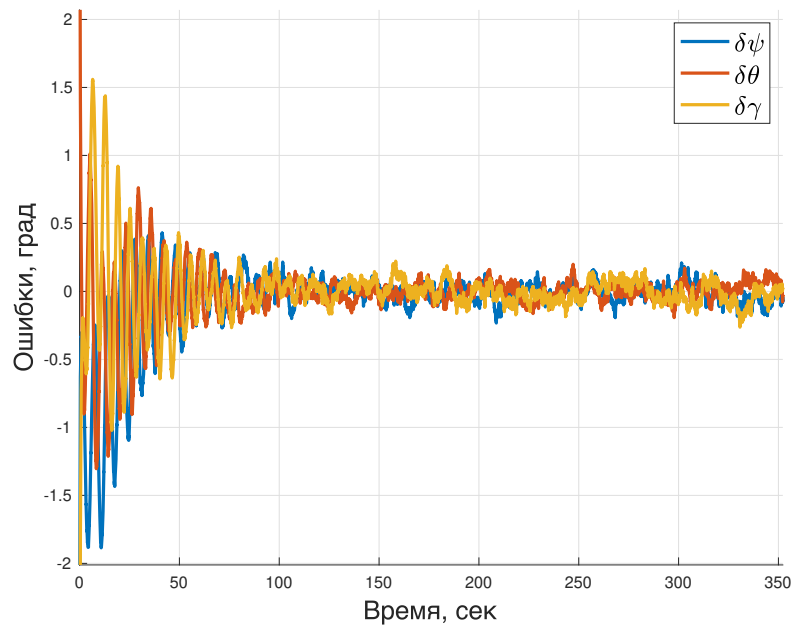


Рис. 2: Ошибки оценок углов ориентации алгоритмом курсовертикали

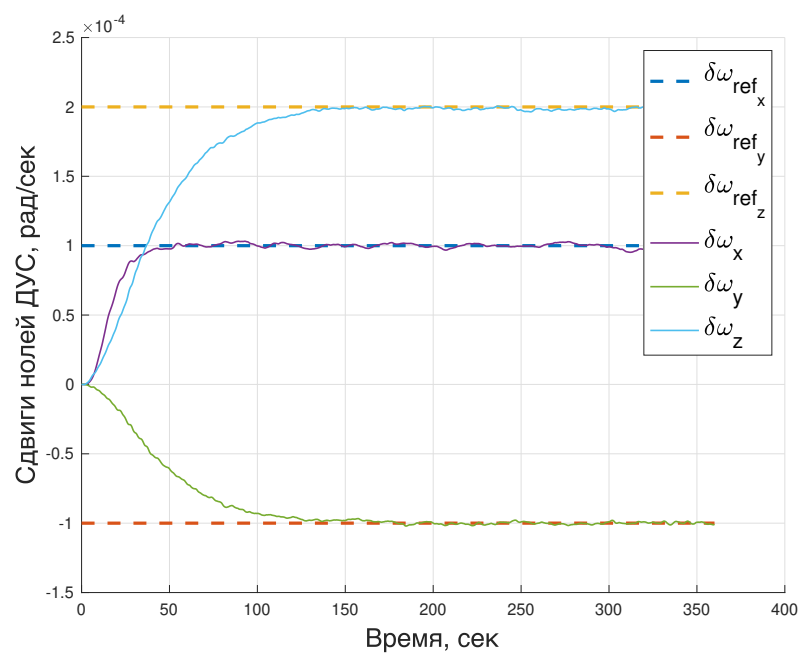


Рис. 3: Оценки сдвигов полей ДУС алгоритмом курсовертикали

1.3 Алгоритм комплексной системы навигации с использованием расширенного фильтра Калмана

1.3.1 Алгоритм навигации с коррекцией от UWB

Комплексная система навигации трехколесного робота с двумя ведущими колесами и третьим пассивным поддерживающим колесом включает датчики скоростей вращения ведущих правого и левого колес, датчик угловой скорости вращения робота вокруг его вертикальной оси. Использование этих источников информации позволяет оценивать координаты, скорости и угол курса робота в навигационной системе координат. Однако, непрерывное интегрирование сигналов датчиков приводит к неконтролируемому нарастанию во времени ошибок определения навигационных параметров робота.

Для компенсации этих ошибок и получения стабильного навигационного решения традиционно используется приемник глобальной навигационной спутниковой системы (ГНСС), информация от которого о координатах и скоростях носителя комплексируется с информацией от одометра и курсового гироскопа. При этом среда, в которой функционирует мобильный робот, зачастую включает помещения, например, ангары, внутри которых получение стабильного навигационного решения приемником ГНСС затруднено.

Для организации непрерывного, корректирующего локального навигационного поля в работе предложено использовать измерения от локальной радионавигационной системы на базе ультраширокополосной технологии ultra-wide band (UWB). Эта радиочастотная технология использует короткие импульсы с максимально возможной полосой пропускания при минимально возможной центральной частоте. Технология используется в устройствах связи, радиолокации, при определении расстояний и позиционировании.

В состав локальной радионавигационной системы входит мобильный приемо-передатчик сигналов, устанавливаемый на трехколесном роботе и несколько стационарных базовых станций, устанавливаемых на объекте, в рамках которого необходимо осуществлять навигацию робота. В результате двустороннего обмена короткими широкополосными радиопакетами между приемо-передатчиком и базовой станцией происходит синхронизация времени на субнаносекундном уровне, что позволяет определить время распространения радиосигнала и, как следствие, вычислить расстояние между этими двумя устройствами с сантиметровой точностью.

Таким образом, в предлагаемой комплексной системе навигации предлагается выполнить сильно связанное комплексирование информации от нескольких разнородных источников - одометра, датчика угловой скорости, локальной системы UWB с целью получения точного и робастного навигационного решения.

На Рис. 4 показаны связанная и навигационная системы координат, схематично приведены расположение элементов системы UWB на роботе и на объекте, в рамках которого предполагается осуществлять навигацию. Координаты базовых станций UWB в навигационной системе координат считаются фиксированными и известными.

Уравнения одометра:

$$v_x^b = \frac{1}{2}(\dot{\phi}_r l_r + \dot{\phi}_l l_l); \quad (16)$$

$$\dot{\theta} = \frac{1}{d}(\dot{\phi}_r l_r - \dot{\phi}_l l_l), \quad (17)$$

где: v_x^b - продольная скорость перемещения робота; $\dot{\theta}$ - угловая скорость разворота робота вокруг вертикальной оси; $\dot{\phi}_r, \dot{\phi}_l$ - угловые скорости вращения правого и левого колес; l_r, l_l - длины окружностей колес; d - длина оси колесной пары.

Угловые скорости вращения колес могут в свою очередь быть получены на основе скорости продольного перемещения и скорости разворота вокруг вертикальной оси следующим

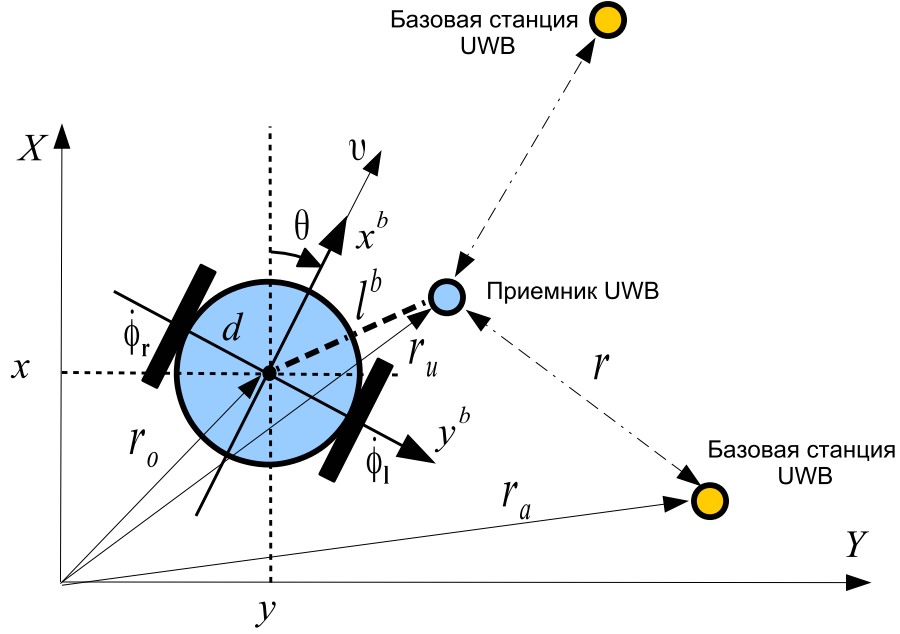


Рис. 4: Система координат

образом:

$$\dot{\phi}_r = \frac{v_x^b}{l_r} + \frac{\dot{\theta}d}{2l_r}; \quad (18)$$

$$\dot{\phi}_l = \frac{v_x^b}{l_r} - \frac{\dot{\theta}d}{2l_r}. \quad (19)$$

Уравнения навигации:

$$\dot{r}^n = C_b^n v^b = C_b^n [v_x^b \ 0 \ 0]^T; \quad (20)$$

$$\dot{C}_b^n = C_b^n \Omega^b; \quad (21)$$

$$C_b^n = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (22)$$

где: Ω^b - кососимметричная матрица вектора $\omega^b = [0 \ 0 \ \omega_z]^T$, ω_z - измерение датчика угловой скорости; $r^n = r_o = [x \ y \ 0]^T$ - координаты центра оси колес в навигационной системе координат.

Уравнения ошибок одометра (ошибок определения скорости продольного перемещения δv_x^b и скорости разворота по углу курса $\delta \dot{\theta}$):

$$\delta v_x^b = \frac{1}{2} (\dot{\phi}_r \delta l_r + \dot{\phi}_l \delta l_l); \quad (23)$$

$$\delta \dot{\theta} = \frac{1}{d} \dot{\phi}_r \delta l_r - \frac{1}{d} \dot{\phi}_l \delta l_l + \frac{1}{d^2} (\dot{\phi}_l l_l - \dot{\phi}_r l_r) \delta d; \quad (24)$$

где: $\delta l_l, \delta l_l$ - ошибки определения длины окружностей колес; δd -ошибка определения длины оси колесной пары.

Динамика ошибок координат, ориентации и сдвига нуля датчика угловой скорости

$$\delta \dot{r}^n = C_b^n \delta v^b + ([C_b^n v^b] \times) \delta \psi^n; \quad (25)$$

$$\delta \dot{\psi}^n = -C_b^n \delta \omega^b; \quad (26)$$

$$\delta \dot{\omega}^b = n_\omega; \quad (27)$$

где: $\delta r^n = [\delta x \ \delta y \ 0]$ - вектор ошибок координат; $\delta v^b = [\delta v_x^b \ 0 \ 0]^T$ -вектор ошибок скорости; $\delta \psi^n = [0 \ 0 \ \delta \psi_z^n]$ - вектор ошибок ориентации; n_ω - шум датчика угловой скорости.

Координаты r_u приемника системы UWB, расположенного на роботе, могут быть записаны так:

$$r_u = r_o + C_b^n l^b \quad (28)$$

где: $r_o = r^n$ - координаты центра оси колесной пары; $l^b = [l_x^b \ l_y^b \ 0]^T$ - вектор смещения точки закрепления приемника UWB относительно центра оси колесной пары.

Измерение расстояния $\|r\|_2$ между стационарной базовой станцией и мобильным приемником, закрепленным на роботе может быть записано так:

$$\|r\|_2 = \|r_a - r_u\|_2 = \|r_a - r_o - C_b^n l^b\|_2 \quad (29)$$

$$\|r\|_2^2 = [r_a - r_o - C_b^n l^b]^T [r_a - r_o - C_b^n l^b] \quad (30)$$

Проварьруем уравнение измерений (15):

$$2r\delta r = [r_a - r_o - \delta r_o - (I - E^n) C_b^n (l^b + \delta l^b)]^T \quad (31)$$

$$[r_a - r_o - \delta r_o - (I - E^n) C_b^n (l^b + \delta l^b)] \quad (32)$$

Тогда ошибку измерения δr расстояния r можно записать через ошибки определения координат, размещения приемника UWB так:

$$\delta r = (r_o^T - r_a^T + (l^b)^T C_n^b) \delta r_o + \quad (33)$$

$$+ ((l^b)^T + (r_o^T - r_a^T) C_n^b) \delta l^b + \quad (34)$$

$$+ (r_o^T - r_a^T + (l^b)^T C_n^b) ([C_b^n l^b] \times) \delta \psi^n \quad (35)$$

Комплексирование навигационной информации от всех источников осуществляется с использованием расширенного фильтра Калмана который оценивает компоненты следующего вектора ошибок навигационной системы трехколесного робота:

$$\mathbf{x} = [\delta x \ \delta y \ \delta \psi_x^n \ \delta l_r \ \delta l_l \ \delta \omega_z \ \delta d \ \delta l_x^b \ \delta l_y^b]^T \quad (36)$$

Вектор и матрица измерений системы UWB:

$$\mathbf{z} = [\hat{r} - r] \quad (37)$$

$$\mathbf{H} = \begin{bmatrix} r_o^T - r_a^T + (l^b)^T C_n^b \\ (r_o^T - r_a^T + (l^b)^T C_n^b) ([C_b^n l^b] \times) \\ 0_{4 \times 1} \\ (l^b)^T + (r_o^T - r_a^T) C_n^b \end{bmatrix}^T \quad (38)$$

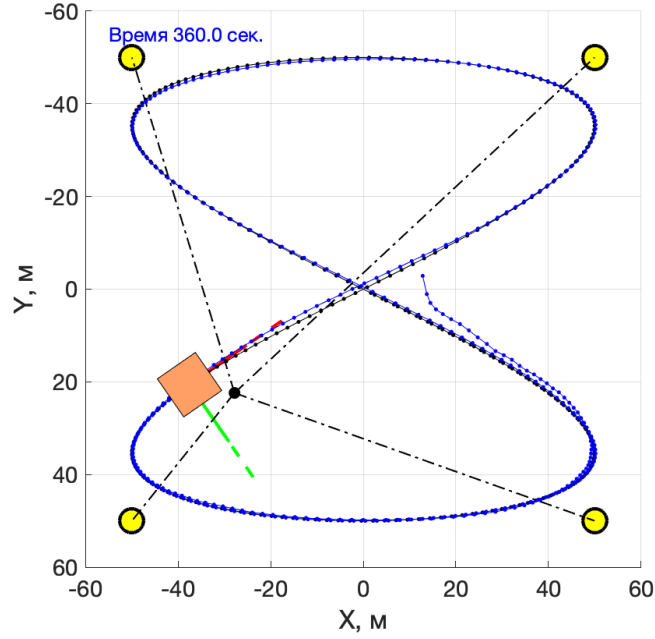


Рис. 5: Модельная траектория и расположение базовых станций UWB

Вектор и матрица измерений одометра (датчики угловой скорости вращения ведущих колес и курсового гироскопа):

$$\mathbf{z} = [\hat{\omega}_z - \dot{\theta}] \quad (39)$$

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & -\frac{\dot{\phi}_r}{d} & \frac{\dot{\phi}_l}{d} & 1 & \frac{1}{d^2}(\dot{\phi}_r l_r - \dot{\phi}_l l_l) & 0 & 0 & 0 \end{bmatrix} \quad (40)$$

1.3.2 Моделирование работы алгоритма навигации с коррекцией от UWB

С целью оценки точности и устойчивости работы комплексной системы навигации был проведен численный эксперимент в котором моделировалось перемещение робота с комплексной системой навигации по замкнутой траектории в рамках покрытия локальной радионавигационной системы UWB с четырьмя базовыми станциями, расположенными так, как показано на Рис. 5.

В результате моделирования продемонстрирована устойчивая работа навигационного комплекса трехколесного робота. На рисунках 6 - 11 приведены результаты моделирования.

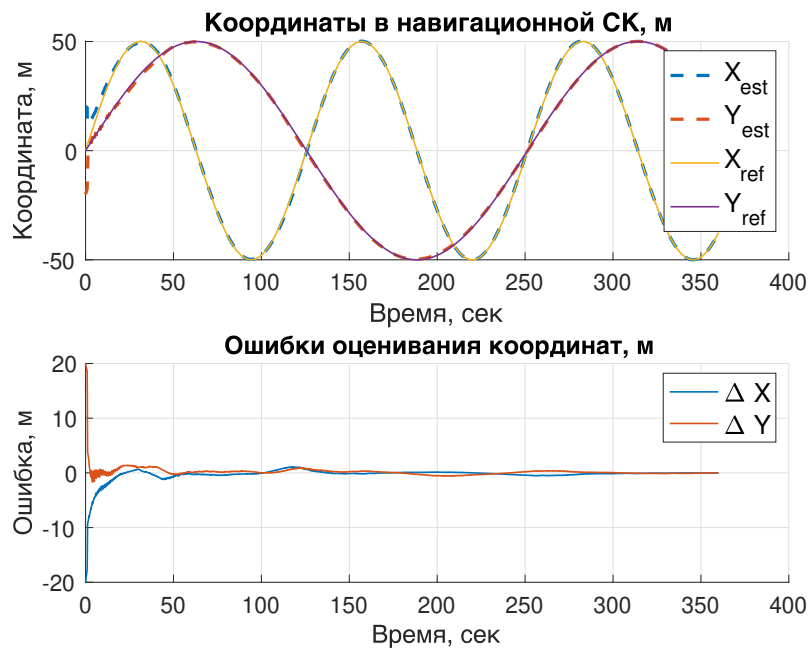


Рис. 6: Координаты робота и оценки их ошибок - UWB коррекция

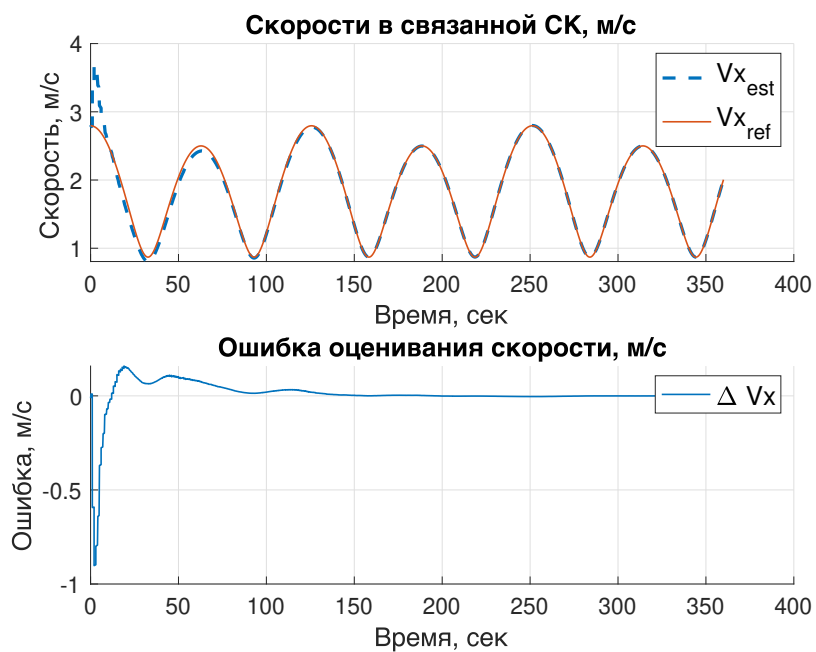


Рис. 7: Скорость продольного перемещения робота и оценка ее ошибки - UWB коррекция

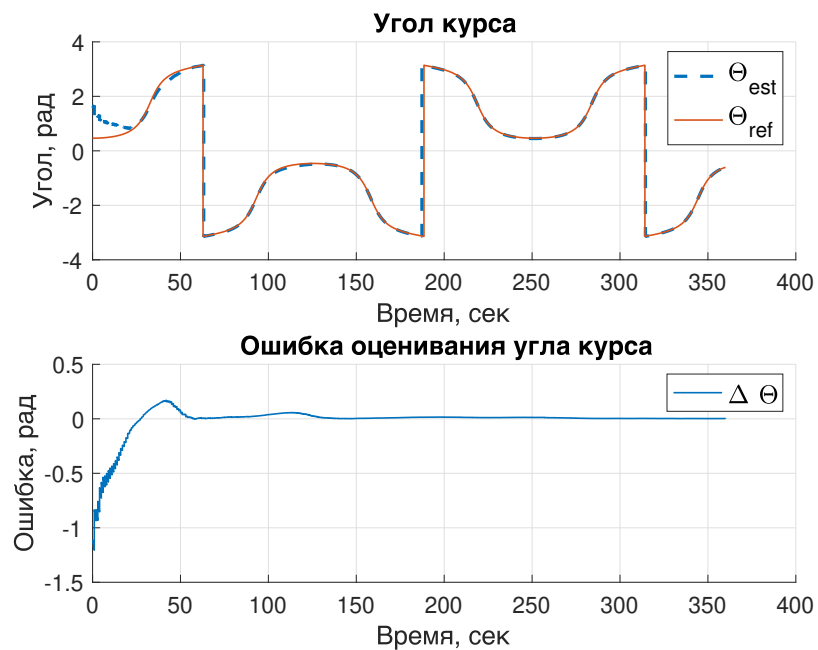


Рис. 8: Угол курса робота и оценка его ошибки - UWB коррекция

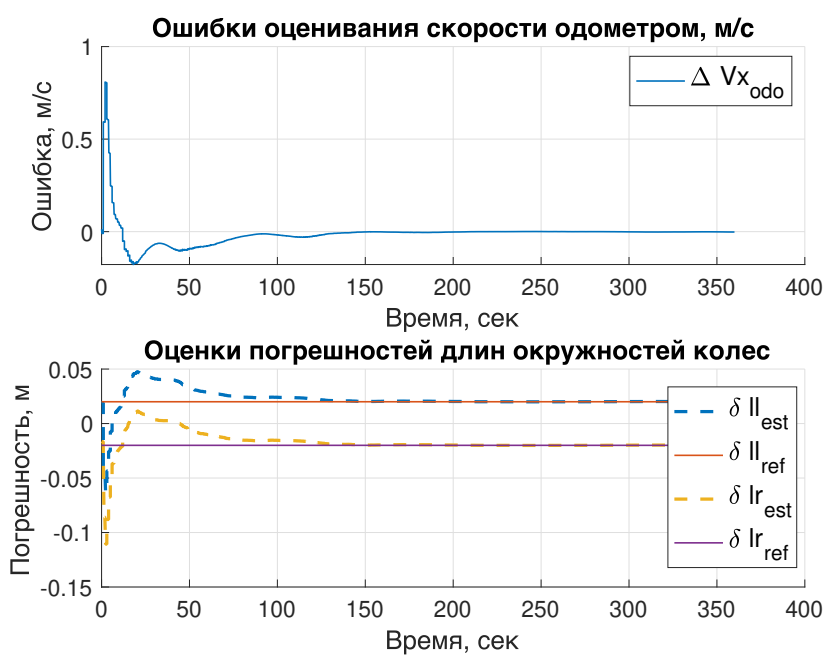


Рис. 9: Оценка ошибок одометра - UWB коррекция

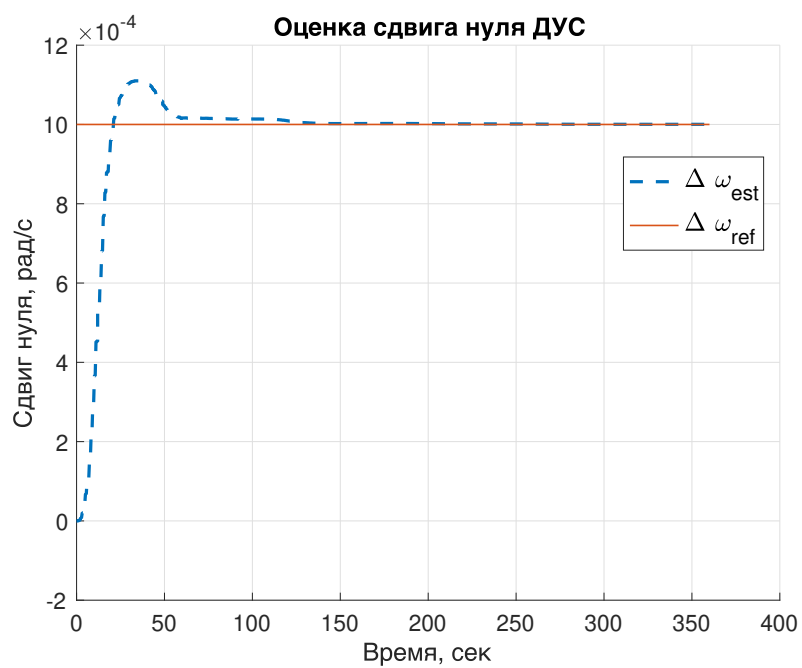


Рис. 10: Оценка ошибок датчика угловой скорости - UWB коррекция

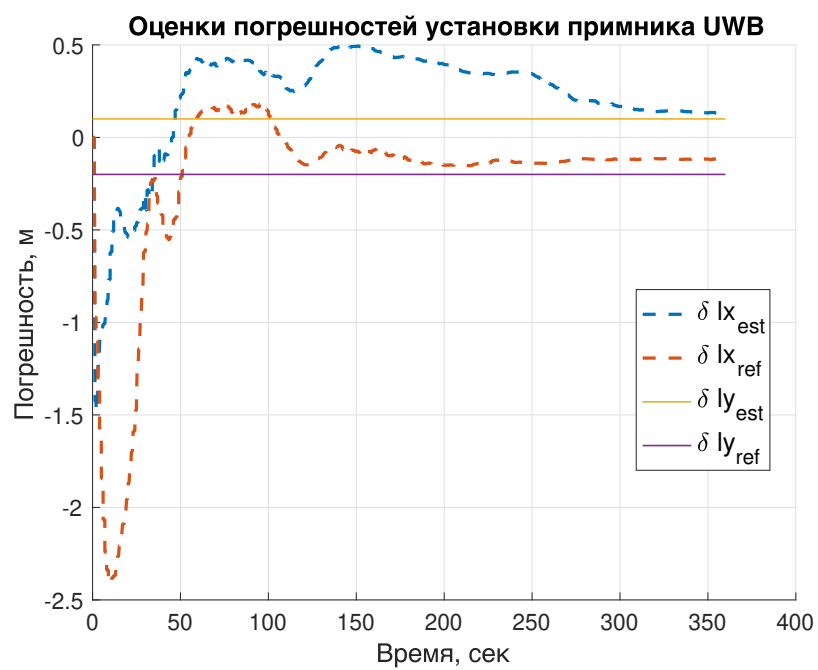


Рис. 11: Оценка погрешностей установки приемника UWB

1.3.3 Алгоритм навигации с коррекцией от спутниковой навигационной системы

В этом варианте алгоритма система координат (Рис. 4) и структура системы остается прежней, с той только разницей, что приемник системы UWB заменяется на приемник спутниковой навигационной системы.

Комплексирование навигационной информации в этом случае также осуществляется с использованием расширенного фильтра Калмана который оценивает компоненты следующего вектора ошибок навигационной системы трехколесного робота:

$$\mathbf{x} = [\delta x \quad \delta y \quad \delta \psi_x^n \quad \delta l_r \quad \delta l_l \quad \delta \omega_z^b]^T \quad (41)$$

Вектор и матрица измерений при использовании в качестве корректирующей спутниковой системы навигации (СНС):

$$\mathbf{z} = \begin{bmatrix} \hat{r}^n - r^n \\ \hat{v}^n - v^n \end{bmatrix}, \quad (42)$$

где \hat{r}^n , \hat{v}^n - оценки векторов координат и скоростей приемника СНС в навигационной СК, формируемые так:

$$\hat{r}^n = r_o + C_b^n l^b \quad (43)$$

$$\hat{v}^n = C_b^n (v^b + [\omega^b \times] l^b) \quad (44)$$

Ошибки оценок координат и скоростей приемника СНС можно записать следующим образом:

$$\delta r^n = \delta r_o + [C_b^n l^b \times] \delta \psi^n \quad (45)$$

$$\delta v^n = -C_b^n [l^b \times] \delta \omega^b + [(C_b^n v^b + C_b^n [\omega^b \times] l^b) \times] \delta \psi^n \quad (46)$$

Вид матрицы измерений H расширенного фильтра Калмана однозначным образом следует из (41), (45), (46).

1.3.4 Моделирование работы алгоритма навигации с коррекцией от СНС

С целью демонстрации работы алгоритма комплексной системы навигации с коррекцией от СНС было проведено его моделирование в рамках которого было задано перемещение модели робота по траектории, аналогичной приведенной на Рис. 5. На рисунках 12 - 16 приведены результаты моделирования, демонстрирующие устойчивость и точность работы алгоритма комплексной системы навигации.

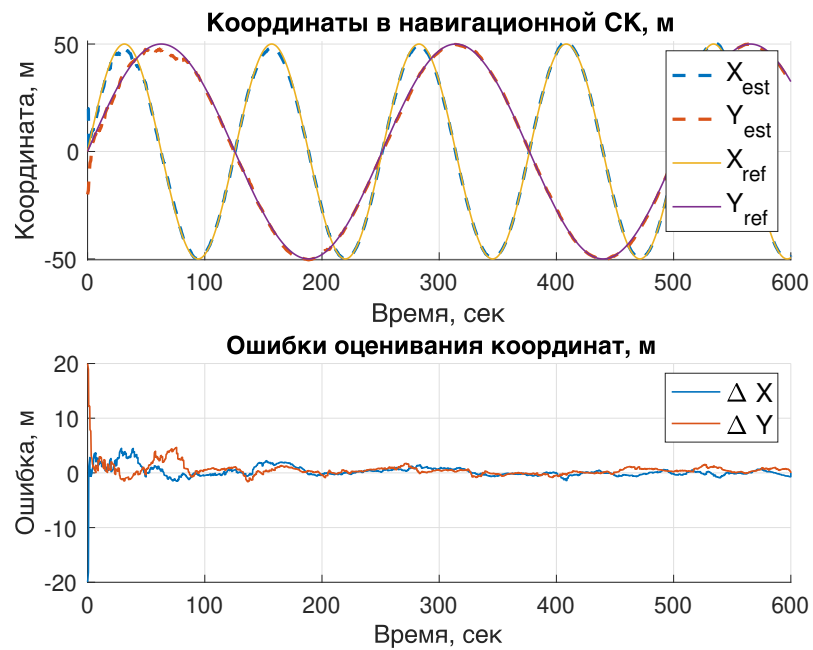


Рис. 12: Координаты робота и оценки их ошибок - СНС коррекция

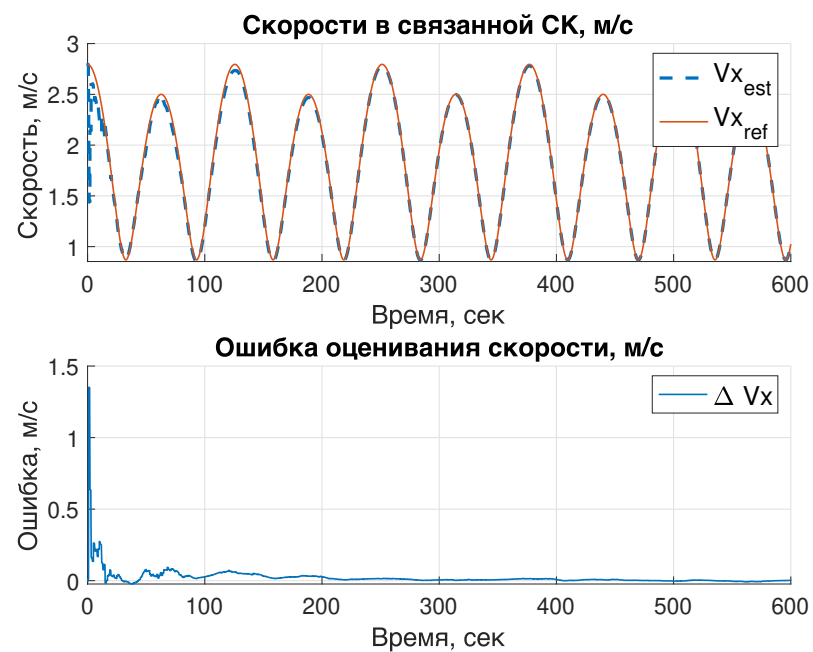


Рис. 13: Скорость продольного перемещения робота и оценка ее ошибки - СНС коррекция

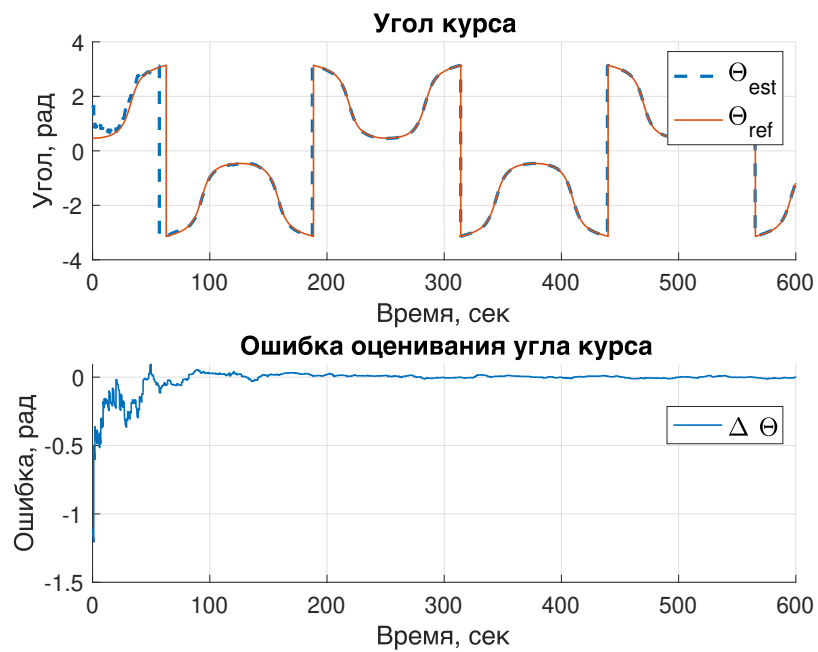


Рис. 14: Угол курса робота и оценка его ошибки - СНС коррекция

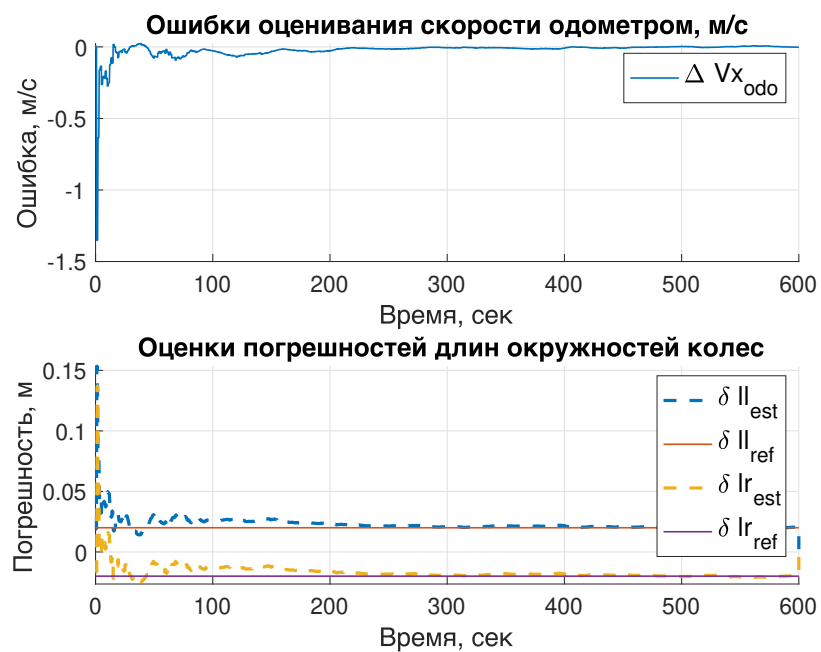


Рис. 15: Оценка ошибок одометра - СНС коррекция

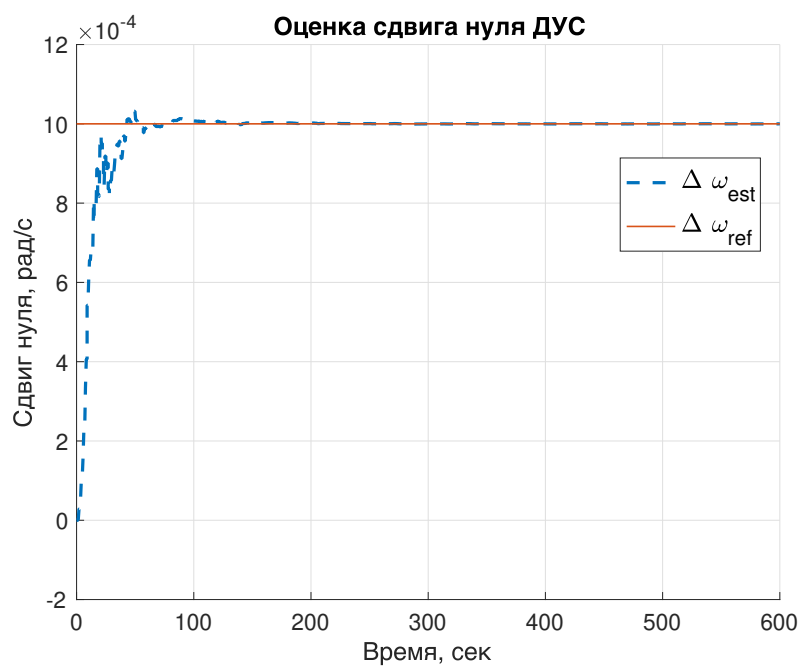


Рис. 16: Оценка ошибок датчика угловой скорости - СНС коррекция

1.4 Алгоритм комплексной системы навигации с использованием нелинейного Байесовского фильтра частиц

Для решения задач оценивания с использованием плотности вероятности распределения оценки произвольного вида (в том числе мультимодальной) и сугубо нелинейных моделей системы и измерений применяется фильтр частиц - Particle Filter (PF), который также является производным от обобщенного Байесовского фильтра.

Фильтр частиц представляет собой непараметрическую реализацию Байесовского фильтра, он аппроксимирует плотность вероятности оценки случайной величины набором частиц, представляющих случайные выборки из этой плотности.

Каждая из множества частиц в PF содержит вектор состояния, оцениваемый фильтром:

$$\chi = \mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \dots, \mathbf{x}_t^{[M]}; \quad (47)$$

где M - общее количество частиц во множестве χ .

Каждая из частиц описывает гипотезу нахождения системы в соответствующей точке пространства состояния и как следствие, чем более плотно “населена” соответствующая область в этом пространстве, тем более вероятным является и существование результирующего вектора оценки в этой области пространства состояния.

Входной информацией для этого алгоритма служит множество частиц χ , вектор измерения z и в общем случае управляющее воздействие u . В общем случае алгоритм фильтра частиц включает следующие шаги:

- Определение нового вектора состояния для соответствующей частицы на основе предыдущего вектора состояния и текущего управляющего воздействия. Новый вектор состояния для частицы определяется исходя из вероятностной модели системы $p(x_t|u_t, x_{t-1})$;
- Вычисление весового коэффициента ω для каждой из частиц с помощью текущего вектора измерений z и вероятностной модели измерения $p(z_t|x_t)$;
- Выборка нового поколения частиц - чем больше вес частицы, тем с большей вероятностью частица перейдет из текущего поколения в следующее. Выборка производится с замещением, т.е. маловероятные частицы удаляются а высоковероятные дублируются.

Из описания алгоритма видно, что фильтр оперирует непосредственно нелинейными вероятностными моделями системы $p(x_t|u_t, x_{t-1})$ и измерений $p(z_t|x_t)$, а распределение плотности вероятности оценки вектора состояния может быть произвольным.

В рамках алгоритма комплексной системы навигации вектор состояния i -й частицы имеет следующий вид:

$$\mathbf{x}_i = [x_i \quad y_i \quad \theta_i] \quad (48)$$

где x_i, y_i - координаты центра оси колесной пары в навигационной СК, θ_i - азимутальный угол разворота робота относительно навигационной СК.

Нелинейная модель системы для фильтра частиц определяется моделью кинематики трехколесного робота:

$$\mathbf{x}_{i,k} = \begin{bmatrix} \cos \theta_{i,k-1} & 0 \\ \sin \theta_{i,k-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x^b \\ \omega_z^b \end{bmatrix}, \quad (49)$$

где v_x^b, ω_z^b - скорости продольного перемещения и азимутального разворота робота, поступающие в качестве управляющих воздействий от системы траекторного управления колесного робота.

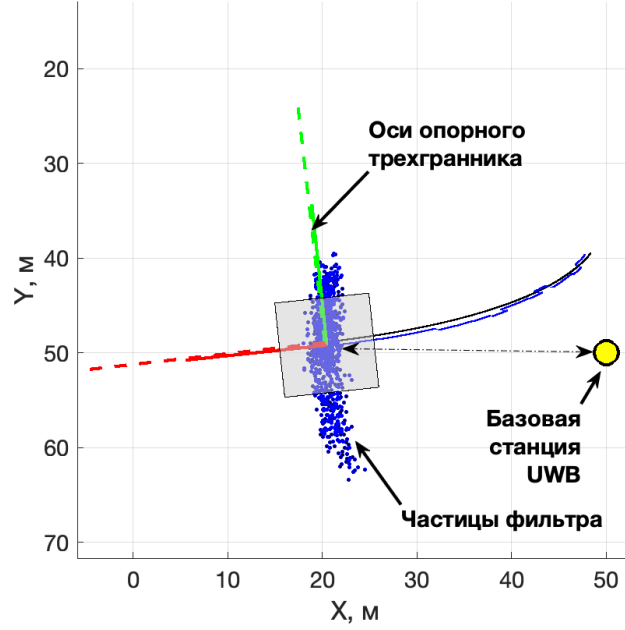


Рис. 17: Пример распределения облака частиц, описывающих плотность вероятности координат робота в навигационной СК

Нелинейная модель измерений для фильтра частиц определяется режимом функционирования локальной радионавигационной системы на базе UWB, доставляющей измерения расстояний между роботом и базовыми станциями, установленными на соответствующей площадке в рамках которой перемещается робот с установленным на нем приемником системы UWB:

$$\mathbf{z}_{i,k} = \begin{bmatrix} \sqrt{(x_{i,k} - x_{UWB,1})^2 + (y_{i,k} - y_{UWB,1})^2} \\ \vdots \\ \sqrt{(x_{i,k} - x_{UWB,n})^2 + (y_{i,k} - y_{UWB,n})^2} \end{bmatrix} \quad (50)$$

где $x_{UWB,n}$, $y_{UWB,n}$ - координаты n -ой базовой станции системы UWB в навигационной СК, n - общее количество базовых станций.

1.5 Моделирование работы алгоритма комплексной системы навигации с использованием нелинейного Байесовского фильтра частиц

Для оценки работы алгоритма было проведено моделирование в рамках которого задавалось перемещение робота, аналогичной представленной на Рис. 5. На рисунках 17 - 19 представлены результаты моделирования демонстрирующие устойчиво сходящийся процесс оценивания навигационных параметров робота.

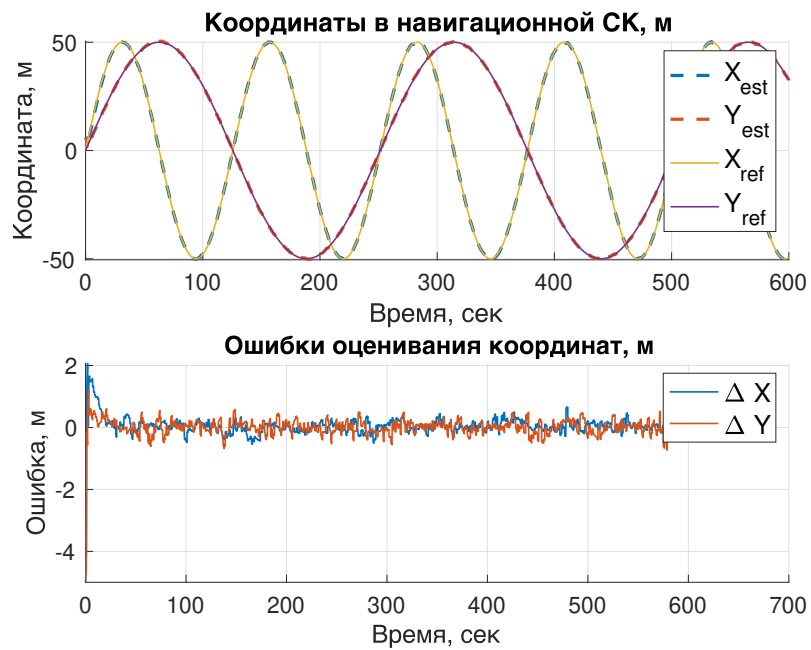


Рис. 18: Координаты робота и ошибки их оценок фильтром частиц

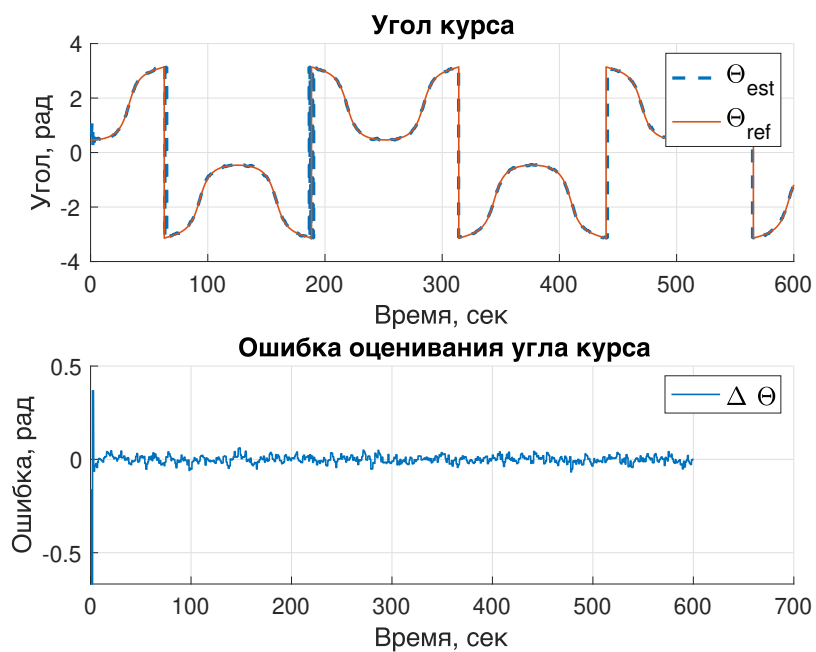


Рис. 19: Угол курса робота и ошибка его оценки фильтром частиц

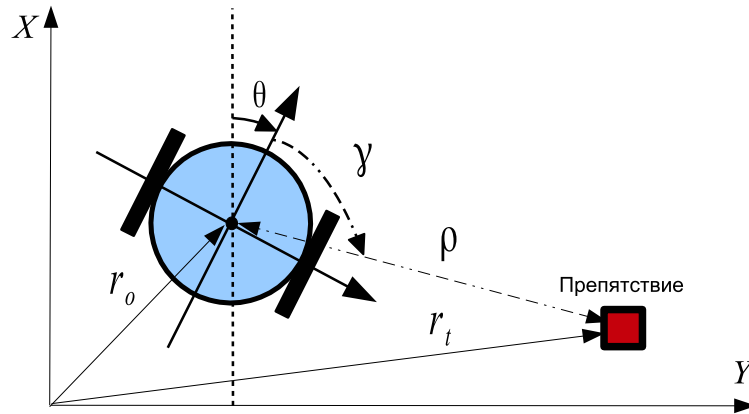


Рис. 20: Конфигурация задачи обнаружения и трекинга препятствий

2 Библиотека алгоритмов обнаружения препятствий - теория

Для построения алгоритма обнаружения препятствий сделаем несколько допущений (Рис. 20):

- Примем, что робот оснащен датчиком, доставляющим измерения угла γ пеленга (в связанной с роботом системе координат) и дальности ρ до подвижного или неподвижного препятствия;
- Координаты робота r_o и его угол θ ориентации в азимуте поступают от комплексной системы навигации;
- Известны идентификаторы препятствий - существует однозначный способ сопоставления текущих измерений пеленга и дальности конкретному препятствию.

Принимая во внимание ряд допущений, приведенных выше, будем строить алгоритм обнаружения препятствий как совокупность индивидуальных алгоритмов-трекеров, построенных на базе расширенного фильтра Калмана (ЕКФ) либо интерактивного многомодельного, расширенного фильтра Калмана (IMM-ЕКФ).

Верхний уровень архитектуры алгоритма составляет логика в рамках которой каждому наблюдаемому препятствию сопоставляется свой трекер оценивающий координаты и скорости препятствия в навигационной системе координат. При обнаружении и идентификации нового препятствия инициализируется новый трекер, после определенного количества наблюдений препятствие считается захваченным и осуществляется его сопровождение. После пропадания препятствия из вида оценка его координат и скоростей продолжает формироваться в соответствующем трекере до того момента, когда препятствие считается потерянным.

Для построения алгоритмов трекеров отдельных препятствий используем алгоритмы, имеющие в своей основе расширенный фильтр Калмана.

2.1 Трекер координат и скоростей препятствия на базе ЕКФ

Задачей трекинга препятствий является задача оценивания их координат и скоростей в навигационной СК. При этом скорости \dot{x}, \dot{y} препятствия не наблюдаются непосредственно, а

его координаты связаны с измерением пеленга и дальности нелинейным образом. Поставим задачу оценивания элементов вектора состояния \mathbf{x} :

$$\mathbf{x} = [x \quad y \quad \dot{x} \quad \dot{y}]^T \quad (51)$$

Вектор состояния \mathbf{x} соответствует кинематической модели перемещения материальной точки по плоскости с постоянной скоростью.

В непрерывной форме уравнение движения точки, перемещающейся по плоскости может быть записано так:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\boldsymbol{\nu} \quad (52)$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_{\dot{x}} & 0 \\ 0 & \nu_{\dot{y}} \end{bmatrix} \quad (53)$$

где ν_x и ν_y - гауссовские шумы ускорений.

$$E[\nu(t)] = 0 \quad (54)$$

$$E[\nu(t)\nu(\tau)] = \sigma^2\delta(t - \tau) \quad (55)$$

Для перехода к дискретной форме записи уравнений системы с периодом дискретизации Δt воспользуемся следующими соотношениями:

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + \boldsymbol{\nu}(k) \quad (56)$$

$$F = e^{A\Delta t} \quad (57)$$

$$\boldsymbol{\nu}(k) = \int_0^{\Delta t} e^{A(\Delta t - \tau)} B \boldsymbol{\nu}(k\Delta t + \tau) d\tau \quad (58)$$

$$Q = E[\boldsymbol{\nu}(k)\boldsymbol{\nu}(k)^T] \quad (59)$$

Тогда для модели (53) матрица перехода $F(k)$ и дискретная матрица ковариации $Q(k)$ в дискретной форме запишутся так:

$$F(k) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (60)$$

$$Q(k) = \begin{bmatrix} (\Delta t^3\sigma)/3 & 0 & (\Delta t^2\sigma)/2 & 0 \\ 0 & (\Delta t^3\sigma)/3 & 0 & (\Delta t^2\sigma)/2 \\ (\Delta t^2\sigma)/2 & 0 & \Delta t\sigma & 0 \\ 0 & (\Delta t^2\sigma)/2 & 0 & \Delta t\sigma \end{bmatrix} \quad (61)$$

Уравнения измерений, связывающие навигационные параметры робота с борта которого осуществляется определение препятствий и координаты препятствия в навигационной СК запишем так:

$$f^b = C_n^b(r_t^n - r_o^n) \quad (62)$$

$$\hat{\gamma} = \arctan\left(\frac{f_y^b}{f_x^b}\right) \quad (63)$$

$$\hat{\rho} = \|r_o - r_t\|_2 \quad (64)$$

Вектор измерений \mathbf{z} и матрица измерений H таковы:

$$\mathbf{z} = \begin{bmatrix} \rho - \hat{\rho} \\ \gamma - \hat{\gamma} \end{bmatrix} \quad (65)$$

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & 0 & 0 \\ H_{2,1} & H_{2,2} & 0 & 0 \end{bmatrix} \quad (66)$$

$$H_{1,1} = \frac{x_o - x_t}{\|r_o - r_t\|_2}$$

$$H_{1,2} = \frac{y_o - y_t}{\|r_o - r_t\|_2}$$

$$H_{2,1} = \frac{\partial \gamma}{\partial x_t}$$

$$H_{2,2} = \frac{\partial \gamma}{\partial y_t}$$

2.2 Трекер координат и скоростей препятствия на базе IMM-EKF

Применяемая в предыдущем разделе кинематическая модель препятствия хотя и является наиболее универсальной, все же не обеспечивает оптимального качества трекинга для неподвижных препятствий, когда нет необходимости использования доступной информации для оценивания скорости препятствия.

Для различных типов траектории перемещения препятствия оптимальными являются разные кинематические модели, к примеру при прямолинейном перемещении с постоянной скоростью предпочтительно использовать модель четвертого порядка, не включающую ускорения, равные нулю в случае равномерного перемещения, и соответственно не “тратить” информацию на их оценивание. Это приводит к улучшению качества трекинга. При маневрировании препятствия предпочтительно использовать кинематическую модель шестого порядка с ускорениями, при координированном развороте (маневр колесного транспортного средства на плоскости) - модель координированного разворота с вектором состояния пятого порядка, включающим координаты, скорости и угловую скорость разворота в азимуте.

Для решения задачи адаптивного выбора применяется алгоритм IMM EKF (Interactive Multiple Model Extended Kalman Filter). В таком алгоритме оценка вектора состояния на каждом шаге рассчитывается для каждой из n моделей соответствующим EKF фильтром, использующим свою комбинацию оценок векторов состояний, полученных на предыдущем шаге работы алгоритма. Вход для фильтра, соответствующего модели j формируется путем взаимодействия n фильтров, которое заключается в смешивании оценок $\hat{x}^i(k-1|k-1)$ с весовыми коэффициентами (вероятностями) $\mu^{i|j}(k-1|k-1)$, называемыми вероятностями смешивания.

Алгоритм IMM EKF состоит из трех основных этапов, выполняемых на каждом шаге k его работы:

- Взаимодействие моделей.

На этом этапе рассчитываются вероятности смешивания $\mu_k^{i|j}$ для каждой пары моделей M^i и M^j из дискретного множества моделей $M = \{M^1, \dots, M^n\}$:

$$\bar{c}_j = \sum_{i=1}^n p_{ij} \mu_{k-1}^i \quad (67)$$

$$\mu_k^{i|j} = \frac{1}{\bar{c}_j} p_{ij} \mu_{k-1}^i \quad (68)$$

где $\mu_k^{i|j}$ - вероятность модели M^i на шаге k , \bar{c}_j - нормализующий фактор.

Затем рассчитываются смешанные входные величины (векторы состояния и матрицы ковариации) для каждого из фильтров по следующим соотношениям:

$$x_{k-1}^{0j} = \sum_{i=1}^n \mu_k^{i|j} x_{k-1}^i \quad (69)$$

$$P_{k-1}^{0j} = \sum_{i=1}^n \mu_k^{i|j} \left\{ P_{k-1}^i + [x_{k-1}^i - x_{k-1}^{0j}] [x_{k-1}^i - x_{k-1}^{0j}]^T \right\} \quad (70)$$

где m_{k-1}^i и P_{k-1}^i - оценки соответственно вектора состояния и матрицы ковариации i -ой модели на шаге работы $k-1$.

- Фильтрация.

На этом этапе сначала выполняется собственно фильтрация каждым из фильтров, входящих в состав алгоритма:

$$[x_k^{-,i}, P_k^{-,i}] = EK F_{predict} \left(x_{k-1}^{0j}, P_{k-1}^{0j}, F_{k-1}^i, Q_{k-1}^i \right) \quad (71)$$

$$[x_k^{+,i}, P_k^{+,i}] = EK F_{update} \left(x_k^{-,i}, P_k^{-,i}, z_k, H_k^i, R_k^i \right) \quad (72)$$

Также вычисляются вероятности измерений для каждого из фильтров:

$$\Lambda_k^i = N(v_k^i, 0, S_k^i) \quad (73)$$

где v_k^i - обновляющая последовательность, S_k^i - матрица ковариации обновляющей последовательности.

Вероятность для каждой из моделей M^i на шаге k рассчитывается как:

$$c = \sum_{i=1}^n \Lambda_k^i \bar{c}_i \quad (74)$$

$$\mu_k^i = \frac{1}{c} \Lambda_k^i \bar{c}_i \quad (75)$$

- Смешивание оценок. На последнем этапе работы алгоритма вычисляется комбинированная оценка вектора состояния \hat{x}_k и матрицы ковариации P_k :

$$\hat{x}_k = \sum_{i=1}^n \mu_k^i x_k^i \quad (76)$$

$$P_k = \sum_{i=1}^n \mu_k^i \left\{ P_k^i + [x_k^i - \hat{x}_k] [x_k^i - \hat{x}_k]^T \right\} \quad (77)$$

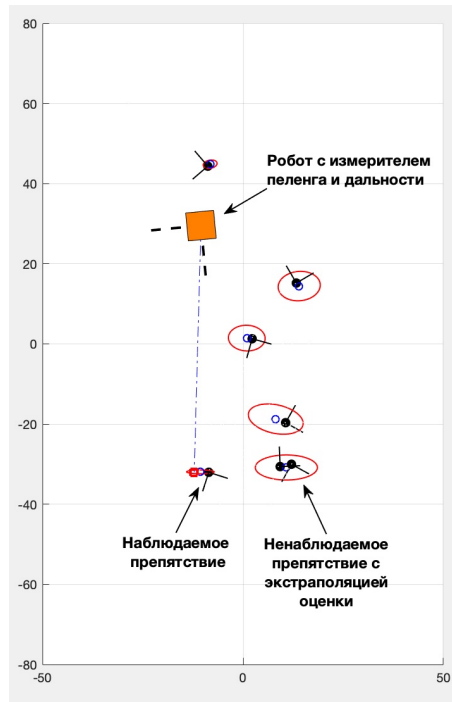


Рис. 21: Модельный сценарий задачи обнаружения и трекинга препятствий

В состав набора моделей IMM-EKF для трекинга препятствий включим две модели для подвижного и неподвижного препятствия:

- Модель четвертого порядка с вектором состояния $\mathbf{x}_1 = [x \ y \ \dot{x} \ \dot{y}]$;
- Модель второго порядка с вектором состояния $\mathbf{x}_2 = [x \ y]$.

2.3 Моделирование работы алгоритма обнаружения препятствий

Сценарий моделирования имеет следующие параметры (Рис. 21):

- Робот-носитель с измерителем пеленга и дальности перемещается по замкнутой траектории;
- Измеритель пеленга и дальности имеет угол обзора 180 градусов вперед, дальность обнаружения 100 метров.
- В области функционирования робота вначале перемещаются, затем становятся стационарными 7 препятствий;

На Рис. 22 приведены результаты определения алгоритмом координат одного из семи препятствий, различимы момента захвата и трекинга препятствия, экстраполяции оценки его координат на начальном этапе пропадания препятствия из поля зрения измерителя.

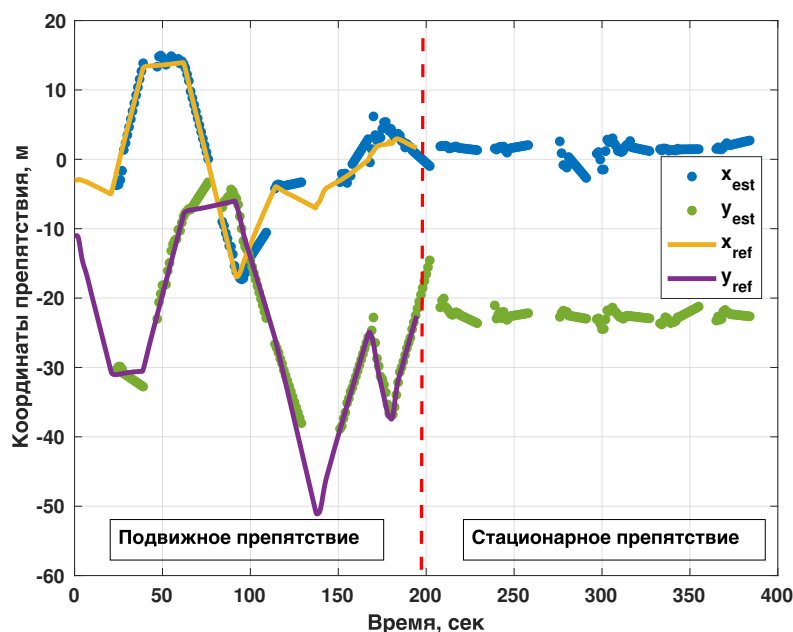


Рис. 22: Координаты препятствия и их оценка алгоритмом обнаружения препятствий

3 Библиотека алгоритмов навигации - программные алгоритмы

Библиотека алгоритмов навигации разработана на языках программирования Matlab и Python.

3.1 Алгоритмы ориентации и навигации на базе расширенного фильтра Калмана

- Алгоритм гироскопирования - Степан
- Алгоритм курсовертикали.

Реализует алгоритм курсовертикали на базе использования измерений от атчиков угловой скорости, ускорения и магнитного поля.

```
[Cbn, P, bw] = ahrs_dcm(Cbn, P, bw, dwb, fb, mb, fn, mn, dt)
```

Входные аргументы: Cbn - матрица направляющих косинусов; P - матрица ковариации фильтра Калмана; bw - вектор оценок сдвигов нулей ДУС; dwb - вектор измерений ДУС; fb - вектор измерений акселерометра; mb - вектор измерений магнитного поля; fn - вектор ускорения свободного падения в навигационной СК; mn - вектор магнитного поля в навигационной СК; dt - шаг интегрирования.

Выходные аргументы: Cbn - матрица направляющих косинусов; P - матрица ковариации фильтра Калмана bw - вектор оценок сдвигов нулей ДУС;

- Алгоритм комплексной системы навигации с коррекцией от СНС.

Реализует процесс оценивания координат, скорости, ошибок датчиков первичной информации для комплексной системы навигации колесного робота, оснащенного одометрами, курсовым гироскопом, приемником СНС.

```
[rn, Cbn, Vb_odometer, Wb_odometer, P, dw, dlr, dll] =
navigation_system_GPS
(
    rn, Cbn, P, dt, dThe, fir, fil, lr, ll, d,
    pos_gps, vel_gps, lb, gps_update, dw, dlr, dll
)
```

Входные аргументы: rn - оценка координат робота; Cbn - оценка матрицы направляющих косинусов ориентации робота; P - матрица ковариации фильтра Калмана; dt - шаг интегрирования; dThe - измерение приращения угла курса гироскопом за шаг интегрирования; fir - скорость вращения правого колеса; fil - скорость вращения левого колеса; lr - длина окружности правого колеса; ll - длина окружности левого колеса; d - длина оси колесной пары; pos_gps - измерение координат приемником GPS, спроецированное в локальную плоскую СК; vel_gps - измерение скоростей приемником GPS, спроецированное в локальную плоскую СК; lb - вектор плеча установки приемника GPS относительно центра колесной пары; gps update - индикатор доступности измерения от приемника GPS; dw - оценка сдвига нуля курсового гироскопа; dlr - оценка ошибки длины окружности правого колеса; dll - оценка ошибки длины окружности левого колеса.

Выходные аргументы: rn - оценка координат робота; Cbn - оценка матрицы направляющих косинусов ориентации робота; Vb odometer - оценки скорости продольного перемещения одометром; Wb odometer - оценка скорости курсового разворота одометром; P - матрица ковариации фильтра Калмана; dw - оценка сдвига нуля курсового гироскопа; dlr - оценка ошибки длины окружности правого колеса; dll - оценка ошибки длины окружности левого колеса.

- Алгоритм комплексной системы навигации с коррекцией от UWB.

Реализует процесс оценивания координат, скорости, ошибок датчиков первичной информации для комплексной системы навигации колесного робота, оснащенного одометрами, курсовым гироскопом, приемником локальной радионавигационной системы UWB.

```
[rn, Cbn, Vb_odometer, Wb_odometer, P, dlu, dw, dlr, dll, dd] =
navigation_system_UWB
(
    rn, Cbn, P, dt, dThe, fir, fil, lr, ll, d,
    Ranges, Anchors, lu, uwb_update, dlu, dw, dlr, dll, dd
)
```

Входные аргументы: rn - оценка координат робота; Cbn - оценка матрицы направляющих косинусов ориентации робота; P - матрица ковариации фильтра Калмана; dt - шаг интегрирования; dThe - измерение приращения угла курса гироскопом за шаг интегрирования; fir - скорость вращения правого колеса; fil - скорость вращения левого

колеса; l_r - длина окружности правого колеса; l_l - длина окружности левого колеса; d - длина оси колесной пары; $Ranges$ - измерения расстояний до базовых станций системы UWB; $anchors$ - координаты базовых станций системы UWB; lu - вектор плеча установки приемника UWB относительно центра колесной пары; uwb_update - индикатор доступности измерения от приемника UWB; d_{lu} - оценка ошибки определения плеча установки приемника UWB; d_w - оценка сдвига нуля курсового гироскопа; d_{lr} - оценка ошибки длины окружности правого колеса; d_{ll} - оценка ошибки длины окружности левого колеса; d_d - оценка ошибки длины оси колесной пары.

Выходные аргументы: gn - оценка координат робота; C_{bn} - оценка матрицы направляющих косинусов ориентации робота; V_b odometer - оценки скорости продольного перемещения одометром; W_b odometer - оценка скорости курсового разворота одометром; P - матрица ковариации фильтра Калмана; d_{lu} - оценка ошибки определения плеча установки приемника UWB; d_w - оценка сдвига нуля курсового гироскопа; d_{lr} - оценка ошибки длины окружности правого колеса; d_{ll} - оценка ошибки длины окружности левого колеса; d_d - оценка ошибки длины оси колесной пары.

3.2 Алгоритм навигации на базе фильтра частиц

- Алгоритм локальной радионавигационной системы на базе UWB.

Реализует процесс оценивания координат и угла курса колесного робота с использованием информации об управляющих воздействиях, поступающих от траекторной системы управления а также с использованием измерений дальностей до локально установленной радиосистемы UWB.

```
[fltr, robot_state] =
radio_navigation_system_UWB
(
fltr, epsilon, dtheta, Ranges, Anchors, update, dt
)
```

Входные аргументы: $fltr$ - структура фильтра частиц; ϵ - управляющее воздействие робота - скорость продольного перемещения; $d\theta$ - управляющее воздействие робота - угловая скорость курсового разворота; $Ranges$ - измерения расстояний до базовых станций системы UWB; $Anchors$ - координаты базовых станций системы UWB; $update$ - индикатор доступности измерения от приемника UWB; dt - шаг интегрирования.

Выходные аргументы: $fltr$ - структура фильтра частиц; $robot_state$ - оценка вектора состояния робота.

4 Библиотека алгоритмов определения препятствий - программные алгоритмы

Библиотека алгоритмов определения препятствий разработана на языке программирования Matlab.

- Алгоритм оценивания координат и скоростей препятствий с изменяющейся динамикой движения.

Реализует процесс оценивания координат и скоростей множества препятствий, дальность и угол пеленга которых определяется измерителем, установленным на подвижном носителе-роботе. Включает в себя базовую логику трекинга, обеспечивающую инициализацию и управление этим процессом за счет использования множества отдельных трекеров на базе алгоритмов EKF и IMM-EKF.

```
detector = basic_obstacle_detector
(
    detector, rn_platform, Cbn_platform,
    targets_range, targets_azimuth, targets_visible
)
```

Входные аргументы: detector - структура детектора препятствий с трекерами на базе EKF и IMM; rn platform - координаты платформы; Cbn platform - углы ориентации платформы; targets range - дальности до препятствий; targets azimuth - углы пеленга препятствий; targets visible - индикаторы видимости препятствий.

Выходные аргументы: detector - структура детектора препятствий с трекерами на базе EKF и IMM.

- Алгоритм трекера на базе EKF.

Реализует процесс оценивания координат и скоростей одного препятствия. Оптимален для подвижного препятствия, движущегося с постоянной скоростью.

```
fltr = ekf_tracker
(
    fltr, rn_platform, Cbn_platform, range, azimuth, update
)
```

Входные аргументы: fltr - структура EKF фильтра; rn platform - координаты платформы; Cbn platform - углы ориентации платформы; range - дальность до препятствия; azimuth - угол пеленга препятствия; update - индикатор включения режима коррекции;

Выходные аргументы: fltr - структура EKF фильтра.

- Алгоритм трекера на базе IMM-EKF.

Реализует процесс оценивания координат и скоростей одного препятствия. Оптимален для препятствия с изменяющейся динамикой движения, стационарно-нестационарного.

```
fltr = ekf_tracker
(
    fltr, rn_platform, Cbn_platform, range, azimuth, update
)
```

Входные аргументы: fltr - структура IMM фильтра; rn platform - координаты платформы; Cbn platform - углы ориентации платформы; range - дальность до препятствия; azimuth - угол пеленга препятствия; update - индикатор включения режима коррекции;

Выходные аргументы: fltr - структура IMM фильтра.