



Оценка погрешности установочных углов MRU и USBL по положению цели на дне моря

Оценка погрешности установочных углов
MRU и USBL по положению цели на дне
моря (функция gNavTaitLever)

Rev. 1.3

Содержание

Введение.....	4
1 Определение систем координат и операций с матрицами вращения	5
1.1 Системы координат	5
1.2 Элементы матрицы трансформации.....	6
1.3 Матрицы вращения для трех координатных осей	7
1.4 Модель манипулятора (последовательность вращений для шарниров).....	8
1.5 Функции MatLab	9
1.6 Резюме к разделу 1.....	10
2 Модель-1: установочные углы USBL, курс и качка	12
2.1 Описание Модели-1	12
2.2 Ошибка установочных углов USBL	13
2.3 Ошибка установочных углов USBL и курс-качка судна.....	14
2.4 Координаты метacentра в GCS	15
2.5 Определение ошибки установочных углов USBL при фиксации координат цели для двух расположений судна	16
2.6 Определение ошибки установочных углов USBL при фиксации координат цели для двух расположений судна при Heading, Pitch, Roll не равных нулю	19
2.7 Суммирование массивов невязок для установочных углов USBL при трех расположениях судна.....	20
2.8 Резюме к разделу 2.....	23
3 Модель-2: установочные углы USBL и MRU, курс и качка	25
3.1 Описание Модели-2	25
3.2 Расширенное описание Модели-2	27
3.2.1 Пересчет плеча USBL из VCS в ECS_MRU-META	27
3.2.2 Пересчет координат цели из ECS_USBL в направление ECS_MRU через VCS.....	28
3.2.3 Код MatLab к Модели-2	28
3.2.4 Влияние ошибки в установочных углах MRU	29
3.2.5 Влияние ошибки в длине плеча LeverUSBL	30
3.2.6 Особый случай Модели-2: ошибки установочных углов MRU и USBL равны	32
3.3 Определение графа преобразования координат.....	33
3.4 Резюме к разделу 3.....	35
4 Заключение	37
Список литературы	38
Приложение 1. Текст функции gNavTaitLever	39
Приложение 2. Скрипт для вычислений по Модели-2	40

История изменений

Версия	Дата	Описание изменений
1.0	2021-05-21	Документ создан // Иван Д.
1.1	2021-05-30	Внесены изменения в Модель-2, откорректирован текст // Иван Д.
1.2	2021-07-06	Дополнен текст по графу преобразования координат; добавлено Приложение 2 // Иван Д.
1.3	2024-07-13	Откорректирован текст // Иван Д.

Список рисунков

Figure 1.1 Оси GCS	5
Figure 2.1 Модель-1, сдвиги и вращения координатных осей.....	12
Figure 2.2 Сечения массива невязок плоскостями [1 0.5 0.8] при ошибках в установочных углах USBL равных [1 0.5 0.8].....	17
Figure 2.3 Отображение большой оси эллипсоида невязок.....	18
Figure 2.4 Сечения массивов невязок плоскостями [1 0.5 0.8] при ошибках в установочных углах USBL равных [1 0.5 0.8].....	20
Figure 2.5 Сечения массива невязок плоскостями [1 0.5 0.8] при расположениях USBL sh1&sh2 ..	21
Figure 2.6 Сечения массива невязок плоскостями [1 0.5 0.8] при расположениях USBL sh1&sh3 ..	21
Figure 2.7 Сечения массива невязок плоскостями [1 0.5 0.8], при расположениях USBL sh2&sh3 ..	22
Figure 2.8 Сечения суммарного массива невязок плоскостями [1 0.5 0.8].....	22
Figure 2.9 Плоскости сечения суммарного массива невязок при dRu=0.8, dPu=0.5, dHu=1	22
Figure 3.1 Модель-2, сдвиги и вращения координатных осей.....	26
Figure 3.2 Сечения массивов невязок плоскостями [-1.5 0 -1.8] при ошибках в установочных углах USBL равных [-0.5 0.5 -1] и MRU равных [1 0.5 0.8]; для расчетов введена ошибка в установочные углы MRU.....	30
Figure 3.3 Сечения массивов невязок плоскостями (а) [0.5 -1 -0.1], (b) [1 0.5 0.8]; введена ошибка в длину плеча USBL – 3 метра по оси X	31
Figure 3.4 Модель-2, ошибки установочных углов для MRU и USBL равны	32
Figure 3.5 Пример графа преобразования координат для Модели-2	34

Введение

В документе рассматривается использование функции `gNavTaitLever` библиотеки `ge0mllib` в контексте оценки ошибки установочных углов MRU и USBL по положению неподвижной цели на морском дне. Такой целью может выступать как маяк-ответчик USBL, так и особенность морского дна (характерная черта рельефа, объект на морском дне), фиксируемая на данных SSS или MBES, которые установлены на устройстве, позиционируемом по USBL.

В первом разделе приводится краткое описание используемого математического аппарата: вводится система координат и система наименований координатных осей для навигации судна; рассмотрены матрицы вращения (Tait-Bryan matrixes) и последовательности матриц вращения («модель манипулятора»).

Во втором разделе рассматривается модель первого приближения (Модель-1), включающая в себя:

- с установочными углами, включающими ошибки установочных углов,
- MRU, который ориентирован точно вдоль геометрических осей корпуса судна.

Также вводятся понятия: `невязок` для оценки установочных углов USBL, `массива невязок (dst)`, `эллипсоида невязок`, `суммарного массива невязок`. Приводятся примеры использования функции `gNavTaitLever`.

В третьем разделе рассматривается модель (Модель-2), включающая в себя:

- USBL с установочными углами, включающими ошибки установочных углов;
- MRU с установочными углами, включающими ошибки установочных углов;
- вынос `LeverUSBL` относительно метацентра.

Рассматривается частный случай, когда установочные углы USBL и MRU равны; приводятся алгоритм и результаты вычисления по «подбору установочных углов USBL» для компенсации ошибки установочных углов.

Все построения, описанные в данном документе, статические; то есть, они выполняются на основе элементарных геометрических построений без учета динамических характеристик, действующих в моделях объектов.

1 Определение систем координат и операций с матрицами вращения

1.1 Системы координат

Для обозначения осей координат введем систему наименований, описанную ниже.

GCS – Global Coordinate System

Оси заданы в географической системе координат (geographic frame); ось X направлена на север, ось Y – на восток, ось Z – вверх, положительные углы поворота отсчитываются по часовой стрелке. Для удобства геометрических расчетов координаты могут задаваться в метрах на прямоугольной проекции с малыми искажениями. Оси используются для выноса позиции различных «объектов» (целей, положения судна, измерений физических полей и др.).

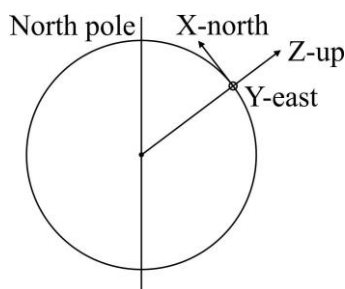


Figure 1.1 Оси GCS

Если оси заданы в направлении GCS, однако начало координат совпадает с расположением того или иного прибора или точки, то к GCS через дефис добавляется название точки положения начала координат:

GCS-META – оси заданы в направлении GCS, однако начало координат совпадает с метacentром.

GCS-MBES – начало координат совпадает с расположением акустического центра MBES;

GCS-MRU – начало координат совпадает с расположением MRU;

GCS-USBL – начало координат совпадает с расположением USBL.

VCS – Vessel's Coordinate System

Оси заданы в геометрии корпуса судна; начало координат располагается в метacentре, ось X направлена в диаметральной плоскости к носу судна, ось Y – к правому борту, ось Z – вверх, положительные углы поворота отсчитываются по часовой стрелке. Оси используются для учета величин, связанных с наклонами, смещениями и ускорениями судна.

Если оси заданы в направлении VCS, однако начало координат совпадает с расположением того или иного прибора или точки, то к VCS через дефис добавляется название точки положения начала координат: VCS-MBES; VCS-MRU; VCS-USBL. Обозначения VCS и VCS-META эквивалентны.

ECS – Equipment Coordinate System

Оси заданы для конкретного прибора в соответствии с его конструктивными особенностями, например, направлением излучающих и приемных антенн (условно примем, что направление осей и начало координат ECS нанесены на корпусе прибора). В обозначениях осей

ECS прописывается название прибора через нижнее подчеркивание: ECS_MBES, ECS_MRU, ECS_USBL; при таком обозначении предполагается, что начало координат находится в точке размещения прибора (обозначение ECS_MRU идентично ECS_MRU-MRU). Если начало координат смещено относительно прибора, то к обозначению через дефис добавляется название точки начала координат; например, если для осей с направлением «осей прибора MRU» начало координат располагается в метациентре, то данные оси обозначим как ECS_MRU-META.

Отклонение направления осей прибора (ECS) от осей корпуса судна (VCS) будем называть установочными углами.

1.2 Элементы матрицы трансформации

Для перехода между координатными осями будем использовать матрицы поворота (частный случай матриц трансформации).

В двумерном случае (ось X вертикальна; ось Y горизонтальна, поворот по часовой стрелке положителен), новые координаты точки (в «старых» координатных осях) рассчитываются, как перемножение матрицы трансформации на матрицу координат:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a1 & a2 & tx \\ b1 & b2 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Вычисления выполняются по формулам:

$$\begin{aligned} x' &= a1x + a2y + tx; \\ y' &= b1x + b2y + ty. \end{aligned}$$

Если условный объект задан множеством точек, то выполняя вычисления для каждой точки, получим следующие трансформации объекта:

- a1 – изменение масштаба по горизонтали (отражение при отрицательных значениях);
- b1 – сдвиг-наклон (shear) по вертикали (изменение направления скашивания при отрицательных значениях);
- a2 – сдвиг-наклон (shear) по горизонтали (изменение направления скашивания при отрицательных значениях);
- b2 – изменение масштаба по вертикали (отражение при отрицательных значениях);
- tx – смещение по оси X;
- ty – смещение по оси Y.

Поворот является комбинацией масштабирования и сдвига-наклона, для сохранения пропорций объекта используются синусы и косинусы:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

При использовании матрицы поворота можно говорить:

- о «повороте вектора» (направленного из начала координат в точку, координаты которой вращаются), когда рассчитываются координаты вектора, повернутого на заданный угол в «неподвижных» координатных осях;

-- о «повороте осей», когда предполагается, что вектор сохраняет свое направление, а вращаются координатные оси; соответственно после расчетов получаем координаты «неподвижного вектора» в повернутых на заданный угол осях.

Во втором случае, при подстановке в формулу, **у угла поворота необходимо поменять знак**. По-другому данный момент можно сформулировать следующим образом: положительные углы поворота отсчитываются по часовой стрелке, если предполагается, что некоторый вектор вращается относительно осей; положительные углы поворота отсчитываются против часовой стрелки, если предполагается, что оси вращаются относительно некоторого вектора.

В случае поворота в трех координатных осях, используется матрица трансформации размером 4x4:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a1 & a2 & a3 & tx \\ b1 & b2 & b3 & ty \\ c1 & c2 & c3 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Для оптимизации вычислений компьютера, вместо «матриц трансформации 4x4» будем использовать «матрицы вращения для трех координатных осей» размером 3x3. Смещение начала координат на величины [tx ty tz], будем выполнять отдельной операцией «переноса начала координат».

1.3 Матрицы вращения для трех координатных осей

Вращение будем осуществлять в определенной ранее системе координат: ось X направлена на север, ось Y – на восток, ось Z – вверх; положительные углы поворота отсчитываются по часовой стрелке. К данной конфигурации координатных осей будем приводить показания всех приборов, пересчитывая, при необходимости, показания приборов, полученные в их собственных координатных осях, отличающихся от описанных.

Примем следующий порядок вращений (the intrinsic rotations are known as: yaw, pitch and roll): (а) поворот осей относительно Z; (б) поворот осей относительно уже повернутой Y; (в) поворот осей относительно уже повернутой X. Обозначим такой порядок как **«прямое вращение»** (forward). Если у нас есть значения [Heading, Pitch, Roll], то мы можем пересчитать «координаты некоторой точки на корпусе судна после поворота судна»; для этого производится вращение вектора относительно осей GCS-META (координаты точки будут рассчитаны также в осях GSC-META).

Обратный порядок поворотов обозначим как **«обратное вращение»**. При измеренных координатах объекта относительно судна в VSC (при помощи USBL), по показаниям MRU мы можем пересчитать координаты объекта в GSC-META. При этом используется «обратное вращение» (в порядке Roll, Pitch, Heading); поскольку производится вращение осей координат, то у углов вращения меняем знак (положительные углы отсчитываются против часовой стрелки).

Обратное вращение (VSC >> GCS-META) выполняется по следующей формуле:

$$R = R_z(\alpha) R_y(\beta) R_x(\gamma) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$

$$R = \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{bmatrix}$$

https://en.wikipedia.org/wiki/Rotation_matrix

В первой строке каждая из трех матриц соответствует вращению вокруг одной из координатных осей (Z1Y2X3). Причем **порядок вращения соответствует порядку умножения матриц углов на матрицу координат** (первой умножается матрица справа). В приведенной формуле вращение сначала производится по Roll (X3), потом по Pitch (Y2), потом по Heading (Z1).

Ниже приведены формулы для иного порядка поворотов вокруг координатных осей. При этом, «прямому вращению [Heading, Pitch, Roll]» соответствует порядок X1Y2Z3, а «обратному вращению [Roll, Pitch, Heading]» (reverse) порядок Z1Y2X3.

Proper Euler angles	Tait-Bryan angles
$X_1 Z_2 X_3 = \begin{bmatrix} c_2 & -c_3 s_2 & s_2 s_3 \\ c_1 s_2 & c_1 c_2 c_3 - s_1 s_3 & -c_3 s_1 - c_1 c_2 s_3 \\ s_1 s_2 & c_1 s_3 + c_2 c_3 s_1 & c_1 c_3 - c_2 s_1 s_3 \end{bmatrix}$	$X_1 Z_2 Y_3 = \begin{bmatrix} c_2 c_3 & -s_2 & c_2 s_3 \\ s_1 s_3 + c_1 c_3 s_2 & c_1 c_2 & c_1 s_2 s_3 - c_3 s_1 \\ c_3 s_1 s_2 - c_1 s_3 & c_2 s_1 & c_1 c_3 + s_1 s_2 s_3 \end{bmatrix}$
$X_1 Y_2 X_3 = \begin{bmatrix} c_2 & s_2 s_3 & c_3 s_2 \\ s_1 s_2 & c_1 c_3 - c_2 s_1 s_3 & -c_1 s_3 - c_2 c_3 s_1 \\ -c_1 s_2 & c_3 s_1 + c_1 c_2 s_3 & c_1 c_2 c_3 - s_1 s_3 \end{bmatrix}$	$X_1 Y_2 Z_3 = \begin{bmatrix} c_2 c_3 & -c_2 s_3 & s_2 \\ c_1 s_3 + c_3 s_1 s_2 & c_1 c_3 - s_1 s_2 s_3 & -c_2 s_1 \\ s_1 s_3 - c_1 c_3 s_2 & c_3 s_1 + c_1 s_2 s_3 & c_1 c_2 \end{bmatrix}$
$Y_1 X_2 Y_3 = \begin{bmatrix} c_1 c_3 - c_2 s_1 s_3 & s_1 s_2 & c_1 s_3 + c_2 c_3 s_1 \\ s_2 s_3 & c_2 & -c_3 s_2 \\ -c_3 s_1 - c_1 c_2 s_3 & c_1 s_2 & c_1 c_2 c_3 - s_1 s_3 \end{bmatrix}$	$Y_1 X_2 Z_3 = \begin{bmatrix} c_1 c_3 + s_1 s_2 s_3 & c_3 s_1 s_2 - c_1 s_3 & c_2 s_1 \\ c_2 s_3 & c_2 c_3 & -s_2 \\ c_1 s_2 s_3 - c_3 s_1 & c_1 c_3 s_2 + s_1 s_3 & c_1 c_2 \end{bmatrix}$
$Y_1 Z_2 Y_3 = \begin{bmatrix} c_1 c_2 c_3 - s_1 s_3 & -c_1 s_2 & c_3 s_1 + c_1 c_2 s_3 \\ c_3 s_2 & c_2 & s_2 s_3 \\ -c_1 s_3 - c_2 c_3 s_1 & s_1 s_2 & c_1 c_3 - c_2 s_1 s_3 \end{bmatrix}$	$Y_1 Z_2 X_3 = \begin{bmatrix} c_1 c_2 & s_1 s_3 - c_1 c_3 s_2 & c_3 s_1 + c_1 s_2 s_3 \\ s_2 & c_2 c_3 & -c_2 s_3 \\ -c_2 s_1 & c_1 s_3 + c_3 s_1 s_2 & c_1 c_3 - s_1 s_2 s_3 \end{bmatrix}$
$Z_1 Y_2 Z_3 = \begin{bmatrix} c_1 c_2 c_3 - s_1 s_3 & -c_3 s_1 - c_1 c_2 s_3 & c_1 s_2 \\ c_1 s_3 + c_2 c_3 s_1 & c_1 c_3 - c_2 s_1 s_3 & s_1 s_2 \\ -c_3 s_2 & s_2 s_3 & c_2 \end{bmatrix}$	$Z_1 Y_2 X_3 = \begin{bmatrix} c_1 c_2 & c_1 s_2 s_3 - c_3 s_1 & s_1 s_3 + c_1 c_3 s_2 \\ c_2 s_1 & c_1 c_3 + s_1 s_2 s_3 & c_3 s_1 s_2 - c_1 s_3 \\ -s_2 & c_2 s_3 & c_2 c_3 \end{bmatrix}$
$Z_1 X_2 Z_3 = \begin{bmatrix} c_1 c_3 - c_2 s_1 s_3 & -c_1 s_3 - c_2 c_3 s_1 & s_1 s_2 \\ c_3 s_1 + c_1 c_2 s_3 & c_1 c_2 c_3 - s_1 s_3 & -c_1 s_2 \\ s_2 s_3 & c_3 s_2 & c_2 \end{bmatrix}$	$Z_1 X_2 Y_3 = \begin{bmatrix} c_1 c_3 - s_1 s_2 s_3 & -c_2 s_1 & c_1 s_3 + c_3 s_1 s_2 \\ c_3 s_1 + c_1 s_2 s_3 & c_1 c_2 & s_1 s_3 - c_1 c_3 s_2 \\ -c_2 s_3 & s_2 & c_2 c_3 \end{bmatrix}$

https://en.wikipedia.org/wiki/Euler_angles#Intrinsic_rotations

1.4 Модель манипулятора (последовательность вращений для шарниров)

«Для описания вращательных и поступательных связей между соседними звеньями манипулятора Денавит и Хартенберг предложили матричный метод последовательного построения систем координат, связанных с каждым звеном кинематической цепи. Смысл представления Денавита-Хартенберга (ДХ-представление) состоит в формировании однородной матрицы преобразования, имеющей размерность 4x4 и описывающей положение системы координат каждого звена относительно системы координат предыдущего звена. Это дает возможность последовательно преобразовать координаты схвата манипулятора из системы отсчета, связанной с последним звеном, в базовую систему отсчета, являющуюся инерциальной системой координат для рассматриваемой динамической системы.» (Краснобаев Е.А.

Модель манипулятора, как последовательности шарнирных соединений (поворот координатных осей) и рычагов (перенос начала координат) является «физически наглядной», поэтому далее будем использовать аналогию с манипулятором, как в тексте, так и в рисунках (например, Figure 2.1).

В расчетах также будут использоваться последовательности матриц вращения координатных осей и сдвигов начала координат (для текущих координатных осей), подобные представлению Денавита-Хартенберга. Однако, «единой матрицы преобразования» формироваться не будет; каждый переход между «шарнирами манипулятора» будет выполняться отдельной операцией.

Для математических построений должно выполняться следующее условие: **при переносе координат между «шарнирами манипулятора», оси координатной системы одного шарнира должны быть развернуты параллельно осям координатной системы другого шарнира.**

1.5 Функции MatLab

Описание моделей будет задаваться как в виде математических формул, так и на основе программного кода MatLab. При этом будет использоваться функция [gNavTaitLever](#) (см. Приложение 1. Текст функции gNavTaitLever), описание которой приведено ниже.

[XYZ_Lev=gNavTaitLever\(Head,Pitch,Roll,Lever,RotDirect\)](#)

где

Head – вектор-строка, который содержит угол вращения в плоскости X0Y в градусах, для времен t1...tn;

Pitch – вектор-строка, который содержит угол вращения в плоскости Z0Y в градусах, для времен t1...tn;

Roll – вектор-строка, который содержит угол вращения в плоскости Y0Z в градусах, для времен t1...tn;

Lever – три строки, содержащие величину выносов в метрах [Lx1... Lxm; Ly1... Lym; Lz1... Lzm];

RotDirect – ключ направления вращения: 'frw'- «прямое» вращение; 'rev'- «обратное» вращение;

XYZ_Lev – координаты Lever после вращения; переменная содержит трехмерный массив: три строки со значениями [Lx(t1)...Lx(tn); Ly(t1)...Ly(tn); Lz(t1)...Lz(tn)] (т.е. координатами выноса после вращения для времен t1...tn); вдоль третьей размерности расположены результаты расчетов по каждому из выносов Lever_1...Lever_m.

Если значение Roll или Pitch, или Head является скаляром, то из него будет сформирован вектор-строка необходимой длины.

Вращение выносов (Lever) осуществляется в системе координат: ось X направлена на север, ось Y – на восток, ось Z – вверх; положительные углы поворота отсчитываются по часовой

стрелке. При вращении осей координат (относительно выносов Lever) у углов вращения необходимо изменить знак.

Также в библиотеке присутствуют две функции `gNavTaitNLever` и `gNavTaitLeverN`. Указанные функции формируют массивы с результатами расчета такой же размерности, как и входные массивы. Также данные функции работают с большими массивами в 100-300 раз быстрее, чем `gNavTaitLever`, но используют больше оперативной памяти. Увеличение скорости работы функций происходит за счет отказа от использования циклов в теле функций. Для массива 301x301x301, например, запуская `gNavTaitTest` (см. Приложение 2. Скрипт для вычислений по Модели-2) для расчетов с использованием `gNavTaitLever`, получаем следующее время вычислений:

```
>> gNavTaitTest;
```

```
Using for-scycle
```

```
Elapsed time is 4256.362563 seconds.
```

Запуская `gNavTaitTest` (см. Приложение 2. Скрипт для вычислений по Модели-2) для расчетов с использованием `gNavTaitNLever` и `gNavTaitLeverN`, получаем увеличение скорости в 366 раз:

```
>> gNavTaitTest;
```

```
Not using for-scycle
```

```
Elapsed time is 11.625237 seconds.
```

`gNavTaitNLever`

Позволяет задавать массивы Head, Pitch, Roll на входе. На выходе получаются массивы X, Y, Z, такого же размера, как и массивы Head, Pitch, Roll. Значения могут рассчитываться для нескольких выносов, при этом результаты для каждой из координат каждого выноса выдаются в отдельной переменной.

`gNavTaitLeverN`

Позволяет задавать массивы выносов Lx, Ly, Lz на входе. На выходе получаются массивы X, Y, Z, такого же размера, как и массивы Lx, Ly, Lz. Значения могут рассчитываться для нескольких углов поворота Heading, Pitch, Roll, при этом результаты для каждого из углов выдаются в отдельной переменной.

Функции `gNavTaitNLever` и `gNavTaitLeverN` используются при работе с большими объемами данных для уменьшения времени расчетов и избегания использования циклов в основной программе.

1.6 Резюме к разделу 1

1) Заданы **направления осей координат**, в которых будут производиться дальнейшие расчеты: ось X направлена на север, ось Y – на восток, ось Z – вверх; положительные углы поворота отсчитываются по часовой стрелке.

Считаем, что к данной конфигурации координатных осей приведены показания всех приборов.

- 2) Введена [система наименований](#) для обозначения осей координат в приложении к навигации судна.
- 3) Рассмотрены матрицы для [«прямого»](#) и [«обратного» вращения](#) в трех осях координат. Разделены понятия [«вращение вектора в осях координат»](#) и [«вращение осей координат относительно неподвижного вектора»](#).
- 4) Декларировано использование при расчетах [последовательности матриц вращения](#) координатных осей и сдвигов начала координат (для текущих координатных осей), подобно представлению Денавита-Хартенберга.
- 5) Описана [функция gNavTaitLever](#) (MatLab), которая используется далее при вычислениях и при описании математических моделей.

2 Модель-1: установочные углы USBL, курс и качка

2.1 Описание Модели-1

Для простоты предположим, что установочные углы приборов, то есть отклонение направления осей прибора (ECS) от осей корпуса судна (VCS), равны нулю. Ошибками в установочных углах прибора будем называть значения dH , dP , dR , на которые поворачиваются оси прибора (в текущих координатных осях) при последовательности вращений dH , dP , dR (если бы на корпусе прибора был установлен MRU, то он бы показал значения углов dH , dP , dR). Далее к обозначениям dH , dP , dR будем добавлять литеру, обозначающую тип прибора.

Рассмотрим модель, в которой присутствует USBL и MRU. USBL установлен с ошибкой в установочных углах (dHu , dPu , dRu). MRU установлен точно по направлению осей корпуса судна (VCS) без ошибок и показывает значения углов курса и качки (Heading, Pitch, Roll). Координаты цели tr определяются по USBL в осях ECS-USBL; необходимо рассчитать координаты цели в осях GCS-META (Figure 2.1).

Для решения задачи будем производить последовательное вращение координатных осей от ECS-USBL в сторону GCS-META так, чтобы направление предыдущих осей совпадало с направлением последующих осей (тогда координаты цели «передадутся между осями»). Будем использовать матрицы вращения 3×3 , при необходимости выполняя сдвиг начала координат, в новую точку вращения.

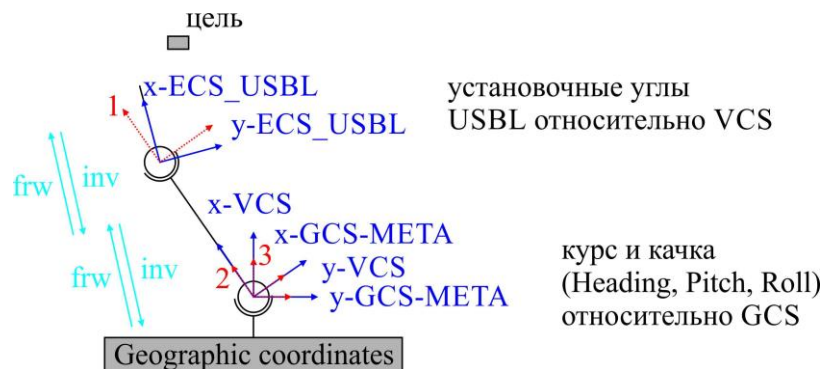


Figure 2.1 Модель-1, сдвиги и вращения координатных осей

Опишем последовательность вращений-сдвигов, показанную на Figure 2.1:

- цель расположена в координатных осях ECS_USBL (оси с ошибками в установочных углах USBL);
- осуществляем поворот осей ECS_USBL на ошибки установочных углов, получаем координаты цели в осях 1, которые расположены в направлении осей судна (более точное определение – пересчитываем координаты цели с осей ECS_USBL на оси 1);
- смещаем оси 1 (точка установки USBL) к осям VCS (метацентр судна), получаем координаты цели в осях 2;

- в результате поворота и смещения начала координат осей получаем координаты цели не в осях ECS_USBL (оси головы USBL с ошибками в установочных углах), а в осях VCS (оси корпуса судна, начало координат в метацентре);
- осуществляем поворот осей 2 на углы курса и качки (Roll, Pitch, Heading) к осям GCS-META (направление осей географических координат), получаем координаты цели в осях 3;
- в результате двух вращений и одного смещения мы пересчитали координаты цели с осей головы USBL (ECS_USBL), в оси, направление которых совпадает с направлением осей географических координат, а начало находится в метацентре судна (GCS-META).

2.2 Ошибка установочных углов USBL

При ошибке в установочных углах USBL координаты цели выдаются в «повернутых» (относительно корпуса судна) осях ECS_USBL; предположение, что направления осей USBL и корпуса судна (VCS) совпадают, приводит к «ошибке в вычислении координат цели за счет ошибки установочных углов USBL».

Пусть USBL установлен в метацентре судна (нет смещения LeverUSBL), а оси корпуса судна (VCS) расположены по направлению GCS-META. Координаты цели в GCS-META – tr=[120; 60; -80].

Если в установочных углах USBL накрутили ошибку для всех углов на +2 градуса (dHu, dPu, dRu), то оси ECS_USBL повернутся на 2 градуса относительно координатных осей VCS; USBL будет показывать координаты цели в ECS_USBL.

Определим координаты цели в осях ECS_USBL при ошибке в установочных углах в +2 градуса, используя следующий код MatLab:

```
tmp001=gNavTaitLever(-2,-2,-2,tr,'frw')
tmp001 =
    124.738496964113
    53.0998924855046
   -77.5932264644392
```

Знак углов поворота изменяется, так как производится вращение координатных осей относительно «неподвижного» заданного вектора. Поворот осей USBL производится как 'frw' (forward – Heading, Pitch, Roll), то есть «оси закручиваются».

При компенсации ошибки установочных углов производится поворот осей USBL как 'rev' (reverse – Roll, Pitch, Heading). При этом у углов поворота [-2 -2 -2] (вращение осей) изменяется знак, так как вращение производится «раскручивая оси»:

```
gNavTaitLever(2,2,2,tmp001,'rev')
ans =
    120
    60
   -80
```

Координаты цели вернулись к исходному значению $tr=[120; 60; -80]$.

2.3 Ошибка установочных углов USBL и курс-качка судна

Пусть USBL установлен в метацентре судна (нет смещения $LeverUSBL$), а оси корпуса судна (VCS) расположены по направлению GCS-META. Координаты цели в GCS-META – $tr=[120; 60; -80]$.

Пусть для судна с ошибкой в установочных углах USBL +2 градуса показания MRU (на момент фиксации координат цели) составили $[10\ 8\ 5]$ градусов (Heading, Pitch, Roll), тогда координаты цели tr можно пересчитать с осей GCS-META в оси ECS_USBL следующим образом:

-- повернем GCS-META к направлению VCS, используя углы MRU:

```
tmp002=gNavTaitLever(-10,-8,-5,tr,'frw')
```

```
tmp002 =
```

```
138.478183407984
```

```
32.7603577655034
```

```
-64.4247753515762
```

-- повернем VCS к направлению ECS_USBL, используя ошибку установочных углов USBL:

```
tr1=gNavTaitLever(-2,-2,-2,tmp002,'frw')
```

```
tr1 =
```

```
141.700536213638
```

```
25.8135117529133
```

```
-60.4534585259582
```

Для перехода от координат цели по ECS_USBL (измеренной по данным USBL) к GCS-META необходимо выполнить:

-- переход от ECS_USBL к VCS, «компенсируя» ошибку установочных углов USBL:

```
tmp002=gNavTaitLever(2,2,2,tr1,'rev')
```

```
tmp002 =
```

```
138.478183407984
```

```
32.7603577655034
```

```
-64.4247753515762
```

-- переход от VCS к направлению GCS-META, «убирая» углы курса и качки, измеренные MRU:

```
tr2=gNavTaitLever(10,8,5,tmp002,'rev')
```

```
tr2 =
```

```
120
```

```
60
```

```
-80
```

При попытке выполнить переход от координат цели по ECS_USBL к GCS-META, используя неправильный порядок ввода поворотов – сначала углы MRU, а далее углы USBL, мы получим неправильный результат:

```
tmp002=gNavTaitLever(10,8,5,tr1,'rev')
tmp002 =
    124.863584386837
    53.4789489131471
   -77.1303268320328
```

```
tr3=gNavTaitLever(2,2,2,tmp002,'rev')
tr3 =
    120.128874737449
    60.3673920212812
   -79.5288088359389
```

То есть, попытка оценки ошибки установочных углов USBL после применения текущих углов MRU некорректна (изменение порядка матриц поворота для USBL и MRU изменяет смысл действия, как и изменение порядка матриц поворота для углов Pitch, Roll, Heading).

2.4 Координаты метацентра в GCS

Пусть USBL установлен в метацентре судна (нет смещения LeverUSBL) с нулевыми ошибками установочных углов. Положение метацентра задано в координатах GCS – sh=[120; 200; 0]. Координаты цели в GCS – tr=[120; 60; -80].

При заданных условиях, до вращений необходимо пересчитать координаты цели tr=[120; 60; -80] из GCS в GCS-META:

```
tr-sh=[0; -140; -80].
```

Пусть для судна показания MRU [10 8 5] градусов (Heading, Pitch, Roll), тогда координаты цели tr в VCS (после ввода значений MRU) будут рассчитываться при помощи следующего кода MatLab:

```
tmp01=gNavTaitLever(-10,-8,-5,tr-sh,'frw')
tmp01 =
   -12.9403062986162
  -144.547923523288
   -70.2740797022897
```

Если мы захотим рассчитать «координаты цели в GCS, после поворота судна на углы MRU [10 8 5]», то необходимо прибавить к смещению «метацентр-цель» (координаты цели в VCS) значение sh2 (координаты метацентра в GCS):

```
tmp02=gNavTaitLever(-10,-8,-5,tr-sh,'frw')+sh
tmp02 =
    107.059693701384
    55.4520764767123
   -70.2740797022897
```

2.5 Определение ошибки установочных углов USBL при фиксации координат цели для двух расположений судна

Пусть USBL установлен в метацентре судна (нет смещения LeverUSBL), с ошибкой установочных углов USBL. Если мы имеем две координаты цели (tr1 и tr2), определенные в GCS при двух расположениях USBL (sh1 и sh2; подразумевается перемещение судна в GCS), то задача определения ошибки установочных углов USBL может быть поставлена следующим образом:

необходимо найти величину установочных углов USBL, при которой расстояние между «откорректированным за установочные углы USBL» положением целей (в GCS) будет минимальным.

В предельном случае, при вводе коррекции за установочные углы, положение целей должно совпасть для обоих расположений USBL (судна).

В матричном виде задача запишется следующим образом:

$$[M2_sh2]*[M1]*(tr2-sh2)+sh2=[M2_sh1]*[M1]*(tr1-sh1)+sh1$$

где

[M2_sh1] – матрица поворота, рассчитываемая по значениям Heading, Pitch, Roll для углов курса и качки, определенных по MRU (reverse, вращение осей координат), для первого расположения USBL;

[M2_sh2] – матрица поворота, рассчитываемая по значениям Heading, Pitch, Roll для углов курса и качки, определенных по MRU (reverse, вращение осей координат), для второго расположения USBL;

[M1] – матрица поворота, рассчитываемая по значениям dHu, dPu, dRu для установочных углов USBL (reverse, вращение осей координат);

sh1 – первое расположение (координаты) USBL в GCS, например [0; 0; 0];

sh2 – второе расположение USBL в GCS, например [120; 200; 0];

tr1 – координаты цели в GCS при первом расположении USBL;

tr2 – координаты цели в GCS при втором расположении USBL.

Если расписать матричное представление к скалярному виду, то мы получим систему из трех уравнений, с тремя неизвестными установочными углами. Для вычисленных значений установочных углов USBL координаты цели для первого и второго расположения USBL будут совпадать.

Примем, что при первом и втором положениях USBL углы Heading, Pitch, Roll равны нулю (судно расположено в направлении географических осей координат); тогда множители [M2_sh1] и [M2_sh2] можно принять равными единице. Для заданных условий расстояние между пересчитанным положением целей (далее – невязку) можно найти, используя следующий код MatLab:

(а) Ошибку установочных углов зададим [1 0.5 0.8] градусов и рассчитаем положение цели в осях ECS_USBL:

```
tr1=gNavTaitLever(-1,-0.5,-0.8,tr-sh1,'frw')+sh1
```

```
tr2=gNavTaitLever(-1,-0.5,-0.8,tr-sh2,'frw')+sh2
```

(b) Вычислим невязку для каждой комбинации установочных углов USBL:

```
r1=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr1-sh1,'rev')+sh1;
```

```
r2=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr2-sh2,'rev')+sh2;
```

```
dst(n)=sqrt(sum((r1-r2).^2));
```

где

r_1 – координаты цели в GCS, при первом расположении USBL и при поправках в установочные углы USBL величиной $dHu(n)$, $dPu(n)$, $dRu(n)$;

r_2 – координаты цели в GCS, при втором расположении USBL и при поправках в установочные углы USBL величиной $dHu(n)$, $dPu(n)$, $dRu(n)$;

$dst(n)$ – невязка (дистанция между «целями» в GCS, при первом и втором расположении USBL).

Выполним расчеты для множества установочных углов $dHu(n)$, $dPu(n)$, $dRu(n)$. Для этого зададим диапазон изменения для всех трех установочных углов от -3 до 3 градусов, с шагом 0.05 градуса, используя следующий код MatLab:

```
[dHu,dPu,dRu]=meshgrid(-3:.05:3,-3:.05:3,-3:.05:3);
```

Для координат $sh1=[0; 0; 0]$; $sh2=[120; 200; 0]$; $tr=[120; 60; -80]$ в результате вычислений получим трехмерный массив невязок dst . На Figure 2.2 отображены сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$; всем значениям невязки больше 10 метров присвоено NaN (прозрачный цвет).

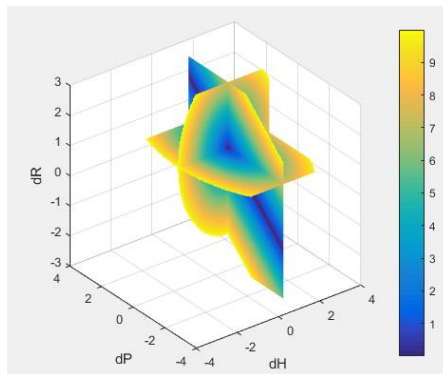


Figure 2.2 Сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$ при ошибках в установочных углах USBL равных $[1 \ 0.5 \ 0.8]$

В плоскости dRu & dPu существует неопределенность решения – множеству пар dRu & dPu соответствуют малые значения невязок в доли миллиметров, которые «выстраиваются в линию». Данная линия является большой осью «эллипсоида невязок» (объемная фигура, по форме близкая к эллипсоиду), на которой решение слабо обусловлено (Figure 2.3).

При этом точное решение, с невязкой «равной» нулю, будет единственным – $[1 \ 0.5 \ 0.8]$.

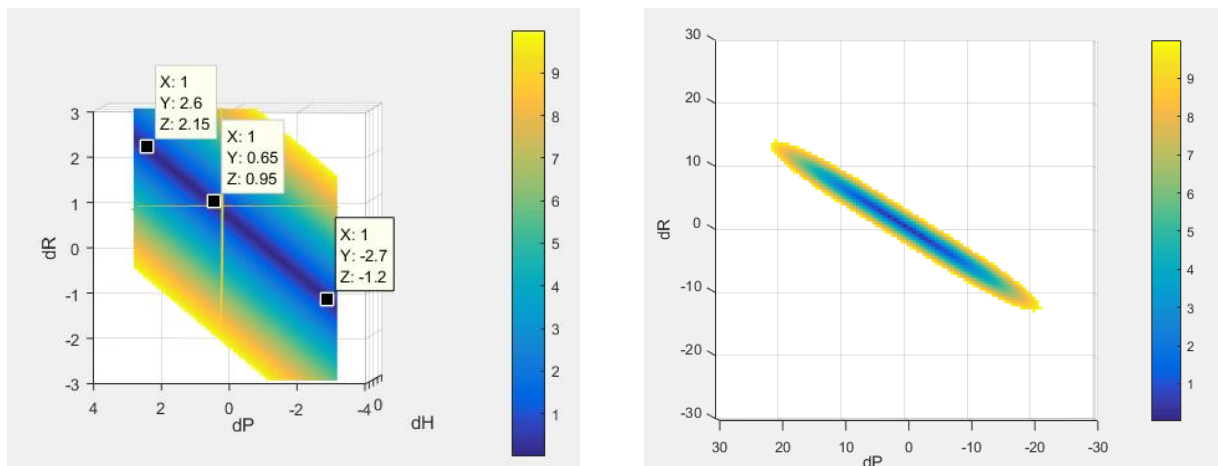


Figure 2.3 Отображение большой оси эллипсоида невязок

На качественном уровне появление большой оси эллипсоида можно объяснить следующим образом. Для двух расположений USBL (судна) в GCS, можно построить две сферы с радиусом «расстояние до USBL». Сферы будут пересекаться по окружности, причем за счет малого изменения координат судов по оси Z, окружность будет лежать в плоскости почти параллельной оси Z. Далее, в некоторых пределах мы можем корректировать dRu&dPu (ошибки в определении углов наклона USBL) так, что координаты цели будут перемещаться близко к окружности без большого отклонения. Область такого перемещения и сформирует «большую ось» на плоскости dRu&dPu.

Рассчитаем положение целей при трех «малых невязках», выбранных на плоскости dRu&dPu (показаны на Figure 2.3 маркерами).

Пересчитанные координаты цели при первом расположении USBL sh1=[0; 0; 0]	Пересчитанные координаты цели при втором расположении USBL sh2=[120; 200; 0]	Расстояние по XYZ между пересчитанными координатами целей
Коррекция невязок: dHu=0;dPu=0;dRu=0 $tr1 = gNavTaitLever(-1, -0.5, -0.8, tr-sh1, 'frw') + sh1$ $tr2 = gNavTaitLever(-1, -0.5, -0.8, tr-sh2, 'frw') + sh2$		
tr1 = 121.722382236441 56.7887438491782 -79.7414586944469	tr2 = 118.254878973433 58.9177376424443 -78.0560683142346	tr1-tr2 = 3.46750326300841 -2.12899379326615 -1.68539038021223
Коррекция невязок: dHu=1;dPu=2.6;dRu=2.15 $r1 = gNavTaitLever(1, 2.6, 2.15, tr1-sh1, 'rev') + sh1$ $r2 = gNavTaitLever(1, 2.6, 2.15, tr2-sh2, 'rev') + sh2$		
r1 = 117.018361302192 61.7919922944682 -82.9967035877334	r2 = 116.888440095016 61.8700462180799 -83.129020583442	r1-r2 = 0.129921207176096 -0.0780539236116411 0.132316995708592
Коррекция невязок: dHu=1;dPu=0.65;dRu=0.95 $r1 = gNavTaitLever(1, 0.65, 0.95, tr1'-sh1, 'rev') + sh1$ $r2 = gNavTaitLever(1, 0.65, 0.95, tr2'-sh2, 'rev') + sh2$		
r1 = 119.788296878018 60.2028040226261 -80.1647448625456	r2 = 119.782783016295 60.2062072630099 -80.3594943301005	r1-r2 = 0.00551386172379864 -0.0034032403837756 0.194749467554885
Коррекция невязок: dHu=1;dPu=-2.7;dRu=-1.2 $r1 = gNavTaitLever(1, -2.7, -1.2, tr1-sh1, 'rev') + sh1$		

$r2=gNavTaitLever(1,-2.7,-1.2,tr2-sh2,'rev')+sh2$		
$r1 =$ 124.417955281592 57.2864277532088 -75.0895305540264	$r2 =$ 124.283714217508 57.3670337868774 -75.0831987980597	$r1-r2 =$ 0.134241064084875 -0.0806060336685874 -0.00633175596675528
Коррекция невязок: $dHu=1;dPu=0.5;dRu=0.8$ $r1=gNavTaitLever(1,0.5,0.8,tr1-sh1,'rev')+sh1$ $r2=gNavTaitLever(1,0.5,0.8,tr2-sh2,'rev')+sh2$		
$r1 =$ 120 60 -80	$r2 =$ 120 59.9999999999999 -80	$r1-r2 =$ -1.4210854715202e-14 6.3948846218409e-14 -1.4210854715202e-14

Наличие «большой оси эллипсоида невязок» делает невозможным поиск точного аналитического решения системы уравнений для определения установочных углов USBL при двух расположениях судна. Такое решение будет заведомо слабо обусловлено – углы будут изменяться на десятки градусов при малой погрешности в исходных данных. Для учета «неопределенности решения» необходимо проводить анализ всех значений трехмерного массива невязок dst .

2.6 Определение ошибки установочных углов USBL при фиксации координат цели для двух расположений судна при Heading, Pitch, Roll не равных нулю

При углах Heading, Pitch, Roll не равных нулю, углы наклона и величина осей эллипсоида невязок будут изменяться в зависимости от углов курса и качки судна.

Выполним расчет для координат $sh1=[0; 0; 0]$; $sh2=[120; 200; 0]$; $tr=[120; 60; -80]$, при этом значения Heading, Pitch, Roll зададим в различных комбинациях для первого и второго расположений USBL (судна). Ошибку установочных углов зададим $[1 \ 0.5 \ 0.8]$ градусов. Для расчетов используем следующий код MatLab:

(а) Пересчитаем положение целей в оси ECS_USBL:

```
tmp1=gNavTaitLever(-Head1,-Pitch1,-Roll1,tr-sh1,'frw');
tr1=gNavTaitLever(-1,-0.5,-0.8,tmp1,'frw')+sh1;
tmp2=gNavTaitLever(-Head2,-Pitch2,-Roll2,tr-sh2,'frw');
tr2=gNavTaitLever(-1,-0.5,-0.8,tmp2,'frw')+sh2;
```

(b) Зададим диапазон изменения установочных углов от -3 до 3 градусов, с шагом 0.05 градуса:

```
[dHu,dPu,dRu]=meshgrid(-3:.05:3,-3:.05:3,-3:.05:3);
```

(c) Рассчитаем невязку для каждой комбинации установочных углов USBL:

```
tmp1=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr1-sh1,'rev');
r1=gNavTaitLever(Head1,Pitch1,Roll1,tmp1,'rev')+sh1;
tmp2=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr2-sh2,'rev');
r2=gNavTaitLever(Head2,Pitch2,Roll2,tmp2,'rev')+sh2;
dst(n)=sqrt(sum((r1-r2).^2));
```

В результате вычислений получим трехмерный массив невязок dst . На Figure 2.4 отображены сечения двух массивов невязок плоскостями $[1 \ 0.5 \ 0.8]$; всем значениям невязки

больше 10 метров присвоено NaN (прозрачный цвет). Первый массив – при всех Heading, Pitch, Roll равных нулю, показан ранее на Figure 2.2.

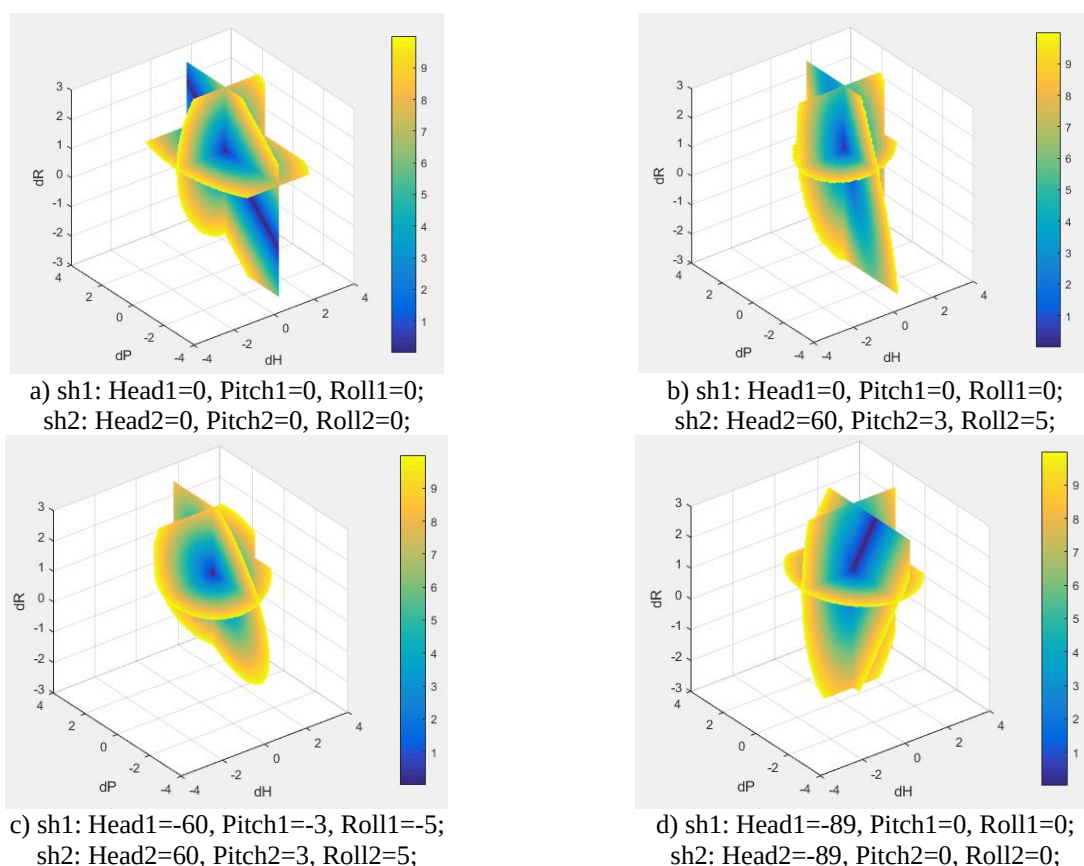


Figure 2.4 Сечения массивов невязок плоскостями [1 0.5 0.8] при ошибках в установочных углах USBL равных [1 0.5 0.8]

2.7 Суммирование массивов невязок для установочных углов USBL при трех расположениях судна

Пусть USBL установлен в метацентре судна (нет смещения LeverUSBL) с нулевыми установочными углами. Пусть заданы три расположения USBL в GCS – sh1=[0; 0; 0]; sh2=[120; 200; 0]; sh3=[200; 0; 0]; истинные координаты цели в GCS – tr=[120; 60; 80]; ошибка установочных углов USBL [1 0.5 0.8]. Примем, что при всех трех положениях USBL, **углы Heading, Pitch, Roll равны нулю** (судно расположено в направлении географических осей координат).

Рассчитаем массивы невязок dst, для комбинаций расположения USBL sh1&sh2, sh1&sh3, sh2&sh3.

Для sh1=[0; 0; 0]; sh2=[120; 200; 0] сечения массива невязок плоскостями [1 0.5 0.8] приведены на Figure 2.5.

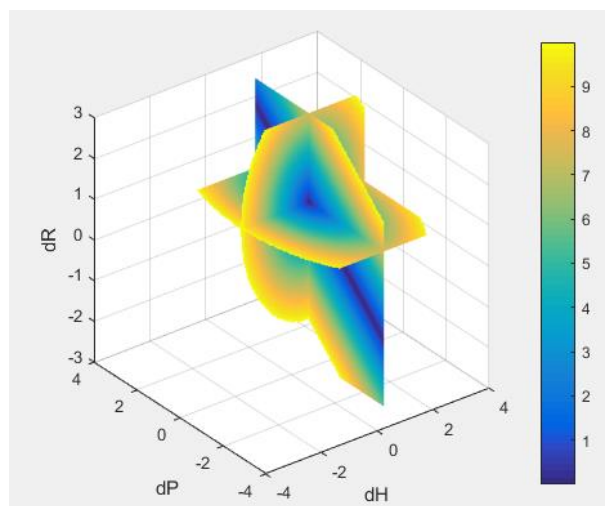


Figure 2.5 Сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$ при расположениях USBL sh1&sh2

Для $sh1=[0; 0; 0]$; $sh3=[200; 0; 0]$ сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$ приведены на Figure 2.6. При расположении судов вдоль оси X большая ось эллипса ориентируется вдоль изменения dRu (перпендикулярно оси X), что соответствует перемещению по линии соприкосновения сфер «расстояние до USBL» (см. описание в Разделе 2.5).

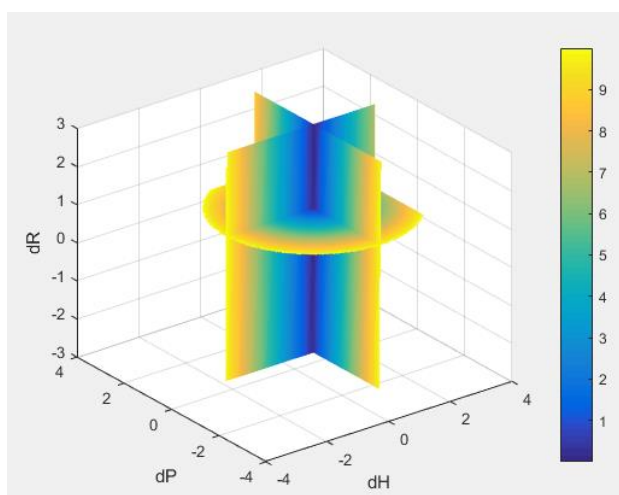


Figure 2.6 Сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$ при расположениях USBL sh1&sh3

Для $sh2=[120; 200; 0]$; $sh3=[200; 0; 0]$ сечения массива невязок плоскостями $[1 \ 0.5 \ 0.8]$ приведены на Figure 2.7.

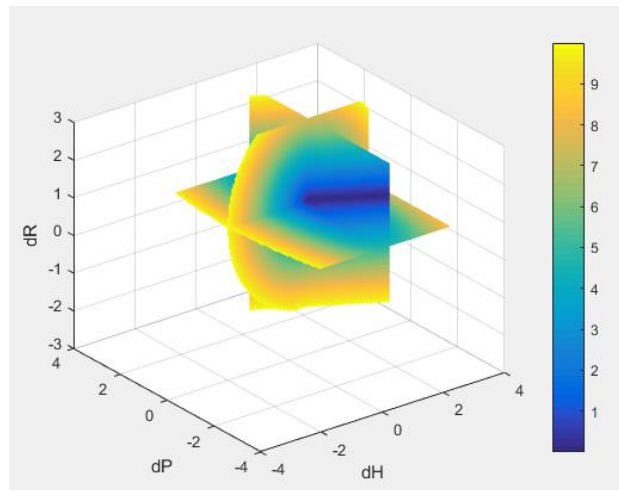


Figure 2.7 Сечения массива невязок плоскостями [1 0.5 0.8], при расположениях USBL sh2&sh3

Сложив все три массива невязок, уменьшим неопределенность нахождения минимальной невязки (Figure 2.8):

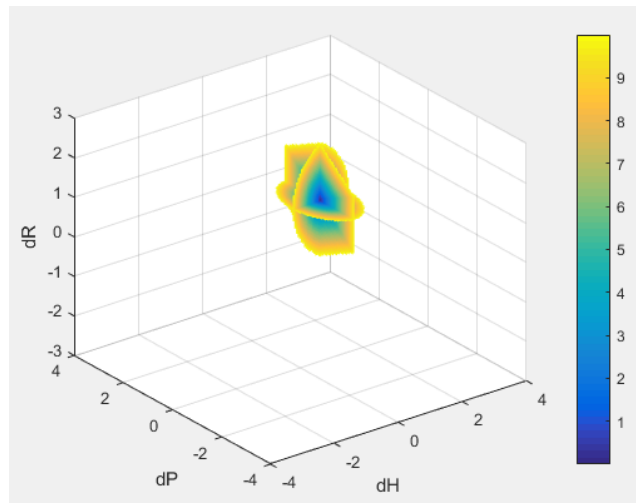


Figure 2.8 Сечения суммарного массива невязок плоскостями [1 0.5 0.8]

На Figure 2.9 показаны сечения суммарного массива невязок, в каждой из плоскостей отдельно.

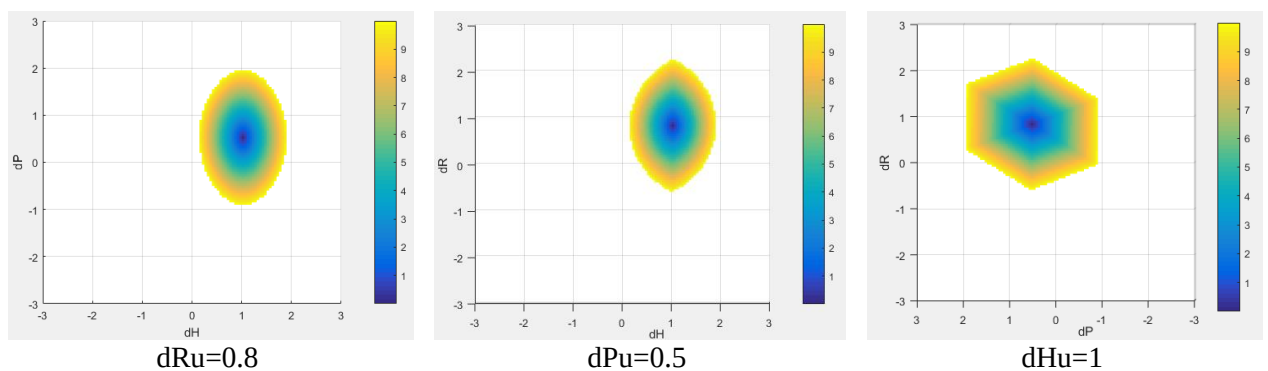


Figure 2.9 Плоскости сечения суммарного массива невязок при $dRu=0.8$, $dPu=0.5$, $dHu=1$

Возможно, в присутствии иных погрешностей, кроме установочных углов USBL (погрешность USBL при определении расстояния до цели, погрешность определения положения метacentра судна), картина распределения малых невязок может быть более сложной чем «эллипсоид невязок».

2.8 Резюме к разделу 2

1) Основной задачей раздела 2 была [оценка ошибки установочных углов USBL по координатам цели на дне](#).

2) Рассмотрена математическая модель первого приближения (Модель-1) для пересчета координат цели из ECS_USBL в GCS-META (Figure 2.1). В модели присутствуют USBL, MRU и «корпус судна» (оси VCS). Для упрощения, во всех вычислениях принималось, что USBL установлен в метacentре судна (нет плеча LeverUSBL).

В модели использованы оси: ECS_USBL, VCS, GCS-META. Координаты цели пересчитывались при последовательном повороте предыдущих осей параллельно последующим; повороты могут осуществляться как forward (Heading-Pitch-Roll) или reverse (Roll-Pitch-Heading).

Порядок поворотов осей ECS_USBL, VCS фиксирован (при изменении порядка поворотов, фактически изменяются углы вращения осей, и получается неверный результат вычислений).

3) Для определения координат цели tr в осях ECS_USBL при внесении ошибки в установочные углы в +2 градуса относительно VCS используется следующий код MatLab:

```
tmp2=gNavTaitLever(-2,-2,-2,tr,'frw')
```

Для пересчета координат цели tr с осей ECS_USBL на оси VCS при компенсации ошибки установочных углов USBL (ошибка +2 градуса) используется следующий код MatLab:

```
tmp1=gNavTaitLever(2,2,2,tr,'rev')
```

4) В качестве [меры оценки «правильности расчета» установочных углов USBL](#) введено расстояние между «расчетными координатами цели после введения поправки в установочные углы USBL»; при этом расчеты производятся для одной и той же цели на дне при двух расположениях судна (USBL).

Введено понятие [массива невязок \(dst\)](#), рассчитанного по заданному множеству установочных углов (dHu, dPu, dRu).

Показано, что для Модели-1, при оценке установочных углов USBL, мы получаем [эллипсоид невязок](#) (объемная фигура по форме близкая к эллипсоиду) на осях dHu, dPu, dRu. В центральной части эллипсоида располагаются значения с малыми невязками. При «вытянутой» форме эллипсоида малые невязки могут получаться для комбинаций установочных углов при изменении этих углов на десятки градусов (Figure 2.3).

Форма эллипсоида невязок зависит от (1) взаимного расположения цели на дне и двух положений судна (USBL) (для которых выполняется расчет невязки), (2) углов Heading, Pitch, Roll для каждого из расположений судна (Figure 2.4).

5) Введено понятие **суммарного массива невязок**, который вычисляется для нескольких расположений судна при фиксировании одной цели (возможно также суммирование при фиксировании разных целей). При суммировании невязок неопределенность нахождения минимальной невязки может уменьшаться (Figure 2.8).

3 Модель-2: установочные углы USBL и MRU, курс и качка

3.1 Описание Модели-2

Направление осей VCS при работе навигационной аппаратуры часто определяется по MRU; при этом предполагается, что направление осей ECS_MRU совпадает с VCS. Однако на практике, MRU устанавливается с «погрешностью установочных углов MRU».

Рассмотрим модель, в которой присутствует USBL и MRU. USBL установлен с ошибкой в установочных углах $[dH_u, dP_u, dR_u]$ относительно VCS или с ошибкой $[dH, dP, dR]$ относительно осей MRU. MRU установлен с ошибкой в установочных углах $[dH_m, dP_m, dR_m]$ относительно VCS. MRU показывает значения углов курса и качки (Heading, Pitch, Roll) относительно своих координатных осей (ECS_MRU). Координаты цели tr определяются по USBL в осях ECS-USBL; необходимо рассчитать координаты цели в осях GCS-META (Figure 3.1).

Ошибкой в установочных углах MRU (относительно осей корпуса судна) будем называть такие значения dH_m, dP_m, dR_m , которые покажет MRU при установке соответствующих осей корпуса судна в направлениях север и отвес. Проводя физические аналогии, можно рассмотреть, как будто показания MRU снимаются с «измерительных лимбов», установленных на корпусе MRU (в левой части лимба расположены положительные значения). При этом все дальнейшие измерения MRU будут выдаваться в «системе повернутых измерительных лимбов». Повороты MRU для компенсации ошибки в установочных углах будут производиться в последовательности dR_m, dP_m, dH_m .

Поскольку показания MRU выдаются относительно осей ECS_MRU, то удобно приводить к осям GCS-META именно оси ECS_MRU-META, а не VCS. Тогда «неправильную установку MRU» можно рассмотреть, как «неправильную установку корпуса судна относительно MRU». В этом случае установочные углы USBL (ECS_USBL) должны оцениваться относительно осей ECS_MRU, а «неправильно установленный корпус» будет влиять на выносы оборудования. Вынос LeverUSBL должен будет пересчитываться из координатных осей VCS в координатные оси ECS_MRU.

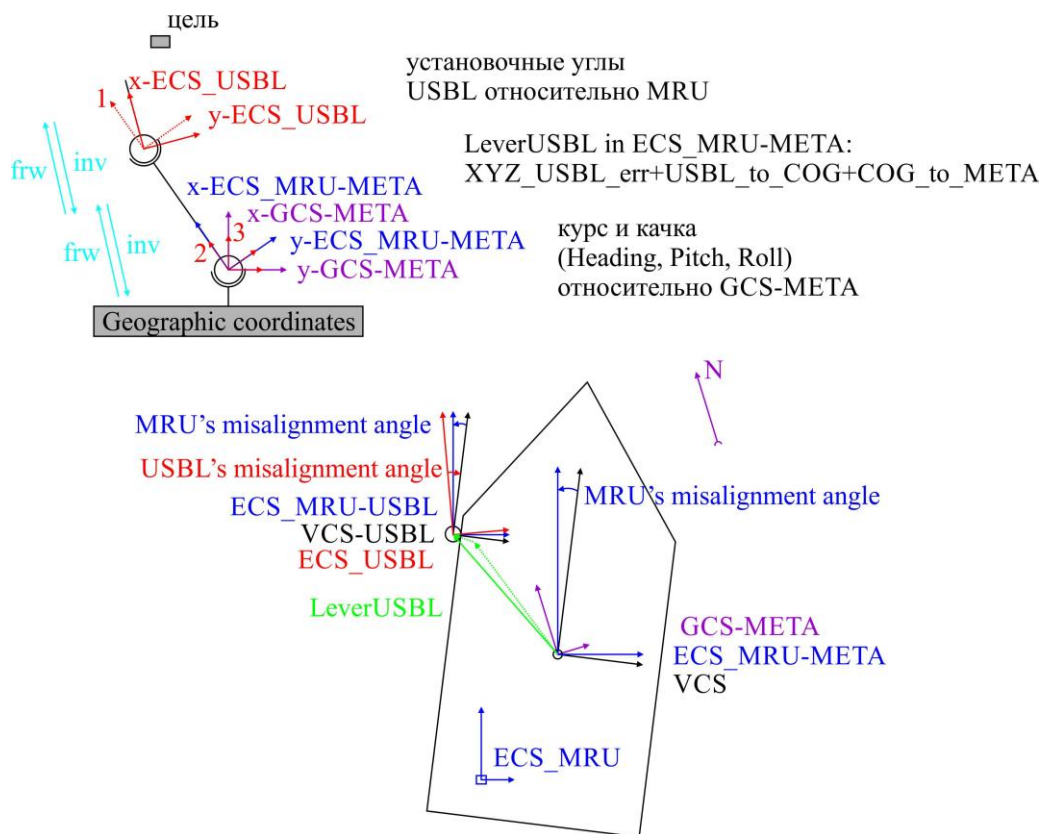


Figure 3.1 Модель-2, сдвиги и вращения координатных осей

Опишем последовательность вращений-сдвигов, показанную на Figure 3.1:

- цель расположена в координатных осях ECS_USBL (оси с ошибками в установочных углах USBL);
- осуществляем поворот осей ECS_USBL на величину ошибок установочных углов USBL относительно осей MRU, получаем оси 1 (данные оси расположены в направлении осей ECS_MRU);
- смещаем оси 1 (точка установки USBL) к осям ECS_MRU-META (метацентр судна), получаем оси 2; величина смещения должна рассчитываться в осях ECS_MRU, а не VCS (для этого необходимо знать ошибку установочных углов MRU относительно корпуса судна, чтобы рассчитать проекцию выносов на ECS_MRU);
- в результате поворота и смещения осей, получаем координаты цели не в осях ECS_USBL (голова USBL с ошибками в установочных углах), а в осях ECS_MRU-META (метацентр судна с осями по MRU);
- осуществляем поворот осей 2 на углы курса и качки (Roll, Pitch, Heading) к осям GCS-META (направление осей географических координат), получаем оси 3;
- в результате двух вращений и одного смещения мы получили координаты цели в осях, направление которых совпадает с направлением осей географических координат, а начало находится в метацентре судна.

Описанная модель реализуется как последовательное умножение матрицы координат на матрицы поворота:

$$\begin{aligned} tr_GCS-META &= [M_rev(Head,Pitch,Roll)]*[M_rev(dH,dP,dR)+LeverUSBL]*tr_ECS-USBL \\ tr_ECS-USBL &= [M_frw(-dH,-dP,-dR)]*[M_frw(-Head,-Pitch,-Roll)-LeverUSBL]*tr_GCS-META \end{aligned}$$

Смещение LeverUSBL используется для задания выноса USBL в текущих координатных осях ECS_MRU-META.

Для расчета координат цели при цепочке вращений-сдвигов, показанных на Figure 3.1, для перехода от GCS-META к ECS_USBL необходимо выполнить расчеты по следующему коду MatLab:

(а) пересчитаем координаты цели tr с осей GCS-META на оси ECS_MRU-META

```
tmp1=gNavTaitLever(-Head,-Pitch,-Roll, tr,'frw')
```

(б) сдвигаем оси из метacentра (META) в точку установки USBL

```
tmp2=tmp1-LeverUSBL
```

при этом значение LeverUSBL вводится в осях ECS_MRU-META

(с) пересчитаем координаты цели с осей ECS_MRU-USBL на оси ECS_USBL (при этом подразумевается ввод ошибки установочных углов USBL именно «относительно осей ECS_MRU-META»)

```
tr1=gNavTaitLever(-dH,-dP,-dR,tmp2,'frw')
```

где tr1 – координаты цели «по данным USBL» (на осях ECS_USBL).

Для расчета координат цели при цепочке вращений-сдвигов, показанных на Figure 3.1, для перехода от ECS_USBL к GCS-META необходимо выполнить расчеты по следующему коду MatLab:

(с)

```
tmp2=gNavTaitLever(dH,dP,dR,tr1,'rev')
```

(б)

```
tmp1=tmp2+LeverUSBL
```

(а)

```
tr2=gNavTaitLever(Head,Pitch,Roll,tmp1,'rev')
```

где tr2 – координаты цели в системе координат GCS-META.

3.2 Расширенное описание Модели-2

Для более полного описания Модели-2 необходимо рассмотреть два вопроса: (1) пересчет LeverUSBL в ECS_MRU-META; (2) поворот осей ECS_USBL в направлении ECS_MRU.

3.2.1 Пересчет плеча USBL из VCS в ECS_MRU-META

Пусть начало вектора LeverUSBL находится в метacentре, а конец – в точке с координатами установки USBL. При этом плечо LeverUSBL можно рассматривать как состоящее из следующих компонент:

-- ошибка определения точки установки USBL,

-- вынос между метacentром и точкой установки USBL,

все компоненты изначально привязываются к корпусу судна с осями VCS-META. В рамках Модели-2 не рассматривается смещение META-COG при неправильном определении метacentра судна.

Определим координаты вектора LeverUSBL, заданного в VCS, на осях ECS_MRU-META при ошибке в установочных углах в +2 градуса относительно VCS. Для этого «подвернем» оси VCS на установочные углы MRU при «неподвижном» векторе LeverUSBL, используя следующий код MatLab:

```
LeverUSBL_ECS_MRU=gNavTaitLever(-2,-2,-2,LeverUSBL_VCS,'frw')
```

или

```
LeverUSBL_ECS_MRU=gNavTaitLever(-dHm,-dPm,-dRm,LeverUSBL_VCS,'frw')
```

3.2.2 Пересчет координат цели из ECS_USBL в направление ECS_MRU через VCS

При повороте осей ECS_USBL к направлению осей ECS_MRU для углов поворота отсутствует смысловая привязка к установочным углам аппаратуры. Целесообразнее осуществлять данный поворот через направление осей корпуса судна (VCS). В этом случае мы сможем отдельно оперировать «установочными углами USBL относительно осей судна» и «установочными углами MRU относительно осей судна».

Разобьем поворот осей ECS_USBL к направлению осей ECS_MRU на два действия:

1) Пересчитаем координаты цели с осей ECS_USBL на оси VCS-USBL. При компенсации ошибки установочных углов USBL (ошибка +2 градуса) производится поворот осей ECS_USBL как 'rev' (reverse – Roll-Pitch-Heading) к направлению осей VCS:

```
tmp1=gNavTaitLever(2,2,2,tr0,'rev')
```

или

```
tmp1=gNavTaitLever(dHu,dPu,dRu,tr0,'rev')
```

2) Определим координаты цели в осях ECS_MRU при внесении ошибки в установочные углы MRU в +2 градуса относительно направления осей VCS. Для этого «подвернем» оси VCS-USBL на установочные углы MRU, используя следующий код MatLab:

```
tmp2=gNavTaitLever(-2,-2,-2,tmp1,'frw')
```

или

```
tmp2=gNavTaitLever(-dHm,-dPm,-dRm,tmp1,'frw')
```

3.2.3 Код MatLab к Модели-2

С учетом дополненного описания для пересчета координат цели из ECS_USBL в GCS-META (при цепочке вращений-сдвигов, показанных на Figure 3.1) выполним расчеты по следующему коду MatLab:

0) пересчет плеча LeverUSBL из VCS в ECS_MRU-META

```
LeverUSBL=gNavTaitLever(-dHm,-dPm,-dRm,LeverUSBL_VCS,'frw')
```

1) компенсация ошибки установочных углов USBL (поворот ECS_USBL к VCS-USBL)

```
tmp1=gNavTaitLever(dHu,dPu,dRu,tr0,'rev')
```

2) внесение ошибки в установочные углы MRU (поворот VCS-USBL к ECS_MRU)

```
tmp2=gNavTaitLever(-dHm,-dPm,-dRm,tmp1,'frw')
```

3) смещение начала координат от USBL к META в координатных осях ECS_MRU-USBL

`tmp3=tmp2+LeverUSBL`

4) учет показаний MRU (поворот ECS_MRU-META к GCS-META)

`tr02=gNavTaitLever(Head,Pitch,Roll,tmp3,'rev')`

где tr02 – координаты цели в системе координат GCS-META.

Для пересчета координат цели из GCS-META в ECS_USBL (при цепочке вращений-сдвигов, показанных на Figure 3.1) выполним расчеты по следующему коду MatLab:

0) пересчет плеча LeverUSBL из VCS в ECS_MRU-META

`LeverUSBL=gNavTaitLever(-dHm,-dPm,-dRm,LeverUSBL_VCS,'frw')`

4) учет показаний MRU (поворот GCS-META к ECS_MRU-META)

`tmp3=gNavTaitLever(-Head,-Pitch,-Roll,tr02,'frw')`

3) смещение начала координат от META к USBL, в координатных осях ECS_MRU-META

`tmp2=tmp3-LeverUSBL`

2) компенсация ошибки в установочных углах MRU (поворот ECS_MRU к VCS-USBL)

`tmp1=gNavTaitLever(dHm,dPm,dRm,tmp2,'rev')`

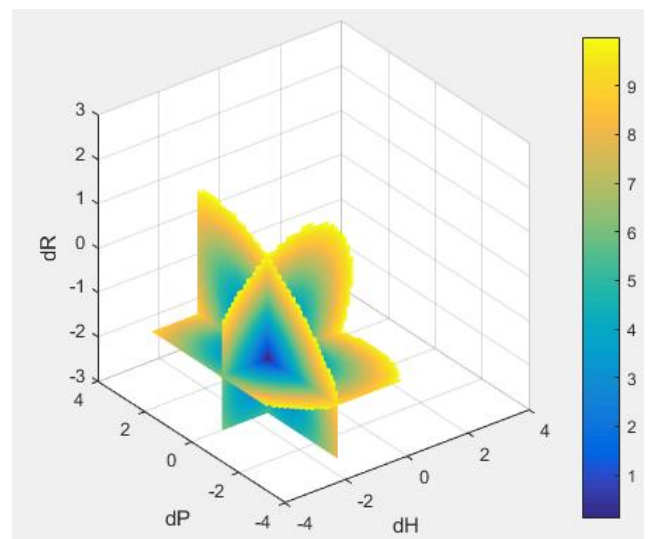
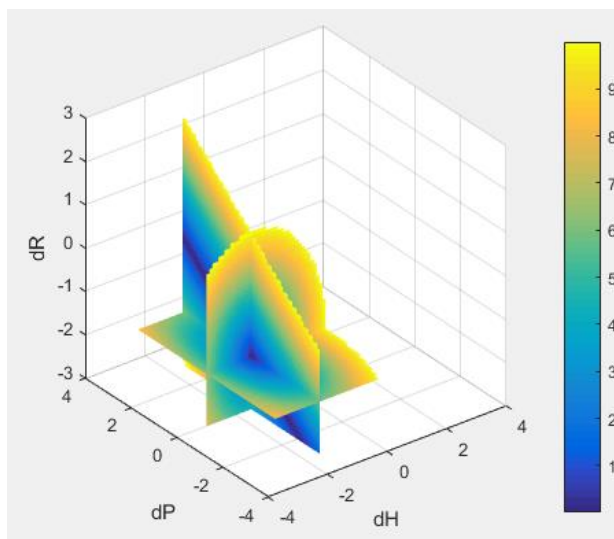
1) внесение ошибки установочных углов USBL (поворот VCS-USBL к ECS_USBL)

`tr2=gNavTaitLever(-dHu,-dPu,-dRu,tmp1,'frw')`

3.2.4 Влияние ошибки в установочных углах MRU

Выполним расчет для координат sh1=[0; 0; 0]; sh2=[120; 200; 0]; tr=[120; 60; -80], при этом значения Heading, Pitch, Roll зададим в различных комбинациях для первого и второго расположений USBL (судна). Ошибку установочных углов USBL зададим [-0.5 0.5 -1] градусов. Величину LeverUSBL_VCS зададим [0; 0; 0] метров. Ошибку установочных углов MRU зададим [1;0.5;0.8] градусов, при расчетах будем использовать «ошибочную» величину [0;0;0] градусов.

Результаты расчетов показаны на Figure 3.2. Как видно из рисунка, форма эллипсоида невязок (для комбинаций углов Heading, Pitch, Roll) приблизительно сохраняется относительно Figure 2.4. Минимальное значение невязки смещается, приблизительно в точку [-1.5 0 -1.8]=[-0.5 0.5 -1]-[1 0.5 0.8] градуса (приблизительно ошибка установочных углов USBL минус ошибка установочных углов MRU).



a) sh1: Head1=0, Pitch1=0, Roll1=0;
 sh2: Head2=0, Pitch2=0, Roll2=0;
 LeverUSBL_VCS=[0;0;0];
 for dst calculation LeverUSBL_VCS=[0;0;0];

b) sh1: Head1=0, Pitch1=-3, Roll1=-5;
 sh2: Head2=60, Pitch2=3, Roll2=5;
 LeverUSBL_VCS=[0;0;0];
 for dst calculation LeverUSBL_VCS=[0;0;0];

Figure 3.2 Сечения массивов невязок плоскостями [-1.5 0 -1.8] при ошибках в установочных углах USBL равных [-0.5 0.5 -1] и MRU равных [1 0.5 0.8]; для расчетов введена ошибка в установочные углы MRU

Приблизительное равенство является следствием малых значений углов невязок. При точном расчете (приняв Heading, Pitch, Roll равными нулю):

```
tmp1=gNavTaitLever(1,0.5,0.8,tr-sh1,'rev');
```

```
tr1=gNavTaitLever(-(-0.5),-0.5,-(-1),tmp1,'frw')+sh1
```

получим координаты цели, пересчитанные из GCS-META в ECS_USBL

```
tr1 =
```

```
118.359476949244
```

```
65.5851230982138
```

```
-78.036054772736
```

При приближенном расчете для [-1.5 0 -1.8]=[-0.5 0.5 -1]-[1 0.5 0.8], получим следующие приближенные координаты цели, пересчитанные из GCS-META в ECS_USBL:

```
gNavTaitLever(-(-1.5),-0,-(-1.8),tr-sh1,'frw')+sh1
```

```
ans =
```

```
118.388262098594
```

```
65.6023877797828
```

```
-77.977856567525
```

Разница в координатах, для условий в приведенном примере, составит сантиметры:

```
tr1-ans
```

```
ans =
```

```
-0.0287851493501989
```

```
-0.0172646815689603
```

```
-0.0581982052109566
```

3.2.5 Влияние ошибки в длине плеча LeverUSBL

Выполним расчет для координат sh1=[0; 0; 0]; sh2=[120; 200; 0]; tr=[120; 60; -80], при этом значения Heading, Pitch, Roll зададим в различных комбинациях для первого и второго расположений USBL (судна). Ошибку установочных углов USBL зададим [1 0.5 0.8] градусов; ошибку установочных углов MRU зададим [1;0.5;0.8] градусов. Исходно величину LeverUSBL_VCS зададим 3 метра; при расчетах будем использовать «ошибочную» величину плеча 0 метров.

Результаты вычислений показаны на Figure 3.3.

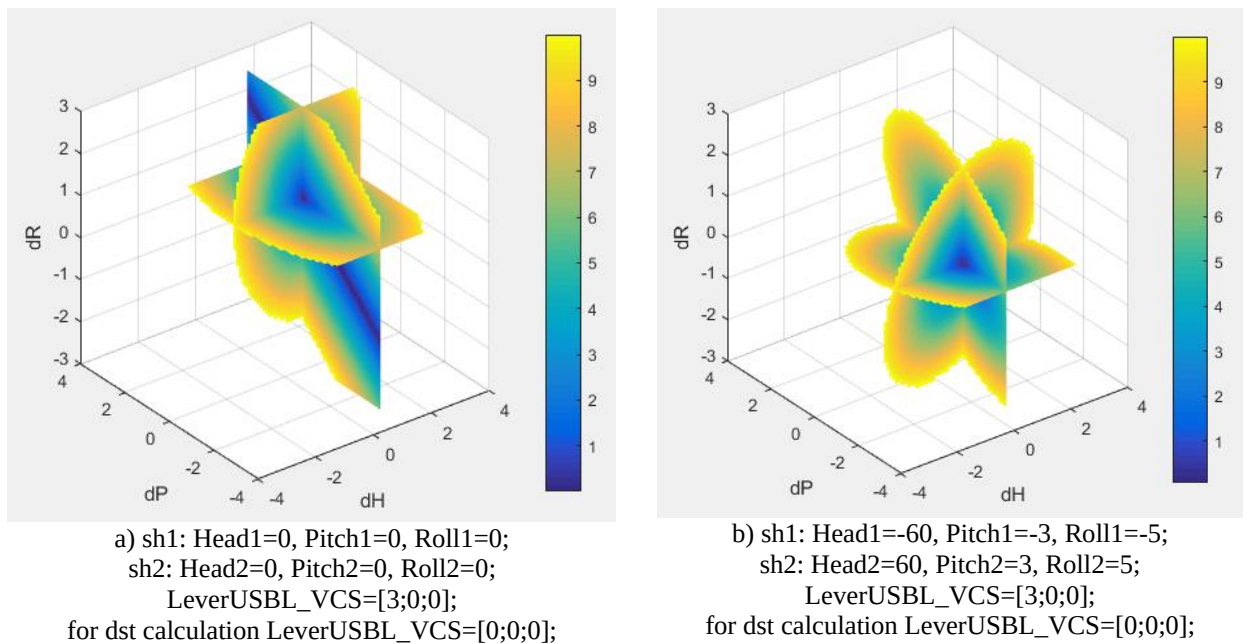


Figure 3.3 Сечения массивов невязок плоскостями (a) [0.5 -1 -0.1], (b) [1 0.5 0.8]; введена ошибка в длину плеча USBL – 3 метра по оси X

Как видно из рисунка, форма эллипсоида невязок (для комбинаций углов Heading, Pitch, Roll) приблизительно сохраняется относительно Figure 2.4.

Если суда (USBL) со-направлены, то погрешности, вносимые в координаты цели tr за счет неправильного значения LeverUSBL, одинаковы. Расчет невязки установочных углов USBL при таких условиях выполняется верно (Figure 3.3, a).

Величина погрешности (расчета невязки установочных углов USBL) возрастает при увеличении разницы в направлениях USBL при двух расположениях судна. Так, в приведенном примере, при ошибке в LeverUSBL на 3 метра ошибка определения невязки составляет [-0.5 -1.5 - 0.9] градуса.

Таким образом:

- влияние ошибки в установочных углах MRU «переносит» значение минимума в массиве невязок dst (невязок для установочных углов USBL) в новую точку постоянную для различных комбинаций Heading, Pitch, Roll,
- влияние ошибки в длине LeverUSBL перемещает минимум в массиве невязок dst для каждой комбинации Heading, Pitch, Roll.

Предположив, что установочные углы MRU направлены по корпусу судна (ошибка в установочных углах MRU равна [0; 0; 0]), найдя минимальное значение в массиве невязок dst (для установочных углов USBL) – мы получим «установочные углы USBL относительно осей корпуса судна». Фактически установочные углы USBL будут определены относительно осей ECS_MRU.

При этом ошибки в проекции LeverUSBL на ECS_MRU (поскольку фактическая ошибка в установочных углах MRU не равна нулю или не полностью компенсирована в

поправках навигационной системы) будут перемещать минимум в массиве невязок dst для каждой комбинации Heading, Pitch, Roll и создавать «разброс минимумов» в суммарном массиве невязок.

3.2.6 Особый случай Модели-2: ошибки установочных углов MRU и USBL равны

Если подобрать ошибки установочных углов USBL или MRU так, что оси ECS_USBL и ECS_MRU совпадут по направлению (Figure 3.4), то повороты на углы $[dHu, dPu, dRu]$ и $[dHm, dPm, dRm]$ (см. Код MatLab к Модели-2) скомпенсируют друг друга. После этого «ошибка расчета координат цели» будет определяться только ошибкой расчета плеча LeverUSBL.

Ошибка LeverUSBL, в свою очередь, будет определяться величиной ошибки установочных углов MRU относительно корпуса судна (VCS); на Figure 3.4 пунктиром показано положение LeverUSBL и позиция USBL при нулевом значении ошибки установочных углов MRU и USBL (при этом корпус судна поворачивается, компенсируя ошибку установочных углов MRU).

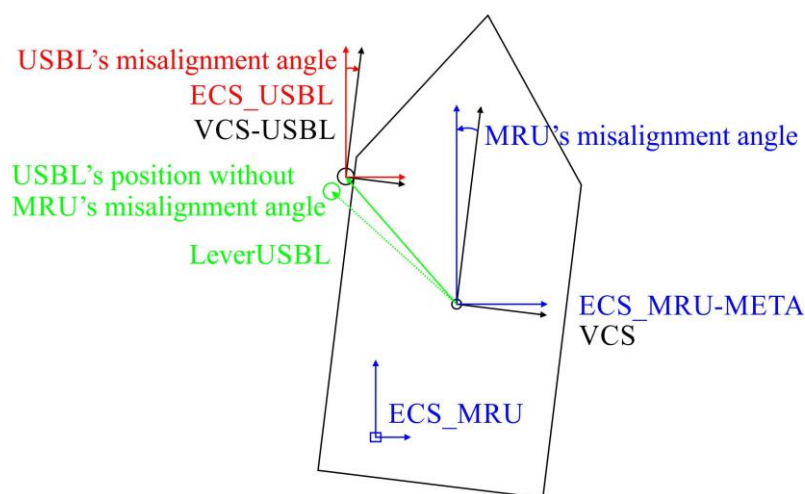


Figure 3.4 Модель-2, ошибки установочных углов для MRU и USBL равны

Таким образом, даже для ошибок установочных углов MRU, значительно отклоняющих оси от VCS, можно подобрать такие углы «компенсации установочных углов USBL», что оси ECS_USBL и ECS_MRU совпадут по направлению (Figure 3.4). Дальнейшее увеличение «точности позиционирования по USBL» коррекцией установочных углов USBL окажется невозможным (или будет работать для определенных курса, интервала глубин и расстояния до цели). Ошибка будет определяться влиянием ошибки установочных углов MRU на проекцию LeverUSBL (на оси ECS_MRU с осями VCS), то есть, дальнейшее увеличение «точности позиционирования по USBL» потребует коррекции углов MRU.

Величина невязки, рассчитываемой для двух положений судна, будет зависеть от комбинации углов Heading, Pitch, Roll; невязка будет уменьшаться для одинакового направления судов (осей USBL) и увеличиваться до максимума при встречных направлениях (осей USBL).

3.3 Определение графа преобразования координат

На основе Модели-2 возможно построение более полной модели навигации судна. Для этого необходимы следующие действия:

- ввести в модель Heave-sensor;
- ввести в модель антенны GPS, чтобы относительно них рассчитывать географические координаты метacentра в GCS (для двух антенн можно рассчитывать Heading, а также другие углы наклона при работе в RTK);
- ввести в модель отдельные понятия META и COG; где META – метacentр, относительно которого происходит фактическое вращение судна; COG – точка относительно которой рассчитываются выносы оборудования и которая принимается за метacentр;
- ввести в модель расчет положения буксируемых устройств по длине вытравленного кабеля, датчикам глубины буксируемого устройства и иным данным, не относящимся к гидроакустическому позиционированию.

При расширении Модели-2 не все взаимосвязи между компонентами модели («звенья манипулятора») будут описываться матрицами поворота и смещением начала координат. Например, буксируемое устройство в первом приближении можно представить как «груз на жестком стержне», однако последующие приближения потребуют иной математической модели. В этом случае, от «схемы манипулятора» целесообразно перейти к **«графу преобразования координат»**; при этом обработка будет осуществляться по следующим правилам:

- каждому узлу графа будет соответствовать «процедура преобразования» (в частном случае это будет матрица поворота координатных осей или смещение начала координат);
- для каждого узла графа должна быть предусмотрена «прямая» процедура преобразования и «обратная» процедура преобразования;
- переход от одного узла графа к другому узлу необходимо выполнять по заданному пути на графе (в установленном порядке, выполняя процедуры преобразования для каждого промежуточного узла).

Пример графа преобразования показан на Figure 3.5. Граф приведен для типовой конфигурации инженерной съемки, при которой одновременно осуществляются – съемка рельефа дна (MBES), сеймопрофилирование (SBP), съемка буксируемыми ГЛБО (SSS) и магнитометром (Magnetometer). При этом, для выполнения съемки используется следующее оборудование:

- на судне установлены MBES, GPS, MRU&Heave, Draft sensor, Cable counter, USBL, выдающий координаты маяка-ответчика в осях ECS_USBL и SBP (сейсмический профилограф);
- через Cable counter буксируется связка SSS и магнитометр; при этом магнитометр снабжен альтиметром и датчиком глубины, а SSS снабжен электронным компасом-наклономером и датчиком глубины;
- на буксировочном кабеле связки закреплен маяк-ответчик USBL.

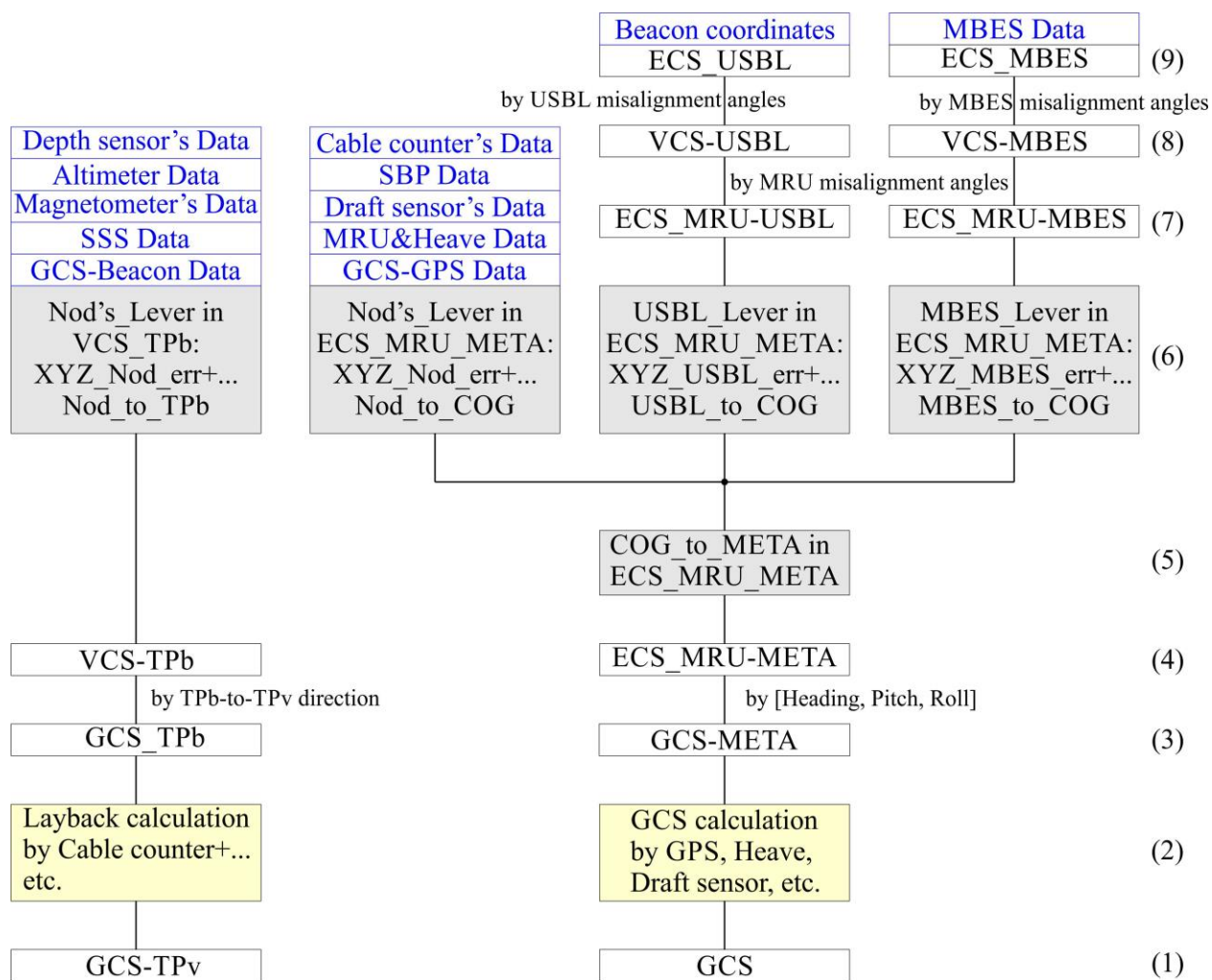


Figure 3.5 Пример графа преобразования координат для Модели-2

На графе, белым прямоугольникам соответствует поворот системы координат. Серым прямоугольникам соответствует перенос начала координат из точки в точку. Желтым прямоугольникам соответствуют процедуры расчета координат, отличающиеся от поворота и переноса начала координат.

Синим цветом, показаны данные с датчиков, которые также могут быть использованы в «преобразовании координат». Например, зная вынос GPS относительно метацентра, мы за счет углов [Heading, Pitch, Roll] можем его пересчитать в оси GCS-META и, далее, основываясь на данных GPS, привязать оси GCS-META к GCS (за счет этого можно пересчитать в оси GCS положение любого нода или цели, координаты которых известны в GCS-META).

Синим цветом показаны данные следующего оборудования, установленного на судне: MBES, GPS, MRU&Heave, Draft sensor, Cable counter, SBP, USBL, выдающий координаты маяка-ответчика в осях ECS_USBL.

Слева на графе показан пример ветви для двух последовательно буксируемых приборов SSS и магнитометра (магнитометр буксируется в piggy-back; при этом магнитометр снабжен альтиметром и датчиком давления-глубины). Позиционирование выполняется при помощи маяка-ответчика USBL, с контролем по алгоритму “Layback”, который использует длину вытравленного

кабеля и иные навигационные данные. При пропадании сигнала на USBL позиционирование будет выполняться по алгоритму “Layback” (с коэффициентами фильтров, подобранными при работе USBL). Курс буксируемых приборов определяется по направлению вектора «позиция маяка-ответчика USBL – позиция точки буксировки после фильтрации»; контроль курса выполняется по электронному магнитному компасу-наклономеру, установленному на SSS.

Дополнительный контроль будет осуществляться:

- по сличению данных эхолота и датчика давления-глубины, установленных на магнитометре, (привязанных к координатам) с рельефом дна по MBES;
- по сличению координат целей SSS на сонограммах, когда одна цель отбивается на различных галсах;
- по сличению координат целей на сонограммах SSS и данных MBES.

3.4 Резюме к разделу 3

1) Рассмотрена математическая модель (Модель-2) для пересчета координат цели из ECS_USBL в GCS-META (Figure 3.1). В модели присутствуют USBL, MRU и «корпус судна» (оси VCS).

Для Модели-2 принималось, что USBL установлен с [выносом LeverUSBL](#) относительно метацентра.

Помимо ошибки [установочных углов USBL](#), в Модели-2 введена ошибка [установочных углов MRU](#). Все установочные углы, в рамках модели, увязаны с направлением осей корпуса судна (VCS).

Показано, что:

- влияние ошибки в установочных углах MRU «переносит» значение минимума в массиве невязок `dst` (невязок для установочных углов USBL) в новую точку постоянную для различных комбинаций Heading, Pitch, Roll,
- влияние ошибки в длине [LeverUSBL](#) перемещает минимум в массиве невязок `dst` для каждой комбинации Heading, Pitch, Roll.

2) Рассмотрен особый случай Модели-2, когда ошибки для [установочных углов MRU и USBL равны](#). В данном случае отсутствует составляющая ошибки установочных углов, которая увеличивается с расстоянием. Ошибка позиционирования по USBL будет зависеть только от неправильной оценки положения выноса [LeverUSBL](#).

Следовательно:

- Для любых ошибок установочных углов MRU (даже значительно отклоняющихся от VCS), можно подобрать такие углы «компенсации установочных углов USBL», что оси ECS_USBL и ECS_MRU совпадут по направлению (Figure 3.4). Дальнейшее увеличение «точности позиционирования по USBL» изменением установочных углов USBL окажется невозможным (или будет работать для определенных курса, интервала глубин и расстояния до цели).

- Ошибка позиционирования цели (когда ошибки установочных углов MRU и USBL равны) будет определяться влиянием ошибки установочных углов MRU на проекцию LeverUSBL (на оси ECS_MRU с осей VCS), то есть, дальнейшее увеличение «точности позиционирования по USBL» потребует коррекции углов MRU.
- Величина невязки, рассчитываемой для двух положений судна (когда ошибки установочных углов MRU и USBL равны), будет зависеть от соотношения углов Heading, Pitch, Roll (при первом и втором положении); невязка будет уменьшаться для одинакового направления судов (осей USBL) и увеличиваться до максимума при встречных направлениях (осей USBL).

3) Введено понятие [графа преобразования координат](#).

4 Заключение

Ошибку в позиционировании по USBL можно разделить на четыре составляющих:

- (a) Случайная аппаратурная ошибка. Для данной составляющей ошибка позиционирования возрастает с увеличением расстояния до цели в соответствии с характеристиками аппаратуры.
- (b) Систематическая ошибка установочных углов USBL и MRU. Для данной составляющей ошибка позиционирования возрастает при увеличении расстояния до цели. Ошибка удаляется коррекцией установочных углов USBL, такой, что они **начинают совпадать** с ошибками установочных углов MRU.

Дальнейшее увеличение точности позиционирования цели коррекцией установочных углов USBL невозможно; требуется коррекция установочных углов MRU.

При работе с двумя MRU (или при переходе на резервный MRU) их установочные углы относительно корпуса судна должны быть одинаковыми, либо необходимо выполнять отдельный поиск «установочных углов USBL» относительно второго MRU. Также, возникновение ошибки в показаниях MRU за счет «режима эксплуатации» будет создавать ошибку в показаниях USBL.

- (c) Систематическая ошибка выносов LeverUSBL и LeverGPS. Для данной составляющей ошибка позиционирования не будет возрастать при увеличении расстояния до цели.

Компонента ошибки будет близка к «постоянному сдвигу» относительно курса судна; величина компоненты будет зависеть от величины ошибки установочных углов MRU и USBL **относительно корпуса судна**. При проходе над целью по одному галсу величина невязки (в положении цели на галсах) будет близка к нулю; при проходе на встречных галсах величина невязки (в положении цели на галсах) будет максимальной.

Ошибка удаляется коррекцией установочных углов MRU, такой, что оси MRU начинают совпадать с осями корпуса судна (VCS), в которых были измерены выносы LeverUSBL и LeverGPS.

- (d) Ошибка «смещения на плечо META_to_COG», которая не рассматривается в рамках Модели-2. Если при систематической ошибке выносов LeverUSBL и LeverGPS (c) рассматривается ошибка в расположении «конца вектора выноса», то при «смещении на плечо META_to_COG» имеет место ошибка в положении «начала вектора выноса». Поскольку, при данной ошибке учет качки и курса будет производиться относительно COG, а не настоящего метacentра судна, то величина ошибки окажется зависимой от курса судна и углов качки.

Все расчеты в настоящем документе выполнены при помощи функции **gNavTaitLever**, библиотеки **ge0mlib**. Текст функции приведен в Приложении 1.

Скрипт для вычислений по Модели-2 на основе функций **gNavTaitNLever**, **gNavTaitLeverN** или **gNavTaitLever** приведен в Приложении 2.

Список литературы

1. <https://www.xometry.com/resources/materials/what-is-brass/>

Приложение 1. Текст функции gNavTaitLever

```
function XYZ_Lev=gNavTaitLever(Head,Pitch,Roll,Lever,RotDirect)
%A number LeverArm's coordinates calculation using scalar Heading,Pitch,Roll
  (3D Tait-Bryan matrixes rotation).
%function XYZ_Lev=gNavTaitLever(Head,Pitch,Roll,Lever,RotDirect), where
%Head - row, rotation angle for x0y in degree [H_t1... H_tn];
%Pitch - row, rotation angle for z0x in degree [P_t1... P_tn];
%Roll - row, rotation angle for y0z in degree [R_t1... R_tn];
%Lever - 3 rows with lever arms coordinates [Lx1...Lxm; Ly1...Lym;
  Lz1...Lzm];
%RotDirect - 'frw' for forward rotation; 'rev' for reverse rotation;
%XYZ_Lev - "moved" lever arms coordinates: column is [Lx;Ly;Lz]; rows is
  t1...tn; 3rd-size is Lever_1...Lever_m
%if Roll or Pitch or Head is scalar, then vector with necessary length will
  be created.
%-----
%Coordinate system is:
%^ x(forward/roll)
%|
%o---> y(right/pitch)
%z(up/head)
%All right/clockwise rotation sign is + (if vector is rotated relatively
  axis);
%All left/unclockwise rotation sign is + (if axis are rotated relatively
  vector);
%-----
%Example: tr=[120;60;-80];tmp001=gNavTaitLever(-2,-2,-2,tr,'frw');

Head=Head./180.*pi;Pitch=Pitch./180.*pi;Roll=Roll./180.*pi;
Len=max([size(Roll,2) size(Pitch,2) size(Head,2)]);Len=Len(1);
if all(size(Head)==1), Head=repmat(Head,1,Len);end;
if all(size(Pitch)==1), Pitch=repmat(Pitch,1,Len);end;
if all(size(Roll)==1), Roll=repmat(Roll,1,Len);end;
if all(size(Head)~=size(Roll))||all(size(Pitch)~=size(Roll)),
  error('Roll,Pitch,Head length must be equal or scalar');end;
if size(Lever,1)~=3, error('Levers must be 3 rows with lever arms
  coordinates');end;
NumL=size(Lever,2);

XYZ_Lev=zeros(3,Len,NumL);%allocate memory for output
for n=1:Len, %calculate rotation for all levers
  if isempty(RotDirect)||strcmp(RotDirect,'frw'), %forward rotation >>
    Head-Pitch-Roll >> RollM*PitchM*HeadM*Lever
    XYZ_Lev(:,n,:)= [1 0 0; 0 cos(Roll(n)) -sin(Roll(n)); 0 sin(Roll(n))
      cos(Roll(n))] * [cos(Pitch(n)) 0 sin(Pitch(n)); 0 1 0; -sin(Pitch(n)) 0
      cos(Pitch(n))] * [cos(Head(n)) -sin(Head(n)) 0; sin(Head(n))
      cos(Head(n)) 0; 0 0 1]*Lever;
  elseif strcmp(RotDirect,'rev'), %reverse rotation >> Roll-Pitch-Head >>
    HeadM*PitchM*RollM*Lever
    XYZ_Lev(:,n,:)= [cos(Head(n)) -sin(Head(n)) 0; sin(Head(n))
      cos(Head(n)) 0; 0 0 1] * [cos(Pitch(n)) 0 sin(Pitch(n)); 0 1 0; -
      sin(Pitch(n)) 0 cos(Pitch(n))] * [1 0 0; 0 cos(Roll(n)) -sin(Roll(n));
      0 sin(Roll(n)) cos(Roll(n))]*Lever;
  else
    error('Incorrect RotDirect value');
  end;
end;
end;
```

%mail@ge0mlib.com 27/05/2021

Приложение 2. Скрипт для вычислений по Модели-2

```
%script gNavTaitTest;
%Calculate "error ellipsoid" for USBL misalignment angles, using 2 ship's
    location with target's coordinates by USBL.

SU=struct('du',[1;0.5;0.8],'dm',[0;0;0],'LeverUSBL',[3;0;0]); %set USBL and
    MRU misalignment error, set LeverUSBL in VCS [-0.5;0.5;-1]
    [1;0.5;0.8] [10;5;0]
%sh1m=[-60;-3;-5];sh2m=[60;3;5]; %set [Head Pitch Roll] for sh1 and sh2
sh1m=[0;0;0];sh2m=[0;0;0]; %set [Head Pitch Roll] for sh1 and sh2
sh1=[0;0;0];sh2=[120;200;0];tr=[120;60;-80]; %set ship coords and target
    coords
LeverUSBL=gNavTaitLever(-SU.dm(1),-SU.dm(2),-SU.dm(3),SU.LeverUSBL,'frw');
    %calculate LeverUSBL in ECS_MRU
%calculate 'true' target coords for ship1
tmp3=gNavTaitLever(-sh1m(1),-sh1m(2),-sh1m(3),tr-sh1,'frw');tmp2=tmp3-
    LeverUSBL;tmp1=gNavTaitLever(SU.dm(1),SU.dm(2),SU.dm(3),tmp2,'rev');
tr1=gNavTaitLever(-SU.du(1),-SU.du(2),-SU.du(3),tmp1,'frw')+sh1;
%calculate 'true' target coords for ship2
tmp3=gNavTaitLever(-sh2m(1),-sh2m(2),-sh2m(3),tr-sh2,'frw');tmp2=tmp3-
    LeverUSBL;tmp1=gNavTaitLever(SU.dm(1),SU.dm(2),SU.dm(3),tmp2,'rev');
tr2=gNavTaitLever(-SU.du(1),-SU.du(2),-SU.du(3),tmp1,'frw')+sh2;

%calculate 'dst-residual set'
dd=3;ddd=0.02;[dHu,dPu,dRu]=meshgrid(-dd:ddd:dd,-dd:ddd:dd,-
    dd:ddd:dd);dst=zeros(size(dHu));%set 'USBL angles error corrections'
    by Pich&Roll&Head grid
dHm=SU.dm(1);dPm=SU.dm(2);dRm=SU.dm(3);%set 'false' MRU's misalignment angles
LeverU=[0;0;0]; %set 'false' LeverUSBL in ECS_MRU // LeverU=LeverUSBL;
tic;
if 0, disp('Using for-scycle');
    for n=1:numel(dHu),
        %target coordinate recalculate with 'angles correction' for ship1
        tmp1=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr1-
            sh1,'rev');tmp2=gNavTaitLever(-dHm,-dPm,-
            dRm,tmp1,'frw');tmp3=tmp2+LeverU;
        r1=gNavTaitLever(sh1m(1),sh1m(2),sh1m(3),tmp3,'rev')+sh1;
        %target coordinate recalculate with 'angles correction' for ship2
        tmp1=gNavTaitLever(dHu(n),dPu(n),dRu(n),tr2-
            sh2,'rev');tmp2=gNavTaitLever(-dHm,-dPm,-
            dRm,tmp1,'frw');tmp3=tmp2+LeverU;
        r2=gNavTaitLever(sh2m(1),sh2m(2),sh2m(3),tmp3,'rev')+sh2;
        %distance between sh1&sh2 target locations
        dst(n)=sqrt(sum((r1-r2).^2));
    end;
else disp('Not using for-scycle');
    [X1,Y1,Z1,X2,Y2,Z2]=gNavTaitNLever(dHu,dPu,dRu,[tr1-sh1 tr2-sh2],'rev');
    [X1,Y1,Z1]=gNavTaitLeverN([-dHm;-dPm;-
        dRm],X1,Y1,Z1,'frw');[X1,Y1,Z1]=gNavTaitLeverN(sh1m,X1+LeverU(1),Y1+L
        everU(2),Z1+LeverU(3),'rev');
    [X2,Y2,Z2]=gNavTaitLeverN([-dHm;-dPm;-
        dRm],X2,Y2,Z2,'frw');[X2,Y2,Z2]=gNavTaitLeverN(sh2m,X2+LeverU(1),Y2+L
        everU(2),Z2+LeverU(3),'rev');
    dst=sqrt((X1+sh1(1)-X2-sh2(1)).^2+(Y1+sh1(2)-Y2-sh2(2)).^2+(Z1+sh1(3)-Z2-
        sh2(3)).^2);
end;
toc;
dst(dst>10)=nan;
%draw figures and slices
figure(1);h=slice(dHu,dPu,dRu,dst,1,0.5,0.8);for
    n=1:numel(h);h(n).EdgeColor='none';end;xlabel('dH');ylabel('dP');zlab
    el('dR');
```

```

figure(2);h=slice(dHu,dPu,dRu,dst,0.5,1.5,1);for
    n=1:numel(h);h(n).EdgeColor='none';end;xlabel('dH');ylabel('dP');zlabel('dR');
figure(3);h=slice(dHu,dPu,dRu,dst,-1.5,0,-1.8);for
    n=1:numel(h);h(n).EdgeColor='none';end;xlabel('dH');ylabel('dP');zlabel('dR');
figure(4);h=slice(dHu,dPu,dRu,dst,0,-0.5,-0.2);for
    n=1:numel(h);h(n).EdgeColor='none';end;xlabel('dH');ylabel('dP');zlabel('dR');
figure(5);h=slice(dHu,dPu,dRu,dst,0.5,-1,-0.1);for
    n=1:numel(h);h(n).EdgeColor='none';end;xlabel('dH');ylabel('dP');zlabel('dR');

```

%mail@ge0mlib.com 07/06/2021