Практика М 4.7

Построение модели системы ориентации и стабилизации МКА

Математическое моделирование и прикладная инженерия космических систем

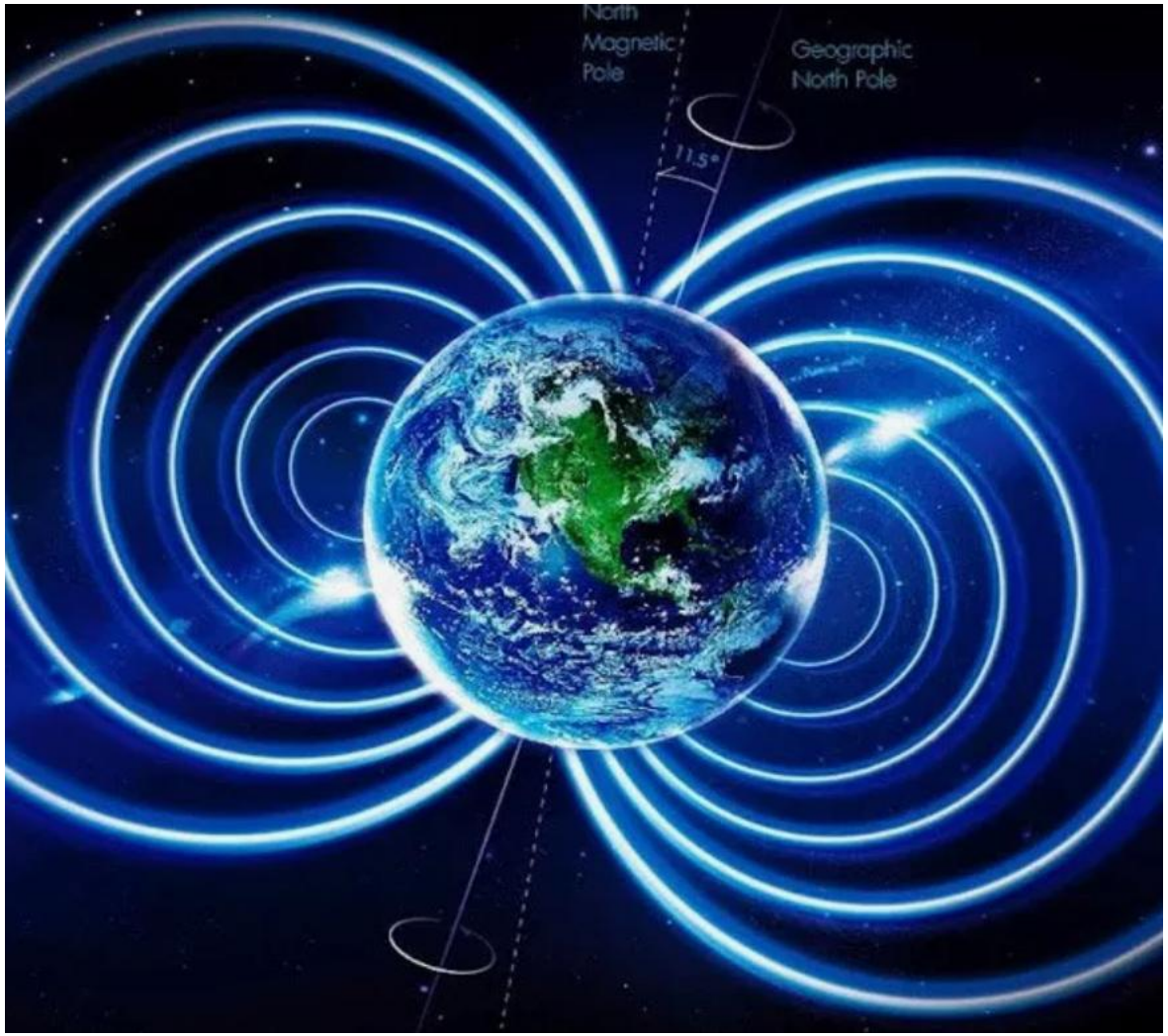
Юдинцев В. В.

кадры для космоса

Содержание

- Магнитное поле Земли
- Магнитная катушка
- Алгоритм B-dot
- Системы координат
- Построение модели в SimInTech

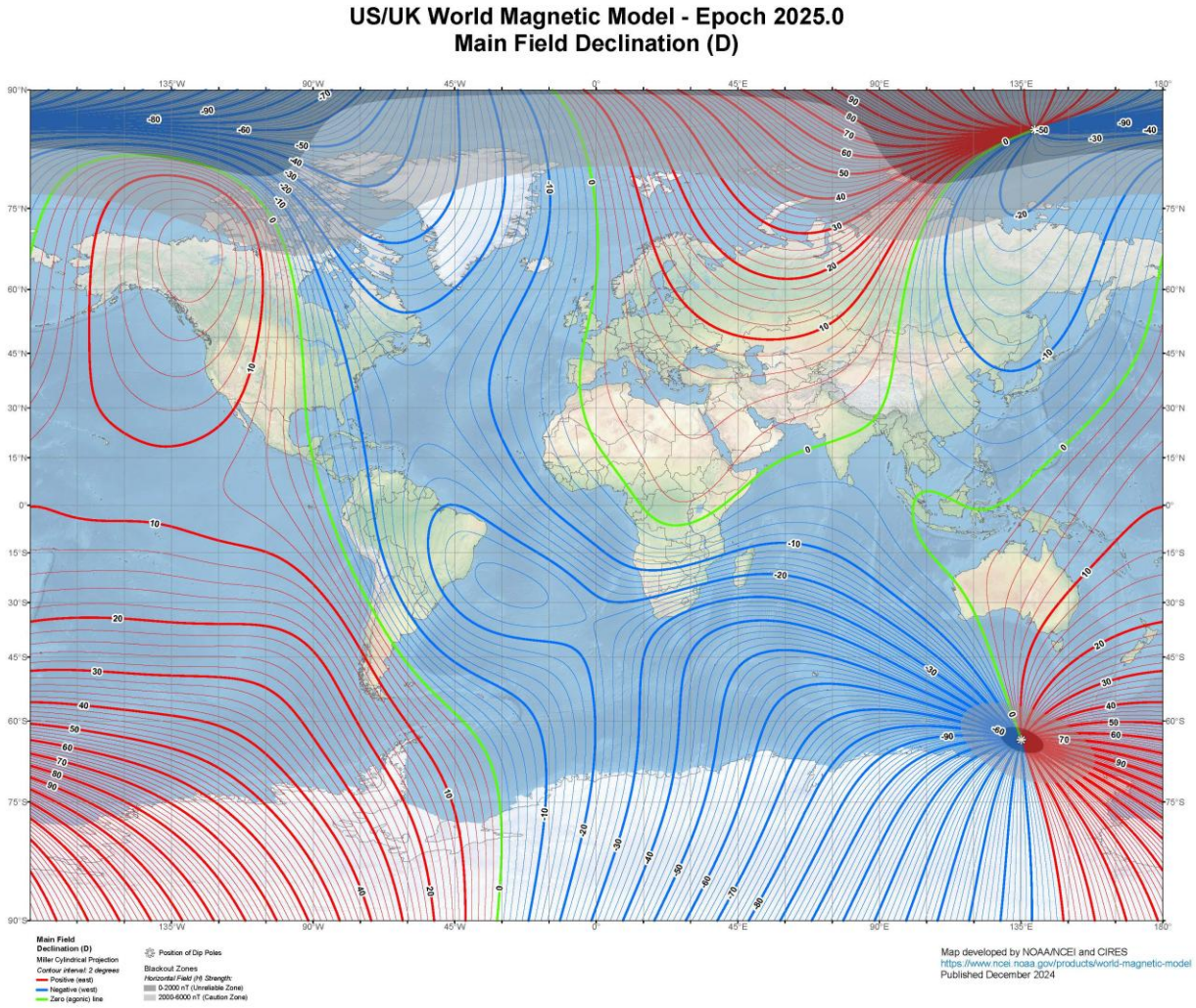
Магнитное поле Земли



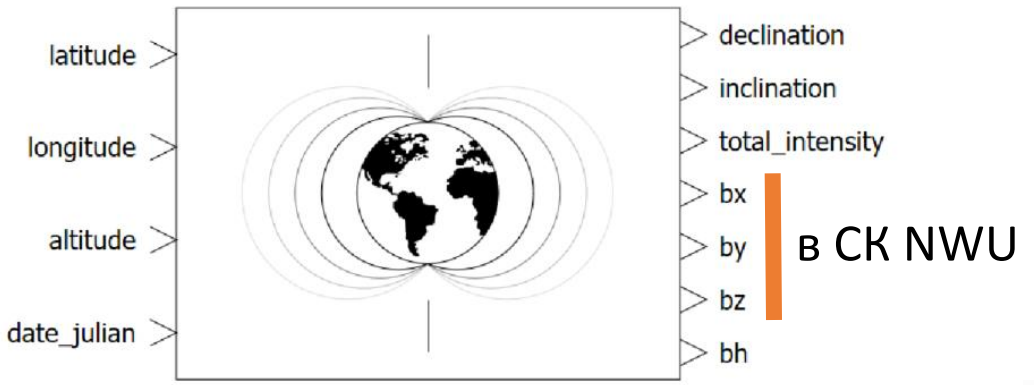
- Для создания управляющего момента без расхода рабочего тела можно использовать магнитные катушки
- Магнитные катушки создают за счёт взаимодействия с магнитным полем Земли
- Момент пропорционален вектору магнитной индукции **B** в точке орбиты
- **B** непрерывно меняется и зависит от положения спутника на орбите и от времени

кадры для космоса

Модели магнитного поля

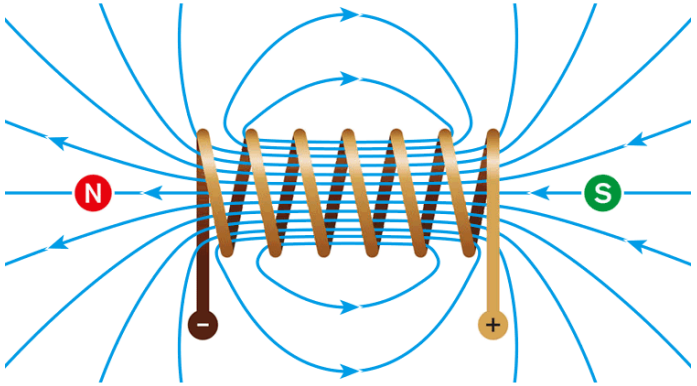


- **GRF** (International Geomagnetic Reference Field) — международная эталонная модель.
- **WMM** (World Magnetic Model) — модель, обновляемая каждые 5 лет (актуальна WMM-2025)
- В SimInTech есть готовый блок модели магнитного поля Земли

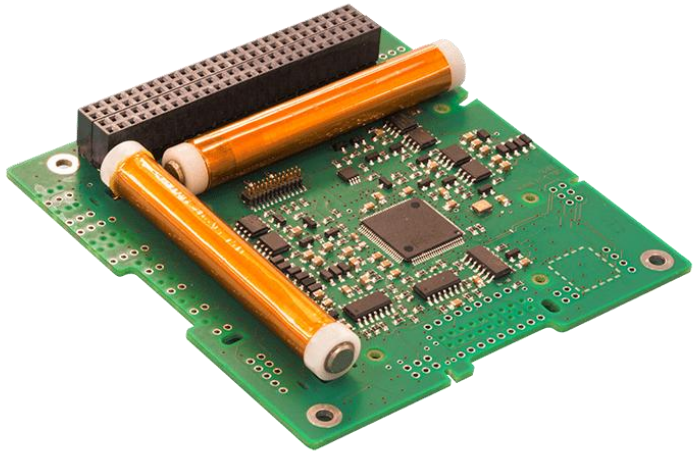


кадры для космоса

Магнитный момент катушки



https://light-fizika.ru/index.php/8-klass?layout=edit&id=314



https://www.winding-technology.com/eu-en/wiki/magnetorquer

- Катушка с током создаёт дипольный момент:

$$\vec{m} = N \cdot I \cdot S \cdot \vec{n} \qquad [A \cdot м^2]$$

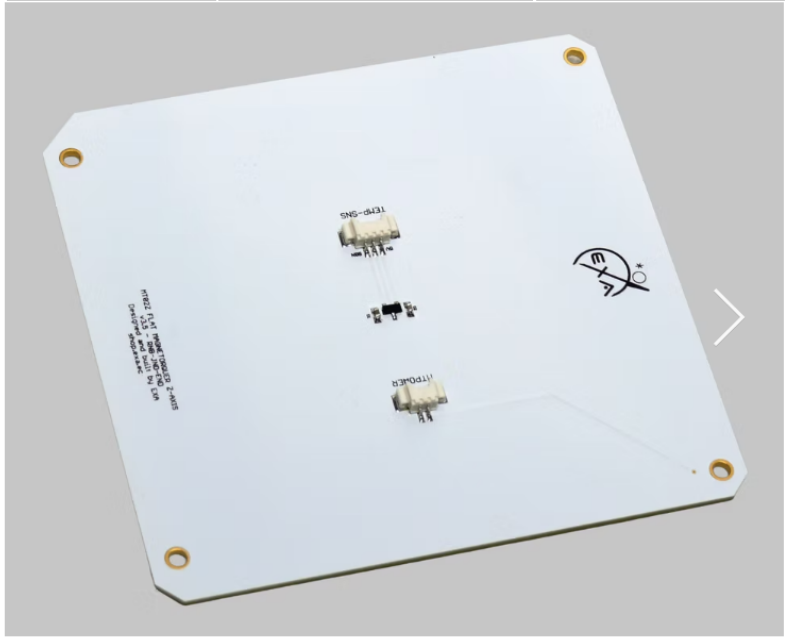
где N — число витков, I — ток, S — площадь витка, $\vec{n}$ — единичная нормаль к плоскости катушки.

- Направление **m** задаётся ориентацией катушки на спутнике.
- Величина **m** управляется током I.
- Три ортогональные катушки позволяют сформировать три независимые компоненты **m**.

кадры для космоса

Магнитная катушка

ITAR FREE | FLIGHT HERITAGE | NASA LISTED









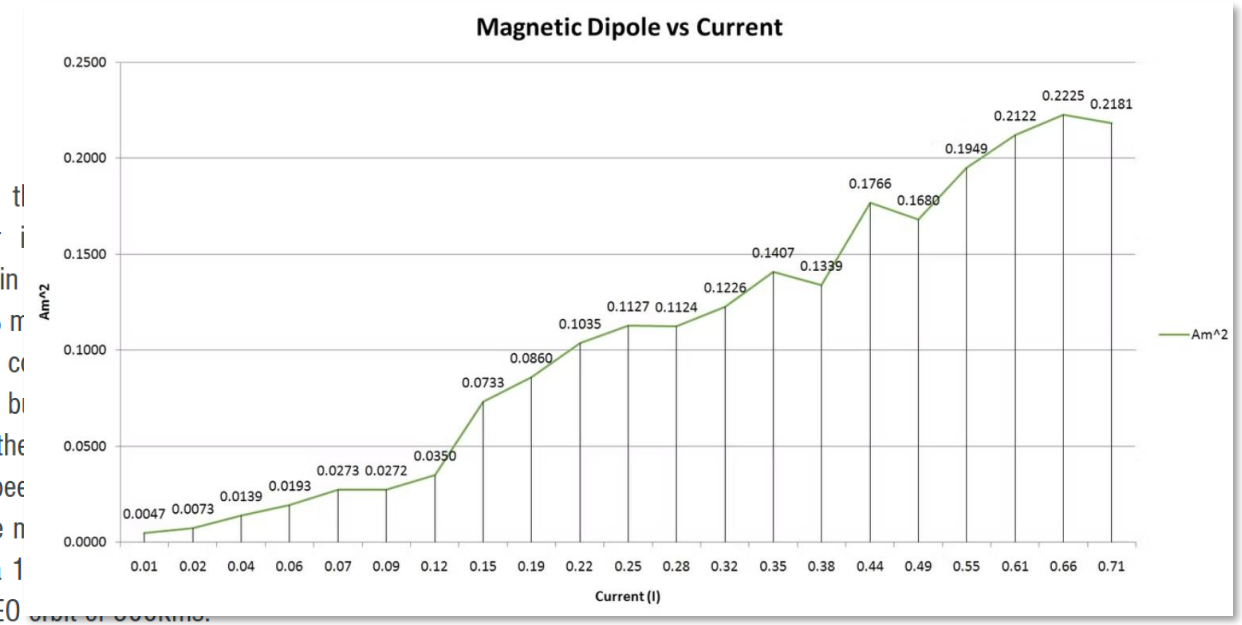


MT02Z Z-axis Flat Cubesat MT

5V to 12V

Up to 0.22 Am²

€ 1,000

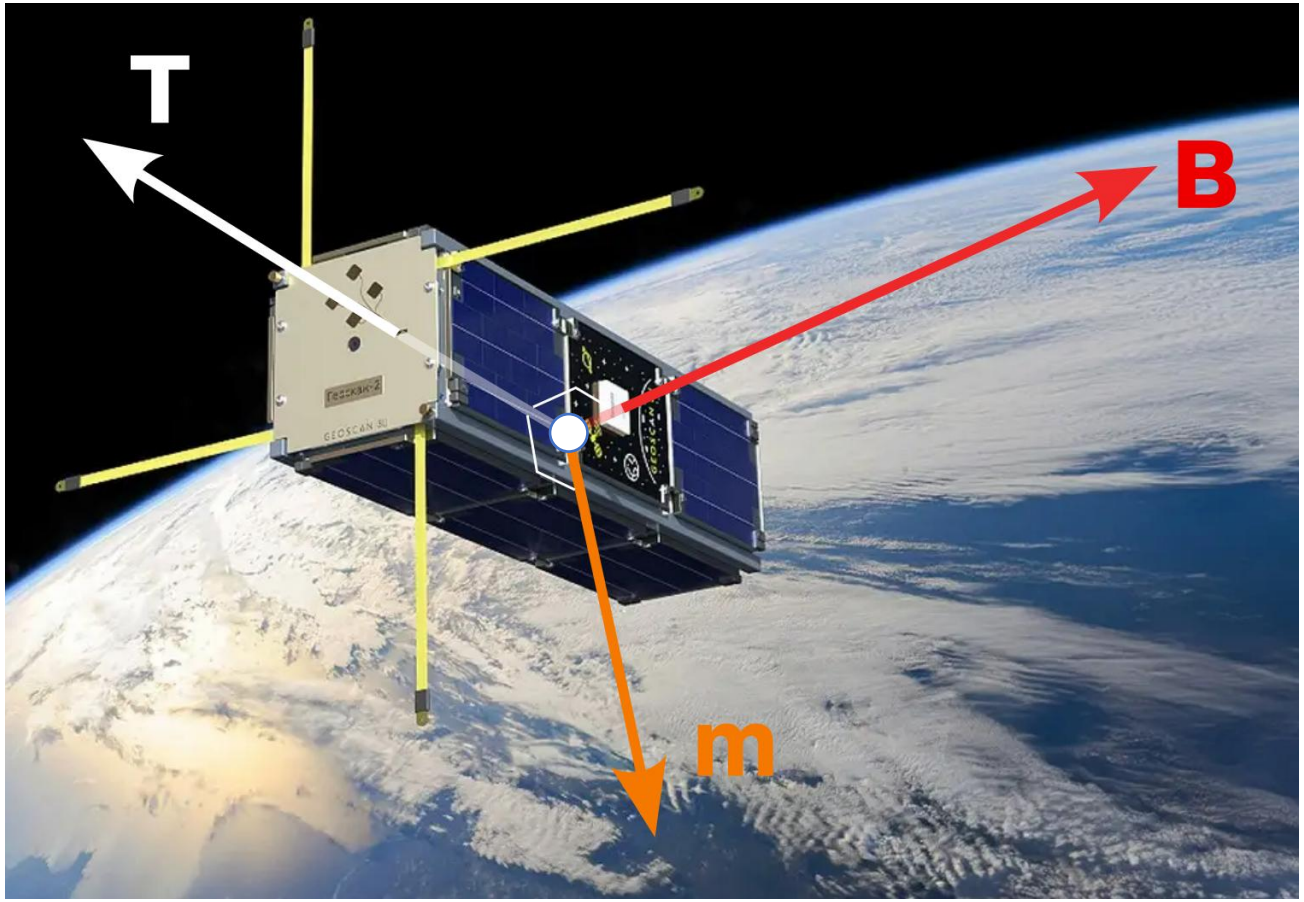


With only 1.5 millimeters t
MT02Z Flat Magnetorquer i
designed for ADCS control in
medium voltage levels, 60% m
an impressive performance c
the mass, power, and area b
those small dimensions the
magnetic moments, turn spee
comparable products on the n
to a minimum: It can turn a 1
using only 0.5 Watts at a LEO

The greatest advantage of the MT02Z is space saving: It can be easily affixed anywhere on your cubesat stack using minimal thickness.

кадры для космоса

Управляющий момент



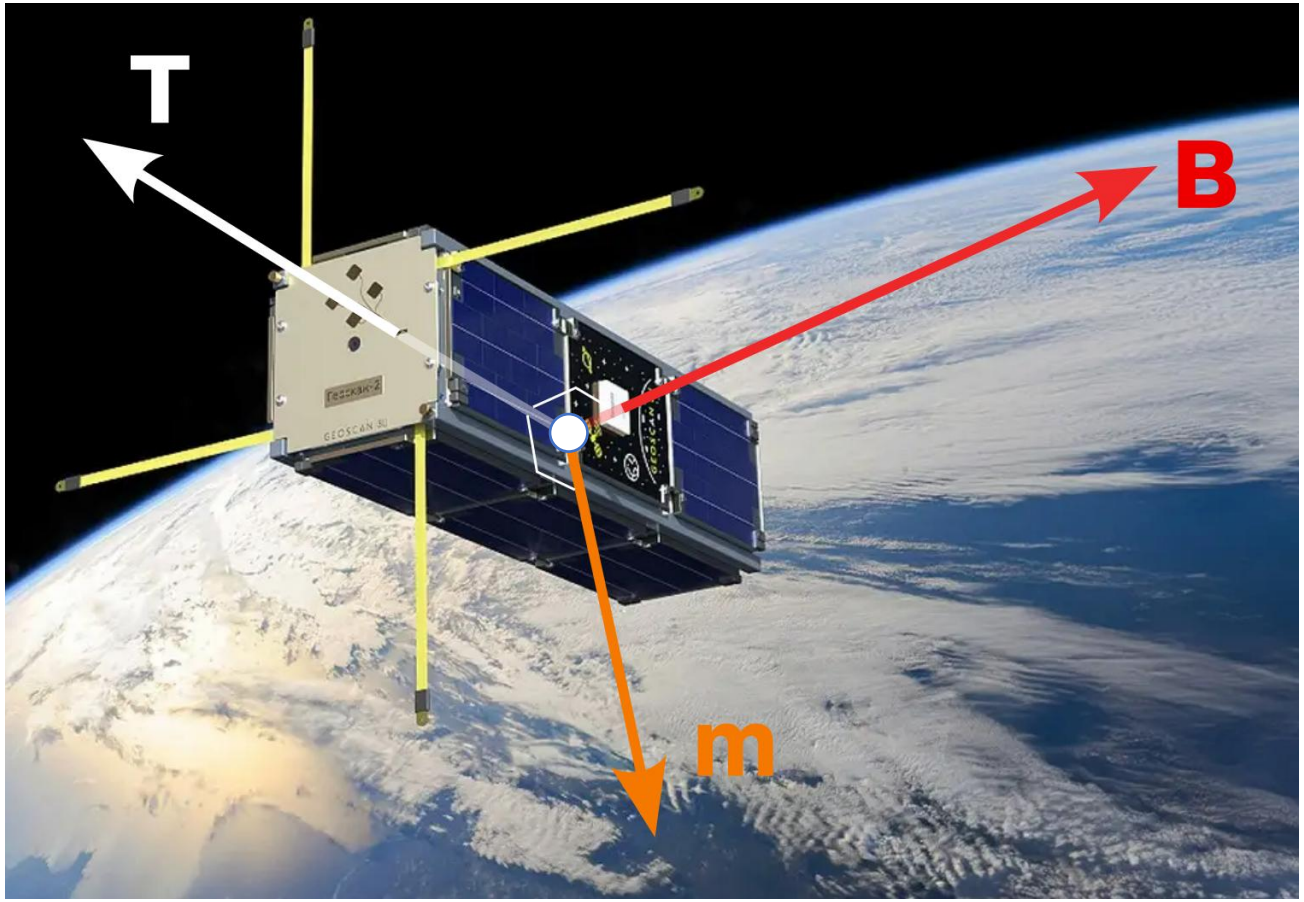
https://topwar.ru/272785-pervyj-sputnik-klassa-cubesat-rossijskij-apparat-geoskan-2-v-rabote.html

- Момент, создаваемый при взаимодействии магнитного дипольного момента катушки с магнитным полем Земли:

$$\vec{T} = \vec{m} \times \vec{B}$$

- Свойства векторного произведения:
  - **T** перпендикулярен и **m**, и **B**.
  - Момент **нулевой**, если **m** || **B**.
  - Максимальный момент достигается, когда вектор **m** перпендикулярен вектору **B**

Управляющий момент

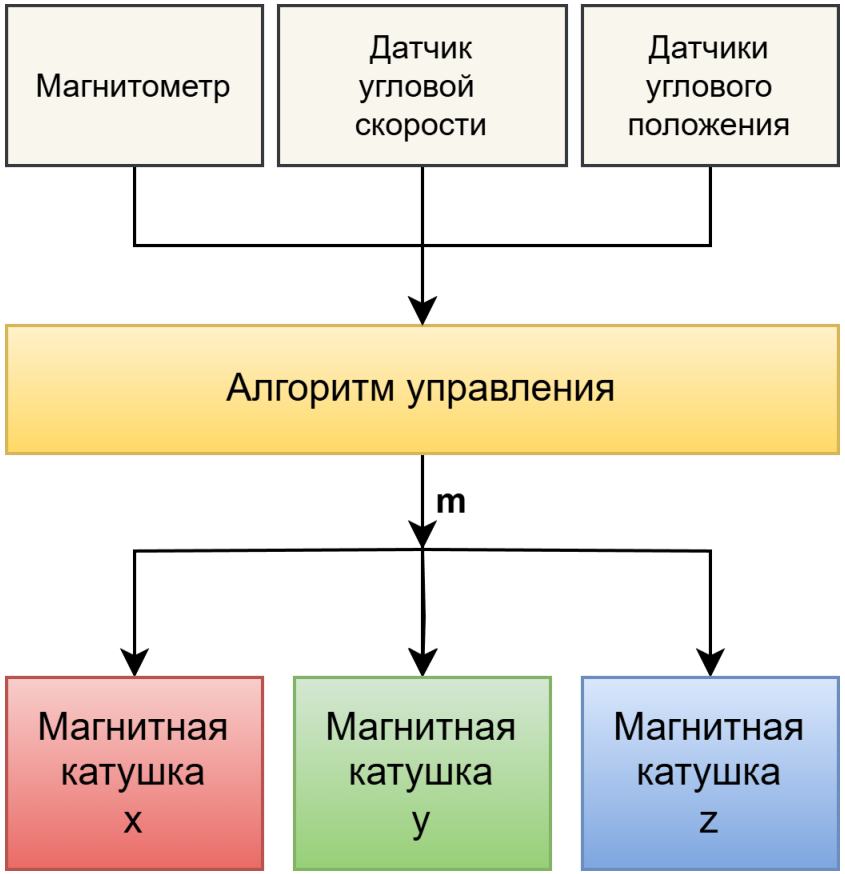


https://topwar.ru/272785-pervyj-sputnik-klassa-cubesat-rossijskij-apparat-geoskan-2-v-rabote.html

- **Нельзя управлять по оси B** (момент вокруг B = 0).
- Момент зависит от |B|: на полярных орбитах поле сильнее, на экваториальных — слабее.
- **B** меняется по орбите → момент меняется даже при постоянном **m**.
- Управление по сути **периодическое**: нужна стратегия усреднения по витку.
- Магнитные катушки — пассивны, не расходуют рабочее тело, но момент мал (мН·м).

кадры для космоса

Управление угловым движением



- **Датчики:** магнитометр (измеряет B), солнечный/звёздный датчик, гироскоп.
- **Алгоритм управления:** вычисляет требуемый **m** по закону управления (например, PD-регулятор).
- **Исполнительный орган:** три магнитные катушки, ток задаётся ШИМ-регулятором.
- **Обратная связь:** магнитометр → оценка B → коррекция m.
- Типовая архитектура: «датчик → фильтр → закон управления → катушки»

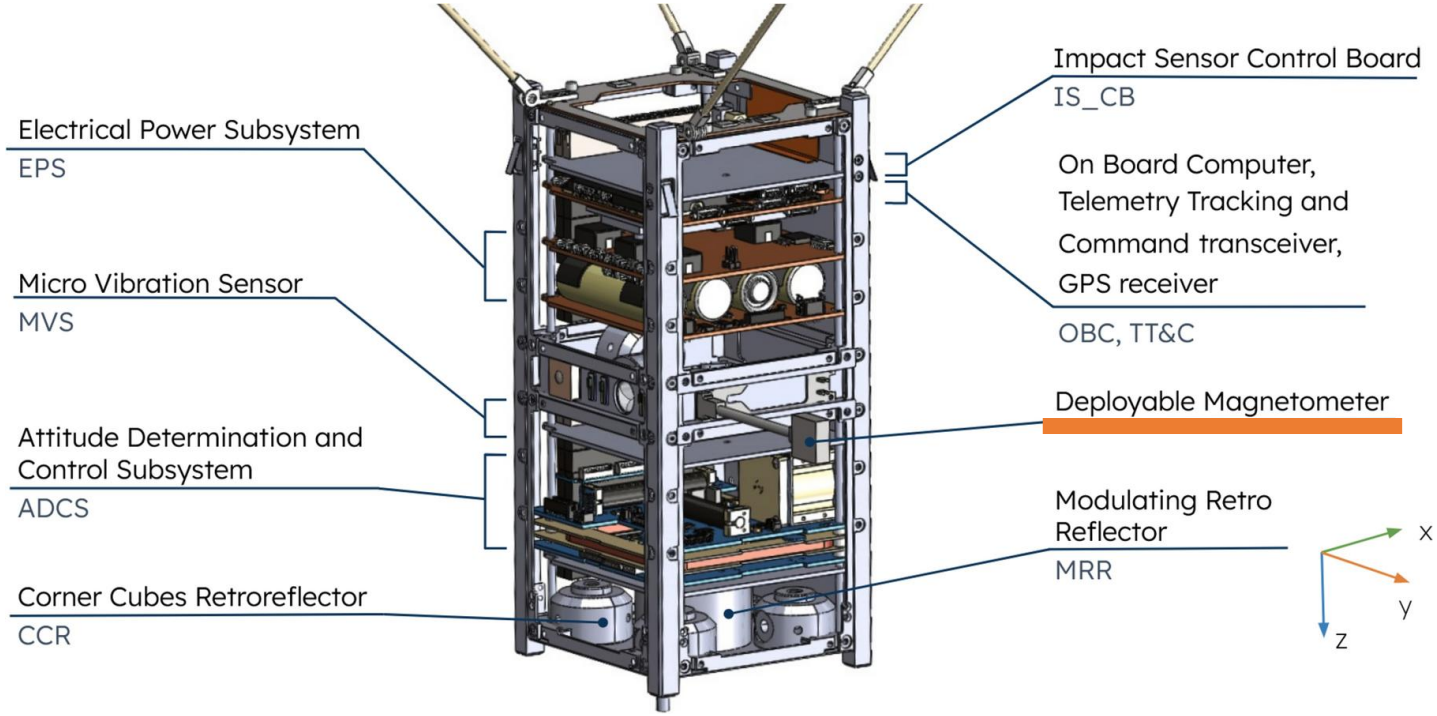
Алгоритм B-dot



- **B-dot** — самый распространённый алгоритм **начального успокоения (detumbling)** спутника после отделения от ракеты-носителя.
- Задача: снять начальную угловую скорость (может достигать десятков и сотен градусов в секунду).
- Идея: создать магнитный момент, **противодействующий изменению магнитного поля** в связанной системе координат.
- Алгоритм использует **только магнитометр** — не нужны гироскопы или звёздные датчики

кадры для космоса

Алгоритм B-dot



- Магнитометр измеряет поле Земли в связанной системе координат КА
- Если КА вращается, то направление вектора B в системе координат КА меняется:

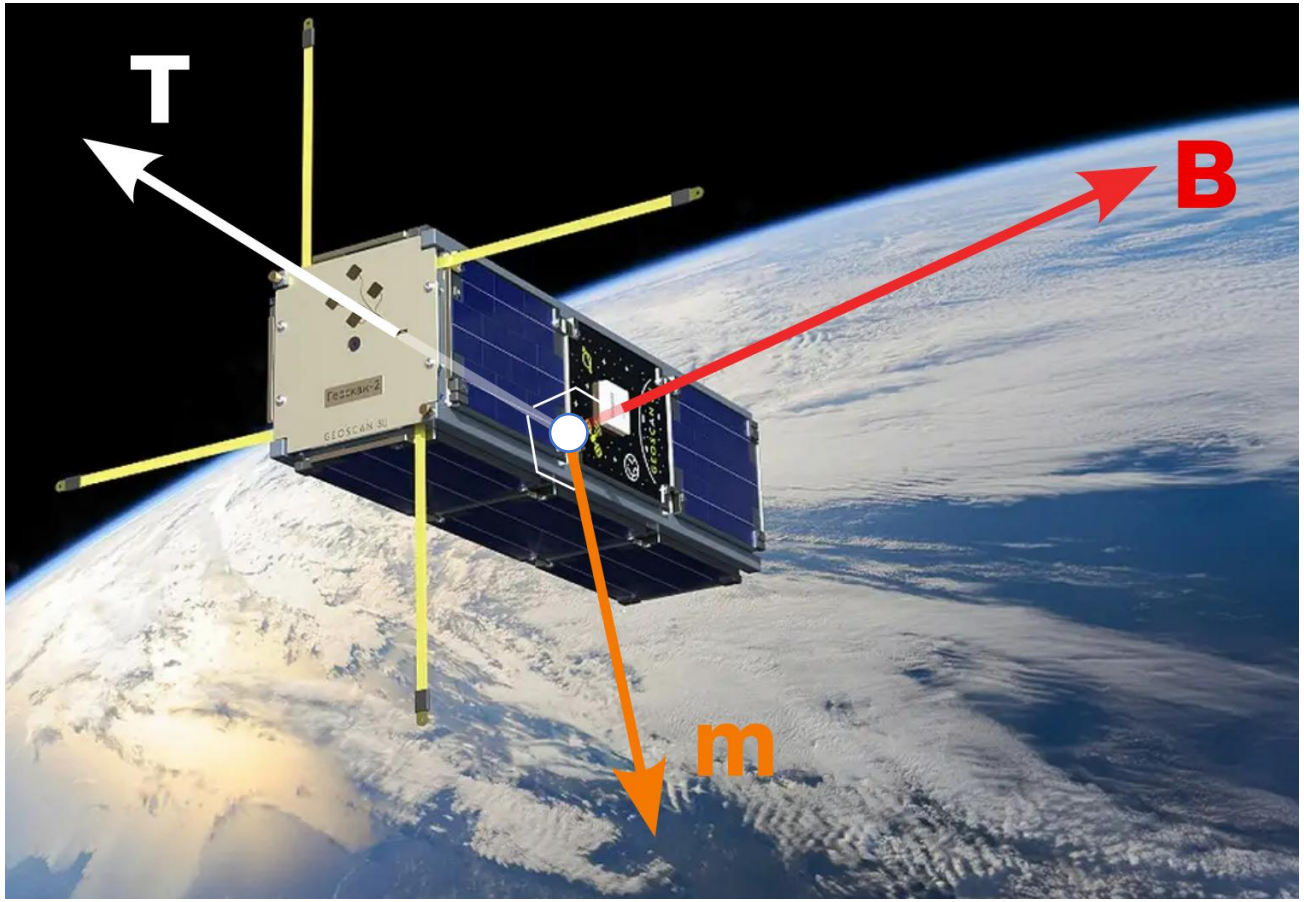
$$\dot{B} \approx -\omega \times B$$

- Быстрое вращение ⇒ большой сигнал B-dot
- Этот сигнал можно использовать для формирования демпфирующего момента

Mozzato, M., Bemporad, G., Enzo, S. et al. Concept and Feasibility Analysis of the Alba Cubesat Mission. Aerotec. Missili Spaz. 103, 377–389 (2024). https://doi.org/10.1007/s42496-024-00205-9

кадры для космоса

Алгоритм B-dot



https://topwar.ru/272785-pervyj-sputnik-klassa-cubesat-rossijskij-apparat-geoskan-2-v-rabote.html

- Производная магнитного поля:

$$\dot{\mathbf{B}} \approx \frac{\mathbf{B}(k)-\mathbf{B}(k-1)}{\Delta t}]$$

- Закон управления:

$$\boldsymbol{m} = -k_B \dot{\mathbf{B}}$$

$k_B > 0$ – коэффициент B-dot

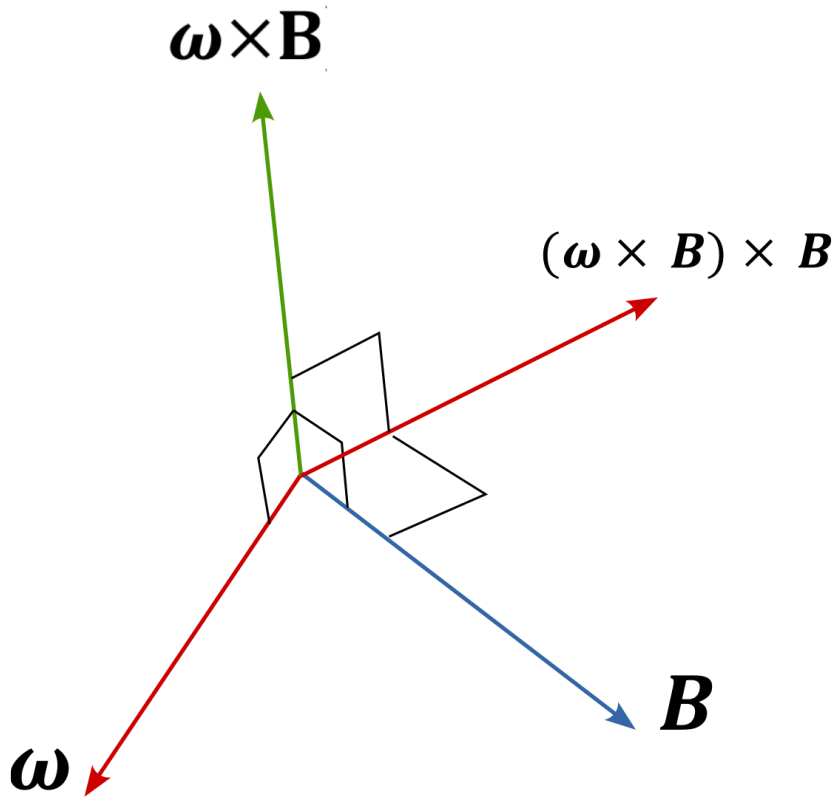
$\boldsymbol{m}$ – команда магнитным катушкам.

- Ограничение:

$$|m_i| \leq m_{i,\max}$$

кадры для космоса

Демпфирование угловой скорости



$$\left.\frac{d\mathbf{B}}{dt}\right|_{ECI} = \left.\frac{d\mathbf{B}}{dt}\right|_{body} + \boldsymbol{\omega} \times \mathbf{B}$$

$$\dot{\mathbf{B}} \approx -\boldsymbol{\omega}\times\mathbf{B}$$

- Тогда:

$$\mathbf{m} = -k_B\dot{\boldsymbol{B}} = k_B(\boldsymbol{\omega}\times\mathbf{B})$$

- Управляющий момент:

$$\mathbf{M} = \mathbf{m}\times\mathbf{B}, \qquad \mathbf{M} = k_B(\boldsymbol{\omega}\times\mathbf{B})\times\mathbf{B}$$

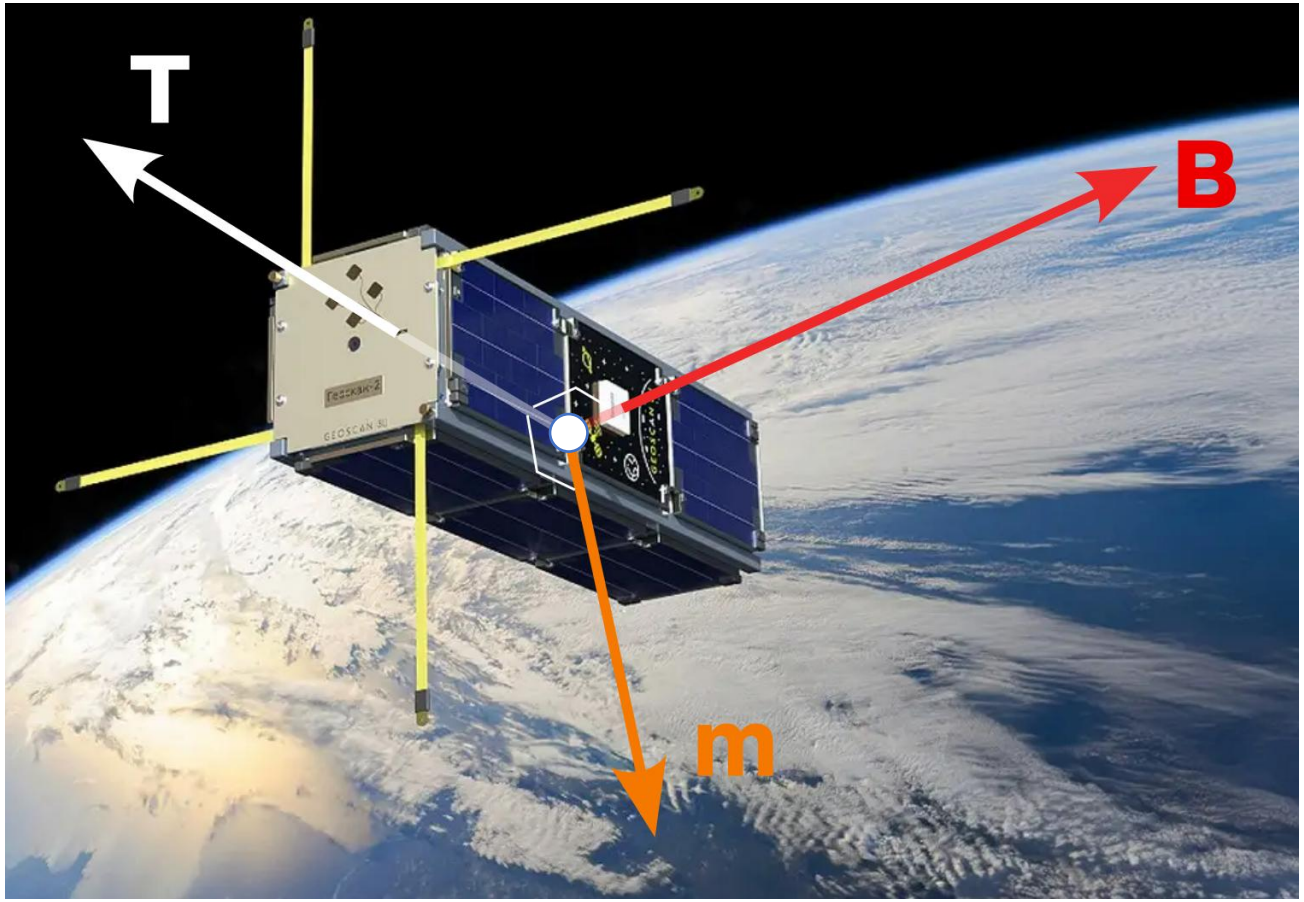
Результат:

$$\mathbf{M}\cdot\boldsymbol{\omega} \le 0$$

Магнитный момент “отбирает” кинетическую энергию вращения.

кадры для космоса

Алгоритм B-dot



https://topwar.ru/272785-pervyj-sputnik-klassa-cubesat-rossijskij-apparat-geoskan-2-v-rabote.html

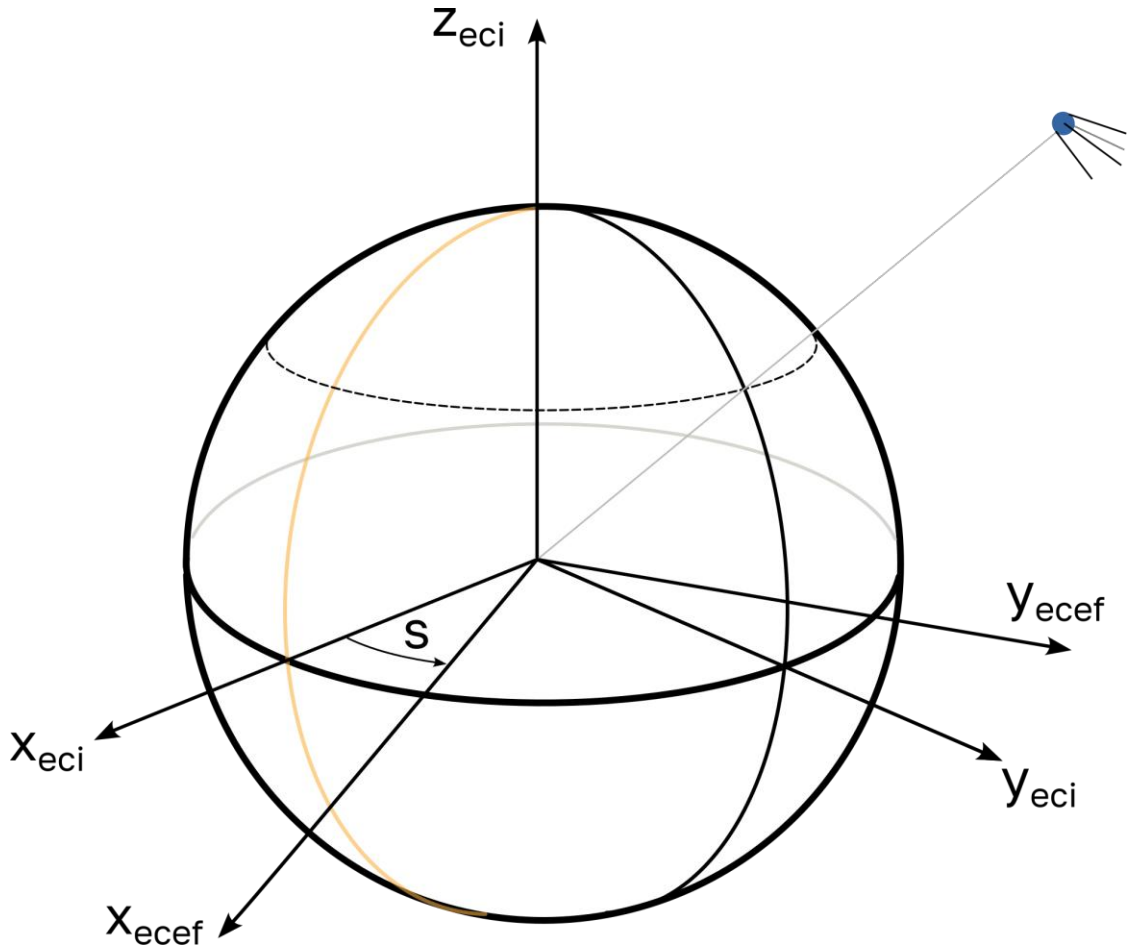
- Измерить $\mathbf{B}(k)$
- Рассчитать $\dot{\mathbf{B}}$.
- Отфильтровать шум.
- Вычислить:

$$\mathbf{m}_{\text{ком}} = -k_B \dot{\mathbf{B}}$$

- Ограничить магнитный момент
- Преобразовать $\mathbf{m}_{\text{ком}}$ в токи катушек
- Повторить через Δt

кадры для космоса

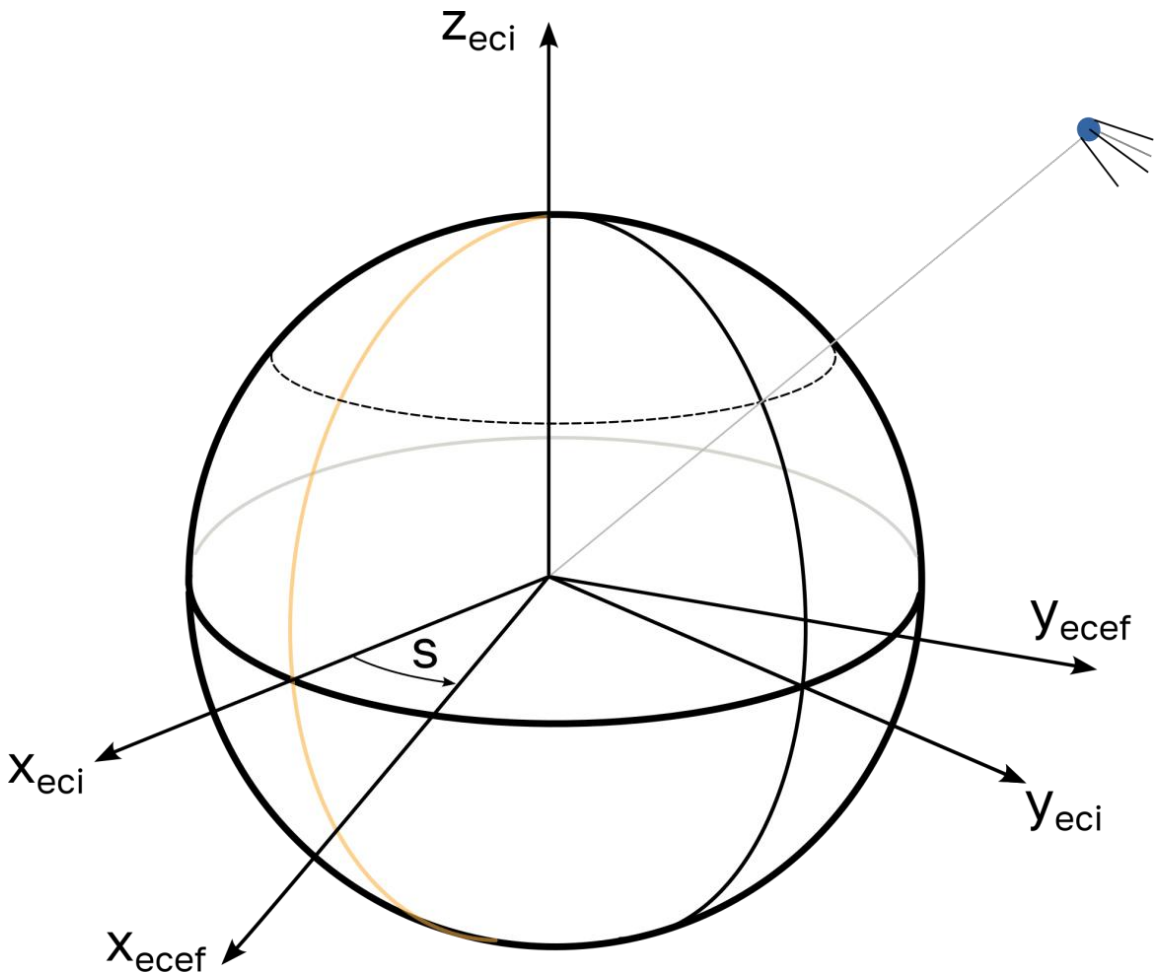
Системы координат



- **ECI (Earth-Centered Inertial)** — геоцентрическая инерциальная система координат.
- **Не вращается вместе с Землёй**
- Начало в центре масс Земли
- Ось x направлена в точку весеннего равноденствия (фиксированное направление в инерциальном пространстве)
- Ось y направлена вдоль оси вращения Земли
- **J2000** (EME2000) - Оси зафиксированы на эпоху 2000-01-01 12:00

кадры для космоса

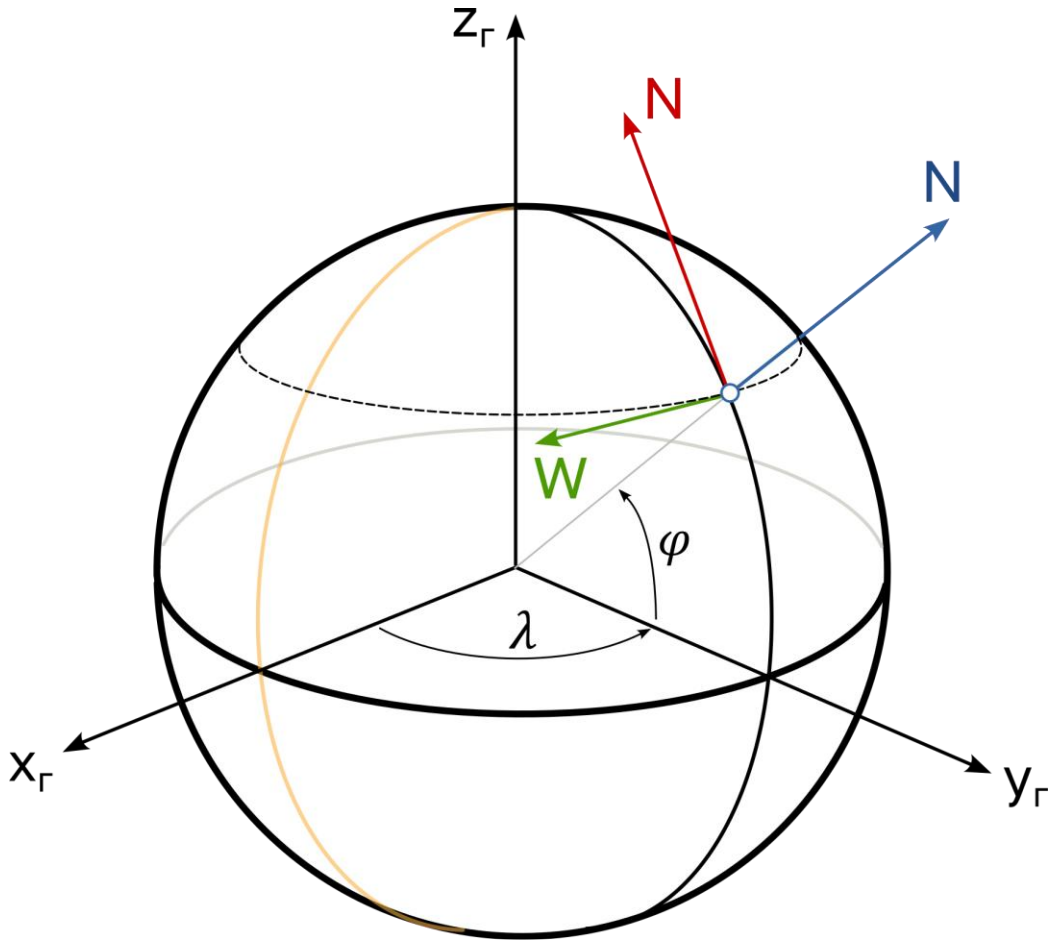
Системы координат



- **ECEF (Earth-Centered, Earth-Fixed) — геоцентрическая гринвичская система координат, жёстко связанная с Землёй.**
  - Начало в центре земли
  - Ось x лежит в плоскости экватора, проходит через Гринвичский меридиан
  - Ось z - вдоль оси вращения Земли (на Северный полюс)
  - Ось y дополняет до правой тройки
- **Система координат вращается вместе с Землёй с угловой скоростью ~7.292115×10^{-5} рад/с (звёздные сутки).**

кадры для космоса

NWU (North-West-Up)

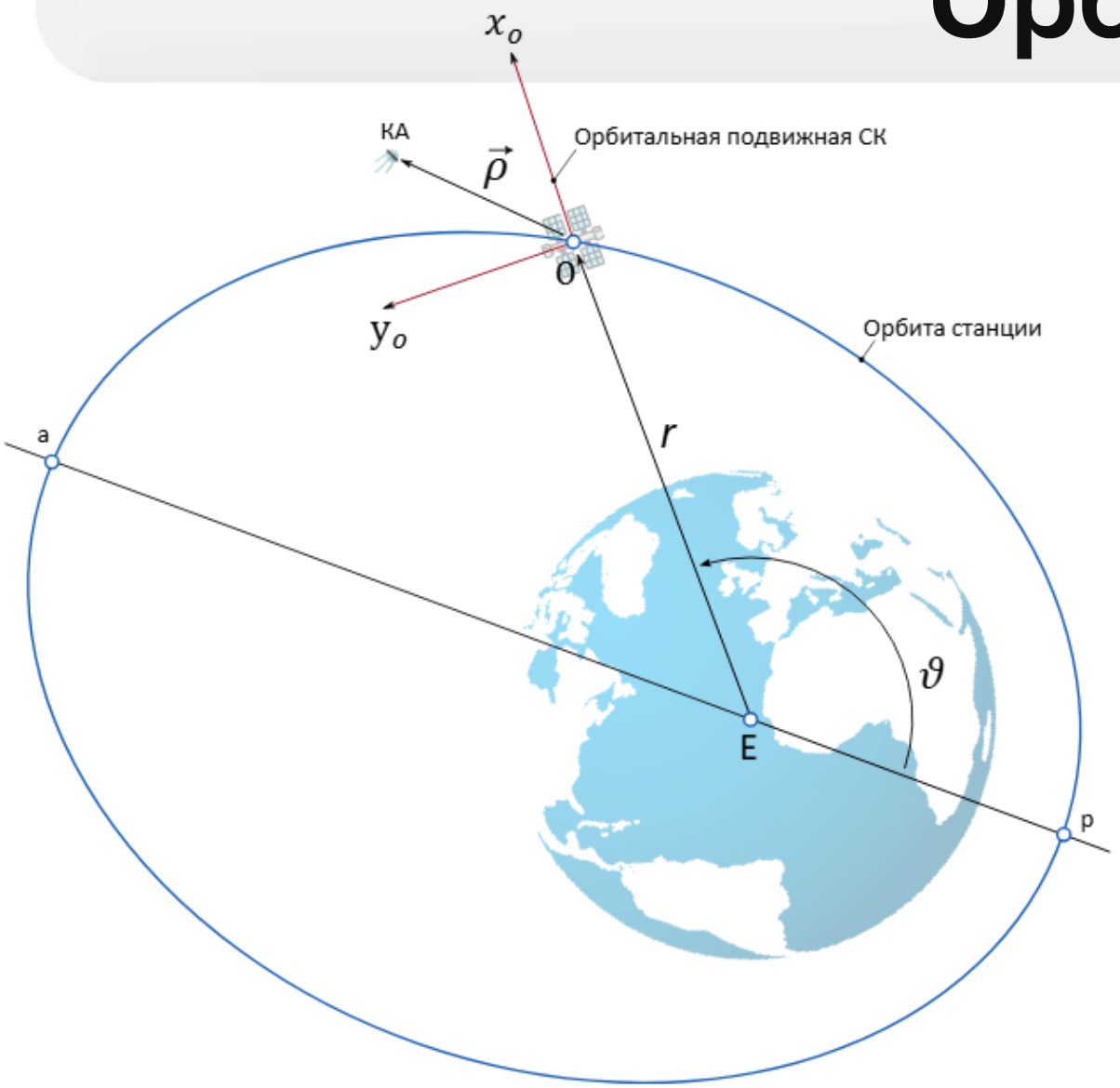


- Начало — в точке спутника (или точки на поверхности).
- Ось **N** — на географический север (касательно к меридиану).
- Ось **W** — на запад (касательно к параллели).
- Ось **U** — вверх, по геоцентрической (или геодезической) вертикали

$$A_{ECEF \to NWU} = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ \sin\lambda & -\cos\lambda & 0 \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{bmatrix}$$

кадры для космоса

Орбитальная подвижная

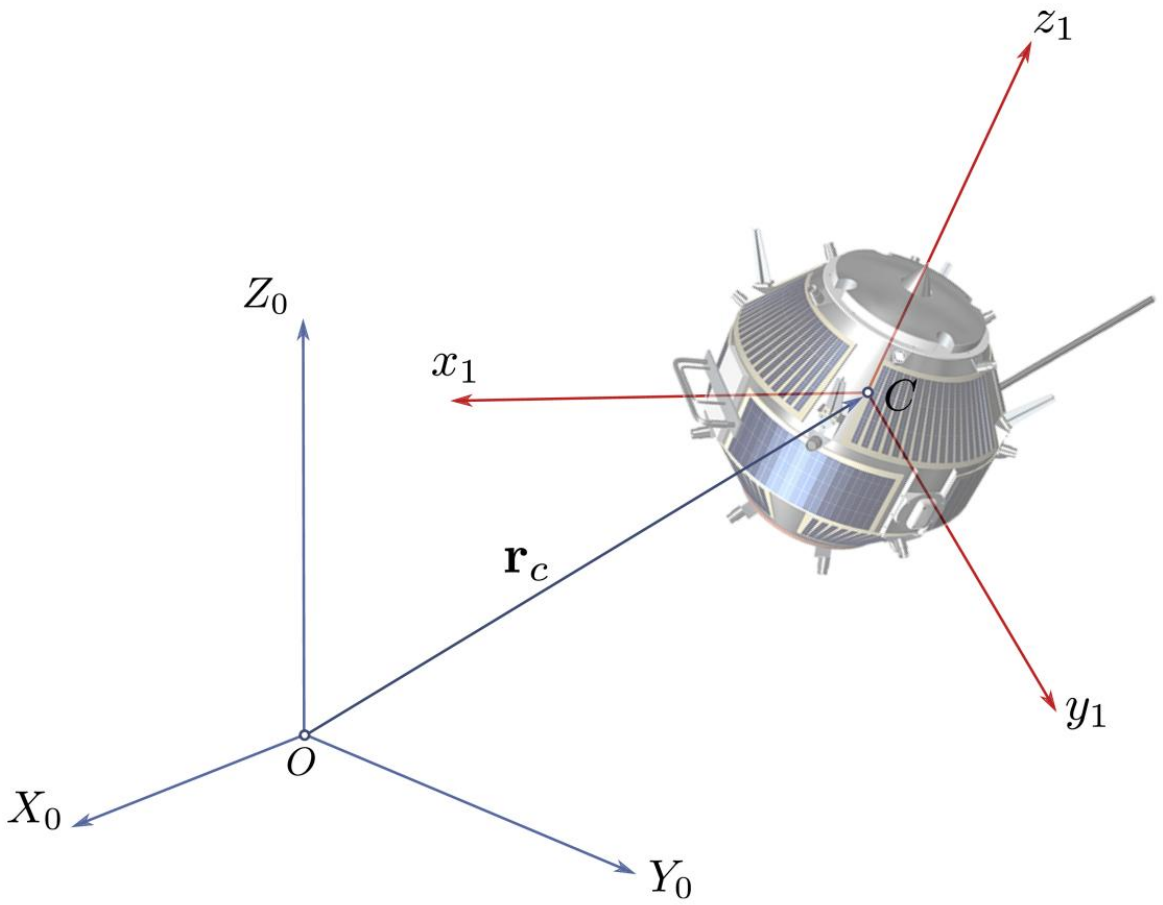


- Ox_o – (R): радиальное направление от Земли
- Oy_o – (T): трансверсальное направление
- Oz_o – (N): нормальное направление, перпендикулярное плоскости орбиты
- Для известных $\vec{r}$ и $\vec{V}$:

$$\vec{e}_{xo} = \frac{\vec{r}}{|\vec{r}|}, \qquad \vec{e}_{zo} = \frac{\vec{e}_{xo} \times \vec{V}}{|\vec{e}_{xo} \times \vec{V}|}, \qquad \vec{e}_{yo} = \vec{e}_{zo} \times \vec{e}_{xo}$$

кадры для космоса

Положение МКА

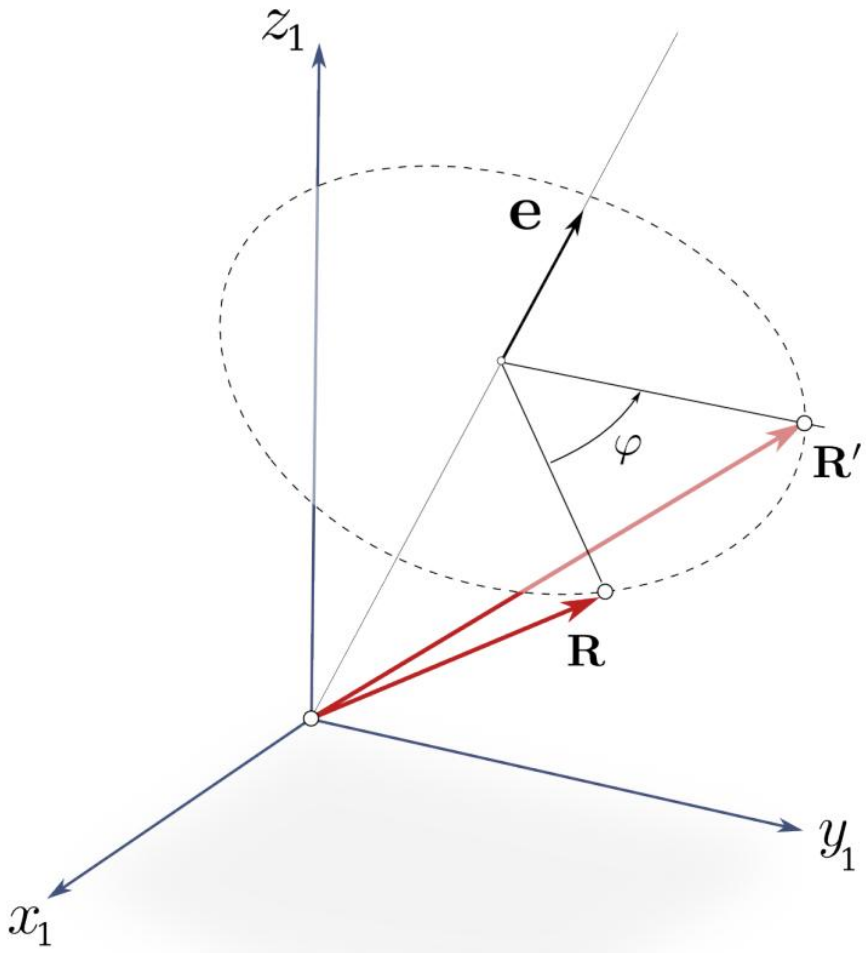


Движение МКА как твёрдого тела определяется

- положением центра масс относительно выбранной системы координат ($Ox_oy_oz_o$)
- скоростью центра масс относительно $Ox_oy_oz_o$
- ориентацией системы координат, связанной с МКА относительно системы координат $Ox_oy_oz_o$
- угловой скоростью МКА

кадры для космоса

Кватернионы



- Ориентация систем координат по отношению друг к другу может задаваться кватернионом, представляющим собой систему гиперкомплексных чисел
- Геометрическое представление кватерниона, описывающее преобразование поворота
  число + вектор

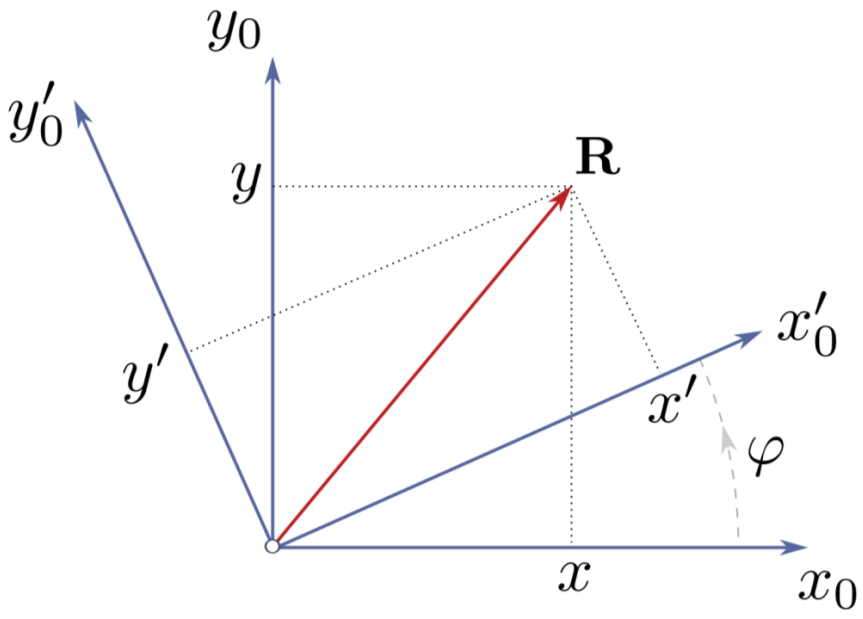
$$\Lambda = \cos\frac{\varphi}{2} + \boldsymbol{e}\sin\frac{\varphi}{2}$$

$\boldsymbol{e}$ – ось вращения (вектор), φ – угол поворота

- Координаты кватерниона

$$\Lambda = \left[\cos\frac{\varphi}{2}, e_x\sin\frac{\varphi}{2}, e_y\sin\frac{\varphi}{2}, e_z\sin\frac{\varphi}{2}\right]$$

Преобразование поворота



- Пусть система координат $Ox'_o y'_o z'_o$ повернута относительно системы координат $Ox_o y_o z_o$ вокруг оси z на угол $\varphi = 30\,°$
- Кватернион преобразования:

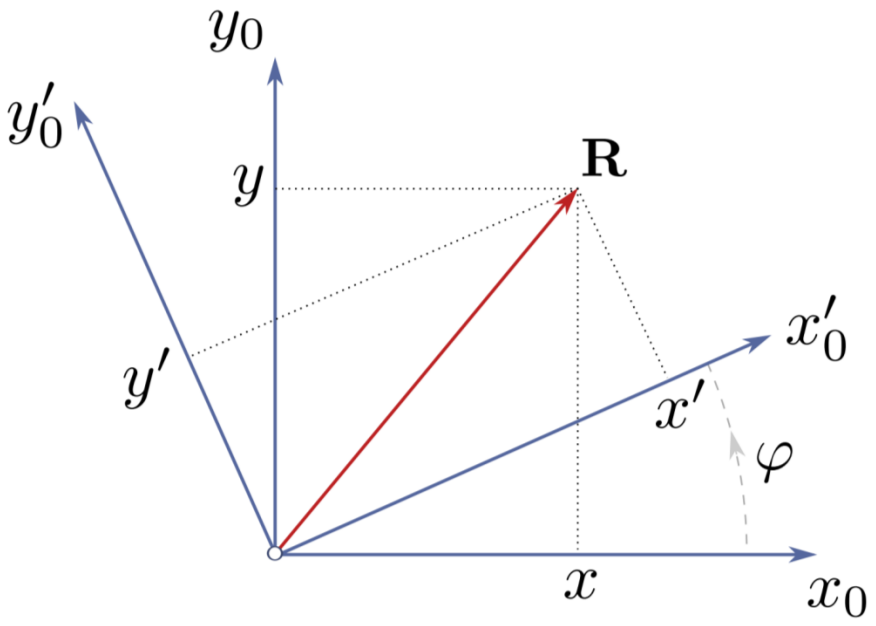
$$\mathbf{\Lambda} = \cos\frac{30\,°}{2} + \begin{bmatrix}0\\0\\1\end{bmatrix}\sin\frac{30\,°}{2}$$

- Координаты кватерниона:

$$\mathbf{\Lambda} = \left[\cos\frac{30\,°}{2}, 0, 0, \sin\frac{30\,°}{2}\right]$$

кадры для космоса

Преобразование поворота



$$\mathbf{\Lambda} = \left[\cos\frac{30\,^\circ}{2}, 0, 0, \sin\frac{30\,^\circ}{2}\right]$$

- Если известны координаты вектора **R** в повернутой системе координат

$$\boldsymbol{R}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

- Координаты вектора в исходной системе координат

$$\boldsymbol{R} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

определяются следующим преобразованием

$$\boldsymbol{R} = \mathbf{\Lambda} \circ \boldsymbol{R} \circ \overline{\mathbf{\Lambda}}$$

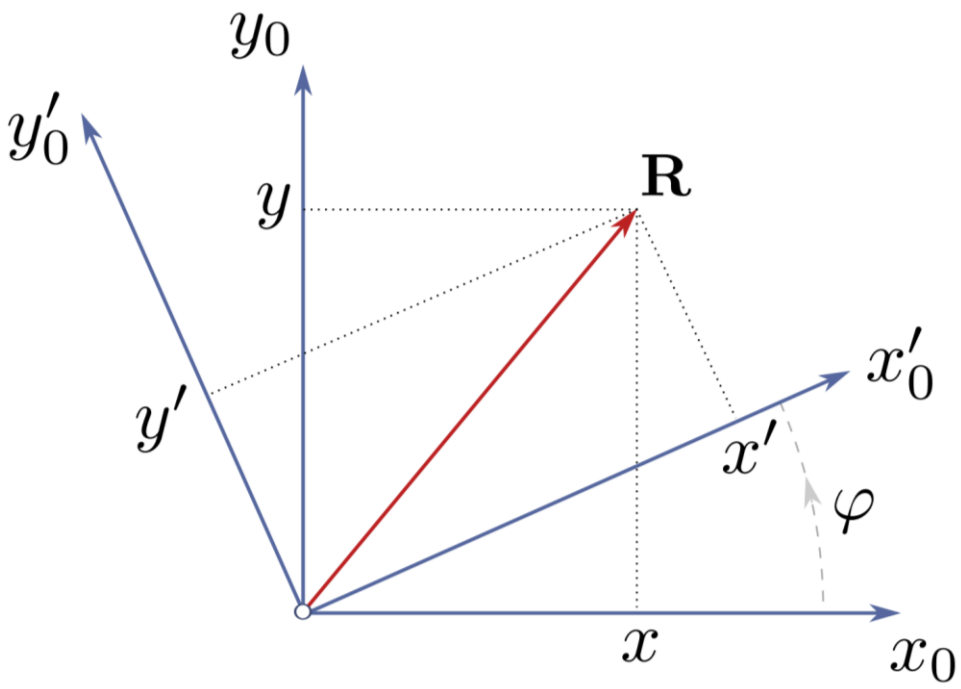
$\mathbf{\Lambda} \circ \boldsymbol{R}$ – произведение кватернионов

$\overline{\mathbf{\Lambda}}$ - сопряженный кватернион

$\boldsymbol{R}$ – в произведении кватернионов участвует как кватернион без скалярной части

кадры для космоса

Преобразование поворота



Вектор, совпадающий с осью Ox’ повернутой СК

```
phi = 30*pi/180;
lambda = [cos(phi/2),0,0,sin(phi/2)];
Rp  = [0,1,0,0];
R = quatmultiply(quatmultiply(lambda, Rp),quatconj(lambda));
R = quatgetvector(R);
```

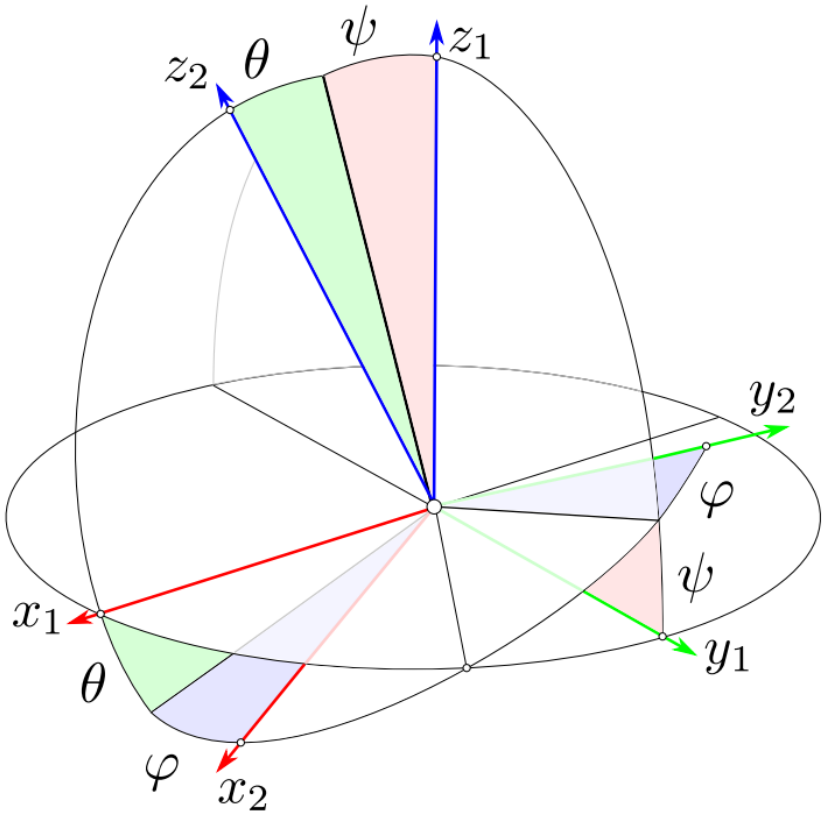
Результат

```
Rp  = [0, 0.866, 0.5, 0]
```

```
function quatrotate(quat,v)
  res = quatmultiply(quatmultiply(quat, [0 & v]), quatconj(quat));
  quatrotate = quatgetvector(res);
end;
```

кадры для космоса

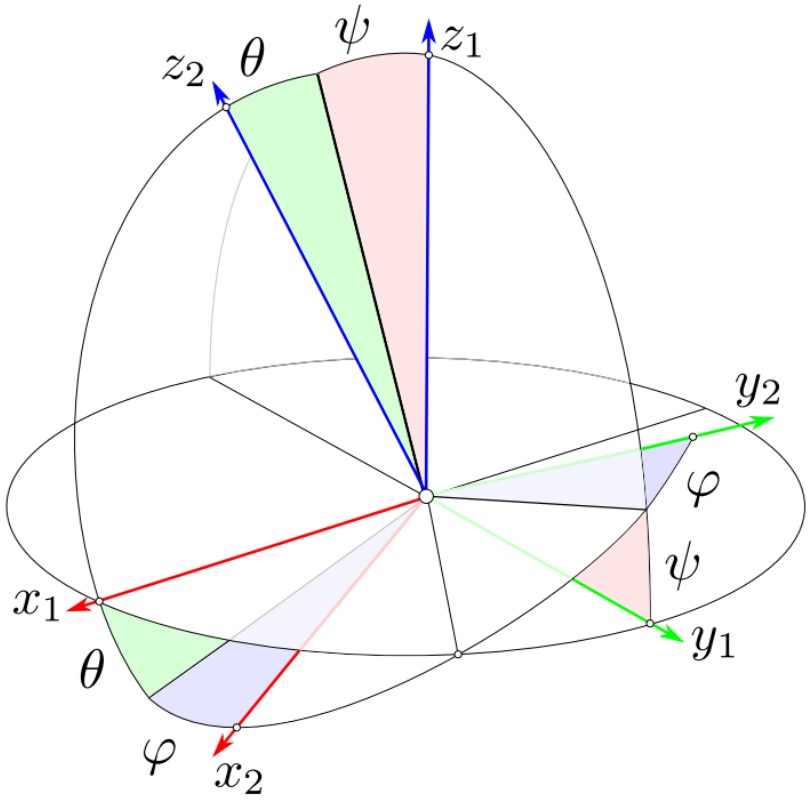
Сложный поворот



- Пусть система координат $Ox_2y_2z_2$повернута относительно системы координат $Ox_1y_1z_1$
- Поворот определяется тремя последовательными плоскими поворотами вокруг осей x, y и z на углы ϑ, ψ и φ соответственно
- Каждый новый поворот производится вокруг нового направления соответствующей оси (после предыдущего поворота)

кадры для космоса

Сложный поворот



- Кватернион первого поворота

$$\mathbf{\Lambda}_\theta = \left[\cos\frac{\theta}{2}, \sin\frac{\theta}{2}, 0, 0\right]$$

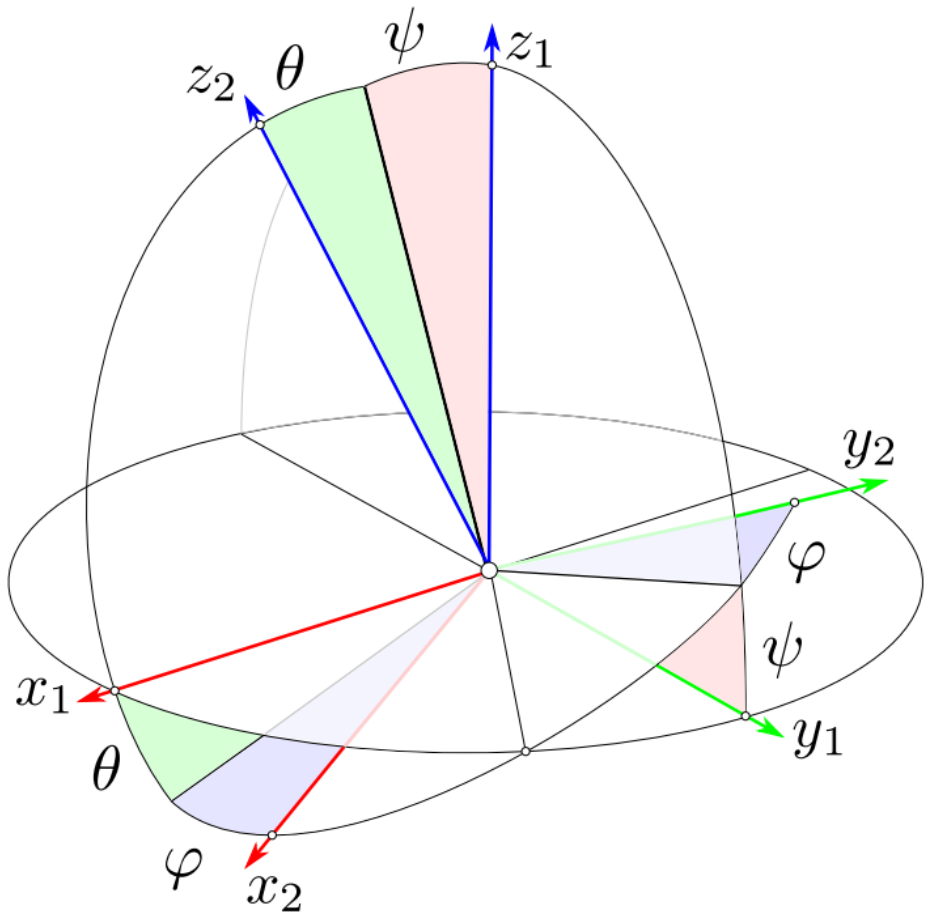
- Кватернион второго поворота

$$\mathbf{\Lambda}_\psi = \left[\cos\frac{\psi}{2}, 0, \sin\frac{\psi}{2}, 0\right]$$

- Кватернион третьего поворота

$$\mathbf{\Lambda}_\varphi = \left[\cos\frac{\varphi}{2}, 0, 0, \sin\frac{\varphi}{2}\right]$$

Сложный поворот



- Кватернион сложного поворота

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_\theta \circ \boldsymbol{\Lambda}_\psi \circ \boldsymbol{\Lambda}_\varphi$$

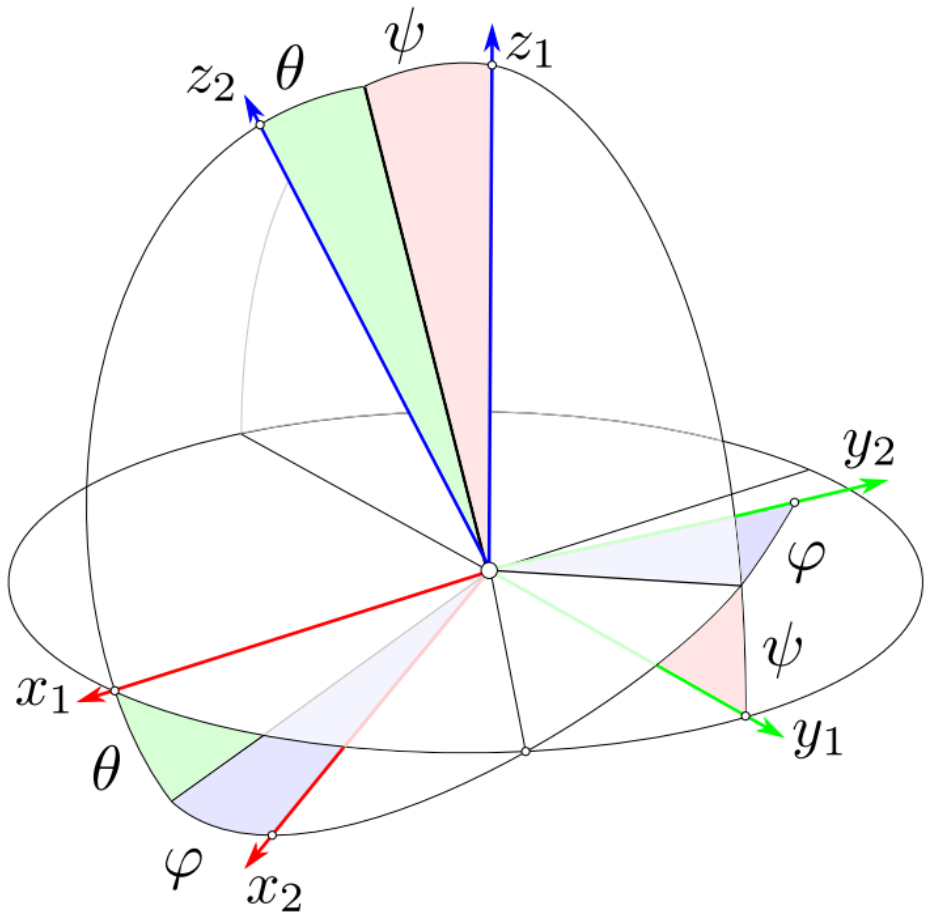
- Преобразование координат вектора из системы координат $Ox_2y_2z_2$ в систему координат $Ox_1y_1z_1$

$$\boldsymbol{R}^{(1)} = \boldsymbol{\Lambda} \circ \boldsymbol{R}^{(2)} \circ \overline{\boldsymbol{\Lambda}}$$

```
psi = 1;
theta = 2;
phi = 3;

q1 = [cos(psi/2),sin(psi/2),0,0]
q2 = [cos(theta/2),0,sin(theta/2),0]
q3 = [cos(phi/2),0,0,sin(phi/2)]
q  = quatmultiply(q1,quatmultiply(q2,q3));

r_in_2 = [1,1,1];
r_in_1 = quatrotate(q, r_in_2);
```

кадры для космоса

Сложный поворот - матрицы



- Матрица первого поворота

$$\boldsymbol{A}_{\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

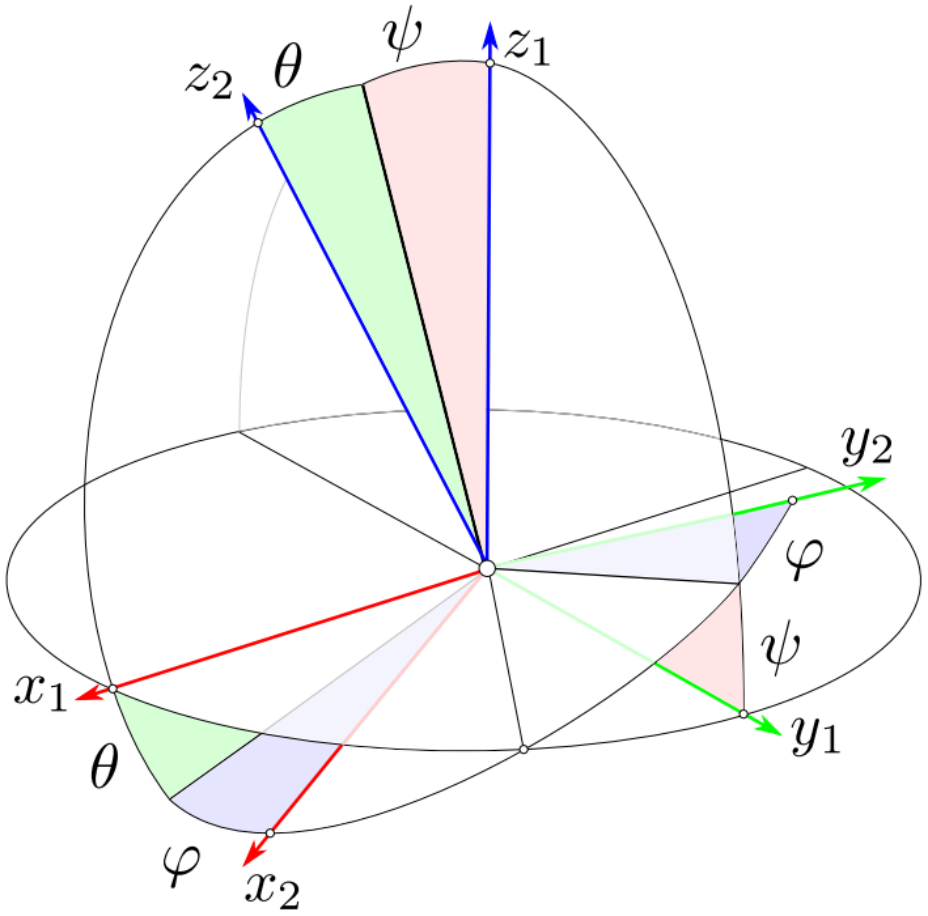
- Матрица второго поворота

$$\boldsymbol{A}_{\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

- Матрица третьего поворота

$$\boldsymbol{A}_{\varphi} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Сложный поворот. Матрицы



- Матрица сложного поворота

$$\boldsymbol{A} = \boldsymbol{A}_\theta \boldsymbol{A}_\psi \boldsymbol{A}_\varphi$$

- Преобразование координат вектора из системы координат $Ox_2y_2z_2$ в систему координат $Ox_1y_1z_1$

$$\boldsymbol{R}^{(1)} = \boldsymbol{A}\boldsymbol{R}^{(2)}$$

```
psi = 1;
theta = 2;
phi = 3;
A1 = rmatx(psi);
A2 = rmaty(theta);
A3 = rmatz(phi);
r_in_2 = [1,1,1];
r_in_1 = A1*A2*A3*transp([r_in_2]);
// или
r_in_2 = [[1],[1],[1]];
r_in_1 = A1*A2*A3*r_in_2;
```

кадры для космоса

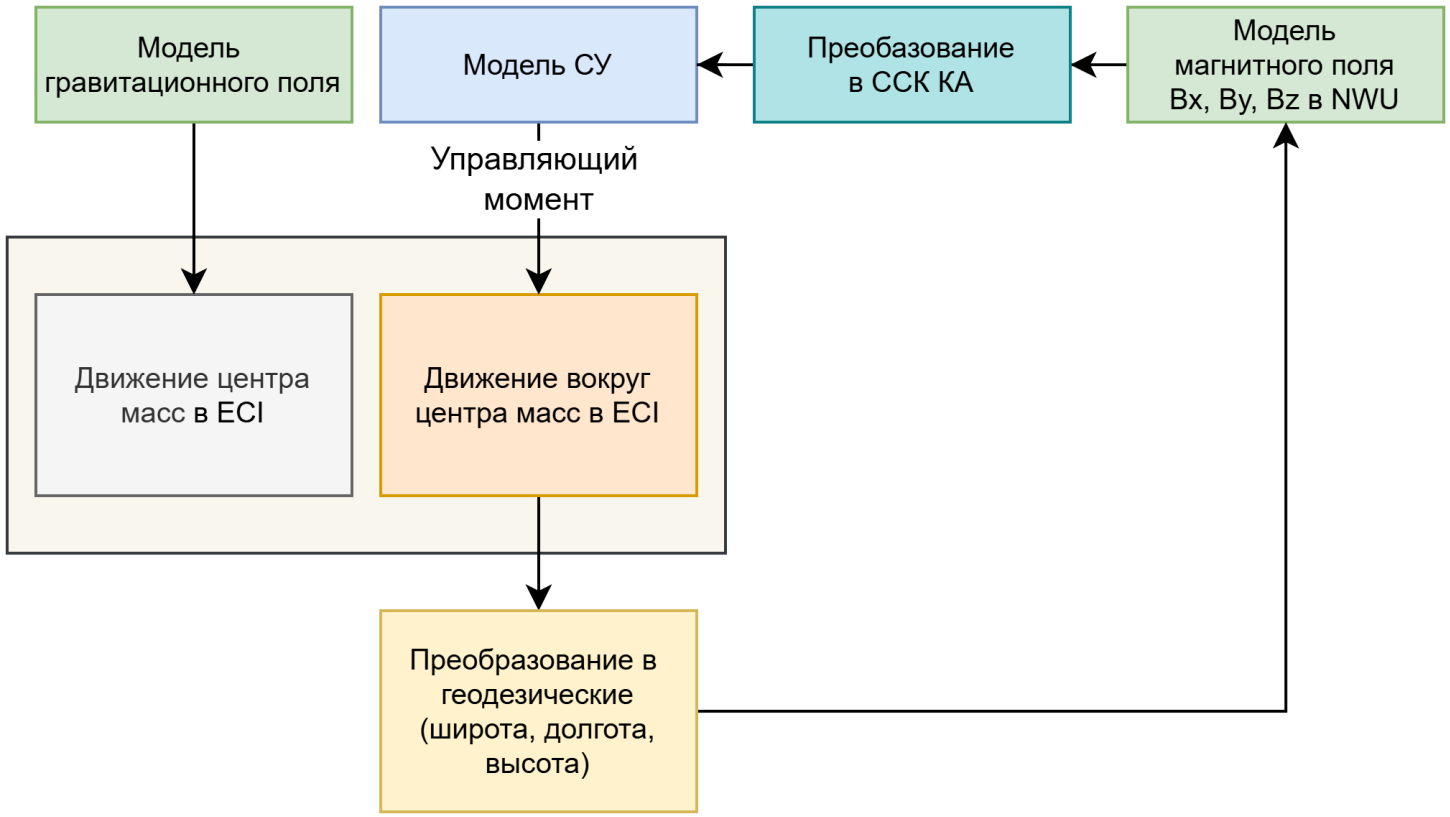
Кинематические уравнения

$$\begin{pmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\tilde{\boldsymbol{\omega}} \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}$$

$$\tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

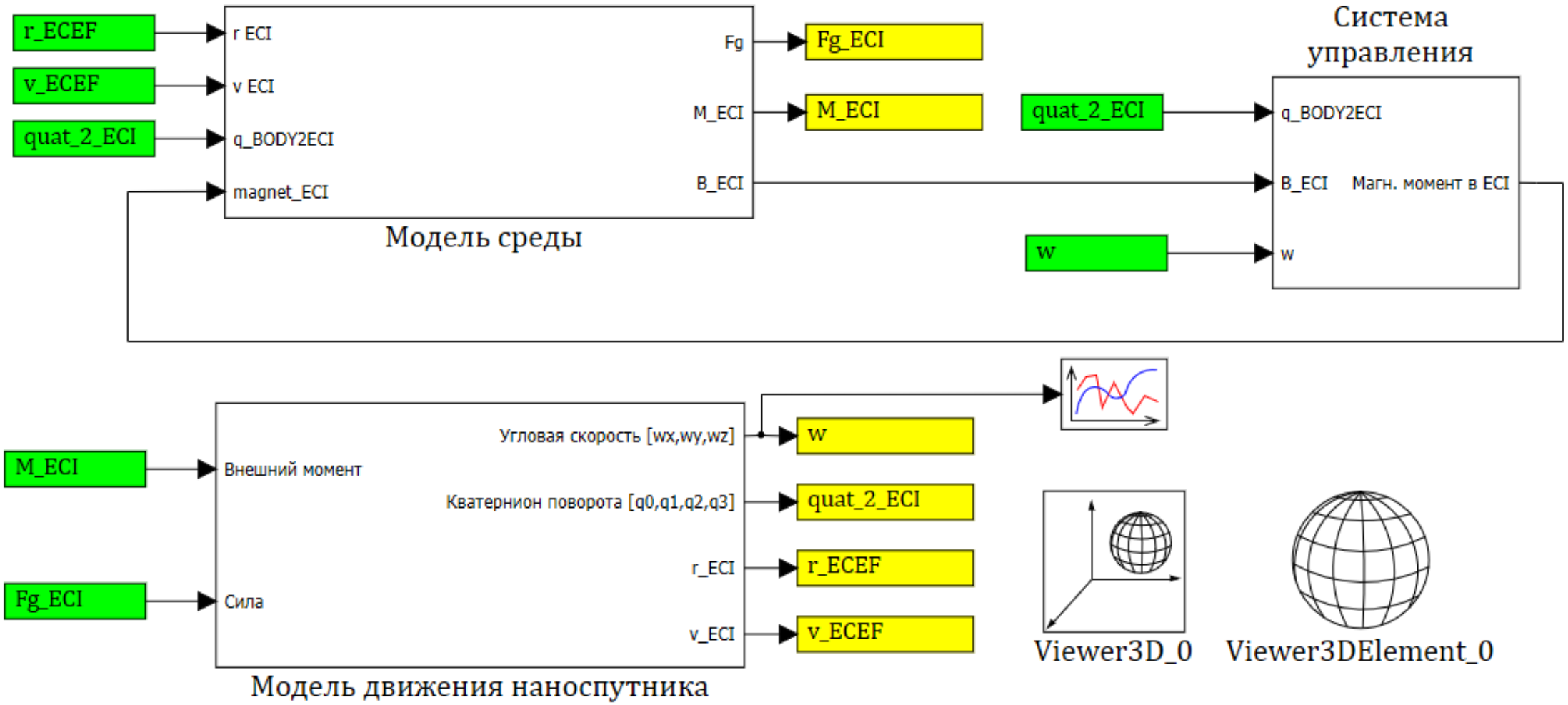
- Кинематические уравнения определяют связь между угловой скоростью тела и производными параметров, определяющих его ориентацию
- Кинематические уравнения интегрируются вместе с динамическими уравнениями для определения углового положения тела (углов Эйлера, матрицы поворота, кватерниона)

Модель движения МКА

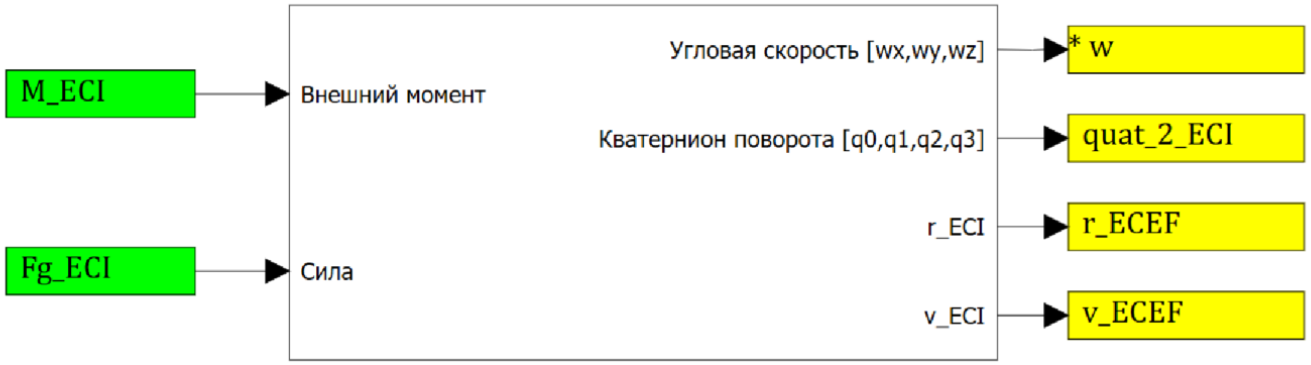


- Модель движения
  - Движение центра масс
  - Движение вокруг центра масс
- Модель магнитного поля Земли
- Модель системы управления
- Модули преобразования координат
- 3D анимация углового движения наноспутника в орбитальной системе координат

Модель движения МКА



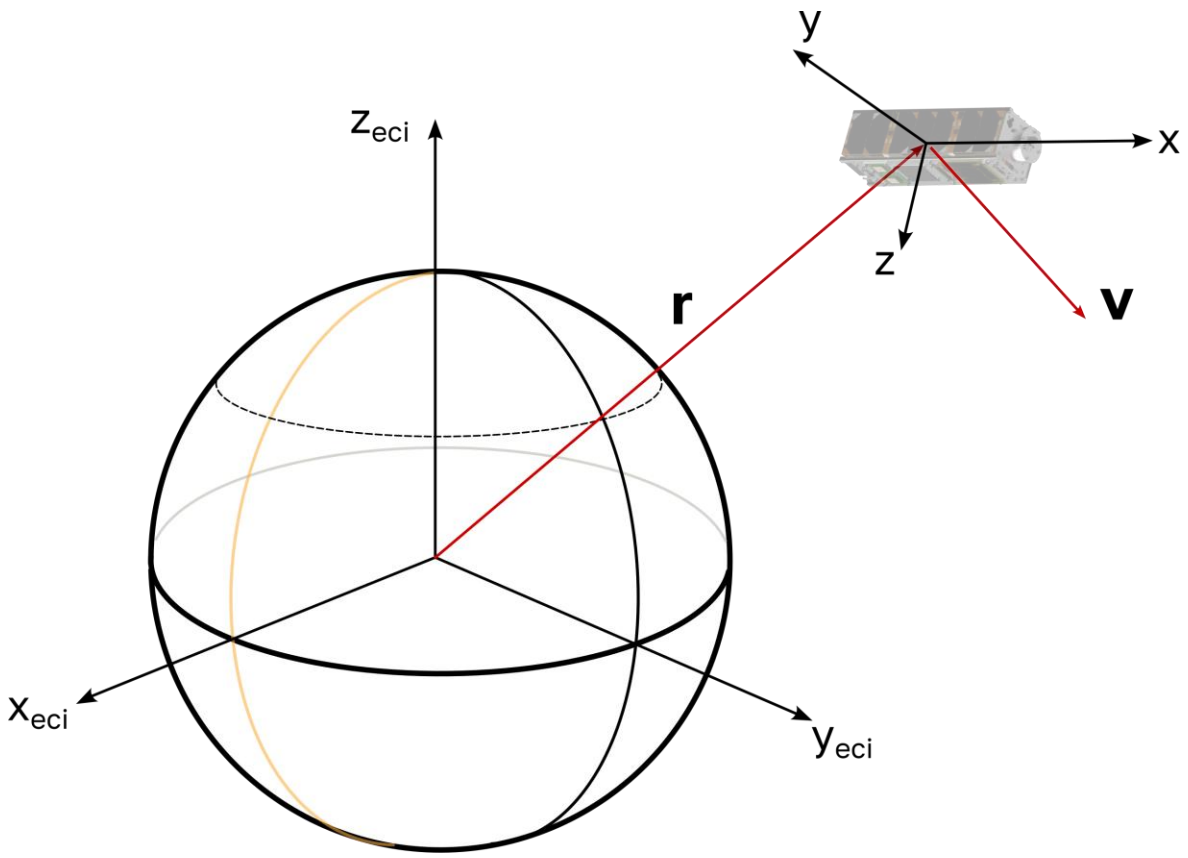
Модель движения МКА



Модель движения наноспутника

- Движение рассматривается относительно геоцентрической инерциальной системы координат
- Угловое положение наноспутника задается кватернионными параметрами
- Движение спутника определяется
  - координатами и скоростью центра масс в ECI
  - кватернионом, определяющим ориентацию связанной СК относительно ECI
  - проекциями угловой скорости на оси связанной системы координат

Уравнения движения



- Уравнение движения центра масс

$$m\,\ddot{\boldsymbol{r}}^{(eci)} = \boldsymbol{F}_g = -m\frac{\mu}{r^3}\boldsymbol{r}^{(eci)}$$

- Уравнение движения вокруг центра масс

$$\boldsymbol{J}^{(c)}\dot{\boldsymbol{\omega}} = \boldsymbol{M}^{(c)} - \boldsymbol{\omega}^{(c)}\times\boldsymbol{J}^{(c)}\boldsymbol{\omega}^{(c)}$$

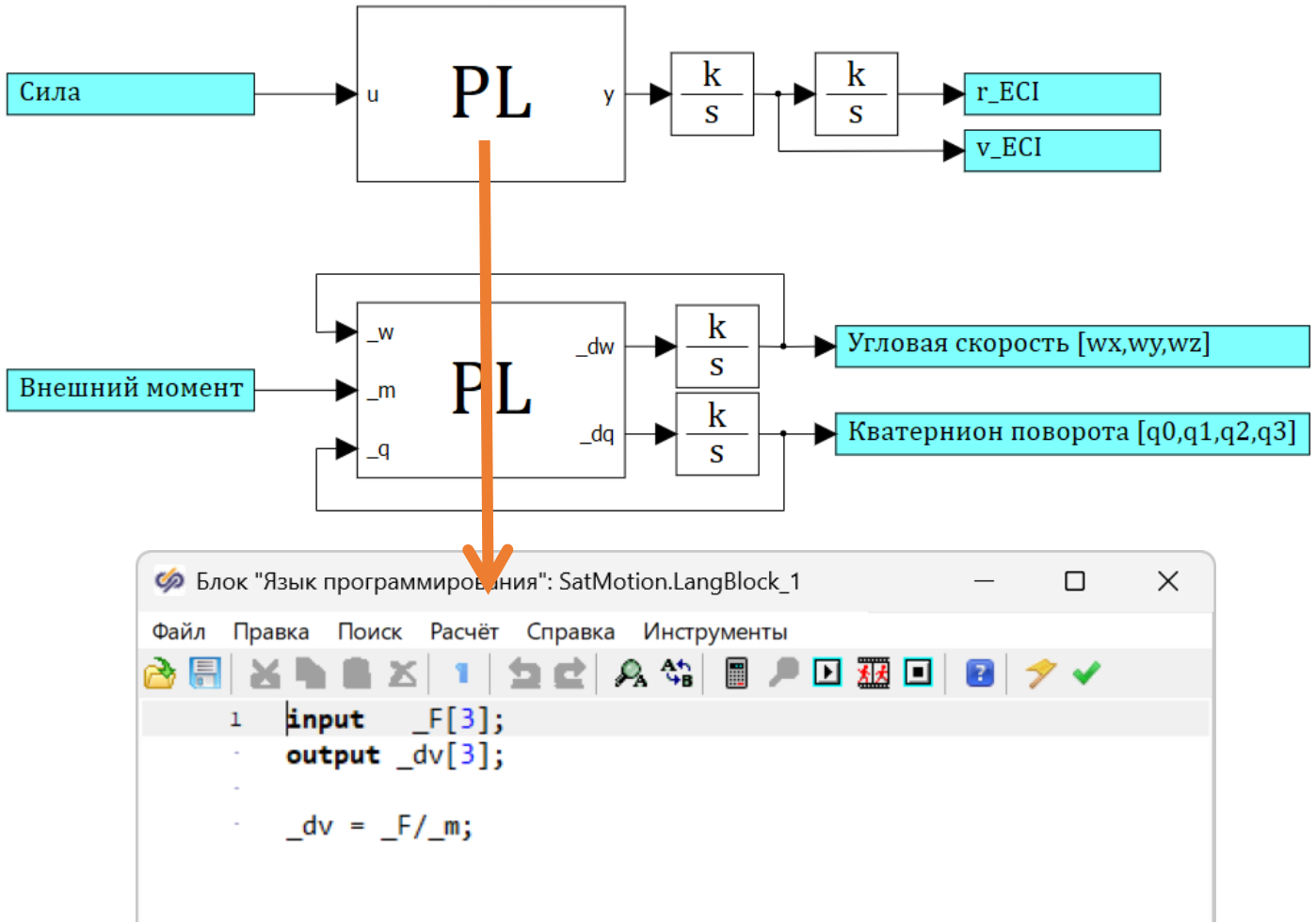
- Кинематические уравнения для кватернионов

$$\begin{pmatrix}\dot{\lambda}_0\\ \dot{\lambda}_1\\ \dot{\lambda}_2\\ \dot{\lambda}_3\end{pmatrix} = \frac{1}{2}\begin{pmatrix}0 & -\boldsymbol{\omega}^T\\ \boldsymbol{\omega} & -\tilde{\boldsymbol{\omega}}\end{pmatrix}\begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\end{pmatrix}$$

$$\tilde{\boldsymbol{\omega}} = \begin{pmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{pmatrix}$$

кадры для космоса

Движение центра масс



- Уравнение движения центра масс записывается в геоцентрической инерциальной СК

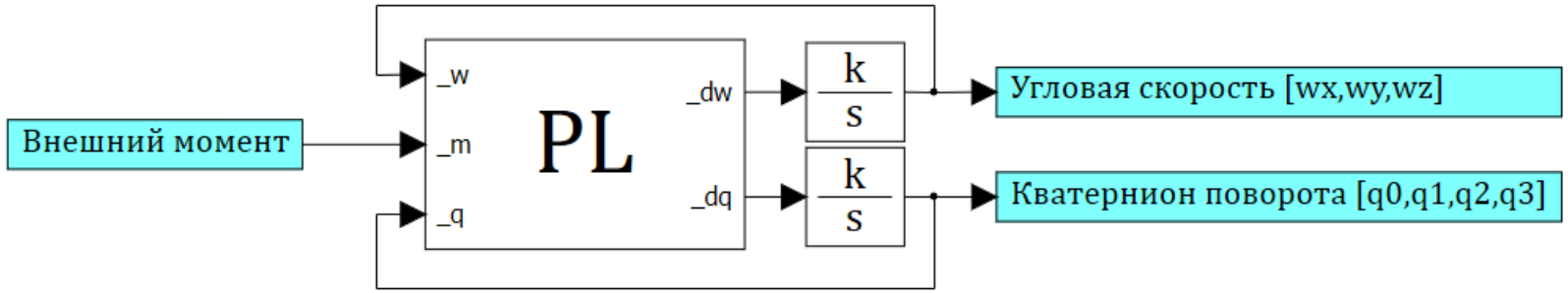
$$\boldsymbol{a}^{(eci)} = \frac{\boldsymbol{F}^{(eci)}}{m}$$

$\boldsymbol{F}^{(eci)}$ - координатный столбец главного вектора внешних сил в геоцентрической инерциальной СК

- Рассматривается действие только силы притяжения Земли (центральное поле)

кадры для космоса

Движение вокруг центра масс



$$\begin{pmatrix}\dot\lambda_0\\ \dot\lambda_1\\ \dot\lambda_2\\ \dot\lambda_3\end{pmatrix} = \frac{1}{2}\begin{pmatrix}0 & -\boldsymbol{\omega}^T\\ \boldsymbol{\omega} & -\tilde{\boldsymbol{\omega}}\end{pmatrix}\begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\end{pmatrix}$$

Блок "Язык программирования": SatMotion.LangBlock_0

Файл Правка Поиск Расчёт Справка Инструменты

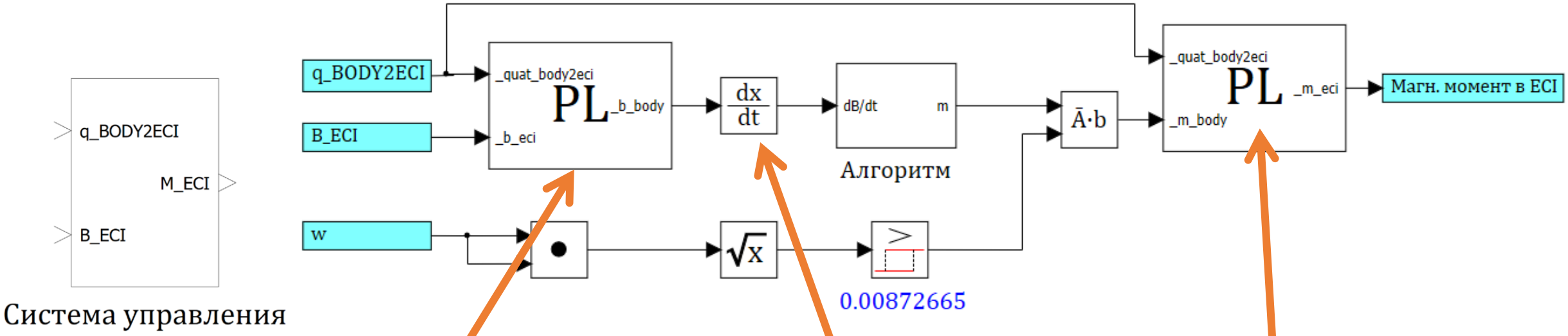
```
input  _w[3],   _M[3], _q[4];
output _dw[3], _dq[4];

// Динамические уравнения
_L    = [0,0,0];
_L    = _J*transp([_w]);
_dw   = _invJ*transp([_M-cross(_w,_L)]);

// Кинематические уравнения
_mat = [[0,-_w[1],-_w[2],-_w[3]],[_w[1],0,-_w[3],_w[2]],[_w[2],_w[3],0,-_w[1]],[_w[3],-_w[2],_w[1],0]];
_dq  = 0.5*_mat*transp([_q]);
```

$$\dot{\boldsymbol{\omega}}^{(c)} = \left[\boldsymbol{J}^{(c)}\right]^{-1}\left[\boldsymbol{M}^{(c)} - \boldsymbol{\omega}^{(c)}\times \boldsymbol{J}^{(c)}\boldsymbol{\omega}^{(c)}\right]$$

1: 1

кадры для космоса

Модель системы управления



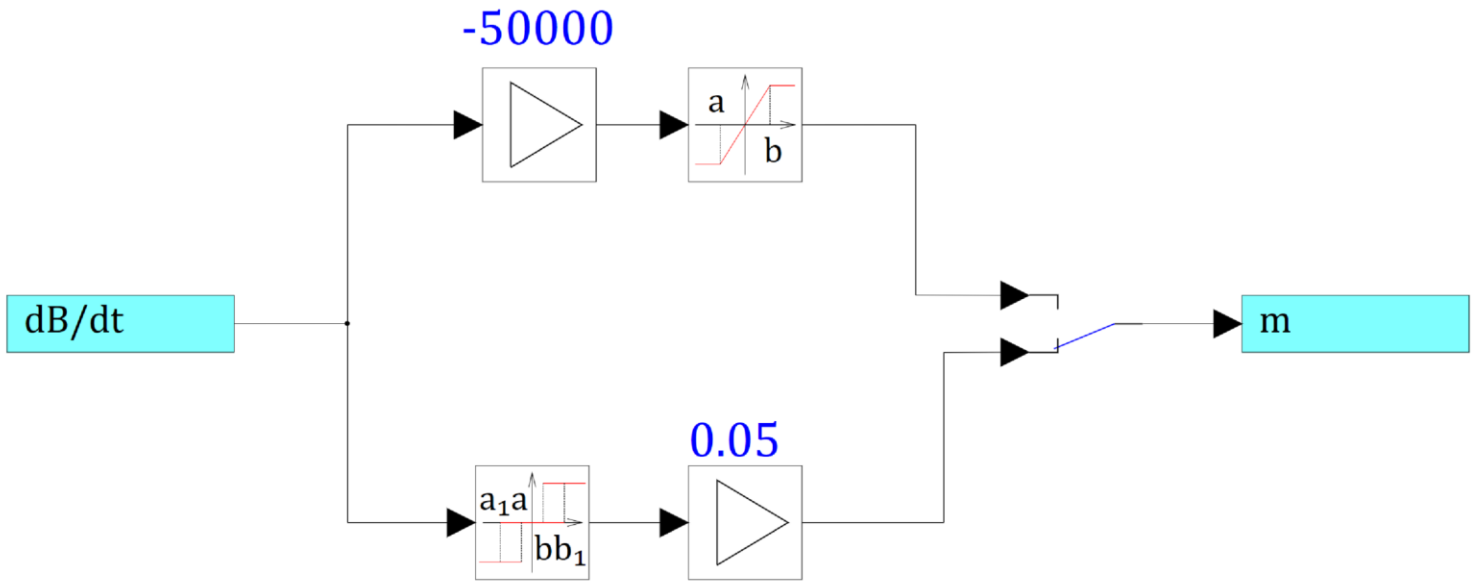
Преобразование вектора B из ECI в связанную СК

Дифференцирование компонент вектора

Преобразование магнитного момента из связанной СК в ECI

кадры для космоса

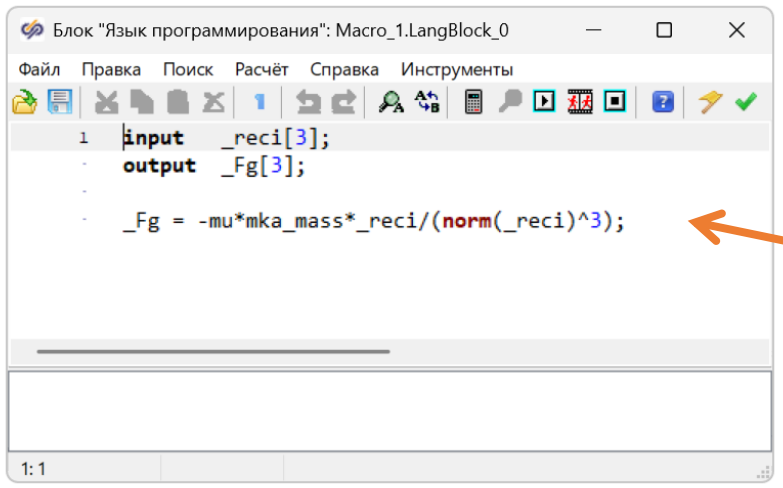
Алгоритм формирования момента

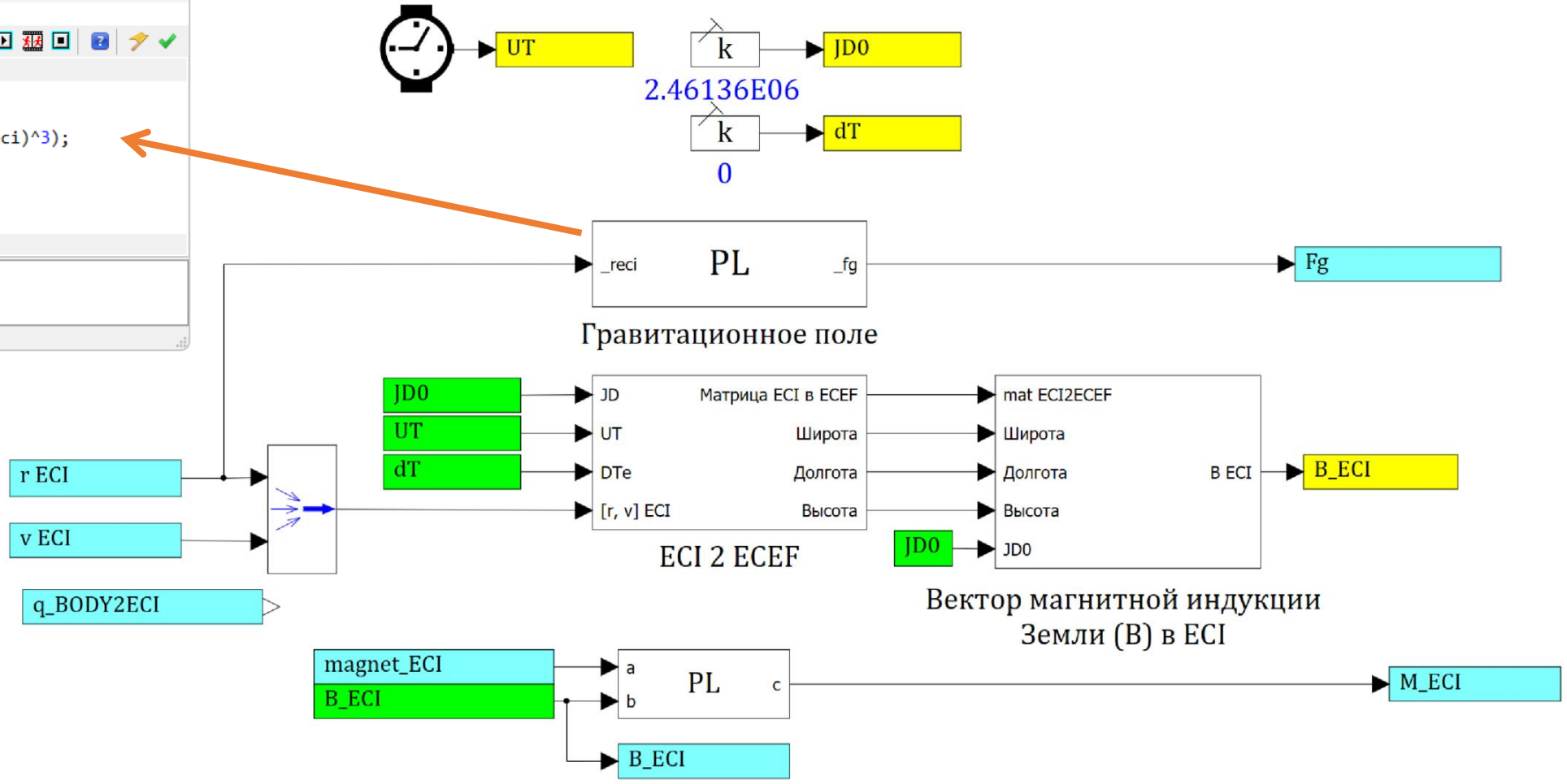


А. Момент, пропорциональный производной $\mathrm{dB/dt}$ с ограничением максимального значения

Б. Момент принимает два значения - вкл. и выкл. в зависимости от знака производной.

Модель внешней среды



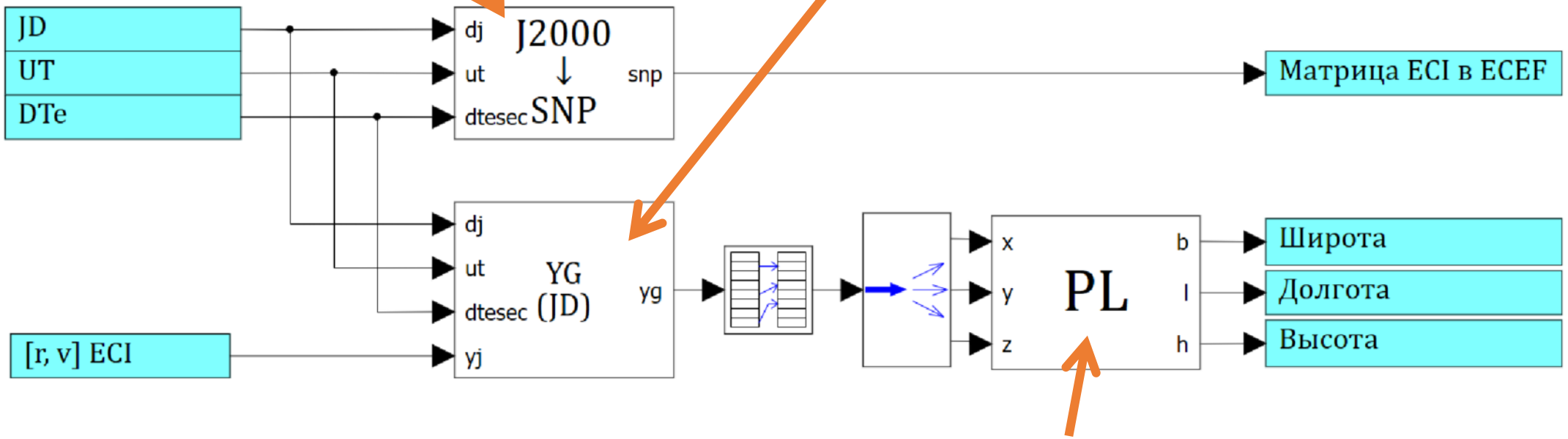


кадры для космоса

Преобразование систем координат

Матрица преобразования из геоцентрической инерциальной в гринвичскую (ECEF)

Преобразование координат и скорости КА из ECI в ECEF



Вычисление широты, долготы и высоты по прямоугольным координатам

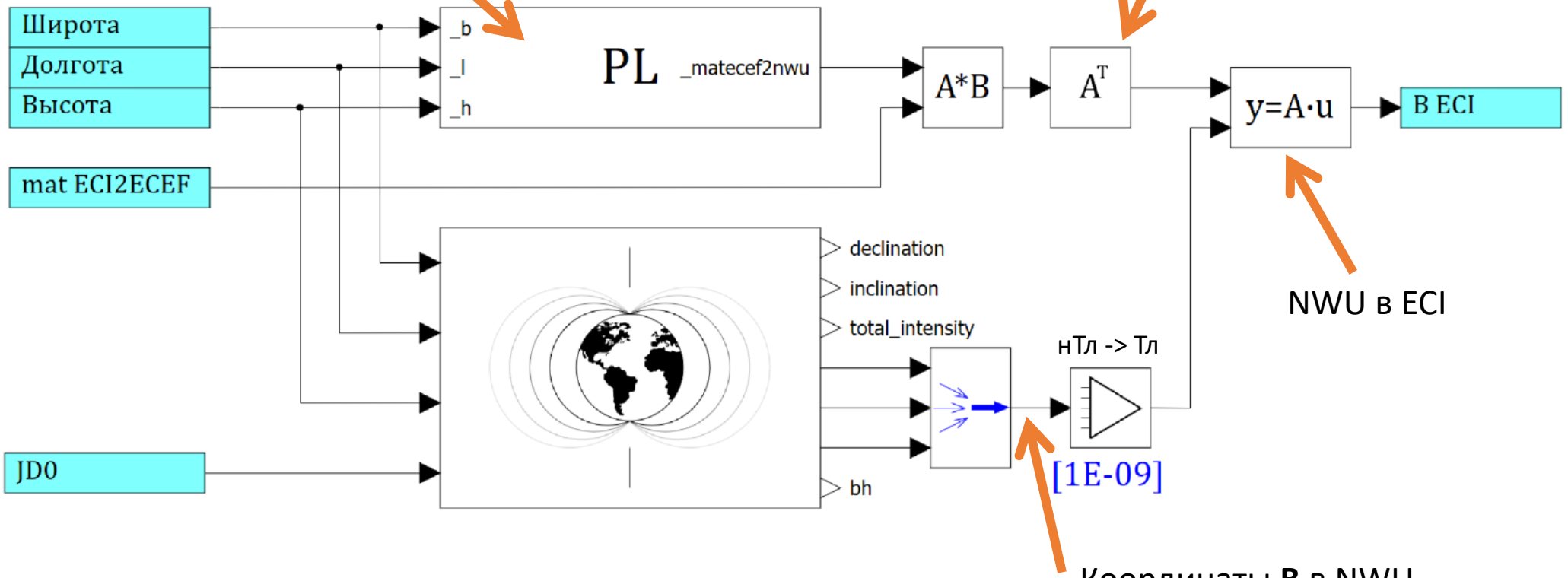
кадры для космоса

Компоненты вектор B в ECI

```
input  _b, _l, _h;
output  _matECEF2NWU[3,3];
_matECEF2NWU[1,1] = -sin(_b)*cos(_l);
_matECEF2NWU[1,2] = -sin(_b)*sin(_l);
_matECEF2NWU[1,3] =  cos(_b);
_matECEF2NWU[2,1] =  sin(_l);
_matECEF2NWU[2,2] = -cos(_l);
_matECEF2NWU[2,3] =  0;
_matECEF2NWU[3,1] =  cos(_b)*cos(_l);
_matECEF2NWU[3,2] =  cos(_b)*sin(_l);
_matECEF2NWU[3,3] =  sin(_b);
```

Матрица перехода из ECEF в NWU

ECI в NWU



NWU в ECI

Координаты **B** в NWU

Скрипт проекта

Скрипт страницы : Главная страница: C:\Users\kidin\Documents\Заметки\Университет\ИОТ-Конкурс\Мод...

Файл Правка Поиск Расчёт Справка Инструменты

```
// Преобразование поворота
function quatrotate(quat,v)
  res = quatmultiply(quatmultiply(quat, [0 & v]), quatconj(quat));
  quatrotate = quatgetvector(res);
end;
// Преобразование кеплеровых элементов орбиты в геоцентрическую инерциальную (ECI)
// a - большая полуось
// e - эксцентрисистет
// raan - долгота восходящего узла
// ap   - аргумент перицентра
// incl - наклонение орбиты
// true_anomaly - угол истинной аномалии
function kep2eci(a, e, raan, ap, incl, true_anomaly, mu)
  r_eci = [0,0,0]; v_eci = [0,0,0];
  p  = a*(1-e*e);
  r  = p/(1+e*cos(true_anomaly));
  vr = sqrt(mu/p)*e*sin(true_anomaly);
  vn = sqrt(mu/p)*(1+e*cos(true_anomaly));
  er    = [0,0,0];
  u     = true_anomaly+ap;
  er[1] = cos(raan)*cos(u)-sin(raan)*sin(u)*cos(incl);
  er[2] = sin(raan)*cos(u)+cos(raan)*sin(u)*cos(incl);
  er[3] = sin(u)*sin(incl);
  en    = [0,0,0];
  en[1] =-cos(raan)*sin(u)-sin(raan)*cos(u)*cos(incl);
  en[2] =-sin(raan)*sin(u)+cos(raan)*cos(u)*cos(incl);
  en[3] = cos(u)*sin(incl);
  r_eci = r*er;
  v_eci = vr*er+vn*en;
  kep2eci = r_eci & v_eci;
end;
// Матрица преобразования из орбитальной RTN в ECI
function mat_LVLH2ECI(r,v)
  mat_LVLH2ECI = matrix(3,3);
  er = r/norm(r);
  en = cross(er,v)/norm(cross(er,v));
  et = cross(en,er);
  mat_LVLH2ECI = transp([ er, et, en ]);
end;
```

40: 33 Изменено

Скрипт страницы : Главная страница: C:\Users\kidin\Documents\Заметки\Университет\И...

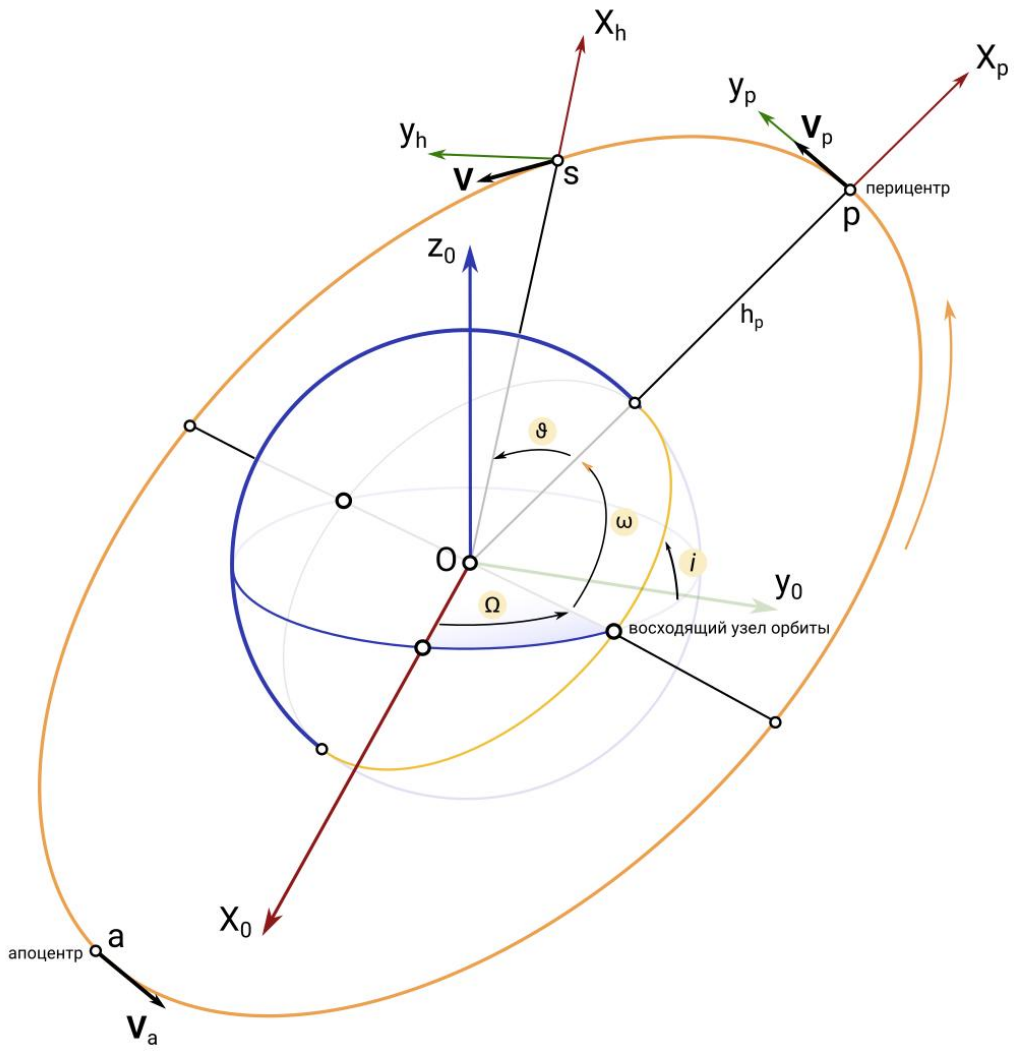
Файл Правка Поиск Расчёт Справка Инструменты

```
initialization
  // Радиус Земли
  Rearth = 6371.0e3;
  // Гравитационный параметр Земли
  mu = 398600.4415e9;
  // Масса КА
  mka_mass = 1.0;
  // Начальные условия в Кеплеровых элементах орбиты
  hp = 350e3; rp = Rearth + hp;  // Высота и радиус перигея
  ha = 350e3; ra = Rearth + ha;  // Высота и радиус апогея
  a  = (rp+ra)*0.5;              // Большая полуось орбиты
  raan = 0;                      // Долгота восходящего узла
  ap   = 0.7;                    // Аргумент перицентра
  incl = 85*pi/180;              // Наклонение орбиты
  true_anomaly = 0;              // Угол истинной аномалии
  e  = (ra - rp)/(rp + ra);      // Эксцентриситет
  // Преобразование ECI
  rv = kep2eci(a, e, raan, ap, incl, true_anomaly, mu);
  r_eci = [rv[1],rv[2],rv[3]];
  v_eci = [rv[4],rv[5],rv[6]];
  // Орбитальная СК
  mat  = mat_LVLH2ECI(r_eci,v_eci);
  quat_LVLH2ECI  = rmat2quat(mat);
  // Начальная ориентация в орбитальной системе координат
  quat_BODY2LVLH = [cos(0.5*pi/2+0.000001),0,sin(0.5*pi/2+0.000001),0];
  // Начальная ориентация в ECI
  quat_BODY2ECI  = quatmultiply(quat_LVLH2ECI,quat_BODY2LVLH);
  // Начальная угловая скорость КА в связанной СК
  w0 = [0.1,-0.3,0.2];
end;
```

40: 33 Изменено

кадры для космоса

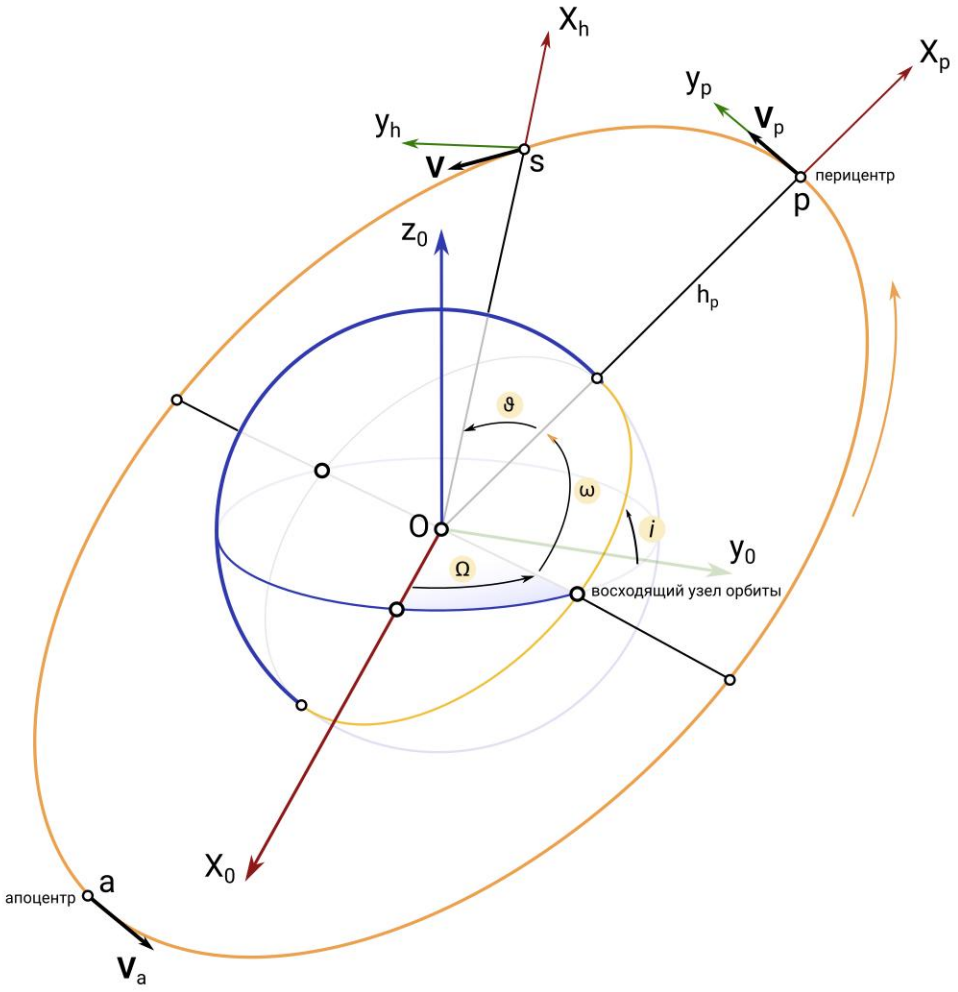
Кеплеровы элементы орбиты



- Долгота восходящего узла - **Поворот** линии узлов в плоскости экватора (от оси X ECI)
- Наклонение орбиты – угол между плоскостью орбиты и плоскостью экватора
- Аргумент перигея – угол в плоскости орбиты от линии узлов до линии апсид
- Угол истинной аномалии – от линии апсид до направления на КА
- Большая полуось орбиты - **размер** орбиты
- Эксцентриситет - **форма** орбиты (0 = круг, 0<e<1 = эллипс)

кадры для космоса

Преобразование



$$p = a(1 - e^2), \quad r = \frac{p}{1 + \cos\vartheta}$$

Радиальная и трансверсальная скорости

$$v_r = \sqrt{\frac{\mu}{p}}\, e \sin\vartheta\,, \quad v_n = \sqrt{\frac{\mu}{p}}\,(1 + e\cos\vartheta),$$

Единичный вектор нормального направления

$$\boldsymbol{e}_r = \begin{bmatrix} \cos\Omega\cos u - \sin\Omega\sin u\cos i \\ \sin\Omega\cos u + \cos\Omega\sin u\cos i \\ \sin u\sin i \end{bmatrix}$$

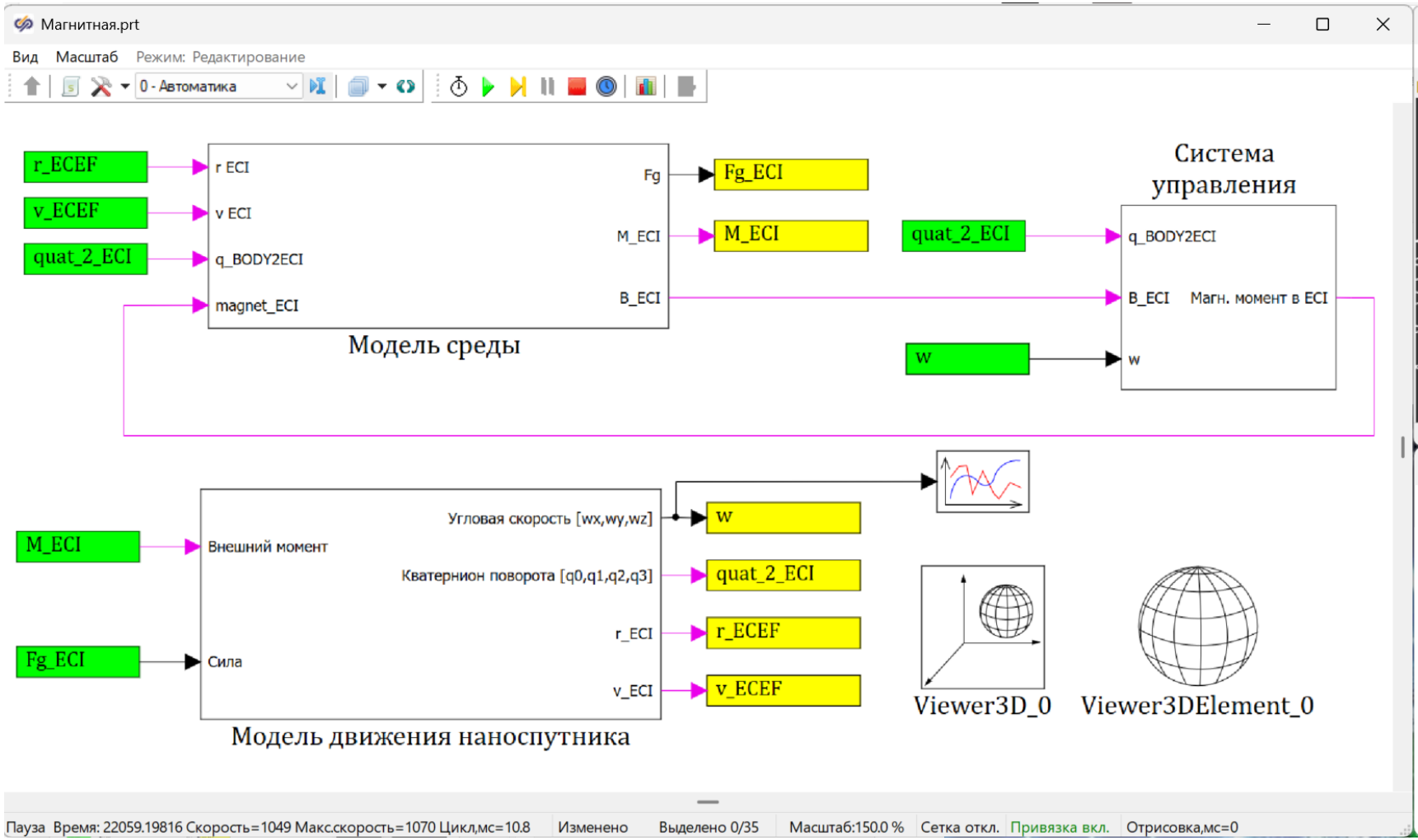
Единичный вектор трансверсального направления

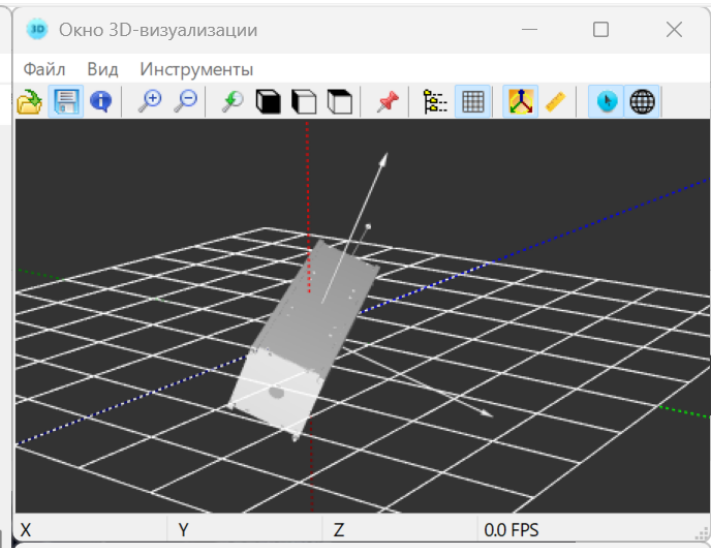
$$\boldsymbol{e}_n = \begin{bmatrix} -\cos\Omega\sin u - \sin\Omega\cos u\cos i \\ -\sin\Omega\sin u + \cos\Omega\cos u\cos i \\ \cos u\sin i \end{bmatrix}$$

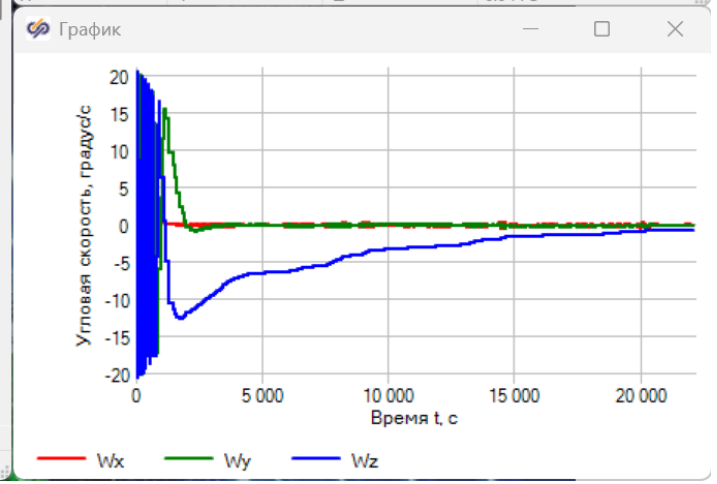
$$r^{(ECI)} = r\,\boldsymbol{e}_r, \boldsymbol{V}^{(ECI)} = v_r\,\boldsymbol{e}_r + v_n\,\boldsymbol{e}_n$$

кадры для космоса

Результаты моделирования

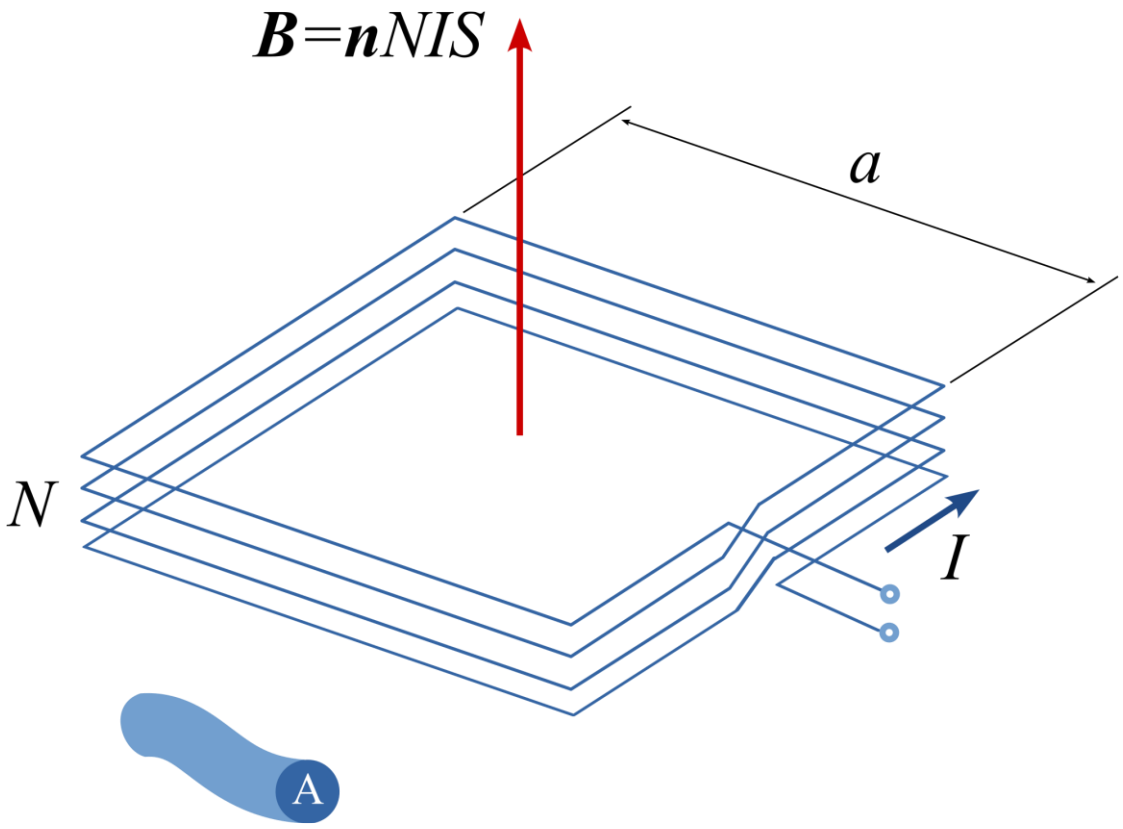






кадры для космоса

Мощность



- Мощность

$$P = I^2 R$$

- Момент

$$m = NIS$$

- Ток для создания момента

$$I = \frac{m}{NS}$$

- Мощность

$$P = \left(\frac{m}{NS}\right)^2 R$$

- Сопротивление катушки

$$R = \rho \frac{4aN}{A}$$

$\rho = 1{,}68 \cdot 10^{-8}$ Ом · м – удельное сопротивление меди

Источники

- **Суханов, А. А.** Астродинамика [Текст] / А. А. Суханов ; Учреждение Российской академии наук, Институт космических исследований РАН. — Москва : ИКИ РАН, 2010.
- Параметры Земли 1990 года (ПЗ-90.11) : справочный документ / Военно-топографическое управление. — Москва, 2014
- Юдинцев, В. В. Ортогональные матрицы : презентация / В. В. Юдинцев. — Самара : Самарский университет, 2024. — 56 слайдов. — URL: https://classmech.ru/assets/files/pdf/mbs/Презентация_Ортогональные_матрицы.pdf (дата обращения: 21.11.2026).
- Юдинцев, В. В. Кватернионы : презентация / В. В. Юдинцев. — Самара : Самарский университет, 2025. — 43 слайда. — URL: https://classmech.ru/assets/files/pdf/mbs/Презентация_Кватернионы.pdf (дата обращения: 21.11.2026).

Актуальная версия презентации и проекты SimInTech размещены по адресу:
https://classmech.ru/pages/simintech/