

Управление в технических системах

Control in technical systems

Научная статья

УДК 004.896 + 004.4

DOI: 10.14529/ctcr250403

ИНТЕРФЕЙС X3D-ПРОТОТИПА СХВАТА МАНИПУЛЯЦИОННОГО РОБОТА

А.И. Телегин¹, *teleginai@susu.ru*

Г.И. Волович², *g_volovich@mail.ru*, <https://orcid.org/0000-0002-3572-1823>

Е.В. Гусев³, *gusev@ susu.ru*, <https://orcid.org/0000-0002-8458-1222>

С.Г. Пудовкина¹, *rudovkinasg@susu.ru*

¹ Южно-Уральский государственный университет, филиал в г. Миассе,
Миасс, Россия

² ООО «Челэнергоприбор», Челябинск, Россия

³ Южно-Уральский государственный университет, Челябинск, Россия

Аннотация. Целью является разработка интерфейсной части X3D-прототипа двухпальцевого схвата манипуляционных роботов для симуляции его функционирования при выполнении различных технологических операций, например, перехода из положения захвата детали в положение его освобождения в рабочей зоне станка с ЧПУ. Методы исследования относятся к трехмерному моделированию, системному анализу и робототехнике. Результаты исследования содержат описание интерфейсной части кода X3D-прототипа схвата, у которого в качестве координат полюса могут выступать декартовы, цилиндрические, сферические, ангулярные или обобщенные координаты и ориентация схвата может описываться параметрами Эйлера, тремя углами (Эйлера, самолетными, корабельными), кватернионами, параметрами Кейли – Клейна. Конечные перемещения схвата могут задаваться винтами, бикватернионами или любыми комбинациями параметров, описывающих целевое положение полюса схвата и его ориентацию. В процессах переходов схвата из одной осевой ориентации в другую могут использоваться не только известные параметры, но и уникальные цифровые коды. Исследованы групповые свойства таких кодов, установлена их связь с параметрами Эйлера и кватернионами. Предоставлена возможность использования любых способов описания положений схвата. Заключение. Полученные результаты используются для X3D-моделирования схвата в цифровой модели манипуляционного робота, входящего в состав роботизированного технологического комплекса, в котором известны технологическое оборудование, а также количество, внешние виды, массо-инерционные и физические свойства объектов манипулирования (грузов, деталей, инструмента), т. е. адаптивных свойств разработанный прототип схвата не имеет.

Ключевые слова: XML, X3D, прототипирование, конечные группы, подгруппы, параметры Эйлера, кватернионы, анимация

Для цитирования: Интерфейс X3D-прототипа схвата манипуляционного робота / А.И. Телегин, Г.И. Волович, Е.В. Гусев, С.Г. Пудовкина // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2025. Т. 25, № 4. С. 43–58. DOI: 10.14529/ctcr250403

INTERFACE OF THE X3D PROTOTYPE OF THE MANIPULATIVE ROBOT GRIP

A.I. Telegin¹, teleginai@susu.ru

G.I. Volovich², g.volovich@mail.ru, <https://orcid.org/0000-0002-3572-1823>

E.V. Gusev³, gusehev@susu.ru, <https://orcid.org/0000-0002-8458-1222>

S.G. Pudovkina¹, pudovkinasg@susu.ru

¹ South Ural State University, Miass, Russia

² LLC Chelenergopribor, Chelyabinsk, Russia

³ South Ural State University, Chelyabinsk, Russia

Abstract. The goal is to develop the X3D interface part, a prototype of a two-finger grip for manipulative robots to simulate its operation during various technological operations, for example, moving from the position of gripping a part to the position of releasing it in the working area of a numerically controlled machines. The research methods relate to three-dimensional modeling, system analysis and robotics. The results of the research contain a description of the interface part and the X3D prototype code of the grip, in which Cartesian, cylindrical, spherical, angular or generalized coordinates can act as pole coordinates, and the orientation of the grip can be described by Euler parameters, three angles (Euler, airplane, ship), quaternions, and Cayley-Klein parameters. The final movements of the grip can be set by screws, biquaternions, or any combination of parameters describing the target position of the grip pole and its orientation. In the processes of grip transitions from one axial orientation to another, not only known parameters can be used, but also unique digital codes. The group properties of such codes are investigated, and their relationship with Euler parameters and quaternions is established. It is possible to use any methods of descrying the positions of the grip. Conclusion. The obtained results are used for X3D modeling of the grip in a digital model of a manipulative robot, which is part of a robotic technological complex in which the technological equipment is known, as well as the number, appearance, mass-inertia and physical properties of objects of manipulation (loads, parts, tools), i.e. adaptive properties, the developed prototype of the grip does not it has.

Keywords: XML, X3D, prototyping, finite groups, subgroups, Euler parameters, quaternions, animation

For citation: Telegin A.I., Volovich G.I., Gusev E.V., Pudovkina S.G. Interface of the X3D prototype of the manipulative robot grip. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics.* 2025;25(4):43–58. (In Russ.) DOI: 10.14529/ctcr250403

Введение

Различные типы захватных устройств (механические, вакуумные, магнитные и прочие) Манипуляционных Роботов (МР) служат для захватывания и удержания объектов манипулирования (грузов), которые могут иметь различную геометрию, размеры, массу и другие параметры. Поэтому их относят к сменным устройствам, которые часто приходится проектировать под решение конкретных задач захвата и удержания различных грузов [1–3]. Схватом называется устройство захвата, представляющее собой механизм, удерживающий груз посредством его зажима рабочими элементами под действием привода [4]. В качестве рабочих элементов в схватах часто выступают два пальца.

Настоящая работа является развитием статьи [5] и содержит результаты исследования возможностей трехмерного графического моделирования управляемых систем тел в Программной Системе (ПС) «СисТел», где используется XML-ориентированный расширяемый язык разметки виртуальных миров X3D и возможности его расширения вплоть до создания собственных (предметно ориентированных) декларативных языков разметки информации путем X3D-прототипирования [6].

Анализ отличий ПС «СисТел» от зарубежных аналогов начат в статье [5]. Первое из них связано с предметной ориентацией ПС «СисТел» и способом расширения функциональных возможностей этой ПС. Сущность этого способа связана с использованием ранее введенных прототипов в прототипировании более сложных систем. Здесь, например, в прототипировании схвата используются прототипы тел, описанные в статье [5].

В ПС, предназначенных для цифрового моделирования систем тел, важно обеспечить несколько свойств. Во-первых, от пользователя не требовать знаний информационных технологий, используемых при создании и функционировании ПС. Пользователь должен знать предметную область, для моделирования которой используется ПС. Во-вторых, подготовка входной информации должна быть простой и на точность моделирования должна влиять только точность входных данных. Например, на точность построения программных движений тел МР должна влиять только точность задания постоянных межполюсных расстояний (длин тел), исходных и целевых значений обобщенных координат, их скоростей и т. д. В-третьих, разрабатываемая ПС должна иметь потенциал своего аддитивного развития (увеличения своего функционала без пересмотра основ). В-четвертых, функционал ПС должен обеспечивать эффективное решение задач механики, управления и синтеза МР с заданными кинематическими, статическими и динамическими свойствами. В-пятых, для описания состояний управляемых систем тел желательно иметь возможность использовать не только известные параметры, но и новые параметры, если, конечно, установлена их связь с известными параметрами. Например, для описания положений полюсов тел желательно предоставить возможность использовать не только декартовы, цилиндрические или сферические координаты, но и обобщенные координаты. Для описания ориентаций тел желательно предоставить возможность использовать не только известные, но и новые (универсальные) параметры. Другие не менее важные свойства ПС, например, удобный интерфейс с пользователем, гибкость, доступность и т. д., в целом больше относятся к коммерческим версиям ПС, чем к его прототипу. Поэтому эти свойства здесь не рассматриваются.

В части методики моделирования тел МР, описанной в статье [5], два первых свойства ПС «СисТел» обеспечены (реализованы), так как пользователю достаточно уметь составлять цифровые коды моделей тел. Для обеспечения третьего свойства ПС геометрические размеры графических объектов, используемых в моделях тел, реализуются через поля Узла Трансформации (УТ), имеющие произвольный тип доступа (`accessType = "inputOutput"`), а не через поля размеров графического объекта, у которых `accessType = "initialiseOnly"` [5]. Четвертое свойство предметно-ориентированной линейки ПС «СисТел» («СисТелСхват», «СисТелМР», «СисТелРТК», «СисТелГПС», «СисТелША», «СисТелШМ») реализуется в тех границах запланированного развития их функциональных возможностей, которые обеспечивают методы теоретической механики систем тел, опубликованные в работах [7, 8]. Реализация пятого свойства ПС «СисТел» продемонстрирована в настоящей статье. В этой связи рекомендуем обратить внимание на статью [9], где для моделирования роботизированных технологических комплексов и гибких производственных систем используются группы, их подгруппы, алгебра полиномов, математическая теория категорий и другие математические методы.

Во многих техпроцессах известны не только целевые ориентации схвата, инструмента, обрабатываемых тел и т. д., но их ориентации направлены вдоль взаимно перпендикулярных осей. Учет этих особенностей упрощает роботизацию таких техпроцессов [9].

Постановка задачи. Разработать и описать интерфейсную часть X3D-прототипа схвата, позволяющую использовать любые известные параметры положения полюса схвата и его ориентации, а также вводить в обращение новые параметры, если установлена их связь с декартовыми координатами полюса схвата и параметрами Эйлера его ориентации.

1. Осевые ориентации схвата

Модель схвата изображена на рис. 1, где С – полюс схвата; 1 – кисть; 2 – большой палец. На рис. 2 изображена модель ориентирующего пространства в виде сферы, разбитой с помощью правого репера $\bar{x}\bar{y}\bar{z}$ на восемь октантов [10], где для первой из них орт \bar{x} направлен горизонтально вправо, орт \bar{y} – вертикально вверх. Осевой Ориентацией Схвата (ООС) назовем такую его ориентацию, в которой орты \bar{x}_c , \bar{y}_c схвата (см. рис. 1) направлены вдоль или навстречу орту репера $\bar{x}\bar{y}\bar{z}$, совмещённых с рёбрами октанта. Наша ближайшая цель – сопоставить ООС с ориентациями репера $\bar{x}\bar{y}\bar{z}$, в которых его орты направлены вдоль рёбер октанта.

Закодируем ООС цифровым кодом ij , где первое число $i \in \{1, 2, \dots, 6\}$ кодирует направления орта \bar{x}_c : 1 – горизонтально вправо, 2 – влево, 3 – вертикально вверх, 4 – вниз, 5 – на нас, 6 – от нас. Аналогично второе число $j \in \{1, 2, \dots, 6\}$ кодирует направление орта \bar{y}_c . Заметим, что для каждого из шести осевых ориентаций орта \bar{x}_c возможны только четыре ориентации орта \bar{y}_c . Таким об-

разом, имеем всего $6 \cdot 4 = 24$ ООС. Например, если орт \bar{x}_c направлен на нас ($i = 5$), то орт \bar{y}_c может быть ориентирован только вправо ($j = 1$) или влево ($j = 2$), а также вверх ($j = 3$) или вниз ($j = 4$), т. е. в случае $i = 5$ возможны следующие коды 5j ООС – 51, 52, 53, 54. Аналогично кодируются остальные ООС. Обозначим ООС, соответствующую коду ij , через $OC(ij)$ и для словесного (вербального) описания этого обозначения поставим ему в соответствие следующие фразы: $OC(13)$ – пальцы вправо, большой вверх; $OC(42)$ – пальцы вниз, большой влево; $OC(53)$ – пальцы на нас, большой вверх и т. д.

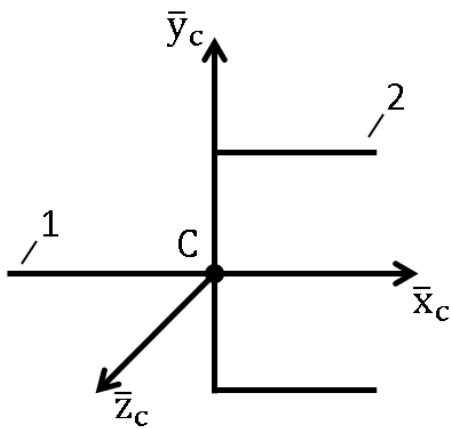


Рис. 1. Модель схвата
Fig. 1. Capture model

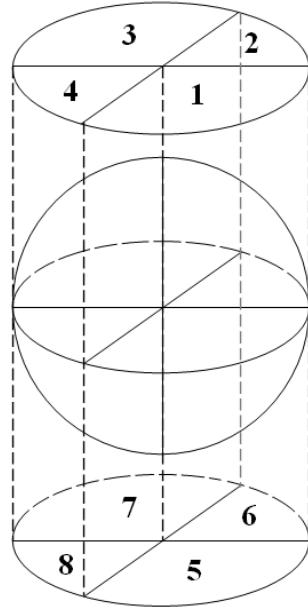


Рис. 2. Модель ориентационного пространства
Fig. 2. Orientation space model

Для визуального (графического) описания ООС построим таблицу из восьми столбцов (по числу октантов) и трёх строк, соответствующих трём возможным ООС в каждом октанте (по числу круговых перестановок имён орт – xuz , zxy , yxz). Обозначим Таблицу ООС через ТОС. Прежде чем заполнить ТОС, напомним [5], что в X3D для ориентации графического объекта относительно Системы Координат X3D-сцены (СКС) используются Параметры Эйлера (ПЭ) c_x, c_y, c_z, q , являющиеся значением поля rotation узла трансформации, который вложен в узел сцены и содержит разметку объекта, например, схвата. Наша цель – записать в нижней части каждой ячейки ТОС ПЭ перехода от $OC(13)$ к $OC(ij)$.

Заполним ТОС за три этапа. На первом в центре ячеек каждого столбца изобразим грани соответствующих октанта. Направление граней очевидно из рис. 2.

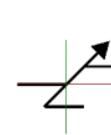
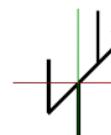
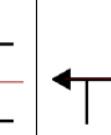
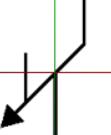
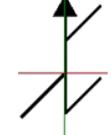
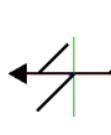
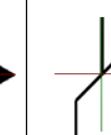
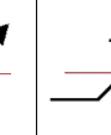
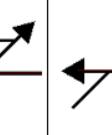
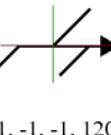
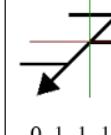
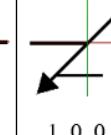
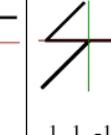
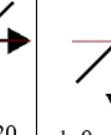
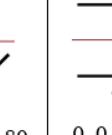
На втором этапе мысленно придадим граням имена (x, y, z). Репер первой ячейки (левая верхняя) должен соответствовать реперу СКС, где орт \bar{x} направлен вправо, орт \bar{y} – вверх, что соответствует $OC(13)$, код которой на третьем этапе запишем в левом верхнем углу первой ячейки первой строки ТОС. Реперы в следующих семи ячейках первой строки ТОС получим путём сохранения ориентации одного орта предшествующего репера, где имя этого орта будем менять круговой перестановкой имен x, y, z . Так, во второй ячейке первой строки орт \bar{x} сохраняет свою ориентацию (вправо) и для перехода во второй октант орт \bar{y} должен принять ориентацию «от нас», что соответствует $OC(16)$. Для этого необходимо повернуть репер из первой ячейки вокруг оси $O\bar{x}$ (O – центр шара на рис. 2) на 270 градусов (или на « -90 » градусов), что соответствует $\Pi\mathcal{E} = (1, 0, 0, -\pi/2)$, который записан в нижней части второй ячейки ТОС. Здесь и далее при записи ПЭ используется положительное значение угла поворота вокруг выбранной оси, например, вместо « -90 » градусов записываем 270° . Значения ПЭ в остальных ячейках будут приведены позже. Здесь же, действуя аналогично, дойдем до последней ячейки первой строки ТОС. Реперы в ячейках второй и третьей строки каждого столбца получим из репера в первой строке этого столбца путём круговой перестановки имен x, y, z .

На третьем этапе в каждой ячейке изобразим ООС по следующему правилу: пальцы направляем вдоль орта \bar{x} и большой палец должен располагаться рядом со стрелкой.

Итоговая ТОС, где в ячейках размещены 2D-виды схвата на рис. 1 (без обозначений полюса, орт и номеров деталей), представлена в табл. 1. ОС(13) с соответствующим ПЭ = (1, 0, 0, 0) будем называть исходной ООС. Она расположена в левом верхнем углу ТОС.

Таблица 1
Возможные осевые ориентации схвата (ООС) манипулятора
Possible axial orientations of the manipulator grip

Table 1

1	2	3	4	5	6	7	8
13  1, 0, 0, 0	16  1, 0, 0, 270	36  -1, -1, 1, 120	53  0, 1, 0, 270	54  1, 0, 1, 180	14  1, 0, 0, 180	42  1, -1, 0, 180	45  1, -1, -1, 120
35  1, 1, 1, 120	63  0, 1, 0, 90	62  1, -1, -1, 240	32  0, 0, 1, 90	41  0, 0, 1, 270	46  -1, 1, -1, 240	26  0, 1, -1, 180	52  1, -1, 1, 120
51  -1, -1, -1, 120	31  -1, -1, 0, 180	23  0, 1, 0, 180	25  0, 1, 1, 180	15  1, 0, 0, 90	61  1, 1, -1, 120	64  1, 0, -1, 180	24  0, 0, 1, 180

Описанный процесс построения ТОС упростил разбиение множества ООС на классы. Если данное множество разбито на попарно непересекающиеся подмножества, дающие в сумме само множество, и элементы каждого подмножества связаны некоторым отношением эквивалентности, обладающим свойствами симметрии, транзитивности и рефлексивности, то для краткости говорят просто о разбиении множества на классы [11]. Будем считать ООС эквивалентными, если они получаются из ОС(13) путем поворота вокруг заданной неподвижной оси, т. е. сохраняют эту ось неподвижной. Очевидно, что ООС с кодами 13, 15, 14, 16 являются эквивалентными, так как они сохраняют неподвижной ось $O\bar{x}$ (первое число каждого кода равно единице). Поэтому в этих ячейках ПЭ = (1, 0, 0, q), где $q \in \{0, 90, 180, 270\}$. Из ТОС видно, что пальцы схвата в ячейках с кодами 13, 15, 14, 16 направлены вправо. Обозначим этот класс через G_x . Аналогично элементы класса $G_y = \{63, 23, 53\}$ сохраняют неподвижной ось $O\bar{y}$, т. е. большой палец схвата в ячейках с кодами 63, 23, 53 направлен вверх и ПЭ = (0, 1, 0, q), где $q \in \{90, 180, 270\}$. Легко проверить, что класс $G_z = \{32, 24, 41\}$ сохраняет неподвижной ось $O\bar{z}$, т. е. ПЭ = (0, 0, 1, q), где $q \in \{90, 180, 270\}$. Действительно, из ТОС видно, что орт \bar{z}_c схвата в ячейках с кодами 32, 24, 41 направлен на нас.

2. Осевые группы G_x , G_y , G_z и их циклические подгруппы

Покажем, что элементы класса G_x , т. е. коды 13, 15, 14, 16, образуют группу, которую обозначим через G_x . Для этого построим таблицу умножения этих кодов. Групповую операцию обозначим символом «*» и введем ее следующим образом. Умножить два кода – значит последовательно выполнить им соответствующие повороты один за другим. При этом будем считать совпадающими два поворота, отличающиеся друг от друга на целое число полных оборотов (360°).

ОС(13) примем за групповую единицу. Это значит, что коду 13 ставится в соответствие поворот схвата на 0° или 360° (тождественный поворот) вокруг оси $C\bar{x}$. Код 15 означает поворот схвата из ОС(13) на 90° вокруг оси $C\bar{x}$. Коды 14 и 16 означают такие повороты на 180° и 270° соответственно.

Очевидно, что для кодов 13, 15, 14, 16 верны следующие операции умножения:

$$\begin{aligned} 13 * 13 &= 0^\circ * 0^\circ = 0^\circ = 13; \quad 13 * 15 = 0^\circ * 90^\circ = 90^\circ = 15; \quad 13 * 14 = 14; \quad 13 * 16 = 16; \\ 15 * 15 &= 90^\circ * 90^\circ = 180^\circ = 14; \quad 15 * 14 = 90^\circ * 180^\circ = 270^\circ = 16; \\ 15 * 16 &= 90^\circ * 270^\circ = 360^\circ = 0^\circ = 13; \quad 14 * 14 = 180^\circ * 180^\circ = 0^\circ = 13; \\ 14 * 16 &= 180^\circ * 270^\circ = 360^\circ * 90^\circ = 0^\circ * 90^\circ = 90^\circ = 15; \\ 16 * 16 &= 270^\circ * 270^\circ = 360^\circ * 180^\circ = 180^\circ * 0^\circ = 180^\circ = 14. \end{aligned}$$

Следовательно, табл. 2 является таблицей умножения группы G_x .

Из табл. 2 видно, что группа G_x абелева (коммутативная). Например, $13 * 15 = 15 * 13$, $14 * 15 = 15 * 14$ и т. д. Каждому элементу группы G_x соответствует обратный ему элемент. Например, $15 * 16 = 13$, т. е. $15^{-1} = 16$, так как по определению, если $ij * ij^{-1} = 13$, где 13 – единица (нейтральный элемент) группы, то ij^{-1} – элемент группы, обратный элементу ij .

Покажем, что группа G_x имеет циклические подгруппы. Напомним, что циклическая подгруппа – это подмножество группы, порождённое одним элементом и содержащее все его степени относительно заданной операции умножения. Например, $H(15)$ – циклическая подгруппа 4-го порядка группы G_x , порожденная ОС(15). Действительно, если элементы подгруппы $H(15)$ обозначить через a_k , то по определению $a_0 = 13 = 15^0$ и далее

$$\begin{aligned} a_1 &= 15^0 * 15 = 13 * 15 = 15 = 15^1; \\ a_2 &= a_1 * 15 = a_1 * a_1 = 15^1 * 15^1 = 15^2 = 15 * 15 = 14; \\ a_3 &= a_2 * 15 = a_2 * a_1 = 15^2 * 15 = 15^3 = 14 * 15 = 16; \\ a_4 &= a_3 * 15 = a_3 * a_1 = 15^3 * 15 = 15^4 = 16 * 15 = 13 = a_0. \end{aligned}$$

Следовательно, с учетом равенств

$$\begin{aligned} a_3 * a_1 &= a_4 = a_0; \quad a_2 * a_2 = a_2 * a_1 * a_1 = a_3 * a_1 = a_4 = a_0; \\ a_2 * a_3 &= a_2 * a_1 * a_1 * a_1 = a_4 * a_1 = a_0 * a_1 = a_1 \end{aligned}$$

получим, что табл. 3 является таблицей умножения группы $H(15)$.

Таблица 2
Таблица умножения
группы G_x
Table 2
Multiplication table
of the G_x group

	13	15	14	16
13	13	15	14	16
15	15	14	16	13
14	14	16	13	15
16	16	13	15	14

Таблица 3
Таблица умножения
циклической подгруппы $H(15)$
Table 3
Multiplication table
of the cyclic subgroup $H(15)$

	a_0	a_1	a_2	a_3
a_0	a_0	a_1	a_2	a_3
a_1	a_1	a_2	a_3	a_0
a_2	a_2	a_3	a_0	a_1
a_3	a_3	a_0	a_1	a_2

Аналогично, считая ОС($1j$) образующим элементом группы $H(1j)$ для $j = 16$ и $j = 14$, получим еще две циклические подгруппы группы G_x . $H(16)$ – циклическая подгруппа 4-го порядка группы G_x , порожденная ОС(16), где

$$\begin{aligned} b_0 &= 13 = 16^0; \quad b_1 = 16^0 * 16 = 16^1; \\ b_2 &= b_1 * 16 = b_1 * b_1 = 16 * 16 = 16^2 = 14; \\ b_3 &= b_2 * 16 = b_2 * b_1 = 16^2 * 16 = 16^3 = 14 * 16 = 15; \\ b_4 &= b_3 * 16 = b_3 * b_1 = 16^3 * 16 = 16^4 = 15 * 16 = 13 = b_0 \end{aligned}$$

и табл. 4 представляет таблицу умножения подгруппы $H(16)$. $H(14)$ – циклическая подгруппа 2-го порядка группы G_x , порожденная ОС(14), где

$$c_0 = 13 = 14^0; \quad c_1 = 14^0 * 14 = 14^1; \quad c_2 = c_1 * 14 = c_1 * c_1 = 14 * 14 = 14^2 = 13 = c_0$$

и табл. 5 представляет таблицу умножения подгруппы $H(14)$.

Таблица 4
Таблица умножения
циклической подгруппы $H(16)$
Table 4
**Multiplication table
of the cyclic subgroup $H(16)$**

	b_0	b_1	b_2	b_3
b_0	b_0	b_1	b_2	b_3
b_1	b_1	b_2	b_3	b_0
b_2	b_2	b_3	b_0	b_1
b_3	b_3	b_0	b_1	b_2

Таблица 5
Таблица умножения
циклической подгруппы $H(14)$
Table 5
**Multiplication table
of the cyclic subgroup $H(14)$**

	c_0	c_1
c_0	c_0	c_1
c_1	c_1	c_0

Группы G_x , G_y , G_z изоморфны, где взаимно однозначные соответствия элементов имеют вид $15 \leftrightarrow 63 \leftrightarrow 32$, $14 \leftrightarrow 23 \leftrightarrow 24$, $16 \leftrightarrow 53 \leftrightarrow 41$.

Эти группы имеют одинаковую структуру и свойства, несмотря на разные обозначения элементов. В них код 13 является единичным элементом, который оставляет другие элементы групп неизменными при умножении. Таблицы умножения для этих групп имеют одинаковый вид, если заменить соответствующие коды. Например, в G_x $15 * 16 = 13$, точно так же в G_y $63 * 53 = 13$, а в G_z $32 * 41 = 13$ и так далее. Таким образом, таблицы умножения для групп G_y и G_z имеют виды табл. 6 и 7.

Таблица 6
Таблица умножения
группы G_y
Table 6
**Multiplication table
of the G_y group**

	13	63	23	53
13	13	63	23	53
63	63	23	53	13
23	23	53	13	63
53	53	13	63	23

Таблица 7
Таблица умножения
группы G_z
Table 7
**Multiplication table
of the G_z group**

	13	32	24	41
13	13	32	24	41
32	32	24	41	13
24	24	41	13	32
41	41	13	32	24

Легко проверить, что группы G_y , G_z имеют циклические подгруппы. Напомним, что группы G_x , G_y , G_z изоморфны. Соответственно, и их циклические подгруппы будут аналогичны G_x . Из этого следует, что

- $H(63)$ – циклическая подгруппа 4-го порядка группы G_y , порожденная ОС(63);
- $H(53)$ – циклическая подгруппа 4-го порядка группы G_y , порожденная ОС(53);
- $H(23)$ – циклическая подгруппа 2-го порядка группы G_y , порожденная ОС(23);
- $H(32)$ – циклическая подгруппа 4-го порядка группы G_z , порожденная ОС(32);
- $H(41)$ – циклическая подгруппа 4-го порядка группы G_z , порожденная ОС(41);
- $H(24)$ – циклическая подгруппа 2-го порядка группы G_z , порожденная ОС(24).

Более того, 24 ООС образуют группу, изоморфную группе поворотов (самосовмещений) куба. Самое простое описание всех самосовмещений куба представлено в популярной книге П.С. Александрова [11]. Это описание можно использовать для выделения оставшихся классов ООС и обоснования значений ПЭ в соответствующих ячейках ТОС. Последнюю цель можно достичь путем использования ПЭ, соответствующих классам G_x , G_y , G_z , и алгебры кватернионов. Реализуем этот путь.

3. Кватернионы переходов

Обозначим ПЭ, описывающий переход из ОС(i) в ОС(ij), через ПЭ(ij). В конце первого раздела статьи получены следующие значения ПЭ(ij):

$$\begin{aligned} \text{ПЭ(13)} &= (1, 0, 0, 0); \quad \text{ПЭ(15)} = (1, 0, 0, 90); \quad \text{ПЭ(14)} = (1, 0, 0, 180); \quad \text{ПЭ(16)} = (1, 0, 0, 270); \\ \text{ПЭ(63)} &= (0, 1, 0, 90); \quad \text{ПЭ(23)} = (0, 1, 0, 180); \quad \text{ПЭ(53)} = (0, 1, 0, 270); \\ \text{ПЭ(32)} &= (0, 0, 1, 90); \quad \text{ПЭ(24)} = (0, 0, 1, 180); \quad \text{ПЭ(41)} = (0, 0, 1, 270). \end{aligned} \quad (1)$$

Наша ближайшая цель – вычислить ПЭ остальных ООС и записать их в нижних частях соответствующих ячеек ТОС. Для этого будем использовать кватернионное описание переходов между ООС и алгебру кватернионов.

Компоненты q_0, q_x, q_y, q_z кватерниона

$$\vec{q} = q_0 + q_x \bar{x} + q_y \bar{y} + q_z \bar{z} \quad (2)$$

связаны с ПЭ c_x, c_y, c_z, q следующими формулами:

$$q_0 = \cos(q/2); \quad q_\xi = c_\xi \sin(q/2); \quad \xi \in \{x, y, z\}, \quad (3)$$

т. е. преобразование ПЭ в \vec{q} можно выполнить по формуле

$$\vec{q} = \cos(q/2) + (c_x \bar{x} + c_y \bar{y} + c_z \bar{z}) \sin(q/2), \quad (4)$$

где $c_x^2 + c_y^2 + c_z^2 = 1$. Из формул (2), (3) также видно, что преобразование \vec{q} в ПЭ можно выполнить по формуле

$$(c_x, c_y, c_z, q) = (q_x/s_q, q_y/s_q, q_z/s_q, q), \quad (5)$$

где $s_q = \sin(q/2)$, $q = 2 \arccos(q_0)$.

В алгебре кватернионов операция умножения кватернионов определена так, что множество кватернионов с операциями умножения и сложения образуют некоммутативное поле. Из определения кватернионного умножения следуют формулы [12]:

$$\bar{\xi} \circ \bar{\eta} = \bar{\xi} \times \bar{\eta} - \bar{\xi} \cdot \bar{\eta}; \quad a \circ \bar{\xi} = \bar{\xi} \circ a = a\bar{\xi}; \quad a \circ b = ab, \quad (6)$$

т. е. кватернионное произведение двух векторов равно разности их векторного и скалярного произведений, а кватернионное умножение скаляра на вектор или двух скаляров равно обычному умножению скаляра на вектор или, соответственно, двух скаляров.

Введем обозначения:

ПЭ(ij, kl) – ПЭ, описывающий переход из ОС(ij) в ОС(kl);

\vec{q}_{ij} – кватернион поворота из ОС(13) в ОС(ij);

\vec{q}_{ij}^{kl} – кватернион поворота из ОС(ij) в ОС(kl).

В формуле (2) орты $\bar{x}, \bar{y}, \bar{z}$ образуют репер, относительно которого осуществляется поворот, заданный кватернионом. Если, например, поворот осуществляется из ОС(13), то $\bar{x}, \bar{y}, \bar{z}$ – орты репера СКС. Для кватерниона \vec{q}_{ij}^{kl} орты $\bar{x}, \bar{y}, \bar{z}$ должны соответствовать ортам $\bar{x}_c, \bar{y}_c, \bar{z}_c$ схвата в ячейках ТОС с кодом ij в левом верхнем углу.

Формулы (6) будем использовать для вычисления кватерниона, описывающего сложные переходы. Например, кватернион \vec{q}_{36} можно вычислить по формуле $\vec{q}_{36} = \vec{q}_{13}^{16} \circ \vec{q}_{16}^{36}$.

В общем случае, если известны ПЭ(ij) и ПЭ(ij, kl), то вычисление \vec{q}_{kl} и его преобразование в ПЭ(kl) можно выполнить по следующему алгоритму (за четыре шага):

1) ПЭ(ij) по формуле (4) преобразуем в кватернион $\vec{q}_{ij} = \vec{q}_{13}^{ij}$;

2) ПЭ(ij, kl) по формуле (4) преобразуем в \vec{q}_{ij}^{kl} ;

3) вычислим по формулам (6) $\vec{q}_{kl} = \vec{q}_{13}^{ij} \circ \vec{q}_{ij}^{kl}$;

4) из \vec{q}_{kl} найдем q_0 и по формуле $q = 2 \arccos(q_0)$ вычислим q . В заключение по формулам (5) получим искомый ПЭ(kl).

Замечание. Отметим, что в X3D значением поля rotation УТ, вложенного в сцену, вместо направляющего косинуса оси поворота можно использовать координаты этой оси в СКС.

Вычислим ПЭ(36). Из (1) имеем ПЭ(16) = (1, 0, 0, 270). Из ТОС видно, что ПЭ(16, 36) описывает поворот схвата на 270° вокруг оси $C\bar{y}_c$, т. е. ПЭ(16, 36) = (0, 1, 0, 270). Следовательно, по описанному алгоритму получим:

1) из ПЭ(16) = (1, 0, 0, 270) по формуле (4) получим

$$\vec{q}_{13}^{16} = \cos(135) + \bar{x} \sin(135) = -1/\sqrt{2} + \bar{x}/\sqrt{2} = (\bar{x} - 1)/\sqrt{2};$$

2) из ПЭ(16, 36) = (0, 1, 0, 270) по формуле (4) получим

$$\vec{q}_{13}^{16} = \cos(135) + \bar{y} \sin(135) = -1/\sqrt{2} + \bar{y}/\sqrt{2} = (\bar{y} - 1)/\sqrt{2};$$

3) по формулам (6) получим

$$\vec{q}_{36} = \vec{q}_{13}^{16} \circ \vec{q}_{16}^{36} = (\bar{x} - 1) \circ (\bar{y} - 1)/2 = (\bar{x} \circ \bar{y} + 1 - \bar{y} - \bar{x})/2 = (1 - \bar{x} - \bar{y} + \bar{z})/2;$$

4) из \vec{q}_{36} имеем $q_0 = 1/2$, т. е. $q = 2 \arccos(1/2) = 2 \cdot 60 = 120$. Следовательно, с учетом замечания после умножения компонент кватерниона на 2 получим ПЭ(36) = (-1, -1, 1, 120).

Вычислим ПЭ(54). Из (1) имеем ПЭ(53) = (0, 1, 0, 270). Из ТОС видно, что ПЭ(53, 54) описывает поворот схвата на 180° вокруг оси $C\bar{x}_c$, т. е. ПЭ(53, 54) = (1, 0, 0, 180). Следовательно, по описанному алгоритму получим:

1) из ПЭ(53) = (0, 1, 0, 270) по формуле (4) получим

$$\vec{q}_{13}^{53} = \cos(135) + \bar{y} \sin(135) = -1/\sqrt{2} + \bar{y}/\sqrt{2} = -(1 - \bar{y})/\sqrt{2};$$

2) из ПЭ(53, 54) = (1, 0, 0, 180) по формуле (4) получим

$$\vec{q}_{53}^{54} = \cos(90) + \bar{x} \sin(90) = \bar{x};$$

3) по формулам (6) получим

$$\vec{q}_{54} = \vec{q}_{13}^{53} \circ \vec{q}_{53}^{54} = -(1 - \bar{y}) \circ \bar{x}/\sqrt{2} = -(\bar{x} - \bar{y} \circ \bar{x})/\sqrt{2} = -(\bar{x} + \bar{z})/\sqrt{2};$$

4) из \vec{q}_{54} имеем $q_0 = 0$, т. е. $q = 2 \arccos(0) = 2 \cdot 90 = 180$. Следовательно, с учетом замечания после умножения компонент кватерниона на $\langle-\sqrt{2}\rangle$ получим ПЭ(54) = (1, 0, 1, 180).

Аналогично вычисляются ПЭ для остальных ООС. Для верификации результатов этих вычислений можно использовать html-код в листинге 1, где ПЭ является значением поля initialDestination. В коде листинга 1 ПЭ = ('1, 0, -1, 3.1415'), что соответствует ОС(64). Этот код работает во всех доступных нам браузерах.

Листинг 1. Разметка поворотов схвата из ОС(13) в ОС(ij) для соответствующего ПЭ

```
<html><head><script src="x3dom.js"></script></head><body>
<X3D width="800px" height="400px"><Scene>
<Background skyColor="1 1 1"></Background><Viewpoint position="0 0 3"></Viewpoint>
<Transform DEF="схват">
<Transform translation="-0.25 0 0">
<Shape><Box size="0.5 0.15 0.15"></Box>
<Appearance DEF="цвет"><Material diffuseColor="0 0 1"></Material></Appearance>
</Shape></Transform>
<Shape><Box size="0.05 0.8 0.3"></Box><Appearance USE="цвет"></Appearance></Shape>
<Transform translation="0 0.5 0">
<Shape><Box size="0.05 0.2 0.05"></Box>
<Appearance USE="цвет"></Appearance></Shape></Transform>
<Transform translation="0.275, 0.3, 0">
<Shape DEF="Палец"><Box size="0.5 0.05 0.25"></Box>
<Appearance USE="цвет"></Appearance></Shape></Transform>
<Transform translation="0.275, -0.3, 0"><Shape USE="Палец"></Shape></Transform>
</Transform>
<OrientationChaser DEF='поворот' initialValue='1 0 0 0'
                     initialDestination='1, 0, -1, 3.1415'>
</OrientationChaser>
<ROUTE fromNode = 'поворот' fromField = 'value_changed'
           toNode = 'схват' toField = 'set_rotation'></ROUTE>
</Scene></X3D></body></html>
```

Подводя итог теоретической части статьи, отметим следующее. Во-первых, если в процессе роботизации производств удается расставить оборудование (МР, станки, накопители и т. д.) так, что ориентации рабочих органов (схватов, сверел, фрез и другого технологического инструмента) образуют конечную группу, то, используя конечное число переходов между ними, можно свести решение задач механики и управления из непрерывной области в дискретную. Например, для рассмотренных здесь осевых ориентаций количество переходов равно ($24 \times 24 = 576$) количеству ячеек в таблице умножения группы ООС. В случае использования осевых ориентаций, изоморфных поворотам правильного икосаэдра, у которого 60 самосовмещений, количество переходов – 3600 [11]. Конечно, количество комбинаций последовательностей переходов возрастает квадра-

тично, но это – цена за возможность дискретного анализа. Во-вторых, если не удается свести проектирование РТК и ГПС к использованию теории конечных групп в части ориентаций оборудования и рабочих органов, то для перехода в дискретную область анализа достаточно на конечном множестве ориентаций оборудования и рабочих органов осуществлять переходы между ними на основе использования X3D-узла <OrientationDamper>, который декларирует плавный разгон в начале перехода и плавное торможение в его конце за заданное время. Множество таких переходов образует группу. С учетом морфизма таких переходов и их описания в виде многочленов времени четвертого порядка, коэффициенты которого вычисляются из заданных четырех краевых условий (с нулевой скоростью на краях), опять появляется возможность дискретного анализа. В-третьих, представление уравнений динамики МР в кватернионах и подстановка в них кватернионов перехода открывает новые возможности решения задач механики и управления. Уже полученные на этом пути результаты позволили нам разработать X3D-прототип схвата, в полях которого заложены различные возможности симуляции его функционирования. При этом пользователям предоставлена возможность разрабатывать собственные способы описания положений схвата в виде цифровых кодов или в виде их списков, содержащих информацию о начале и окончании каждого перехода, а также наличии или отсутствии тех или иных объектов манипулирования в схвате. Эти возможности позволяют просто и быстро кодировать симуляцию сложных техпроцессов. Конечно, для реализации такой гибкости необходимо для каждого уникального способа цифрового кодирования техпроцесса разработать соответствующий интерпретатор на любом из языков программирования, поддерживаемых в X3D. Мы для этого используем JavaScript (JS).

4. Прототипирование схвата

Для освоения методики X3D-прототипирования достаточно владеть основами XML и X3D, которые можно изложить в нескольких абзацах, например, в следующих трех абзацах изложены основы XML.

XML – это метаязык, в котором изложены синтаксис и семантика языков разметки информации (декларативных языков программирования) в различных предметных областях. XML-код состоит из элементов, записываемых в следующем формате:

<имяОткрывающегоТэга атрибуты> содержимое элемента **</имяЗакрывающегоТэга>**

Имя открывающего и закрывающего тэга должно полностью (с учетом регистра) совпадать. Для элемента без содержимого можно использовать укороченную форму записи

<имяТэга атрибуты/>

Элемент может не иметь атрибутов или иметь один или несколько атрибутов, отделенных друг от друга пробельным символом. Каждый атрибут описывает отдельное свойство (параметр, показатель) элемента и записывается в следующем формате:

имяАтрибута = ‘значение’

Вместо одинарных кавычек, в которых записывается значение атрибута, могут использоваться двойные кавычки, например, *ООО = “ ‘AAA’ ”*. Элемент не должен иметь атрибуты с одинаковыми именами.

В качестве содержимого элемента может выступать один или несколько элементов, т. е. элементы можно вкладывать друг в друга по правилу вложения матрешек друг в друга.

Таким образом, каждый XML-ориентированный язык основан на трех правилах – правиле записи элементов, правиле записи атрибутов и правиле правильного вложения элементов. Сложность XML-языков заключается в количестве элементов и их атрибутов. Например, в X3D сотни элементов, которые принято называть узлами, и каждый из них может иметь десятки атрибутов, которые принято называть полями. Это естественно, так как виртуальные миры – сложная и разнообразная предметная область. Наша предметная область (роботизация производств) проще. Поэтому на этапе создания прототипа ПС «СисТел» мы используем пару десятков X3D-узлов, в которых используем несколько полей. Более того, при помощи X3D-прототипов путем расширения X3D мы создаем предметно-ориентированный язык, предназначенный для моделирования МР, ША, ШМ, РТК, ГПС и других управляемых систем тел с решением задач механики и управления с анимацией. В статье [5] разработанprotoузел с именем "прототипТела", который имеет единственное поле с именем "код". В листинге 2 приведен пример его использования.

Листинг 2. Пример правильного вложения X3D-узлов

```
<?xml version = "1.0" encoding = "UTF-8"?>
<X3D version = "3.2"><WorldInfo title = 'груз_1.x3d'>
<Scene> <Background skyColor = "1 1 1"/> <Viewpoint position = "0, 0, 0.1"/>
<ExternProtoDeclare name = "прототипТела" url = "прототипТела.x3d"/>
<Transform>
  <ProtoInstance DEF = "винт" name = "прототипТела"><fieldValue name = "код"/> </ProtoInstance>
</Transform>
<Script url = "груз_1.js"/>
</Scene>
</X3D>
```

В X3D имена узлов, как правило, начинаются с заглавной буквы. Имена полей, как правило, начинаются со строчной буквы. В листинге 2 первая строка называется преамбулой. Во второй строке записан открывающий тэг корневого узла и укороченный узел с информацией об имени файла, в котором сохранена разметка. В 3-й строке открыта сцена и в значениях полей следующих за ней двух укороченных узлов указано, что фон сцены белый (RGB-код цвета имеет значение "1, 1, 1") и за сценой наблюдают на удалении сто миллиметров вдоль оси аппликат СКС. Условимся для сокращения записи X3D-кода первые три строки и последние две опускать (не записывать), как это сделано в листинге 3.

Листинг 3. Пример неправильного вложения X3D-узлов

```
<ExternProtoDeclare name = "прототипТела" url = "прототипТела.x3d"/>
<Transform> <ProtoInstance DEF = "тело" name = "прототипТела"></Transform>
<fieldValue name = "код"/> </ProtoInstance>
```

Из листинга 3 видно, что узел `<Transform>` закрылся раньше, чем закрылся вложенный в негоprotoузел `<ProtoInstance>`, т. е. нарушено правило – первым открылся, последним закрылся (как в матрешках).

Остальные детали X3D опишем по мере необходимости.

Процесс разработки и описания созданного прототипа рекомендуем разбивать на этапы. Здесь рассматриваются два их них.

В любой разработке все начинается с формулировки целей и задач, решение которых обеспечивает достижение целей. Заканчивается разработка описанием путей решения задач и реализацией одного из них. В случае разработки X3D-прототипов желательно, чтобы каждый этап разработки приближал нас к целевым X3D и JS-кодам. Поэтому на первом этапе в качестве формулировки целей мы предлагаем представить экземпляр прототипа со значениями протополей по умолчанию и кратко описать цели их использования с привязкой к целям разработки. Для прототипа схвата такой экземпляр представлен в листинге 4.

Листинг 4. Декларация экземпляра, определенного как "схватПоУмолчанию", прототипа схвата с именем "проСхват"

```
<ProtoInstance DEF = "схватПоУмолчанию" name = "проСхват">
<fieldValue name = "место" value = "0, 0, 0"/>
<fieldValue name = "ориентация" value = "1, 0, 0, 0"/>
<fieldValue name = "положение" value = "0, 0"
<fieldValue name = "разводПальцев" value = "0"/>
<fieldValue name = "жатьКлавиши" value = "false"/>
<fieldValue name = "перенос" value = "0"/>
<fieldValue name = "поворот" value = "0"/>
<fieldValue name = "переходы" value = "0"/>
</ProtoInstance>
```

Здесь и далее в соответствии с принципами XML текстовые значения полей и имен полей расширения записываются на языке основных пользователей (на русском языке). В XML рекомендуется вводить имена полей так, чтобы они были короткими и информационно емкими. Со своей стороны мы стремимся не перегружатьprotoузлы протополями (полями расширения). Например, в прототипе "проСхват" их меньше десяти.

Поле DEF является полем общего назначения. Его используют для идентификации узлов, включая и protoузлы, т. е. узлы расширения языка X3D. Заметим, что имя DEF поля начинается с заглавной буквы (это исключение из правила именования полей) и это имя образовано из термина definition (определение).

Для описания назначения protopolей и их соответствий целям разработки использованы термины "место", "ориентация" и "положение" управляемого тела, принятые в монографиях [13, 14]. Поле с именем "место" содержит Декартовы Координаты (ДК) полюса схваты, которые направляются в поле translation УТ, содержащего разметку схваты, меняя таким образом место схваты на сцене. Поле с именем "ориентация" содержит значение ПЭ, которое направляется в поле rotation УТ для изменения ориентации схваты. Из двух десятков полей УТ здесь использованы только два из них (translation и rotation).

Поле "положение" содержит Цифровой Код (ЦК), описывающий одно или несколько положений схваты. Это поле предоставляет пользователям возможность вводить собственные параметры положения схваты, если разработан JS-код вычисления ДК и ПЭ через эти параметры. Таким образом, реализуется важнейшая цель разработки – предоставить пользователям возможность использовать любые параметры для описания положения схваты в пространстве сцены. Мы предлагаем первым числом ЦК кодировать тип координат полюса схваты: 0 – ДК; 1 – цилиндрические; 2 – сферические; 3 – ангулярные; 4 – обобщенные. Вторым кодировать тип параметров ориентации: 0 – ПЭ; 1 – углы Эйлера; 2 – углы Брайнта; 3 – кватернион; 4 – целевые координаты двух точек схваты, не лежащих на одной прямой с его полюсом; 5 – код ООС. Можно ввести в рассмотрение и другие параметры ориентации, например, параметры Кейли – Клейна. Для этого необходимо разработать JS-функцию, вычисляющую ПЭ через параметры Кейли – Клейна. Более того, можно использовать и другие параметры положения, например, бикватернионы, разработав для этого JS-код, вычисляющий на их основе ДК полюса схваты и ПЭ его ориентации, а также в качестве первых двух чисел ЦК положения принять, например, числа 5, 6. Если ЦК положения схваты содержит два нуля, то, по умолчанию, ДК = (0,0,0) и ПЭ = (1,0,0,0), что соответствует положению схваты в центре сцены, где пальцы направлены горизонтально вправо, большой вертикально вверх.

Значение поля "разводПальцев" содержит расстояние (в метрах) между пальцами схваты. Логическое значение поля "жатьКлавиши" управляет запуском режима управления анимацией при помощи клавиш.

Варианты использования полей "перенос", "поворот", "переходы" и интерпретация их ЦК определяются пользователями. Мы рекомендуем следующие варианты их использования. Поле "перенос" содержит ЦК, описывающий целевые координаты полюса схваты относительно Системы Координат (СК) его предка (по умолчанию сцены). Первое число кода указывает на тип координат полюса схваты, за которым следуют значения соответствующих параметров (смотрите описание поля "положение"). Видно, что по умолчанию используются ДК. Если записано одно значение (ноль), то остальные три числа ДК (абсцисса, ордината и аппликата) по умолчанию равны нулю. Поле "поворот" содержит ЦК, описывающий целевую ориентацию схваты в СК его предка (смотрите описание поля "положение"). Поле "переходы" содержит список, элементы которого описывают последовательность положений схваты и соответствующих им моментов времени, отсчитываемых от события запуска переходов. Код каждого положения аналогичен описанному ранее коду протополя "положение".

На втором этапе описания разрабатываемого прототипа рекомендуем представить его без разметки тела прототипа <ProtoBody>, т. е. достаточно привести только интерфейсную часть. Такой вариант прототипа схваты представлен в листинге 5.

Листинг 5. Разметка интерфейсной части прототипа "проСхват"

```
<ProtoDeclare name = "проСхват">
<ProtoInterface>
<fieldname = "место" type = "SFVec3f" accessType = "inputOutput" value = "0, 0, 0"/>
<field name = "ориентация" type = "SFRotation" accessType = "inputOutput" value = "1,0,0,0"/>
<field name = "положение" type = "MFFloat" accessType = "inputOutput" value = "0"/>
<field name = "разводПальцев" type = "SFFloat" accessType = "inputOutput" value = "0"/>
<field name = "жатьКлавиши" type = "SFBool" accessType = "inputOutput" value = "false"/>
<field name = "перенос" type = "MFFloat" accessType = "inputOutput" value = "0"/>
<field name = "поворот" type = "MFFloat" accessType = "inputOutput" value = "0"/>
<field name = "переходы" type = "MFFloat" accessType = "inputOutput" value = "0"/>
</ProtoInterface>
<ProtoBody><!-- разметка тела прототипа --></ProtoBody>
```

В отличие от листинга 4 здесь для каждого прототипа представлены тип его значения и тип доступа к этому полю. В X3D все типы данных разбиты на два класса: Single-Value (однозначный) и Multiple-Value (многозначный). Имена типов полей класса Single-Value начинаются с символов SF и хранят только одно значение (число, 2-, 3- или 4-мерный вектор и т. п.). Имена типов полей класса Multiple-Value начинаются с символов MF и могут содержать список перечисленных значений. Например, значения поля version относятся к типу SFFloat (одно действительное число). В листинге 5 использованы следующие типы. SFBool – одно логическое (булево) значение true (истина) или false (ложь). SFFloat, MFFloat – одно или несколько вещественных чисел. SFRotation – один ПЭ. SFVec3f – один трехмерный вектор (тройка действительных чисел).

Каждое поле имеет один из четырех типов доступа. При создании новых (авторских) узлов, расширяющих X3D-возможности, типы доступа к новым полям задаются в поле accessType. Если accessType = "initializeOnly", то поле является закрытым, т. е. содержит неизменяемые исходные значения свойств узла. Поле, у которого accessType = "inputOnly", называют in-полем (входным полем). Это поле является контейнером для изменяемых данных, которые могут поступать в это поле от выходных полей других узлов. Для выходного (out-) поля accessType = "outputOnly". Out-поле содержит значения, которые можно отправить в другие поля. Если accessType = "inOutPut", то имеет место inOut-поле, в которое можно записать новые значения и прочитать изменившиеся значения, т. е. inOut-поле является одновременно in-полем и out-полем. Таким образом, декларативный язык программирования X3D строго типизирован (значения полей относятся к одному из тридцати типов) и каждое поле имеет один из четырех типов доступа. Мы считаем, что на этапе разработки прототипа ПС преждевременно рассматривать вопросы необходимости и/или достаточности режима доступа к полям, а также защиты полей от несанкционированного доступа. Поэтому из четырех возможных режимов (типов) доступа к полям мы используем последний (универсальный) тип.

Коротко остановимся на путях решения задач анимации программных движений схвата и операций захвата и отпуска (взять/отдать) объектов манипулирования (грузов) в пальцах схвата. Отметим, что решение этой задачи тесно связано с типом схвата. Разница между простейшим двухпальцевым и универсальным пятипалцевым схватором большая. В намеченном пути развития ПС «СисТел» прототипом первой системы тел является «проСхват» и последним (возможно крайним) будет «проШМ», т. е. прототип многоногой Шагающей Машины (ШМ), основанный на прототипах МР, используемых в качестве ног ШМ. Очевидно, что между универсальным схватором и пятиногой ШМ возможен морфизм [9]. Поэтому на текущем этапе разработки рано стремиться к обобщениям и универсализации. Тем более что в промышленной робототехнике пока доминируют простые схваторы. Возможности экземпляров прототипа схвата «проСхват» достаточны для моделирования его работы в качестве схваторов МР в составе РТК межоперационного транспортирования, покраски и даже сборки, если известны внешние виды объектов манипулирования и их количество.

Анимировать можно, например, цвет тела или его деформацию. Для анимации поступательного и/или вращательного движений схвата можно использовать несколько способов и им соответствующих узлов. Выбор способов в основном зависит от преследуемых целей и решаемых

задач для их достижения. Достаточно подробно анимация тел и их систем с помощью X3D-узлов рассмотрена в книге [15]. В целом рассматривать, анализировать и рекомендовать эффективные способы анимации схватка без решения прямой и обратной задач кинематики схватка в составе МР – не продуктивно.

Для изменения положения схватка в процессе анимации достаточно синхронно менять значения полей translation и rotation УТ схватка, направив в них измененные значения ДК полюса схватка и ПЭ его ориентации. Перед этим если первые два числа ЦК протополя "положение" не равны нулю, то необходимо сначала вычислить ДК и ПЭ через используемые параметры положения. Задачи вычисления можно решать, например, в JS-функциях. В прототипе «проСхват» формулы вычисления ДК полюса схватка через полярные (для МР на плоскости), цилиндрические и сферические координаты взяты из справочника [16].

Формулы вычисления ДК через ангулярные и обобщенные координаты взяты из книги [17]. Формулы вычисления ПЭ через углы и кватернионы взяты из книг [18, 19]. Формулы вычисления кватерниона, описывающего ориентацию схватка через начальное положение двух точек схватка, не лежащих на одной прямой с его полюсом, и их конечным положением представлены в работе [12]. ПЭ через коды ООС определяются по табл. 1.

В отличие от рис. 1 X3D-модель схватка на рис. 3 на левом конце содержит выступ вверх в виде бокса. Во-первых, он как и стрелка на рис. 1 используется в словесном описании ориентации схватка, например, фраза "пальцы вправо выступ вверх" описывает ОС(13). Возможность такими фразами задавать

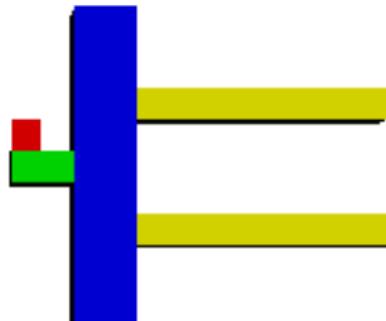


Рис. 3. X3D-модель схватка
Fig. 3. X3D model of the gripper

ориентацию схватка позволит (в дальнейшем) реализовать в ПС «СисТел» речевой ввод информации при помощи XML-ориентированного языка VoiceXML. Во-вторых, этот выступ позволяет визуально наблюдать работу механизма байонета в процессе смены схватка [20].

Заключение

Разработанный и описанный здесь прототип схватка используется в составе прототипов МР, работающих в различных системах координат (декартовой, цилиндрической, сферической, ангулярной). Варианты его использования разнообразны. Возможности введения собственных параметров описания положений схватка через их цифровое кодирование и разработку соответствующих интерпретаторов на любом из поддерживаемых в X3D языках программирования позволяет расширять это разнообразие.

Здесь, как и в заключении статьи [5], мы констатируем, что использовать прототип схватка «проСхват» можно без знания языка X3D. Действительно, внешний вид (геометрия) схватка и груза в нем описываются цифровыми кодами. Анимация движений схватка в техпроцессах описывается соответствующими списками цифровых кодов, интерпретаторы которых используют JS-коды, реализующие авторские способы анимации. Например, в текущей версии тела прототипа «проСхват» реализованы три авторских способа (тик-, клик- и вычисляемой) анимации, изложение которых выходит за пределы заявленной темы статьи.

Список литературы

1. Захватные устройства промышленных роботов: учеб. пособие / К.А. Украженко, Ю.В. Янчевский, А.А. Кулебякин, А.Ю. Торопов. Ярославль: ЯГТУ, 2007. 88 с. ISBN 5-230-20645-4.
2. Игнатьев Н.П. Схваты автоматических манипуляторов и промышленных роботов: справочно-методическое пособие. 2012. 120 с.
3. Схваты роботов манипуляторов. URL: <https://kobotek.ru/katalog/zahvaty>.
4. Электромеханический схват с системой силового чувствования / А.А. Маргун, Д.Н. Базылев, К.А. Зименко, П.Д. Вахвиянова // Изв. вузов. Приборостроение. 2019. Т. 62, № 1. С. 93–96. DOI: 10.17586/0021-3454-2019-62-1-93-96
5. X3D-прототипирование моделей тел манипуляционных роботов / А.И. Телегин, Е.В. Гусев, В.Л. Кодкин, В.И. Ширяев // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2024. Т. 24, № 4. С. 16–30. DOI: 10.14529/ctcr240402

6. Brutzman D., Daly L. X3D: Extensible 3D graphics for web authors. Morgan Kaufmann Publishers Inc., 2007. 472 p. ISBN 978-0-12-088500-8.
7. Телегин А.И. Уравнения математических моделей механических систем. Челябинск: Изд-во ЮУрГУ, 1996. 182 с.
8. Телегин А.И. Основы теоретической механики систем тел. С приложениями в робототехнике: учеб. пособие для вузов. СПб.: Лань, 2023. 252 с. ISBN 978-5-507-45089-3.
9. Лескин А.А. Алгебраические модели гибких производственных систем. Л.: Наука, 1986. 150 с.
10. Борисенко Л.А. Исследование манипуляционных свойств механизма двухосной ориентации // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. 2006. № 10. С. 152–158.
11. Александров П.С. Введение в теорию групп. М.: Наука. Гл. ред. физ.-мат. лит., 1980. 144 с. (Библиотека «Квант». Вып. 7).
12. Голубев Ю.Ф. Алгебра кватернионов в кинематике твердого тела // Препринты ИПМ им. М.В. Келдыша. 2013. № 39. 23 с.
13. Коренев Г.В. Введение в механику человека. М.: Наука, 1977. 264 с.
14. Коренев Г.В. Введение в механику управляемого тела. М.: Наука, 1964. 568 с.
15. Телегин А.И., Тимофеев Д.Н., Читалов Д.И. X3D-моделирование механических систем: опыт использования X3D в моделировании механических систем. Южно-Уральский государственный университет, 2014. 66 с. URL: <http://elibraru.ru/item.asp?id=22585700>.
16. Бронштейн И.Н., Семенджев К.А. Справочник по математике. М., 1962. 608 с.
17. Попов Е.П., Верещагин А.Ф., Зенкевич С.Л. Манипуляционные роботы: динамика и алгоритмы. М.: Наука, 1978. 400 с.
18. Кирличников С.Н., Новоселов В.С. Математические аспекты кинематики твердого тела. Л.: Изд-во ЛГУ, 1986. 249 с.
19. Челноков Ю.Н. Кватернионные и бикватернионные модели и методы механики твердого тела и их приложения. Геометрия и кинематика движения. М.: Физматлит, 2006. 512 с. ISBN 5-9221-0680-5.
20. Крайнев А.Ф. Словарь-справочник по механизмам. 2-е изд. М.: Машиностроение, 1987. 560 с.

References

1. Ukrashenko K.A., Yanchevskiy Yu.V., Kulebyakin A.A., Toropov A.Yu. *Zakhvatnye ustroystva promyshlennykh robotov: ucheb. posobie* [Gripping devices for industrial robots. Training manual]. Yaroslavl: Yaroslavl State Technical University, 2007. 88 p. (In Russ.) ISBN 5-230-20645-4.
2. Ignat'ev N.P. *Skhvaty avtomaticheskikh manipulyatorov i promyshlennykh robotov: spravochno-metodicheskoe posobie* [Grips for automatic manipulators and industrial robots. Reference and methodological manual], 2012. 120 p. (In Russ.)
3. *Skhvaty robotov manipulyatorov* [Grips of robot manipulators]. (In Russ.) Available at: <https://kobotek.ru/katalog/zahvaty>.
4. Margun A.A., Bazylev D.N., Zimenko K.A., Vakhviyanova P.D. Electromechanical gripper with force sensing system. *Journal of Instrument Engineering*. 2019;62(1):93–96. (In Russ.) DOI: 10.17586/0021-3454-2019-62-1-93-96
5. Telegin A.I., Gusev E.V., Kodkin V.L., Shiriaev V.I. X3D prototyping of manipulation robot body models. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2024;24(4):16–30. (In Russ.) DOI: 10.14529/ctcr240402
6. Brutzman D., Daly L. X3D: Extensible 3D graphics for web authors. Morgan Kaufmann Publishers Inc., 2007. 472 p. ISBN 978-0-12-088500-8.
7. Telegin A.I. *Uravneniya matematicheskikh modeley mekhanicheskikh sistem* [Equations of mathematical models of mechanical systems]. Chelyabinsk: South Ural St. Univ. Publ.; 1999. 182 p. (In Russ.)
8. Telegin A.I. *Osnovy teoreticheskoy mekhaniki sistem tel. S prilozheniyami v robototekhnike: uchebnoe posobie dlya vuzov* [Fundamentals of theoretical mechanics of body systems. With applications in robotics: textbook for universities]. St. Petersburg: Lan; 2023. 252 p. (In Russ.) ISBN 978-5-507-45089-3.
9. Leskin A.A. *Algebraicheskie modeli gibkikh proizvodstvennykh sistem* [Algebraic models of flexible manufacturing systems]. Leningrad: Nauka Publ., 1986. 150 p. (In Russ.)
10. Borisenko L.A. [Investigation of the manipulative properties of the biaxial orientation mechanism]. *Vestnik Polotskogo gosudarstvennogo universiteta. Seriya C. Fundamental'nye nauki*. 2006;(10):152–158. (In Russ.)

11. Aleksandrov P.S. *Vvedenie v teoriyu grupp* [Introduction to the theory of groups]. Moscow: Nauka. Main Editorial Board of Physics and Mathematics Literature, 1980. 144 p. (Library "Kvant". Iss. 7).
12. Golubev Yu.F. [Algebra of quaternions in rigid body kinematics]. *Preprints of the Keldysh Institute of Applied Mathematics*. 2013, no. 39. 23 p. (In Russ.)
13. Korenev G.V. *Vvedenie v mekhaniku cheloveka* [Introduction to human mechanics]. Moscow: Nauka Publ., 1977. 264 p. (In Russ.)
14. Korenev G.V. *Vvedenie v mekhaniku upravlyayemogo tela* [Introduction to the mechanics of a controlled body]. Moscow: Nauka Publ., 1964. 568 p. (In Russ.)
15. Telegin A.I., Timofeev D.N., Chitalov D.I. *X3D-modeling of mechanical systems: experience in using X3D modeling of mechanical systems*. South Ural State University; 2014. 66 p. (In Russ.) Available at: <http://elibra.ru/item.asp?id=22585700>.
16. . Bronshteyn I.N., Semendyaev K.A. *Spravochnik po matematike* [Handbook of Mathematics]. Moscow. Moscow, 1962. 608 p. (In Russ.)
17. Popov E.P., Vereshchagin A.F., Zenkevich S.L. *Manipulyatsionnye roboty: dinamika i algoritmy* [Manipulation robots: dynamics and algorithms]. Moscow: Nauka Publ., 1978. 400 p. (In Russ.)
18. Kirpichnikov S.N., Novoselov V.S. *Matematicheskie aspekty kinematiki tverdogo tela* [Mathematical aspects of rigid body kinematics]. Leningrad: Leningrad State University Publ., 1986. 249 p. (In Russ.)
19. Chelnokov Yu.N. *Kvaternionnye i bikvaternionnye modeli i metody mekhaniki tverdogo tela i ikh prilozheniya. Geometriya i kinematika dvizheniya* [Quaternion and biquaternion models and methods of solid mechanics and their applications. Geometry and kinematics of motion]. Moscow: Fizmatlit Publ., 2006. 512 p. (In Russ.) ISBN 5-9221-0680-5.
20. Kraynev A.F. *Slovar'-spravochnik po mekhanizmam* [Dictionary and reference book on mechanisms]. 2nd ed. Moscow: Mashinostroenie Publ., 1987. 560 p. (In Russ.)

Информация об авторах

Телегин Александр Иванович, д-р физ.-мат. наук, проф., проф. кафедры автоматики, Южно-Уральский государственный университет, филиал в г. Миассе, Миасс, Россия; teleginai@susu.ru.

Волович Георгий Иосифович, д-р техн. наук, проф., директор, ООО «Челэнергоприбор», Челябинск, Россия; g_volovich@mail.ru.

Гусев Евгений Васильевич, д-р техн. наук, проф., проф. кафедры цифровой экономики и информационных технологий, Южно-Уральский государственный университет, Челябинск, Россия; guselev@susu.ru.

Пудовкина Светлана Геннадьевна, канд. техн. наук, доц., доц. кафедры прикладной математики и ракетодинамики, Южно-Уральский государственный университет, филиал в г. Миассе, Миасс, Россия; pudovkinasg@susu.ru.

Information about the authors

Aleksandr I. Telegin, Dr. Sci. (Phys. and Math.), Prof., Prof. of the Department of Automation, South Ural State University, Miass, Russia; teleginai@susu.ru.

Georgiy I. Volovich, Dr. Sci. (Eng.), Prof., Director, LLC Chelenergopribor, Chelyabinsk, Russia; g_volovich@mail.ru.

Evgeny V. Gusev, Dr. Sci. (Eng.), Prof., Prof. of the Department of Digital Economics and Information Technologies, South Ural State University, Chelyabinsk, Russia; guselev@susu.ru.

Svetlana G. Pudovkina, Cand. Sci. (Eng.), Ass. Prof., Ass. Prof. of the Department of Applied Mathematics and Rocket Dynamics, South Ural State University, Miass, Russia; pudovkinasg@susu.ru.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

Contribution of the authors: the authors contributed equally to this article.

The authors declare no conflicts of interests.

Статья поступила в редакцию 22.08.2025

The article was submitted 22.08.2025