

25 ноября 2019 года в Военном инновационном технополисе «ЭРА» (г. Анапа) проведена конференция на тему «Состояние и перспективы развития современной науки по направлению «Технологии энергообеспечения. Аппараты и машины жизнеобеспечения».

Цель конференции – Организация обмена информацией о новых научно-технических разработках, объединения ведущих научных школ, поиска партнёров в области разработки перспективных технологии энергетики, аппаратов и машин систем жизнеобеспечения в Вооружённых Силах Российской Федерации.

Задачи конференции:

- создание условий для эффективного взаимодействия органов военного управления с предприятиями ОПК на площадке ВИТ «ЭРА»;
- обмен опытом в области инновационных решений по направлению энергетики, технологий, аппаратов и машин систем жизнеобеспечения;
- обмен мнениями и уточнение приоритетных направлений развития химмотологии топлив, масел, смазок и специальных жидкостей в интересах Вооружённых Силах Российской Федерации, разработки композитных конструкционных материалов, аппаратов и машин систем жизнеобеспечения объектов военной инфраструктуры, источников электропитания и систем распределения энергетических ресурсов;
- уточнение вопросов формирования совместных научных коллективов для эффективного проведения исследований в областях деятельности лаборатории.

В конференции приняли участие как доктора и кандидаты наук, докторанты и адъюнкты (аспиранты) образовательных учреждений и научных организаций, так и операторы научных рот.

Основные результаты работы участников конференции отражены в сборнике статей. Содержание статей представлено в авторском изложении.

Ответственный редактор
капитан-лейтенант Ржавитин В.Л.
Компьютерная верстка
Минасян М.А.
Репин Д.В.

| | |
|---|-----|
| АНАЛИЗ СУЩЕСТВУЮЩИХ КОМПОЗИТНЫХ МАТЕРИАЛОВ И ОПРЕДЕЛЕНИЕ ПЕРСПЕКТИВ ИХ ДАЛЬНЕЙШЕГО РАЗВИТИЯ. | 181 |
| Фролов А.В., Плотникова Я. Р. | |
| ИСПОЛЬЗОВАНИЕ КОМПОЗИЦИОННЫХ МАТЕРИАЛОВ В АВИАЦИИ | 188 |
| Шайдуллин И.Н., Смелик А.А., Шевченко Я.В., Губанов Е.В. | |
| ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЗРАЧНЫХ БРОНЕМАТЕРИАЛОВ | 196 |
| Бакеев М.М., Фролов А.В. | |
| НАПРАВЛЕНИЯ СОВЕРШЕНСТВОВАНИЯ ТЕХНИЧЕСКИХ СРЕДСТВ СЛУЖБЫ ГОРЮЧЕГО АРМИЙ СТРАН НАТО | 203 |
| Трусов Д.Н., Вдовичев А.А., Ржавитин В.Л., Смелик А.А. | |
| АНАЛИЗ РАЗВИТИЯ КОНСТРУКЦИИ ТОПЛИВНЫХ ФОРСУНОК | 209 |
| Марков А.Р., Горшков С.Н., Иконников А.В. | |
| АЛГОРИТМЫ ОЦЕНКИ КАЧЕСТВА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ | 220 |
| Горшков С.Н., Маслов Н.С. | |
| АНАЛИЗ БИБЛИОТЕК BOOST.STATECHART И BOOST.META STATE MACHINE ДЛЯ РЕШЕНИЯ ЗАДАЧИ УПРАВЛЕНИЯ БПЛА КОПТЕРНОГО ТИПА | 225 |
| Иконников А.В., Марков А.Р., Горшков С.Н. | |
| ПОДХОДЫ В ОБРАБОТКЕ ИЗОБРАЖЕНИЙ ПРИ РАСПОЗНАВАНИИ ОБЪЕКТОВ | 231 |
| Усеинов И.А., Щербанев А.Ю., Кириченко А.А., Коваленко Р.В., Горшков С.Н. | |
| ПРИНЦИП РАБОТЫ С НАВИГАЦИОННОЙ АППАРАТУРОЙ ПРИ ОРИЕНТИРОВАНИИ НА МЕСТНОСТИ | 238 |
| Прокофьев М.А., Поляков Р.Г., Горшков С.Н. | |
| АНАЛИЗ СТОХАСТИЧЕСКОГО ПОДХОДА К МОДЕЛИРОВАНИЮ БОЕВЫХ ДЕЙСТВИЙ | 245 |
| Захаренков И.Г., Горшков С.Н. | |
| МЕТОДИКА РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ МОБИЛЬНЫХ ТЕРМИНАЛОВ ОПЕРАТОРА РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ..... | 253 |
| Поляков Р.Г., Горшков С.Н., Прокофьев М.А. | |
| СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ БОЕВЫХ ДЕЙСТВИЙ | 268 |

АНАЛИЗ БИБЛИОТЕК BOOST.STATECHART И BOOST.META STATE MACHINE ДЛЯ РЕШЕНИЯ ЗАДАЧИ УПРАВЛЕНИЯ БПЛА КОПТЕРНОГО ТИПА

Горшков С.Н., специалист, старший оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: sergik-gorshkov@mail.ru, 89097633050

Маслов Н.С., магистр, оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: webconn@mail.ru, 89263743420

Аннотация:

В статье рассматриваются вопросы связанные с использованием конечных автоматов для решения задачи управления БПЛА коптерного типа. Выполнен анализ специализированных библиотек, которые предоставляют все необходимые функциональные возможности для разработки полноценной машины состояний.

Ключевые слова: конечный автомат, программирование, БПЛА, квадрокоптер, C++.

Конечный автомат (или попросту FSM — Finite-state machine) это модель вычислений, основанная на гипотетической машине состояний. В один момент времени только одно состояние может быть активным. Следовательно, для выполнения каких-либо действий машина должна менять свое состояние.

Использование конечного автомата в робототехнических комплексах широко распространено. Существуют различные технологии реализации конечного автомата для задач программирования робототехнических комплексов, в частности летательных аппаратов коптерного типа.

Список необходимых состояний и подсостояний летательного аппарата коптерного типа:

1. Состояние удаленного управления.
2. Состояние успешной инициализации.
3. Состояние ошибки.
4. Состояние режима управления с бортового компьютера:
 1. Состояние инициализации управления с бортового компьютера.
 2. Состояние нахождения на земле.
 3. Состояние взлета.
 4. Состояние парения.
 5. Состояние движения.
 6. Состояние приземления.

Переходы между состояниями внутри конечного автомата происходят при помощи событий. Список событий конечного автомата летательного аппарата коптерного типа:

1. Событие взлета.
2. Событие начала приземления.
3. Событие зависания над точкой.
4. Событие успешного приземления.
5. Событие движения к точке.
6. Событие поворота.
7. Событие ошибки.
8. Событие переключения режима управления.

Для ведения лога полета коптера необходимо отслеживать изменение состояний внутреннего конечного автомата. Решение задачи логирования достигается путем записи момента входа и выхода состояний.

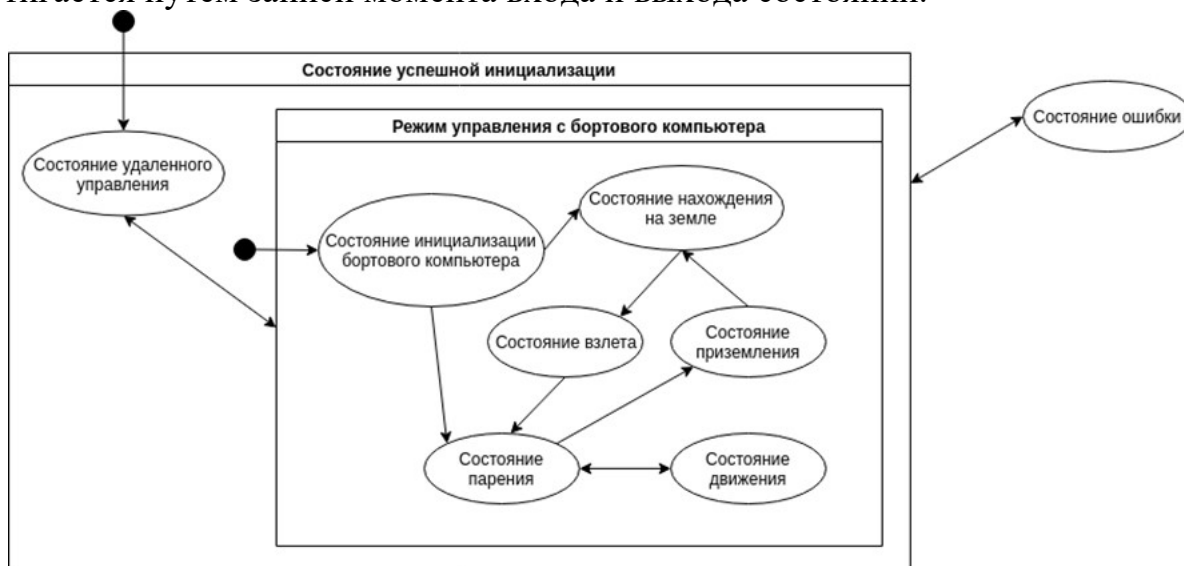


Рисунок 1 – Схема взаимодействия состояний конечного автомата.

В качестве основных инструментов по созданию конечного автомата летательного аппарата коптерного типа были выбраны Boost.StateChart и Boost.Meta State Machine. В ходе разработки и тестирования конечного автомата при помощи обеих библиотек были выявлены их недостатки и преимущества.

Недостатки Boost.StateChart:

1. Высокое потребление вычислительных ресурсов.
2. Некорректная работа нескольких активных состояний.

Преимущества Boost.StateChart:

1. Высокая расширяемость.

Недостатки Boost.Meta State Machine:

1. Менее расширяемая чем Boost.StateChart.

Преимущества Boost.Meta State Machine:

1. Низкое потребление вычислительных ресурсов.

2. Возможность работы нескольких активных состояний.
3. Больше возможностей для управления состояниями и событиями.

В ходе анализа установлено, что более эффективной является библиотека Boost.Meta State Machine.

Пример создания описания машины состояний основанной на применении библиотеки Boost.Meta State Machine.

```
struct stateMachine_
: public front::state_machine_def<stateMachine_> {
    Context _ctxt;
    stateMachine_() {}
    stateMachine_(Context &ctxt) {
        _ctxt = ctxt;
    }
    //Шаблонный метод входа в машину состояний. В шаблоне первый параметр - событие, вызвавшее переход, 2й - сама машина состояний
    template<class Event, class FSM>
    void on_entry(Event const &, FSM &) {
        BOOST_LOG_TRIVIAL(info) << "MetaAI: Entering: StateMachine";
    }
    template<class Event, class FSM>
    void on_exit(Event const &, FSM &) {
        BOOST_LOG_TRIVIAL(info) << "MetaAI: Leaving: StateMachine";
    }
    template<class FSM, class Event>
    void no_transition(Event const &e, FSM &, int state) {
        BOOST_LOG_TRIVIAL(info) << "no transition from state " << state << " on event " << typeid(e).name();
    }
};
```

Пример создания описания события конечного автомата основанного на применении библиотеки Boost.Meta State Machine:

```
namespace Events {
...
struct TakeOff {};
...
struct Move {
    offcon::Point3 _target;
    explicit Move(offcon::Point3 target)
        : _target(target) {}
};
...
} // namespace Events
```

В примере, показано создание 2х событий: событие взлета и событие движения.

Пример создания описания состояния конечного автомата основанного на применении библиотеки Boost.Meta State Machine.

```
//Состояние управления с пульта
struct RemoteController
: public front::state<> {
    //Шаблон FSM - объект машины состояний, к которой принадлежит состояние
    template<class Event, class FSM>
    void on_entry(Event const &, FSM &fsm) {
        BOOST_LOG_TRIVIAL(info) << "MetaAI: Entering: RemoteController";
    }
    template<class Event, class FSM>
    void on_exit(Event const &, FSM &) {
        BOOST_LOG_TRIVIAL(info) << "MetaAI: Leaving: RemoteController";
    }
};
```

Пример создания описания действий, которые происходят при вызове события.

```
//Действие, выполняемое при вызове события TakeOff
struct do_takeoff {
    template<class EVT, class FSM, class SourceState, class TargetState>
    void operator()(EVT const &, FSM &fsm, SourceState &, TargetState &) {
        fsm._ctxt.mController->takeOff(fsm._ctxt.params->behaviour.height);
    }
};
```

Пример создания правил перехода между состояниями.

//Таблица переходов и действий между состояниями подмашины offboard_

```
struct transition_table
: mpl::vector<
    // Start Event Next Action Guard
    // +-----+-----+-----+-----+-----+
-----+
    _row < Init, Events::Ground, Grounding>,
    front::Row<Init, Events::Hover, Hovering, offboard_::do_hover>,
    // +-----+-----+-----+-----+-----+
--++
    front::Row<Grounding, Events::TakeOff, TakeOffing, offboard_::do_take-
off>,
    // +-----+-----+-----+-----+-----+
--++
```

```

    _row<TakeOffing, Events::Hover, Hovering>,
    // +-----
--+
    front::Row<Hovering, Events::Land, Landing, offboard_::do_land>,
    front::Row<Hovering, Events::Move, Moving, offboard_::do_move>,
    front::Row<Hovering,      Events::MoveSelf,      Moving,      off-
board_::do_move_self>,
    front::Row<Hovering, Events::Turn, Moving, offboard_::do_turn>,
    front::Row<Hovering,      Events::TurnSelf,      Moving,      off-
board_::do_turn_self>,
    // +-----
--+
    front::Row<Moving, Events::Move, Moving, offboard_::do_move>,
    front::Row<Moving,      Events::MoveSelf,      Moving,      off-
board_::do_move_self>,
    front::Row<Moving, Events::Turn, Moving, offboard_::do_turn>,
    front::Row<Moving, Events::TurnSelf, Moving, offboard_::do_turn_self>,
    front::Row<Moving, Events::Hover, Hovering, offboard_::do_hover>,
    // +-----
--+
    _row<Landing, Events::LandOk, Grounding>> {};
```

Пример объявления типа состояния для создания объекта состояния, содержащего подсостояния.

//Создание типа машины состояний из описания (offboard_ - это описание конечного автомата)

```
typedef msm::back::state_machine<offboard_> Offboard;
```

Пример объявления типа машины состояний для создания объекта конечного автомата.

```
typedef msm::back::state_machine<stateMachine_> StateMachine;
```

Таким образом, для решения задачи управления БПЛА на основе машины состояний наиболее эффективным является использование библиотеки BoostMeta State Machine. Она позволяет разрабатывать сложный и производительный конечный автомат, с широкими функциональными возможностями, которые необходимы в задаче управления БПЛА.

Литература

1. Конечный автомат: теория и реализация [Электронный ресурс]. URL: <https://tproger.ru/translations/finite-state-machines-theory-and-implementation/>. (дата публикации: 25.09.2015)

2. Basic front-end. Chapter 3 Tutorial [Электронный ресурс]. URL: https://www.boost.org/doc/libs/1_64_0/libs/msm/doc/HTML/ch03s02.html. (дата обращения: 08.01.2020).

3. Functor front-end. Chapter 3 Tutorial
[Электронный ресурс]. URL: https://www.boost.org/doc/libs/1_72_0/libs/msm/doc/HTML/ch03s03.html. (дата обращения: 08.01.2020).

4. Гилл А. Введение в теорию конечных автоматов // Издательство «Наука» // Москва. – 1966. – 272 с.