

## ***JMCAD (JMCADRTS, JMCADRTC)***

- 1   Опис**
- 2   Застосування**
- 3   Структура**
- 4   Встановлення системи**
- 5   Запуск і використання**
  - 5.1 JMCAD**
  - 5.2 JMCADRTS**
  - 5.3 JMCADRTC**
- 6   Розробка**
  - 6.1 Структура класів**
  - 6.2 Компіляція**
  - 6.3 Створення візуальних елементів**
  - 6.4 Створення і локалізація документації**
  - 6.5 Локалізація інтерфейсу користувача**
- 7   Підручник**
  - 7.1 Графічний інтерфейс JMCAD**
  - 7.2 Візуальні елементи та принципи роботи з нити**
  - 7.3 Створення моделі**
  - 7.4 Запуск та відлагодження моделі**
  - 7.5 Створення графічного інтерфейсу для керування моделлю**
  - 7.6 Математичні розрахунки**

# 1 Опис

Програмний комплекс **JMCAD** призначений для аналізу динаміки і проектування різноманітних систем і пристроїв. За своїми можливостями він є альтернативою аналогічним програмним продуктам **LabView**, **Simulink**, **VisSim**, **MBTU** і ін. Зручний редактор структурних схем, велика бібліотека типових блоків і вбудована мова програмування дозволяють реалізовувати моделі практично будь-якого ступеня складності, забезпечуючи при цьому наочність їх представлення. Програмний комплекс **JMCAD** успішно застосовується для проектування систем автоматичного керування, приводів що стежать і роботів-маніпуляторів, теплових енергетичних установок, а також для вирішення нестационарних крайових задач (теплопровідність, гідродинаміка та ін.).

Систему широко використовується в навчальному процесі, моделюючи різні явища у фізиці, електротехніці, в динаміці машин і механізмів і ін. Програмний комплекс може функціонувати у кластерах, в тому числі і в режимі віддаленого доступу до технологічних та інформаційних ресурсів.

Для користувачів зручність роботи з **JMCAD** обумовлено також локалізацією інтерфейсу на різні мови і наявністю документації.

Версії **JMCAD** доступні з вихідними текстами ядра, бібліотек і є відкритою системою з повною документацією і набором демонстраційних прикладів. Також до складу комплексу входять модулі для забезпечення максимальної продуктивності і контролю в реальному часі (**JMCADRTS**, **JMCADRTC**).

Програмний комплекс **JMCAD** розроблений з використанням мови **Java** (<http://java.sun.com>) і може бути використаний у різних операційних системах (**Windows**, **Linux**, **Solaris**, **Unix** і ін.).

## 2 Застосування

Програмний комплекс **JMCAD** реалізує наступні режими роботи:

- **МОДЕЛЮВАННЯ**, що забезпечує:
  - моделювання процесів у безперервних, дискретних та гібридних динамічних системах, в тому числі і за наявності обміну даними з зовнішніми програмами та пристроями;
  - редагування параметрів моделі в режимі «on-line»;
  - розрахунок у реальному часі або в режимі масштабування модельного часу;
  - рестарт і відтворення результатів моделювання;
  - динамічну обробку сигналів.
- **ОПТИМІЗАЦІЯ**, що дозволяє виконувати завдання:
  - мінімізації (максимізації) заданих показників якості;
  - знаходження оптимальних параметрів проектованої системи в

багатокритеріальної постановці за наявності обмежень на показники якості та параметри, що оптимізуються.

- АНАЛІЗ, що забезпечує:
  - розрахунок і побудову характеристик статичних і динамічних систем;
  - розрахунок передавальних функцій;
  - візуалізацію результатів аналізу статично і динамічно.
- СИНТЕЗ, що дозволяє конструювати регулятори:
  - за заданими бажаними частотними характеристиками;
  - по заданому розташуванню домінуючих полюсів.
- КОНТРОЛЬ І УПРАВЛІННЯ, що дозволяє створювати віртуальні прототипи:
  - пультів керування з вимірювальними приладами і керуючими пристроями;
  - мнемосхем з мультимедійними і анімаційними ефектами.

До переваг **JMCAD** відносяться:

- відкритість за рахунок використання мови **Java** і реалізації декількох механізмів обміну даними з зовнішніми програмами;
- можливість використовувати у різних операційних системах (**Windows**, **Linux**, **Solaris**, **Unix** і ін.);
- простота побудови складних моделей завдяки використанню вкладених структур, векторизації сигналів і алгоритмів типових блоків, зручним засобам завдання параметрів і рівнянь;
- ефективні чисельні методи;
- велика кількість навчальних та демонстраційних прикладів з докладними коментарями.

## 3 Структура

Програмний комплекс має три окремих незалежних блоки **JMCAD**, **JMCADRTS**, **JMCADRTC**. Кожен блок може працювати самостійно, а також при створенні розподілених систем можна використовувати їх спільно. Спільне використання дозволяє створювати складні розподілені системи з можливістю швидкого і легкого розвитку системи. Така функціональна особливість дозволяє вносити зміни в працюючу систему без її зупинки. Для цього досить створити розгалуження і дублюючий блок, на якому і проводити розробку, а після закінчення і тестування цей блок включити в систему як основний. Замінивши тим самим старий блок без зупинки роботи всієї системи.

**JMCAD** – основний блок для створення і редагування моделей (рис. 3.1). Також можливий варіант його використання для запуску моделі в режимі роботи (рис. 3.2). Запуск моделі в режимі роботи проводиться через командний рядок за допомогою параметра **-single**.

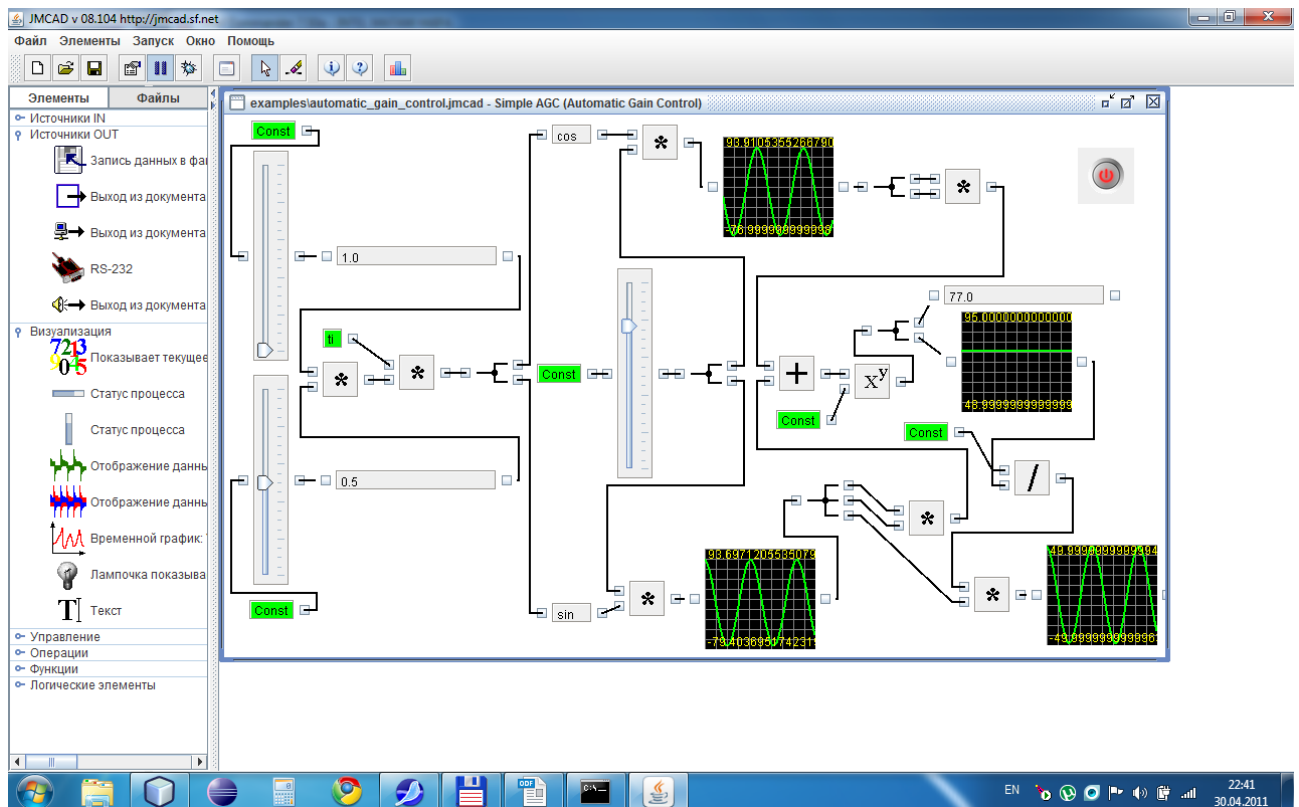


Рис. 3.1

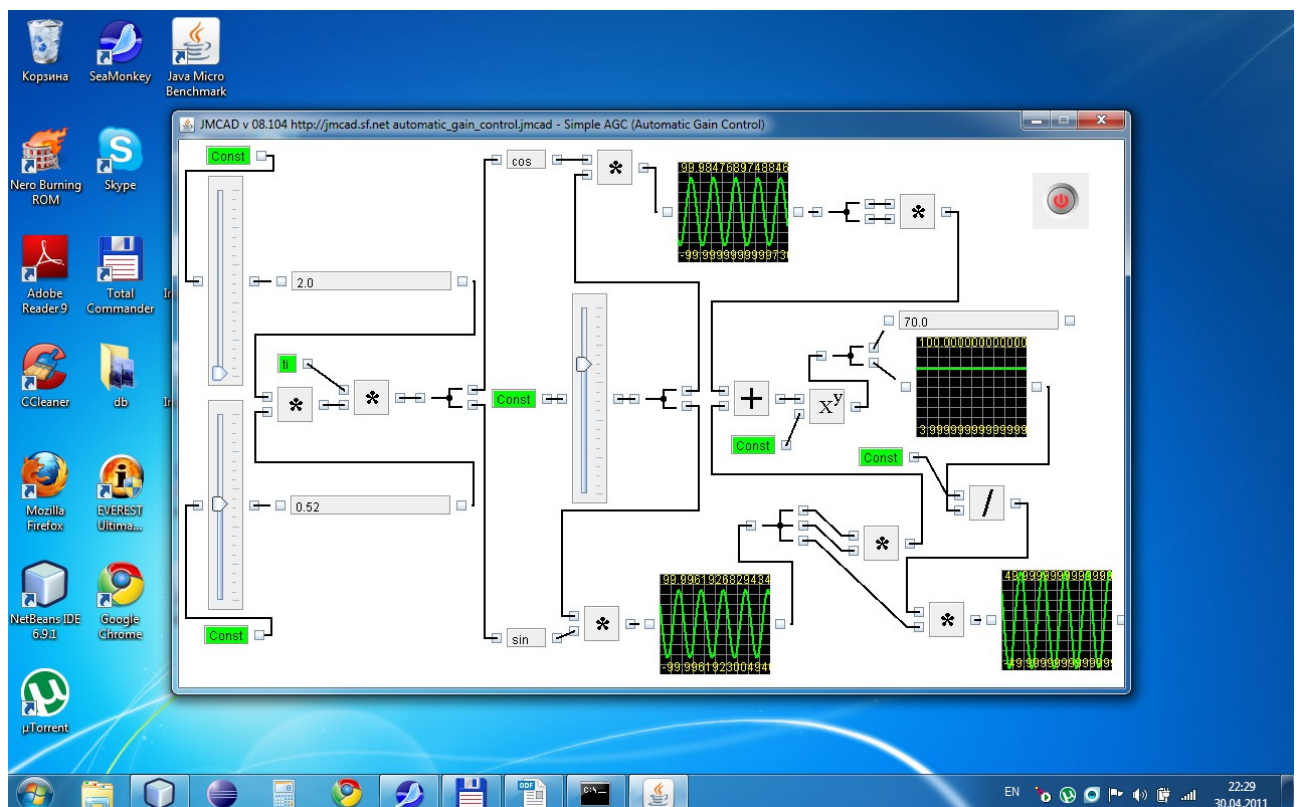


Рис. 3.2

**JMCADRTS** - блок для запуска модели в режиме работы (рис. 3.2). Запуск модели в режиме работы проводится через командный рядок.

**JMCADRTC** - блок для запуска интерфейсу контролю та управління

моделлю (рис. 3.3). Запуск інтерфейсу контролю та управління моделлю проводиться через командний рядок.

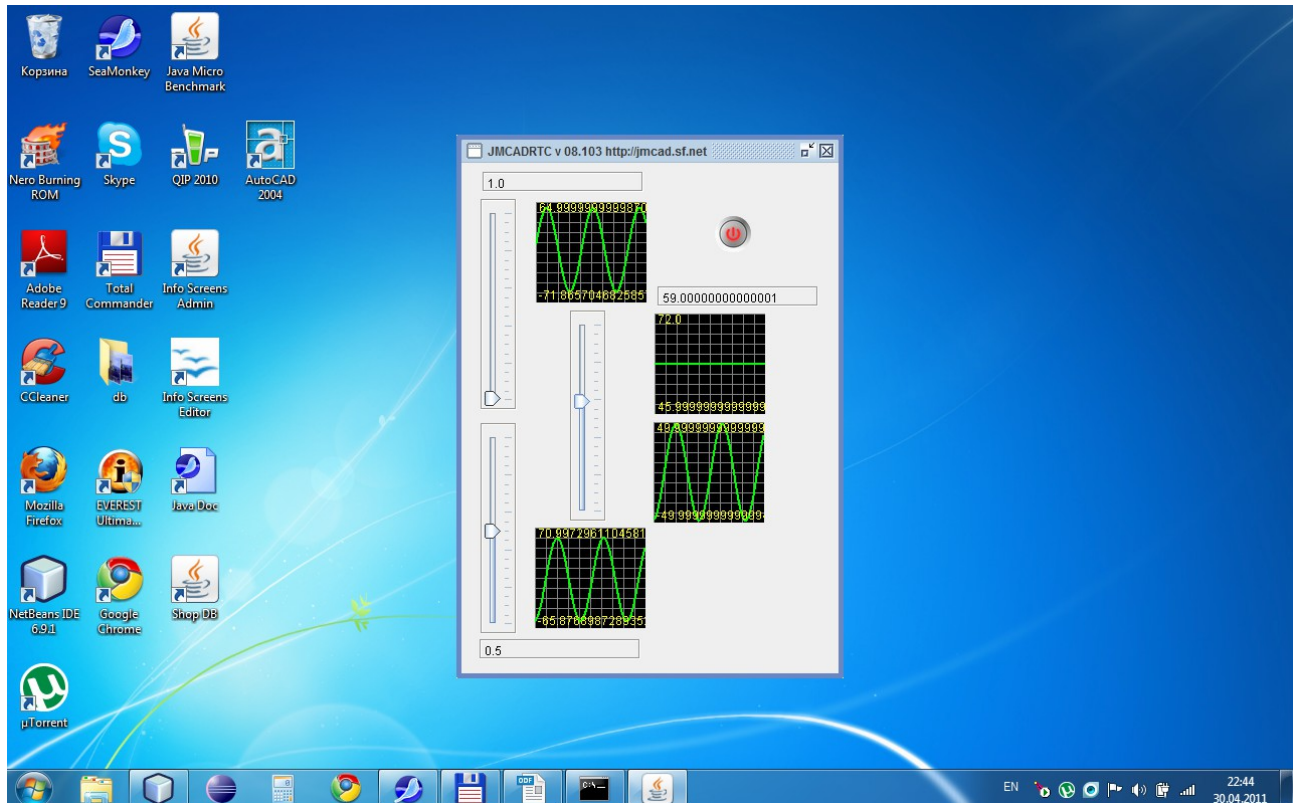


Рис. 3.3

## 4 Встановлення системи

Програмний комплекс **JMCAD** може бути встановлений на будь-який комп'ютер як локально так і з можливістю роботи по мережі.

Для установки системи **JMCAD** необхідно встановити **Java** (<http://java.sun.com/javase/downloads/>). Рекомендується встановлювати версію **Java SE Runtime Environment 7 (JRE)** або вище для запуску програмного комплексу, а для розробки використовувати версію **Java SE Development Kit 7 (JDK)** або вище.

Програмний комплекс **JMCAD** можна завантажити з сервера <http://jmcad.sf.net>. На сервері доступні два варіанти дистрибутива:

- **JMCAD-XX.XXX-bin.zip** – дистрибутив програмного комплексу готовий для роботи;
- **JMCAD-XX.XXX-all.zip** - дистрибутив програмного комплексу для роботи і розробників, що містить вихідні коди.

Для установки **JMCAD** необхідно розпакувати файл **JMCAD-XX.XXX-bin.zip** або **JMCAD-XX.XXX-all.zip** на жорсткий диск. При розархівуванні дистрибутиву буде створена директорія з тим же ім'ям що і у файлу архіву. У директорії будуть знаходитися файли відповідні призначенням дистрибутива.



## 5 Запуск і використання

Для запуску програмного комплексу використовуються файли скриптів **\*.bat** (**\*.sh**). Приклад файлів скриптів для запуску **JMCAD**, **JMCADRTS**, **JMCADRTC** знаходяться в директорії **examples**.

### 5.1 JMCAD

Для запуску програмного комплексу **JMCAD** використовуються файли скриптів **\_\_JMCAD\*.bat** (**\_\_JMCAD\*.sh**). У місці символу **\*** вибирається файл, що відповідає операційній системі де буде використовуватися програмний комплекс.

Також можливий варіант його використання для запуску моделі в режимі роботи (рис. 3.2). Запуск моделі в режимі роботи проводиться через командний рядок за допомогою параметра **-single**. Командний рядок для запуску моделі в режимі роботи виглядає наступним чином:

**java [parameters] <CLASSPATH> JMCAD -single <MODEL>**

де:

**[parameters]** — параметри для віртуальної машини **Java**. Для великих моделей при нестачі оперативної пам'яті необхідно вказувати значення параметра **-Xmx1000m**, який визначає доступний розмір оперативної пам'яті для віртуальної машини **Java**. Також можна використовувати параметр **-server** для збільшення швидкості розрахунку. Більш детально про значення та їх застосуванні можливо дізнатися з документації по віртуальній машині **Java**. Дані параметри є необов'язковими і можуть не вказуватися.

**<CLASSPATH>** - використовується для підключення бібліотек класів. Всі бібліотеки які необхідні для роботи моделі повинні бути доступні віртуальній машині **Java**. Існує кілька варіантів вказівки віртуальній машині **Java** де шукати бібліотеки класів. Детально це описано в документації по віртуальній машині **Java**. Для цього випадку в рядку опису обов'язково повинні бути присутніми бібліотеки які необхідні для запуску та роботи моделі. Основна з усіх бібліотек класів є бібліотека, що знаходиться в **jar** архіві **JMCAD.jar** і містить клас для запуску **JMCAD**.

**<MODEL>** - параметр командного рядка який визначає файл моделі (**\*.jmcad**) для запуску.

### 5.2 JMCADRTS

Запуск моделі в режимі роботи проводиться через командний рядок. Командний рядок для запуску моделі в режимі роботи виглядає наступним

ЧИНОМ:

***java [parameters] <CLASSPATH> JMCADRTS <MODEL>***

где:

***[parameters]*** — параметри для віртуальної машини ***Java***. Для великих моделей при нестачі оперативної пам'яті необхідно вказувати значення параметра ***-Xmx1000m***, який визначає доступний розмір оперативної пам'яті для віртуальної машини ***Java***. Також можна використовувати параметр ***-server*** для збільшення швидкості розрахунку. Більш детально про значення та їх застосуванні можливо дізнатися з документації по віртуальній машині ***Java***. Дані параметри є необов'язковими і можуть не вказуватися.

***<CLASSPATH>*** - використовується для підключення бібліотек класів. Всі бібліотеки які необхідні для роботи моделі повинні бути доступні віртуальній машині ***Java***. Існує кілька варіантів вказівки віртуальній машині ***Java*** де шукати бібліотеки класів. Детально це описано в документації по віртуальній машині ***Java***. Для цього випадку в рядку опису обов'язково повинні бути присутніми бібліотеки які необхідні для запуску та роботи моделі. Основна з усіх бібліотек класів є бібліотека, що знаходиться в ***jar*** архіві ***JMCADRTS.jar*** і містить клас для запуску ***JMCADRTS***.

***<MODEL>*** - параметр командного рядка який визначає файл моделі (***\*.jmcad***) для запуску.

## 5.3 JMCADRTC

Запуск інтерфейсу контролю та керування моделлю проводиться через командний рядок, що виглядає наступним чином:

***java [parameters] <CLASSPATH> JMCADRTC <MODEL>***

где:

***[parameters]*** — параметри для віртуальної машини ***Java***. Для великих моделей при нестачі оперативної пам'яті необхідно вказувати значення параметра ***-Xmx1000m***, який визначає доступний розмір оперативної пам'яті для віртуальної машини ***Java***. Також можна використовувати параметр ***-server*** для збільшення швидкості розрахунку. Більш детально про значення та їх застосуванні можливо дізнатися з документації по віртуальній машині ***Java***. Дані параметри є необов'язковими і можуть не вказуватися.

***<CLASSPATH>*** - використовується для підключення бібліотек класів. Всі бібліотеки які необхідні для роботи моделі повинні бути доступні віртуальній машині ***Java***. Існує кілька варіантів вказівки віртуальній машині ***Java*** де шукати бібліотеки класів. Детально це описано в документації по віртуальній машині ***Java***. Для цього випадку в рядку опису обов'язково повинні бути присутніми бібліотеки які необхідні для запуску та роботи моделі. Основна з усіх бібліотек

класів є бібліотека, що знаходиться в *jar* архіві **JMCADRTC.jar** і містить клас для запуску **JMCADRTC**.

**<MODEL>** - параметр командного рядка який визначає файл моделі (**\*.jmcad**) для запуску.

## 6 Розробка

Програмний комплекс **JMCAD** є відкритою системою з можливістю розробки нових модулів. Система розроблена з використанням мови **Java** (<http://java.sun.com>), що дозволяє використовувати її в різних операційних системах (**Windows**, **Linux**, **Solaris**, **Unix** и т.д.).

В основі системи знаходиться ядро яке контролює роботу системи. Ядро відповідає за запуск, роботу і зупинку моделі. У програмному комплексі використовується три ядра - **JMCAD**, **JMCADRTS**, **JMCADRTC**. Принцип роботи полягає в тому що ядро контролює передачу даних між елементами, а всі дії в елементі виконуються в автономно. Ядро синхронізує передачу даних між елементами і працює в три етапи (запуск, робота, зупинка):

- **запуск** — ядро запускає метод **calc\_pre()**; для всіх елементів моделі. Метод **calc\_pre()**; використовується для підготовки елемента до роботи;
- **робота** — ядро запускає метод **calc()**; в циклі для всіх елементів моделі із зазначеною затримкою. Для елементів які є джерелами викликається метод **start(long t0, long ti, long dt)**; Якщо вказано значення 0 то система працює в реальному часі;
- **зупинка** — ядро запускає метод **calc\_post()**; для всіх елементів моделі. Метод **calc\_post()**; використовується для підготовки елемента до зупинки роботи.

### 6.1 Структура класів

В основу системи покладено модульний принцип. Такий підхід дозволяє легко розробляти нові елементи. Призначення і функції елементів системи розподіляються по різних файлів:

- **JMCAD\*.java** — ядро системи і графічний інтерфейс;
- **JMCAD\_Internationalize\_xx\_XX.properties** — локалізація інтерфейсу. Замість символів **xx** і **XX** вказуються символи, що визначають країну і мову;
- **JMCAD.menu** — меню візуальних елементів;
- **\_\*.java** — візуальні елементи. Також для зручності файли які відносяться до даного візуальному елементу мають аналогічне ім'я.



## 6.2 Компіляція

Для компіляції та упаковки всіх класів в *jar* архів використовується пакетний файл `__make_jar.bat` (`__make_jar.sh`). Ці пакетні файли знаходяться для кожного блоку в директорії з вихідними файлами даного блоку:

- *src* – *JMCAD* (*JMCAD.jar*);
- *src\_rts* – *JMCADRTS* (*JMCADRTS.jar*);
- *src\_rtc* – *JMCADRTC* (*JMCADRTC.jar*).

## 6.3 Створення візуальних елементів

Всі моделі в *JMCAD* створюються з візуальних елементів, які визначають ефективність цієї моделі. У системі присутній великий набір стандартних типових елементів. Але завжди є об'єктивна необхідність збільшувати набір стандартних типових елементів, додаючи нові елементи або модифікуючи вже наявні. Для цього в програмному комплексі *JMCAD* передбачений простий спосіб створення нових елементів.

Для створення базової структури візуального елемента використовується спадкування базового класу для всіх візуальних елементів *JMCADObject*. Цей клас містить змінні і методи необхідні для створення нових візуальних елементів.

Основні змінні:

- *in* — масив вхідних значень. За замовчуванням розмір масиву дорівнює 0. Розмір масиву визначає кількість входів в елемент. Отримання значення на вхід зберігається в комірці масиву відповідне індексу входу;
- *out* — масив вихідних значень. За замовчуванням розмір масиву дорівнює 0. Розмір масиву визначає кількість виходів з елемента. Після того як на всі входи елемента *in* отримані значення запускається метод *calc()*; який виконує описані в ньому дії і після чого всі значення, що знаходяться в масиві *out* передаються на інші елементи;
- *in\_text* — масив імен вхідних значень. Розмір масиву повинен збігатися з масивом вхідних значень *in*. Імена описуються латинськими літерами і цифрами. Повинні починатися з букви і не містити спецсимволів;
- *out\_text* — масив імен вихідних значень. Розмір масиву повинен збігатися з масивом вихідних значень *out*. Імена описуються латинськими літерами і цифрами. Повинні починатися з букви і не містити спецсимволів;
- *ToolTipText* — текстова підказка, що з'являється при наведенні курсору миші на елемент;
- *w, h* — ширина і висота елемента. Значення задаються в конструкторі;
- *isGeneranor* — змінна визначає тип елемента стандартний або джерело. За замовчуванням змінна має значення *false*, що визначає елемент як стандартний і ядро запускає метод *calc()*; При вказівці змінній значення

*true* елемент має тип джерела і ядро запускає метод *start(long t0, long ti, long dt)*;

- *isVisual* — змінна визначає видимість елемента при використанні його в інтерфейсі контролю та керування моделлю (рис. 3.3). За замовчуванням має значення *false*, що робить елемент невидимим в інтерфейсі контролю та керування моделлю;
- *panel\_c* — центральна панель елемента;
- *edit\_panel* — панель створення графічного інтерфейсу, який може використовуватися для редагування властивостей елемента. При подвійному клацанні на елементі, в режимі редагування моделі, відкривається діалогове вікно, що містить цю панель.

Основні методи:

- *calc\_pre()* — метод використовується для підготовки елемента до роботи;
- *calc()* — метод буде викликатися на кожному кроці роботи моделі із зазначеною затримкою. Для елементів які є джерелами викликається метод *start(long t0, long ti, long dt)*. Якщо вказано значення 0 то система працює в реальному часі;
- *calc\_post()* — метод використовується для підготовки елемента до зупинки роботи;
- *start(long t0, long ti, long dt)* — метод для елементів які є джерелами і буде викликатися на кожному кроці роботи моделі із зазначеною затримкою. Якщо вказано значення 0 то система працює в реальному часі;
- *edit\_pre()* — метод використовується для створення графічного інтерфейсу при редагування властивостей елемента. Графічний інтерфейс розташовується на панелі *edit\_panel*, яка потім відображається при подвійному клацанні на елементі в діалоговому вікні.
- *edit\_post()* — метод викликається якщо при зміні властивостей елемента в діалоговому вікні була натиснута кнопка підтвердження змін;
- *paint\_info(Graphics g)* — метод забезпечує доступ до графічних методів даного елемента;
- *parse(String pst)* — метод забезпечує зчитування даних з моделі. Тіло методу повинне починатися з рядка *super.parse(pst)*;
- *write(RandomAccessFile fout)* — метод забезпечує збереження параметрів елемента в моделі. Тіло методу повинне починатися з рядка *super.write(fout)*.

## 6.4 Створення і локалізація документації

Програмний комплекс *JMCAD* містить алгоритм автоматичного відображення документації для кожного елемента і формує загальну документацію.

При подвійному натисканні на елементі, в режимі редагування моделі, відкривається діалогове вікно, що містить кнопку виклику вікна з

документацією по даному елементу.

Для відображення документації в директорії **help** повинен знаходитися файл з документацією по даному елементу. Файл має формат представлення даних **HTML** і повинен мати таке ж ім'я як клас елемента. Для локалізації документації до ім'я файлу додається префікс **\_xx\_XX**. Де **xx** визначає країну, а **XX** визначає мову.

## 6.5 Локалізація інтерфейсу користувача

Локалізація інтерфейсу робиться стандартними методами мови **Java**. Рядки для локалізації знаходяться у файлах для локалізації інтерфейсу користувача **JMCAD\_Internationalize\_xx\_XX.properties**. Де в місці символів **xx** і **XX** вказуються символи визначають країну і мову. Більш детально про методики локалізації можна дізнатися з документації для програмування на **Java**.

# 7 Підручник

## 7.1 Графічний інтерфейс JMCAD

Графічний інтерфейс програмного комплексу **JMCAD** призначений для візуального створення, запуску та тестування моделей (рис. 7.1).

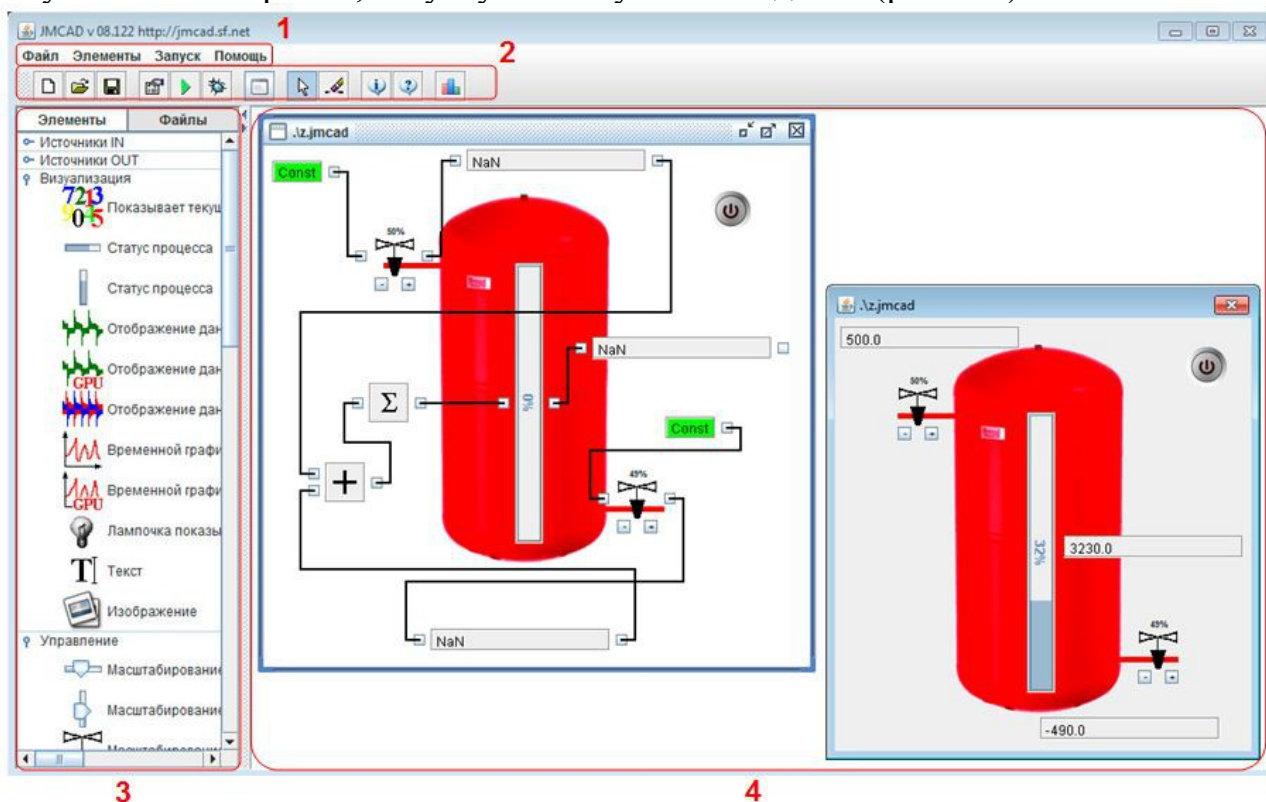


Рис. 7.1

Графічний інтерфейс *JMCAD* складається з чотирьох зон: 1- головне випадаюче меню; 2 - кнопочке меню інструментів; 3 - панель елементів та каталог моделей; 4 - робоча зона для створення та редагування моделей.

## 7.2 Візуальні елементи та принципи роботи з НИТИ

Модель формується з візуальних елементів між якими встановлюються зв'язки. Зв'язки визначають напрямок передачі даних між елементами. Дані передаються дійсними числами подвійної точності (double), а також невизначеним числовим значенням (Double.POSITIVE\_INFINITY).


Елементи моделі бувають трьох типів:

- джерело вхідних даних (елемент має зелений колір) - створює данні або отримує данні від пристроїв ЕОМ і може бути чотирьох типів:
  - звичайне джерело вхідних даних, що має один чи декілька виходів;
  - кероване джерело вхідних даних, що має один чи декілька виходів та один чи декілька входів для керування;
  - комбіноване джерело вхідних даних з джерелом вихідних даних;
  - умовне джерело вхідних даних для створення блоків моделей, що в подальшому будуть використовуватися як один елемент.
- джерело вихідних даних (елемент має жовтий колір) - отримує данні та передає їх на пристроїв ЕОМ і може бути трьох типів:
  - звичайне джерело вихідних даних, що має один чи декілька входів;
  - комбіноване джерело вихідних даних з джерелом вхідних даних;
  - умовне джерело вихідних даних для створення блоків моделей, що в подальшому будуть використовуватися як один елемент.
- звичайний елемент, що проводить обробку даних, виконує керування, візуалізацію, тощо.


Зверніть увагу, що елементи працюють тільки тоді коли на всі їх входи подаються значення. Винятком є елемент - комбіноване джерело вхідних даних з джерелом вихідних даних. В цьому елементі дозволяється використовувати одну або обидві частини комбінованого елемента.

Кожен елемент має певні властивості, які можливо редагувати. Для редагування властивостей елемента необхідно двічі клацнувши курсором по елементу і у діалоговому вікні змінити властивості елемента. У діалоговому вікні властивостей елемента виводиться опис елемента та є кнопка виклику детальної документації для цього елемента.

## 7.3 Створення моделі

Для створення нової моделі у програмному комплексі *JMCAD* необхідно скористатися командою, що знаходиться у меню “Файл” або на панелі кнопочного меню інструментів “Новий” . При використанні даної команди

створюється нове вікно документу моделі, у якому в подальшому розміщуються елементи моделі між якими встановлюються зв'язки.

Також для обробки даних важливо встановити інтервал часу через який буде проводитися повторна обробка даних для усіх елементів моделі на наступному кроці роботи моделі (п. 7.4). Для цього використовується команда, що знаходиться у меню “Запуск” або на панелі кнопкового меню інструментів “Налаштування параметрів розрахунку” . Виконати цю команду можливо двічі клацнувши курсором по вільному місцю у вікні моделі.

Розглянемо покрокове створення моделі на прикладі складання синусоїдального та прямокутного сигналу (рис. 7.2):

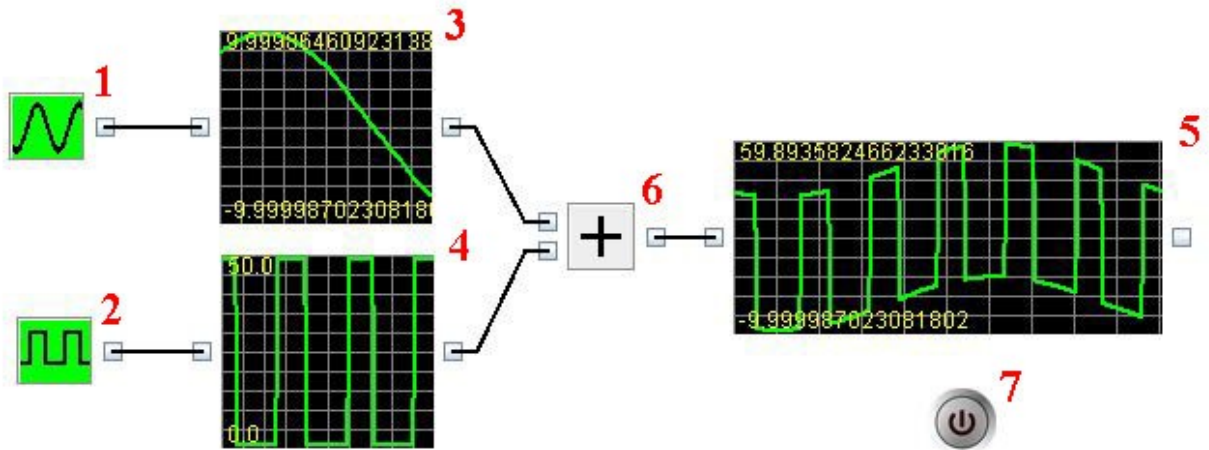











Рис. 7.2


- створюємо нове вікно документу моделі командою “Новий”  з меню “Файл” або панелі кнопкового меню інструментів;
- з меню “Елементи” розділ “Джерела IN” в модель додаємо генератори синусоїдального  1 та прямокутного сигналу  2;
- з меню “Елементи” розділ “Візуалізація” додаємо елементи візуалізації сигналу  3, 4, 5;
- з меню “Елементи” розділ “Операції” додаємо елемент складання  6;
- з меню “Елементи” розділ “Управління” додаємо елемент запуску і зупинки моделі  7;
- з'єднуємо входи, виходи елементів і за допомогою елемента  7 запускаємо (зупиняємо) модель.

## 7.4 Запуск та відлагодження моделі

Модель у програмному комплексі **JMCAD** запускається (зупиняється)




командою, що знаходиться у меню “Запуск” або на панелі кнопкового меню інструментів “Запуск” . Одночасно можливо запустити необмежену кількість моделей. Кількість одночасно запуснених моделей обмежується тільки потужністю ЕОМ. Для перевірки та відлагодження моделі використовується команда, що знаходиться у меню “Запуск” або на панелі кнопкового меню інструментів “Відладка” . Ця команда працює разом з командою “Запуск”.

Налаштування режиму роботи та відлагодження виконується командою, що знаходиться у меню “Запуск” або на панелі кнопкового меню інструментів “Налаштування параметрів розрахунку” . Виконати цю команду можливо двічі клацнувши курсором по вільному місцю у вікні моделі.


Налаштування роботи моделі задається параметром “Затримка по часу”. Параметр задає інтервал часу через який буде проводитися повторна обробка даних, для усіх елементів моделі, на наступному кроці роботи моделі і задається в мілісекундах.

Налаштування відлагодження моделі задається параметром “Затримка по часу для відлагодження”. Параметр задає інтервал часу затримки трасування даних для кожного елементу і задається в мілісекундах. Рекомендований інтервал від 50 до 500 мілісекунд.


Для запуску і зупинки моделі у програмному комплексі **JMCAD** можливо використовувати елемент запуску і зупинки моделі , з меню “Елементи” розділ “Управління”.

## 7.5 Створення графічного інтерфейсу для керування моделлю

В програмний комплекс **JMCAD** входять модулі для забезпечення максимальної продуктивності і контролю в реальному часі (**JMCADRTS**, **JMCADRTC**). Блок **JMCADRTC** дозволяє запускати інтерфейс контролю та управління моделлю як окрему програму (п. 5.3). Запуск інтерфейсу контролю та управління моделлю проводиться через командний рядок або за допомогою файлу скриптів.

Створення графічного інтерфейсу для моделі виконується командою, що знаходиться на панелі кнопкового меню інструментів “Графічний інтерфейс для управління та відображення даних” .

Для створення графічного інтерфейсу моделі (рис. 7.3) необхідно виконати наступну послідовність дій:

- створити модель і зберегти її на жорсткому диску;
- у властивостях елементів встановити параметр “Візуальний Елемент” так (ні);
- виконати команду “Графічний інтерфейс для управління та відображення даних” .
- у вікні графічного інтерфейсу виконати коригування позицій

- розташування та зображення елементів;
- закрити вікно графічного інтерфейсу з збереженням.

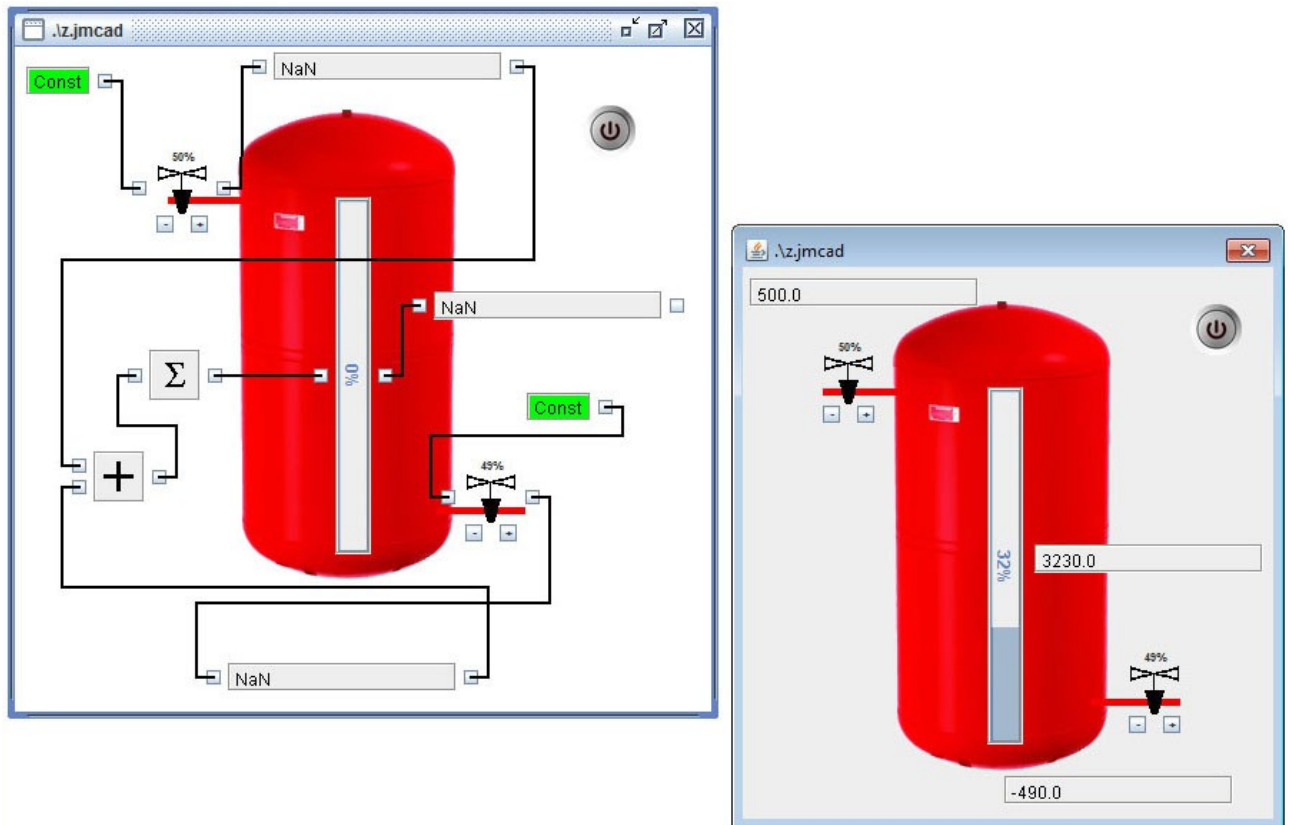


Рис. 7.3

## 7.6 Математичні розрахунки

Одне з можливих застосувань програмного комплексу це проведення математичних розрахунків. Приклад розрахунку залежності  $D=(A+B)^C$  (рис. 7.4). Дана модель може обчислювати значення, що задані як початкові, а також якщо під час роботи моделі змінювати значення констант A,B,C то відповідно буде перераховуватися значення D.

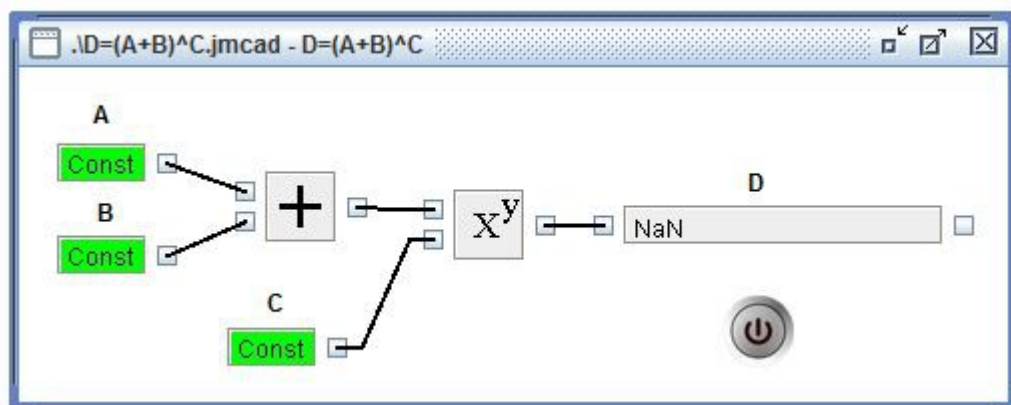



Рис. 7.4

Обчислення моделі буде проводитися постійно з певним кроком, що задається параметром “Затримка по часу” (п. 7.4). Якщо не потрібне постійне обрахування моделі і достатньо одного обрахунку можливо використати елемент зупинки моделі .