

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ЕКОНОМІКО-ГУМАНІТАРНИЙ
УНІВЕРСИТЕТ ІМЕНІ АКАДЕМІКА
СТЕПАНА ДЕМ'ЯНЧУКА**

Р.М.ЛІТНАРОВИЧ, Ю.Г.ЛОТЮК

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

НАВЧАЛЬНО-МЕТОДИЧНИЙ ПОСІБНИК

КНИГА 1



Рівне, 2010

ББК 22.193.73

Л 64

УДК 378.147.31

Літнарівич Р.М., Лотюк Ю.Г. Комп'ютерне моделювання.
Навчально-методичний посібник. Книга 1. МЕГУ, Рівне,
2010,-127 с.

Вивчення дисципліни включає лекційні,
практичні заняття під керівництвом викладача, а також
самостійну роботу за комп'ютером, що забезпечує
закріплення теоретичних знань, сприяє набуттю
практичних навичок і розвитку самостійного наукового
мислення.

Для студентів МЕГУ освітньо-кваліфікаційного
рівня «Магістр інформаційних технологій».

Litnarovich R.M., Lotyuk Yu.G. Computer design.
Educational- methodic manual. Book 1. IEGU, Rivne, 2010-
127 p.

The study of discipline includes a lecture
laboratory employments under the direction of teacher, and also
independent work at the computer, which provides fixing of
theoretical knowledges, instrumental in acquisition of practical
skills and development of independent scientific thought.

For students IEGU of educationally qualifying level
master's «Degree of information technologies».

Рецензенти: В.О.Боровий, доктор технічних наук, професор
В.Г.Бурачек, доктор технічних наук, професор
Є.С.Парняков, доктор технічних наук, професор

Відповідальний за випуск:

Й.В.Джунь, доктор фізико-математичних наук, професор

© Літнарівич Р.М., Лотюк Ю.Г., 2010 р.
ISBN 978-966-8647-16-4 ©“Тетіс“, Рівне, 2010 р.

ЗМІСТ

Передмова.....	5
1.Програма спецкурсу«Засобиомп'ютерного моделювання у вивченні складних природних явищ»	6
2. Лекційний курс.....	19
Лекція 1. Введення в курс. Загальне знайомство з системою «Model Vision Studium 3.2».....	19
...Лекція 2. Установка пакета і загальні угоди.....	21
... Лекція 3. Створення проекту Новий проект.....	22
...Лекція 4. Створюємо безперервну модель.....	25
4.1. Введення змінних, параметрів і констант	26
4.2. Введення рівнянь	28
4.2.1. Введення рівнянь. Додаткова інформація	30
4.3. Створення та запуск виконуваної моделі	32
4.4.Експерименти з візуальною моделлю	33
4.4.1. Запуск і рестарт моделі	34
4.4.2. Тимчасова та фазова діаграми	35
4.4.3. 3D-анімація	38
4.4.4. Інтерактивне втручання	40
4.4.5. 2D-анімація	43
4.4.6. Зупинка за умовою	46
4.4.7.Формульний калькулятор	48
4.4.8. План	48
Лекція 5. Створюємо гібридну модель	49
5.1. Створення карти поведінки	52
5.2. Редагування карти поведінки	53
5.2.1. Редагування вузлів	56
5.2.2. Редагування переходів	61
5.3.Візуальна модель відривається маятника	67
Лекція 6. Створюємо компонентну модель.....	70
6.1. Зовнішні перемінні	71
6.2. Структурна схема	74

6.2.1. Додавання локального блоку	74
6.2.2. Редагування локального блоку	76
6.2.3. Додавання зв'язку	78
6.2.4. Редагування зв'язку	81
6.3. Експеримент з компонентної моделлю	82
Лекція 7. Експорт та імпорт класів.....	83
7.1. Експорт класу в інший проект	83
7.2. Імпорт класу з бібліотеки класів	84
7.3. Створення власної бібліотеки класів	86
3. Практичні роботи.....	87
Практична робота 1.....	87
Практична робота 2.....	92
..Практична робота 3.....	99
...Практична робота 4.....	103
....Практична робота 5	105
....Практична робота 6.....	106
...Практична робота 7.....	108
...Практична робота 8.....	111
...Практична робота 9.....	113
Література.....	126

ПЕРЕДМОВА

Навчальний посібник призначений для швидкого освоєння основних принципів та придбання первинних навичок побудови моделей у програмному середовищі MVS (Model Vision Studium). Посібник орієнтований і на кваліфікованих, і на малопідготованих людей, у тому числі студентів-заочників та школярів, що бажають швидко та ефективно створювати наочні моделі і з їх допомогою ілюструвати рішення фізичних завдань і вивчати поведінку реальних пристроїв. Посібник може бути корисний й тим викладачам і вчителям, хто тільки недавно отримав доступ до комп'ютера й надолужує згаяне.

Мета дисципліни – поглиблення бази знань магістрантів з програмування, теорії програмування, сформованої у попередніх курсах, засвоєння студентами основних принципів програмування, набуття вмінь використання мов програмування на практиці з метою побудови комп'ютерних моделей складних природних і соціальних явищ..

Вивчення дисципліни включає лекційні, практичні заняття під керівництвом викладача, а також самостійну роботу за комп'ютером, що забезпечує закріплення теоретичних знань, сприяє набуттю практичних навичок і розвитку самостійного наукового мислення.

1. Програма спецкурсу «Засоби комп'ютерного моделювання у вивченні складних природних явищ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ЕКОНОМІКО-ГУМАНІТАРНИЙ
УНІВЕРСИТЕТ ім. акад. С.Дем'янчука

УДК 378.14

«ЗАТВЕРДЖУЮ»
Проректор з навчально-
методичної роботи

Підпис _____ Кузьменко А.П.
(підпис, прізвище, ініціали)

«10» вересня 2010 р.

РОБОЧА НАВЧАЛЬНА ПРОГРАМА
для підготовки магістрів факультету Кібернетики

..... СПЕЦКУРС:

«ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ
У ВИВЧЕННІ СКЛАДНИХ ПРИРОДНИХ ЯВИЩ»

За напрямом «ІНФОРМАТИКА»
спеціальності «ІНФОРМАТИКА»

Денна і заочна форми навчання

Рівне, 2010

Нормативні дані . Форма навчання	Курс	Семестр	Лекції	Практичні	СРС	Аудиторні	Всього	Залік (семестр)	Іспит (семестр)
Денна	2	3	18	18	72	36	108	3	
Заочна	2	3	4	6	98	10	108	-	

ОПИС ПРЕДМЕТУ КУРСУ

Курс 1	Напрямок, освітньо-Кваліфікаційний рівень	Характеристика навчального курсу
Кількість кредитів ECTS:3 Модулів:1 Змістовних модулів:1 Загальна кількість годин:108 Тижневих годин:1,6	Магістр	Спецкурс 3 семестр Лекції(теоретична Підготовка): 18/4 год Практичні:18/4 год Самостійна робота:72/98

Примітка: В знаменнику приведені години для заочного відділення

Робоча програма складена доцентом, к.т.н. Літнарівичем Р.М. на основі типової програми, розробленої у МЕГУ.

Затверджена на засіданні кафедри математичного моделювання “ 07 “ вересня 2010, протокол №1.

Завідувач кафедрою

Джунь Й.В. Підпис
доктор фізико-математичних наук, професор

1. АНОТАЦІЯ ДО КУРСУ

Робоча програма складена на основі «Нормативних програм дисциплін фундаментального циклу підготовки магістрів з інформатики» Київ: КНУ ім.Т.Г.Шевченка, 1997, 232 с., освітньо-професійної програми вищої освіти за професійним спрямуванням 0802 «Прикладна математика» циклу підготовки магістра, затвердженої рішенням Вченої Ради ПВНЗ «МЕГУ» ім.академіка Степана Дем'янчука, протокол №6 від 24.02.2006.

2. МЕТА ТА ЗАВДАННЯ ДИСЦИПЛІНИ, ЇЇ МІСЦЕ В НАВЧАЛЬНОМУ ПРОЦЕСІ

2.1. Мета дисципліни – поглиблення бази знань магістрантів з програмування, теорії програмування, сформованої у попередніх курсах, засвоєння студентами основних принципів комп'ютерного моделювання, набуття вмінь використання комп'ютерного моделювання у вивченні складних природних явищ на практиці.

2.2. Вивчення дисципліни включає лекційні,

практичні заняття під керівництвом викладача, а також самостійну роботу за комп'ютером, що забезпечує закріплення теоретичних знань, сприяє набуттю практичних навичок і розвитку самостійного наукового мислення.

2.3. Міжпредметні зв'язки та їх реалізація при викладанні дисципліни (структурно-логічна схема).

Спецкурс «Засоби комп'ютерного моделювання у вивченні складних природних явищ» є продовженням спецкурсу «Математичне моделювання та системний підхід до вивчення складних природних та соціальних явищ», і вимагає базової підготовки з дисциплін «Програмування», «Системне програмування», «Теорія програмування», аплікативні системи, комп'ютерна алгебра, сучасні проблеми прикладної математики. Знання та навички, отримані студентами при вивченні дисципліни, успішно можуть використовуватись при написанні кваліфікаційних робіт (магістерських дисертацій).

3. ФОРМУВАННЯ ПРАКТИЧНИХ НАВИЧКІВ

Студент повинен знати суть комп'ютерного моделювання в системі Model Vision Studium та напрямки його застосування, освоїти основні принципи та придбати первинні навички побудови моделей у програмному середовищі MVS (Model Vision Studium).

4. ЗМІСТ ДИСЦИПЛІНИ 3 С Е М Е С Т Р

4.1.ЛЕКЦІЇ, НАЙМЕНУВАННЯ ТЕМ ЗА ЇХ ЗМІСТОМ

№	Т Е М А	Ле Кці я	Са Мос тійна
---	---------	----------------	--------------------

1.	Лекція 1. Введення в курс. Загальне знайомство з системою «Model Vision Studium 3.2»	2/2	4/5
2.	Лекція 2. Установка пакета і загальні угоди	2	4/5
3.	Лекція 3. Створення проекту 3. Новий проект.	2/2	4/5
4.	Лекція 4. Створюємо безперервну модель 4.1. Введення змінних, параметрів і констант 4.2. Введення рівнянь 4.2.1. Введення рівнянь. Додаткова інформація 4.3. Створення та запуск виконуваної моделі 4.4. Експерименти з візуальною моделлю 4.4.1. Запуск і рестарт моделі .10 4.4.2. Тимчасова та фазова діаграми 4.4.3. 3D-анімація . 4.4.4. Інтерактивне втручання 4.4.5. 2D-анімація 4.4.6. Зупинка за умовою 4.4.7. Формульний калькулятор 4.4.8. План	6	24/23
	Лекція 5. Створюємо гібридну модель ... 5.1. Створення карти поведінки 5.2. Редагування карти поведінки 5.2.1. Редагування вузлів 5.2.2. Редагування переходів 5.3. Візуальна модель відривається	2	6/6

маятника		
Лекція 6. Створюємо компонентну модель	2	6/6
6.1. Зовнішні перемінні		
6.2. Структурна схема		
6.2.1. Додавання локального блоку		
6.2.2. Редагування локального блоку		
6.2.3. Додавання зв'язку		
6.2.4. Редагування зв'язку		
6.3. Експеримент з компонентної моделлю		
Лекція 7. Експорт та імпорт класів	2	6/6
7.1. Експорт класу в інший проект		
7.2. Імпорт класу з бібліотеки класів		
7.3. Створення власної бібліотеки класів		
Всього за 1 семестр	18/4	54/56
4.2. ПРАКТИЧНІ РОБОТИ		
Практична робота 1.....	2	2/4
Практична робота 2.....	2	2/4
..Практична робота 3.....	2	2/4
....Практична робота 4.....	2/2	2/5
....Практична робота 5	2/2	2/5
....Практична робота 6.....	2/2	2/5
...Практична робота 7.....	2	2/5
...Практична робота 8.....	2	2/5
...Практична робота 9.....	2	2/5
Здача Модуля		
Всього за 1 семестр.....	18/6	18/42

Примітка: знаком « / » позначені години для денної і заочної форм навчання.

5. ФОРМИ КОНТРОЛЮ

- 5.1.Оцінка виконання лабораторних робіт.
- 5.2. Оцінка виконання домашніх завдань.
- 5.3.Оцінка виконання самостійної роботи.
- 5.4. Оцінка засвоєння теоретичного курсу.
- 5.5.Залік за семестр.

6.ПЕРЕЛІК ПИТАНЬ ДО ЗАЛІКУ

Лекція 1. Введення в курс.

Загальне знайомство з системою «Model Vision Studium 3.2»

Лекція 2. Установка пакета і загальні угоди

Лекція 3. Створення проекту

3. Новий проект.

Лекція 4. Створюємо безперервну модель

4.1. Введення змінних, параметрів і констант

4.2. Введення рівнянь

4.2.1. Введення рівнянь. Додаткова інформація

4.3. Створення та запуск виконуваної моделі

4.4.Експерименти з візуальною моделлю

4.4.1. Запуск і рестарт моделі .10

4.4.2. Тимчасова та фазова діаграми

4.4.3. 3D-анімація .

4.4.4. Інтерактивне втручання

4.4.5. 2D-анімація

4.4.6. Зупинка за умовою 4.4.7.Формульний калькулятор

4.4.8. План

Лекція 5. Створюємо гібридну модель ...

5.1. Створення карти поведінки

- 5.2. Редагування карти поведінки
- 5.2.1. Редагування вузлів
- 5.2.2. Редагування переходів
- 5.3. Візуальна модель відривається маятника

Лекція 6. Створюємо компонентну модель

- 6.1. Зовнішні перемінні
- 6.2. Структурна схема
 - 6.2.1. Додавання локального блоку
 - 6.2.2. Редагування локального блоку
 - 6.2.3. Додавання зв'язку
 - 6.2.4. Редагування зв'язку
- 6.3. Експеримент з компонентної моделлю

Лекція 7. Експорт та імпорт класів

- 7.1. Експорт класу в інший проект
- 7.2. Імпорт класу з бібліотеки класів
- 7.3. Створення власної бібліотеки класів

7. ПИТАННЯ ГАРАНТОВАНОГО РІВНЯ ЗНАНЬ

1. Знайомство з середовищем MVS.

1.1. Основні інструменти MVS.

Головне вікно проекту: Редактор моделі - середовище реалізації алгоритму моделі.

Головне вікно візуальної моделі: Випробувальний стенд - графічна середовище створення і функціонування візуальної моделі.

1.2. Створення написи у вікні 2D-анімації.

1.2.1. Створення папки для збереження проекту.

1.2.2. Створення написи в полі вікна 2D-анімації випробувального стенду.

1.3. Збереження проекту і моделі. Відкриття існуючого проекту.

1.3.1. Збереження проекту.

- 1.3.2. Компіляція і збереження моделі.
- 1.3.3. Відкриття існуючого проекту.
- 1.4. Завдання для самостійного виконання.
- 2. Створення найпростіших моделей.

- 2.1. Фонарик_1. Статична модель.
 - 2.1.1. Математичне формулювання задачі.
 - 2.1.2. Побудова алгоритму роботи моделі.
 - 2.1.3. Реалізація алгоритму в середовищі MVS.
 - 2.1.4. Побудова моделі, запуск і робота з нею.
 - 2.1.5. Збереження проекту.

- 2.2. Фонарик_2. Карта поведінки.
 - 2.2.1. Поняття про роботу моделі, керованої подіями.
 - 2.2.2. Карта поведінки у програмі MVS.
 - 2.2.3. Створення проекту Фонарик_2.
Математичне формулювання задачі.
Побудова алгоритму.

Введення змінних.

- Складання карти поведінки моделі.
Початковий вузол Init. Вхідні дії вузла Init. Додавання нового вузла. Додавання переходів і умов їх спрацювання.
- 2.2.4. Створення візуальної моделі ліхтарика.
- 2.2.5. Збереження проекту.

Висновок.

- 2.3. Фонарик_3. Побудова динамічної моделі.
 - 2.3.1. Математичне формулювання задачі.
 - 2.3.2. Побудова алгоритму.
 - 2.3.3. Перетворення проекту Фонарик_2 в проект Фонарик_3.

Збереження проекту під новою назвою. Перейменування об'єктів проекту та моделі.

Введення змінної і параметра.

Запис рівнянь вузлів.

Зміна вхідних дій вузлів.

Доопрацювання моделі ліхтарика.

Перевірка працездатності моделі. Перший запуск моделі.
Дослідження моделі. Вікно діаграми (осцилографа).

Корекція алгоритму.

Корекція вхідних дій вузлів і дій переходів.

Тестування моделі.

2.4.Опис моделі ліхтарика з використанням диференціального рівняння.

2.4.1. Математичне формулювання задачі.

2.4.2. Реалізація алгоритму.

2.4.3. Перетворення проекту Фонарик_3 в проект Фонарик_31.

2.4.4. Коригування рівняння для вузла Фонарик_світить.

2.4.5.Тестування динамічної моделі,описуваної диференціальним рівнянням.

2.4.6. Завдання для самостійної роботи.

Висновок.

3. Проста анімація.

3.1. Створення проекту моделі.

3.2. Побудова 3D-анімації.

3.2.1. Висновок вікна нової 3D-анімації.

3.2.2. Побудова фігури ліхтарика з примітивів.

3.2.3. Скріплення змінних з властивостями примітивів у вікні 3D-анімації.

3.3. Тестування моделі.

7.НАУКОВО-ДОСЛІДНА РОБОТА СТУДЕНТІВ

Всі студенти виконують комплексну науково-дослідну роботу на тему: «Побудова і дослідження математичної моделі пункту спостереження GPS методом статистичних випробувань Монте Карло.»

8. ЛІТЕРАТУРНІ ДЖЕРЕЛА

1. <http://www.exponenta.ru/soft/others/mvs/mvs.asp> ,

а в упакованном виде:

2.

http://www.exponenta.ru/soft/others/mvs/archive/MVS_R_FREE_3.1.5.zip

3. Кострікин А.І. Введення в алгебру. - М., Наука, 1977. - 495 с.

4. Кук Д.,Бейз Д. Компьютерная математика: Учебник.пер. с англ.-М.: Наука, 1990.-384 с.

5. Компьютерная алгебра: Символьные и алгебраические вычисления: Учебник.Перев. с англ./Под ред..Б.Бухбергера и др..-М.:Мир, 1986.-392 с.,илл.

6. Курош а.Г. Теория групп. - М., Наука, 1967. - 648 с.

7. Літнарлович Р.М. Ілюстрований самовчитель по MAPLE. Ч.15. Пакети лінійної алгебри і функціональних систем. МEGУ, Рівне, 2009,- 46 с.

8. Половко А.М.,Бутусов П.Н. MATLAB для студента:учебник СПб.:БХВ-Петербург, 2005.-320 с.

9. Якимчук А.Й. Побудова і дослідження математичної моделі пункту GPS спостережень методом статистичних випробувань Монте Карло. Множинний регресійний аналіз . Модель ДА – 50. МEGУ, Рівне, 2010, -112 с.

Додаткова література

1. Літнарлович Р.М. Конструювання і дослідження математичних моделей. Онтодидактика поліноміальної апроксимації. Частина 3. МEGУ, Рівне, 2009,-32 с.

9.РОЗПОДІЛ БАЛІВ ЗА ОДИН ЗМІСТОВИЙ МОДУЛЬ, ПРИСВОЮВАНИХ СТУДЕНТАМ

Аудиторна робота				Самостійна робота				Індивідуальна робота				Підсумкова атестація	Сума
												Здача заліку, екзамену, підсумкової ККР	До 100 балів. За рахунок індивідуальної роботи можливо і більше 100

										тощо	балів
Не більше 20 балів	Не більше 20 балів	До 20 більше 20 балів	та	Не більше 40 балів							

Примітка: Кількість балів з навчальної дисципліни визначається як середнє арифметичне за кожен модуль дисципліни.

10.Шкала оцінювання:

90-100 балів – відмінно (A);

75-89 балів – добре (BC);

60-74 балів – задовільно (DE);

35-59 балів – незадовільно з можливістю повторного складання (FX);

1-34 балів – незадовільно з обов'язковим курсом (F).

11.Зміни та доповнення ,внесені в робочу програму на 201__ рік

Затвержені на засіданні кафедри _Математичного моделювання

Протокол № ___ від _____ 201_ р.

Зав. кафедрою _____

Внесені зміни та доповнення затверджую:

Проректор з навчальної роботи

«___» _____ 201_ р
на 201___ - 201___ навчальний рік

Затвержені на засіданні кафедри _Математичного моделювання

Протокол № ___ від _____ 200 р.

Зав. кафедрою _____

Внесені зміни та доповнення затверджую:

Проректор з навчальної роботи

«___» _____ 201_ р
на 201___ - 201___ навчальний рік

12. Оцінка навчальної діяльності студента

Шкала ECTS	МОН України	100 - бальна	МОН України
A	Відмінно	90...100 балів	зараховано
B	Добре	80...89 балів	зараховано
C	Добре	75...79 балів	зараховано
D	Задовільно	60...75 балів	зараховано
E	Задовільно	50...59 балів	не зараховано
FX	Незадовільно	Менше 50 балів	не зараховано
F	Незадовільно	Менше 30 балів	не зараховано

..... ЛЕКЦІЙНИЙ КУРС

..... Лекція 1. Введення в курс.

Загальне знайомство з системою «Model Vision Studium 3.2»

Метою даного Навчального посібника є навчання основним прийомам роботи з пакетом ModelVision 3.2 на ряді типових прикладів, описаних у документі «Моделювання в прикладах». Більш детально можливості пакета описані в довідковій системі.

Навчальний посібник призначений для швидкого освоєння основних принципів та придбання первинних навичок побудови моделей у програмному середовищі MVS (Model Vision Studium). Посібник орієнтований і на кваліфікованих, і на малопідготованих людей, у тому числі студентів-заочників та школярів, що бажають швидко та ефективно створювати наочні моделі і з їх допомогою ілюструвати рішення фізичних завдань і вивчати поведінку реальних пристроїв. Посібник може бути корисний й тим викладачам і вчителям, хто тільки недавно отримав доступ до комп'ютера й надолужує згаяне.

Model Vision Studium (MVS) - комп'ютерна програма, призначена для створення і дослідження наочних візуальних моделей простих і складних фізичних і технічних об'єктів і систем. Програма MVS розрахована на науковців та інженерів, але досить легко освоюється студентами та школярами, і може бути застосована як для моделювання складних динамічних об'єктів і систем, так і для проведення школярами і студентами віртуальних експериментів з фізики. .

Зазначимо, що моделі можуть бути створені і в інших програмах, у тому числі і в програмах-компіляторах мов високого рівня Borland C ++, Visual C ++, Pascal та інших. Відмінною особливістю MVS є те, що ця програма

виконує за людину велику й копітку роботу по запису кодів програм, зберігаючи можливість застосування потужного і гнучкого математичного апарату для побудови моделей. При створенні простих моделей велику частину роботи можна виконати мишкою, майже не торкаючись клавіатури.

З методичної точки зору програма MVS корисна тим, що дозволяє природним чином, невимушено ввести непідготовленого школяра в коло питань об'єктно-орієнтованого програмування (ООП) та моделювання (ООМ), звільняючи його від необхідності займатися дрібними, хоча і важливими, технічними деталями.

.....Грунтовний виклад питань встановлення програми MVS на комп'ютер і її застосування дано в поданій до програми документації. Навчальний посібник призначений для прискороного знайомства з основними особливостями і прийомами побудови моделей в середовищі MVS і підготовки користувачів до самостійної роботи, а також до вивчення документації і прикладів, супроводжуючих програму.

Програма MVS безперервно вдосконалюється. Наведений посібник написаний для версії MVS Free 3.1.5, призначеної для вільного використання. Нові версії цієї програми, завжди сумісні з усіма попередніми можна знайти на сайті:

<http://www.exponenta.ru/soft/others/mvs/mvs.asp>,

а в упакованому вигляді:

http://www.exponenta.ru/soft/others/mvs/archive/MVS_R_FREE_3.1.5.zip.

Автори висловлюють подяку розробникам програми MVS Колесову Ю.Б. і Сеніченкову Ю.Б. за обговорення методики побудови моделей та корисні поради, що дозволили поліпшити виклад цієї настанови початківця користувача MVS, Федосову Б.Т. з Рудненського індустріального інституту, Казахстан.

Лекція 2. Установка пакета і загальні угоди.

Для установки пакета запустіть програму setup.exe і вкажіть місце на жорсткому диску для розміщення пакета. Версія 3.2.x займає близько 25 Мбайт на жорсткому диску. Для видалення пакета увійдіть в «Панель управління / Установка і видалення програм», знайдіть «Model Vision Studium 3.2» і видаліть його.

Інтегрована оболонка пакету є додатком з багатовіконним інтерфейсом (MDI). Багатовіконний інтерфейс припускає наявність головного вікна програми і довільного числа дочірніх вікон (див. Рис 3.1). Тема головного вікна містить найменування пакету і шлях до відкритого в даний момент проектом. Проект - це сукупність даних, що відносяться до однієї моделі. Дані проекту зберігаються в декількох файлах, розташованих в папці даного проекту. Основний файл проекту (база даних проекту) має розширення *. mvb.

Наявність декількох дочірніх вікон дозволяє одночасно працювати з декількома об'єктами проекту або з різними аспектами одного і того ж об'єкта. Головне вікно містить меню (дочірні вікна не можуть мати меню) і панель кнопок (інструментальну панель). З усіх дочірніх вікон лише одне в кожен момент часу має фокус вводу, заголовок цього вікна виділений кольором (див. Рис 3.1). Щоб перенести фокус введення в це вікно, потрібно натиснути ліву клавішу миші на його зображенні.

Дочірні вікна інтегрованої оболонки за типом візуалізації відносяться до двох типів: вікна з деревами об'єктів (вікно проекту, вікно класу, вікно системи рівнянь) і вікна редакторів схем (вікно структури, вікно карти

поведінки). У дочірньому вікні можна виділити один або декілька об'єктів. На дереві об'єктів виділений об'єкт позначається кольором, а на схемі - спеціальними квадратами по контуру цього об'єкта (див. Рис 3.1). Операції, зіставлені командам головного меню або відповідним кнопок на панелі кнопок, відносяться або до проекту в цілому, або до виділеного об'єкту у вікні, що має фокус вводу.

По натисненню на зображенні об'єкта правої клавіші миші з'являється спливаюче меню, яке містить команди, які застосовуються до даного об'єкта.

Значна частина дій в пакеті виконується методом «drag and drop» («перенеси і кинь»), коли необхідний об'єкт виділяється натисканням лівої клавіші миші, а потім переміщається за допомогою миші з притиснутою лівою клавішею і поміщається в потрібну область того ж або іншого дочірнього вікна після відпускання клавіші миші. Під час переміщення курсор миші індикуює області, в які можна перемістити цей об'єкт.

Лекція 3. Створення проекту

3. Новий проект.

Входимо в інтегровану оболонку пакету і натискаємо



кнопку або виконуємо команду головного меню «Проект \ Новий ...». У діалозі вибираємо шлях до папки проекту (наприклад, "C: \ Мої_моделі \"), вводимо ім'я проекту (наприклад, «Маятник») і натискаємо кнопку «Створити». Після цього з'явиться папка "C: \ Мої_моделі \ Маятник" і в ній файл бази даних проекту «Маятник.mvb».

Зауваження. Передбачається, що для нового проекту відкривається нова папка (якщо використовується існуюча, то всі дані в ній знищуються). У той же час за допомогою команди «Проект / Зберегти як» ви можете зберігати в тій же папці різні модифікації даного проекту (наприклад, Маятник1.mvb).

Зауваження. Надалі в цій папці з'являться ще два файли - «Маятник.ini» і «Маятник_em.ini» - в яких зберігаються відповідно установки проекту і виконуваної моделі.

Після цього в інтегрованому середовищі з'являться такі вікна (див. Рис 3.1):

1) вікно управління проектом (зліва вгорі) містить дерево основних складових проекту. До складових проекту відносяться: класи блоків, глобальні (тобто видимі у всіх складових проекту) константи, глобальні процедури і функції, віртуальний стенд і імпортовані бібліотеки класів. За допомогою спливаючого по правій кнопці миші меню можна додавати нові компоненти, видаляти і редагувати існуючі. У новому проекті за замовчуванням доданий клас безперервного блоку «Маятник» (ім'я класу збігається з ім'ям проекту).

2) вікно віртуального стенду (у центрі вгорі) містить структурну схему системи, що моделюється, тобто екземпляри блоків і зв'язку між ними. За замовчуванням у віртуальний стенд поміщений екземпляр класу «Маятник» з ім'ям «Маятник_1» (наступний екземпляр цього класу в цій схемі отримає за замовчуванням ім'я «Маятник_2» і т.д.). Управляти структурними компонентами можна з допомогою спливаючого по правій клавіші миші меню. Щоб додати нового блоку досить схопити мишкою у вікні управління проектів потрібний клас, буксирувати його у вікно структури і кинути в потрібному місці.

Зауваження. Розглянута модель маятника є ізольованою системою і в загальному немає особливої необхідності в

структурній схемі і блоках. Проте дана версія пакету орієнтована на блокове моделювання і тому ізольована система подається як блок без входів і виходів, а її модель у вигляді віртуального стенду з примірником такого блоку.

3) вікно редагування доданого за замовчуванням класу (в даному випадку класу «Маятник») містить дерево складових опису класу. Оскільки даний блок передбачається безперервним, то за замовчуванням в нього додана порожня система рівнянь з ім'ям «Система_уравнень_1».

Зауваження. У даній версії пакету чисто безперервна система розглядається як вироджений випадок гібридного автомата, головна карта поведінки якого містить тільки один вузол, якому і приписали відповідна система рівнянь. Тому якщо ви відкриєте головну карту поведінки такого класу (з допомогою подвійного клацання миші на відповідному вузлі дерева), ви побачите, що вона складається з єдиного вузла Init (успадкованого від загального предка всіх блоків класу CDevice), якому і приписали «Система_рівнянь_1».

4) вікно редагування системи рівнянь, відкрите на порожній «Система_рівнянь_1».

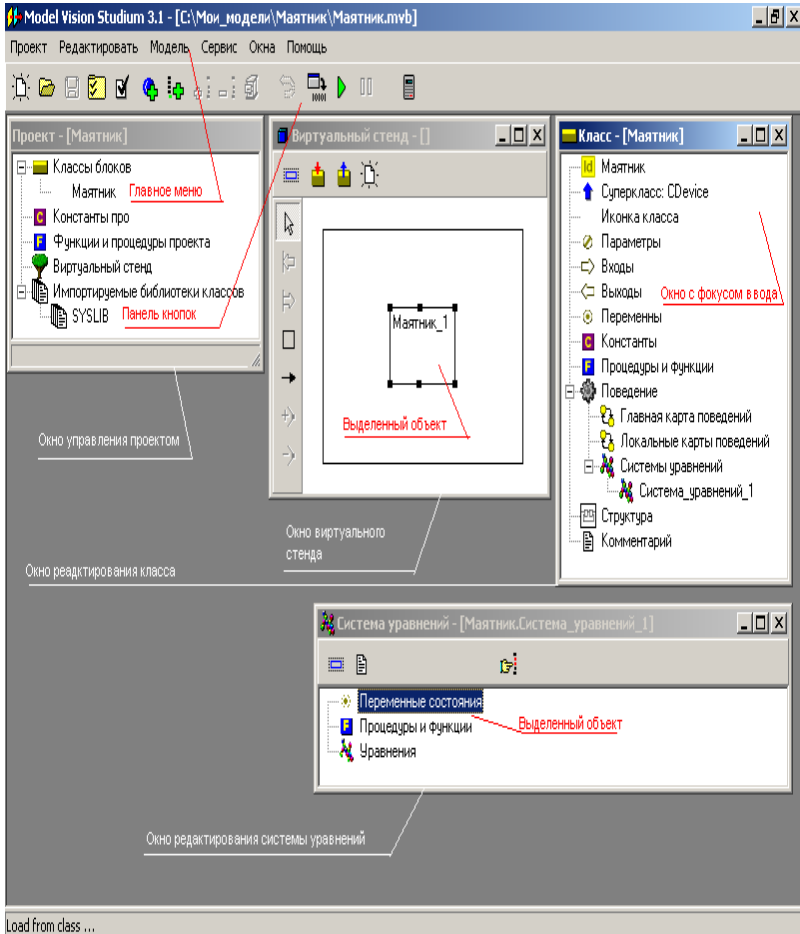


Рис 3.1

Лекція 4.

4. Створюємо безперервну модель.

Розглянемо модель простого математичного маятника (матеріальна точка, підвішена на невагомому нерозтяжній стрижні). Це модель чисто безперервної ізольованої системи. Тому для неї в якості основи чудово підходить

клас, що створюється за замовчуванням при відкритті нового проекту. У нього тепер необхідно додати необхідні змінні і рівняння. Даний проект ми назвемо «Маятник.mvb».

Шлях до готового прикладом «Моделі \ Маятник \ Маятник.mvb».

4.1. Введення змінних, параметрів і констант.

Переходимо до вікна редагування класу «Маятник», виділяємо в дереві об'єктів вузол «Параметри», натисканням правої клавіші миші викликаємо спливаюче меню і виконуємо команду «Додати». Те ж саме можна зробити за допомогою команди «Редагувати / Додати»

головного меню або кнопки  на панелі кнопок.

У діалозі вводимо ідентифікатор змінної α , залишаємо заданий за замовчуванням тип `double`, задаємо початкове значення $-\pi/2$ і коментар (див. Рис 4.1).

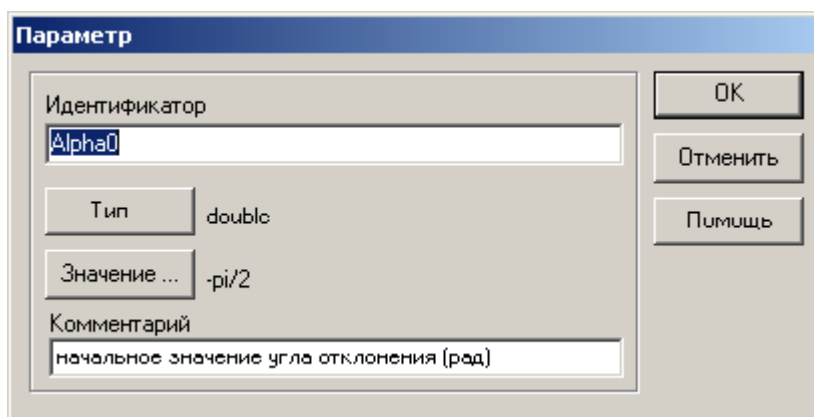


Рис 4.1

Аналогічним чином додаємо параметр L . Далі виділяємо вузол «Внутрішні змінні» і додаємо змінні α , ω , X , Y , а потім виділяємо вузол «Константи» і додаємо константу g (див. Рис 4.2).

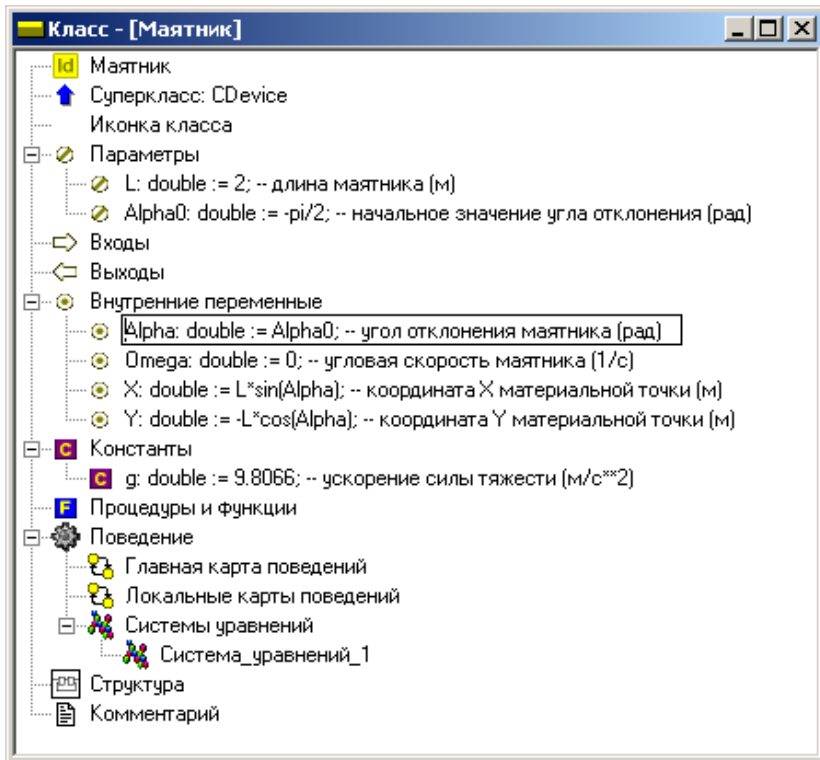



Рис 4.2

Ви можете змінити введені визначення двічі клацнувши на них мишею, а також за допомогою команди «Редагувати» спливаючого меню, команди «Редагувати / Змінити» головного меню або кнопки  на панелі кнопок. Крім того, ви можете редагувати вибраного визначення як текст за допомогою команди «Редагувати / Редагувати як текст (F2)» головного меню або одноразового клацання миші. На Рис 4.2 в режимі текстового редагування знаходиться визначення змінної Alpha. Завершення редагування відбувається по клавіші «Enter» або клацання миші за межами поточного рядка.

Скасування редагування відбувається після натискання клавіші «Esc».

Ви можете видалити введені визначення за допомогою команди «Видалити» спливаючого меню, команди «Редагувати / Видалити» головного меню або кнопки



на панелі кнопок.

4.2. Введення рівнянь.

У вікні редагування рівнянь за допомогою подвійного клацання миші на вузлі «Рівняння» або команди «Змінити» спливаючого меню викликаємо спеціальний редактор формул, який дозволяє вводити математичні вирази в близькому до природного вигляді, і вводимо необхідні рівняння (Рис 4.3).

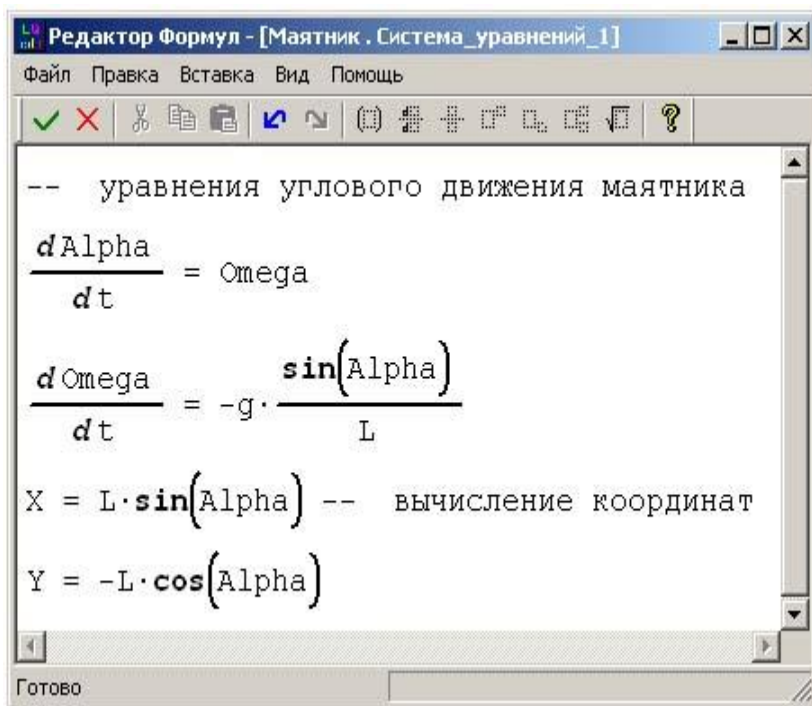


Рис 4.3

Система рівнянь може включати в себе звичайні диференціальні рівняння першого порядку і алгебраїчні рівняння. Форма введення рівнянь в даній версії пакету досить вільна. Наприклад, останні два рівняння можна представити у вигляді

$$\begin{aligned}XL * \sin(\text{Alpha}) &= 0 \\ Y + L * \cos(\text{Alpha}) &= 0 \\ \text{find } X, &\end{aligned}$$

Слід, однак, пам'ятати, що така форма рівнянь зажадає від пакету значно більше зусиль, ніж подання на Рис 4.3. Аналітично дозволені рівняння (формули) за відсутності в них циклів зводяться просто до послідовності операторів присвоювання.

Ви можете також безпосередньо використовувати в рівняннях першу і другу похідні, не вводячи додаткових змінних. Наприклад, рівняння для маятника з в'язким опором можуть бути записані так (Рис 4.4).

$$\frac{d^2 \text{Alpha}}{dt^2} = -g \cdot \frac{\sin(\text{Alpha})}{L} - \mu \cdot \frac{d \text{Alpha}}{dt}$$

$$X = L \cdot \sin(\text{Alpha})$$

$$Y = -L \cdot \cos(\text{Alpha})$$

$$\text{find } \text{Alpha}, X, Y$$

Рис 4.4

Якщо ви не вказали, які змінні є шуканими (оператор find), то транслятор спробує визначити їх сам, виходячи з контексту завдання: шуканими не можуть бути параметри,

константи і входи. Транслятор також прагне зробити шуканими (якщо це можливо) інтегруються змінні (Alpha) та змінні, що стоять в лівих частинах формул (X, Y). Якщо ж транслятору я не можу вибрати шукані змінні, то він викликає спеціальний діалог і пропонує зробити це вручну.

Крім власне рівнянь і формул в систему рівнянь можуть бути також включені необхідні коментарі. Коментар починається символом «-» (подвійний мінус) і триває до кінця рядка (Рис 4.3).

4.2.1. Введення рівнянь. Додаткова інформація.

Розглянемо нескладну додаткову модель: одномірне рух матеріальної точки з масою m під дією допустимої сили F і в'язкого тертя з коефіцієнтом μ (Шлях до готового прикладом «Моделі \ Рух \ Двіженіє.mvb»).

Рівняння руху може бути записано в природному «фізичній» формі

$$m \cdot \frac{d^2 x}{dt^2} = F - \mu v \cdot \frac{dx}{dt}$$

Рис 4.5

Примусова сила подається на вхід з генератора синусоїдального сигналу (Рис 4.6), тому шуканої змінної однозначно є змінна x .

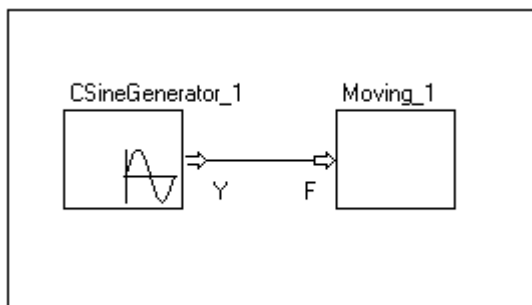


Рис 4.6

Транслятор сам перетворює це рівняння до «канонічної» форми, необхідної для чисельного рішення. Ви можете подивитися на це подання за допомогою команди «Показати рівняння в канонічній формі» спливаючого меню. Для даного рівняння це буде

$$x: d(x) / dt = x';$$

$$x': d(x') / dt = x'';$$

$$x'': m * x'' = F - \mu * x';$$

(Перед двокрапкою показана змінна, яка визначається з цього рівняння).

У даній версії пакету похідні видимі тільки в межах даної системи рівнянь (це означає, що не можна, наприклад, використовувати швидкість у діях переходу або в тілі функції).

Тепер, не міняючи рівняння руху, зробимо вхідної змінної x , а не F (Рис 4.7).

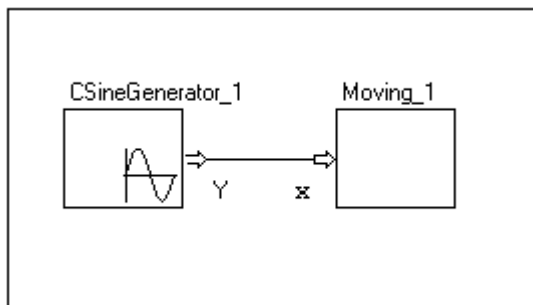


Рис 4.7

Тепер шуканої змінної є сила F (тобто, ми вирішуємо зворотну задачу: яку силу потрібно прикладати, щоб тіло рухалося по синусоїді). Однак, для цього завдання ми будемо мати зовсім іншу «канонічну» систему рівнянь:



$$x': x' = d(x) / dt;$$

$$x'': x'' = d(x') / dt;$$

$F: m * x'' = F - \mu * x'$;

Для визначення сили необхідно подвійне диференціювання змінної x . Бо у цій версії пакету аналітичні перетворення не підтримуються, зробити це можна тільки за допомогою чисельного диференціювання. Це дуже погано для більшості чисельних методів і тому транслятор попереджає користувача.

4.3. Створення та запуск виконуваної моделі.


Створення моделі проводиться за допомогою команди «Модель / Створити» головного меню або кнопки  на панелі кнопок. Запуск (і створення, якщо потрібно) моделі проводиться за допомогою команди «Модель / Пуск» головного меню або кнопки  на панелі кнопок.

Пакет MVS відноситься до категорії компілюються: для елементів опису модельованої системи генерується "код" на проміжному мовою програмування, який потім компілюється в машинний код і зв'язується з Run Time Library (RTL) MVS за допомогою штатного компілятора командного рядка цієї мови. У даній версії пакету в якості проміжного мови використовується Borland Object Pascal (Delphi 3).


Пакет MVS може створювати моделі двох видів: візуальну у формі додатку Win32 і «приховану» у формі Win32 DLL. Далі розглядається тільки робота з візуальними моделями. Питання використання «прихованих» моделей див. у відповідному розділі.

При створенні моделі ви побачите спочатку вікно генератора коду, а потім одне за іншим два вікна консольних додатків - компілятора ресурсів (необхідний для приєднання іконки класу) і компілятора Object Pascal Delphi. Вся робота з проміжним кодом проводиться в локальній папці ... \Tmp (у даному прикладі в C: \

Мої_моделі \ Маятник1 \ Temp), в ній же формується файл model.exe. Зазвичай це займає кілька секунд. При виході з інтегрованого середовища весь вміст цієї папки автоматично знищується. Тому, якщо ви хочете залишити проміжний код для незалежного використання (наприклад, щоб використовувати діалоговий відладчик Delphi),

скористайтеся кнопкою  або командою «Модель / Створити». Для того, щоб зберегти модель як незалежну від оболонки виконувану програму, скористайтеся командою «Модель / Зберегти як» головного меню.

Зауваження. Якщо папка проекту недоступна для запису (розташована на CD або закрита на запис для даного користувача), то всі тимчасові файли і файл виконуваної моделі формуються в робочій папці Windows (наприклад, C: \ Temp).

Виконання команди «Пуск» завершується запуском створеної програми model.exe. У будь-який момент ви можете перервати виконання моделі з інтегрованого середовища (наприклад, у випадку її зависання) за допомогою кнопки  або команди «Модель / Стоп».

У виконувану модель включається тільки вміст вікна «Віртуальний стенд». Перед генерацією коду проводиться комплексний контроль його правильності і в разі виявлення помилки створення виконуваної моделі скасовується.




4.4. Експерименти з візуальною моделлю.

На Рис 4.8 показано головне вікно візуальної моделі після першого запуску. Візуальна модель також як і інтегрована оболонка є багатовіконний додатком. У заголовку головного вікна вказується ім'я виконуваної моделі, яке ви можете змінити у вікні віртуального стенду інтегрованої

оболонки (за умовчанням це ім'я проекту). У лівій частині інструментальної панелі відображається поточне значення модельного часу (початкове значення 0). У нижній частині головного вікна виводиться інформація про поточні типи сукупної системи рівнянь і загальному числі рівнянь (у випадку системи з перемикальними вони можуть змінюватися в ході прогону моделі).

У лівому верхньому куті розташоване вікно віртуального стенду, яке відображає структуру моделі і є «провідником» по ній. Для всіх блоків, присутніх в моделі (в даному випадку є один блок Маятник_1), автоматично відкриваються вікна змінних. У заголовку дочірніх вікон вказується блок або елемент блоку, який відображається в цьому вікні. Зверніть увагу на те, що після створення екземпляра цього пристрою його параметри взяли вказані значення і фазові змінні ініціалізовані зазначеними виразами.

4.4.1. Запуск і рестарт моделі.

Запустимо виконання моделі за допомогою кнопки  (на інструментальній панелі головного вікна моделі) або за допомогою команди «Моделювання / Пуск» головного меню. Ви побачите, що почне змінюватися модельне час (в темпі реального, якщо у вашого комп'ютера достатня продуктивність) і значення фазових змінних. Однак, спостереження за цифровими значеннями мало що дає (тільки констатацію, що в моделі взагалі щось обчислюється). Тому зупинимо виконання за допомогою кнопки  або команди «Моделювання / Стоп» і потім закінчимо цей прогін за допомогою кнопки  або команди «Моделювання / Рестарт». У результаті цих дій даний екземпляр випробуваної системи буде знищений і

створений новий, знову з початковими значеннями змінних. Модельне час знову буде дорівнює 0.

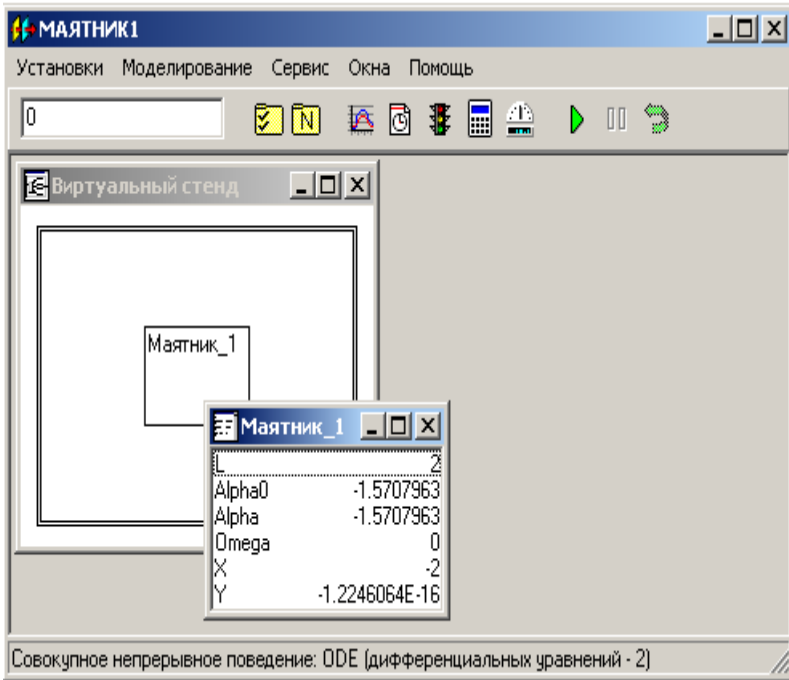




Рис 4.8

4.4.2. Тимчасова та фазова діаграми.

За допомогою кнопки  або команди «Вікна / Нова діаграма» створимо вікно діаграми (за умовчанням це буде тимчасова діаграма, тобто по осі абсцис будуть відкладатися значення модельного часу). Методом "drag-and-drop" перенесемо у вікно діаграми з вікна змінних змінні α і ω . Запустимо модель і отримаємо наступний графік (Рис 4.9).

Зауваження. Може виявитися, що на вашому комп'ютері ці нескладні рівняння вирішуються так швидко, що ви просто не встигаєте нічого помітити. За допомогою кнопки  або команди «Установки / Модель» викличте діалог редагування установок. На станиці «Виконання»

перемкніть параметр «Співвідношення модельного та реального часу» з положення «так швидко як можна» на позицію «число» (за замовчуванням це 1, тобто моделювання в реальному часі). Змінюючи це число, ви можете прискорювати або сповільнювати прогін моделі.

Нам тепер хотілося б побачити на графіку залежність, тобто отримати фазову діаграму. Для цього створимо нову діаграму, перетягнемо в неї ті ж самі змінні, а потім правою кнопкою миші відкриємо на ній спливаюче меню і виконаємо команду «Налаштування». У діалозі налаштувань вкажемо за допомогою подвійного клацання мишею в полі, що по осі абсцис відкладаються значення змінної Alpha (див. Рис 4.10). Запустивши модель знову, ми отримаємо наступний графік (Рис 4.12).

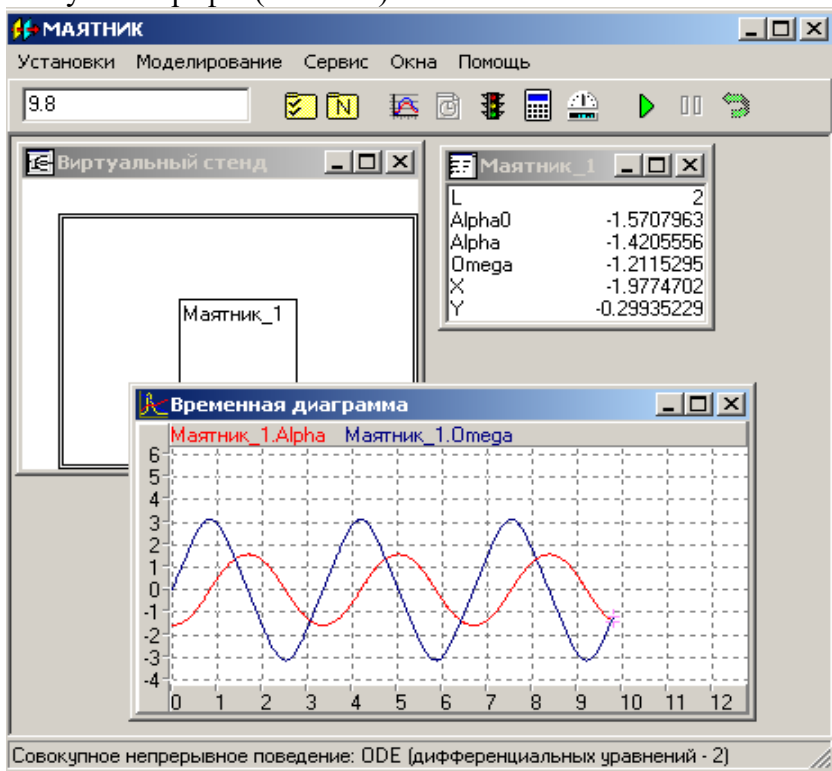


Рис 4.9

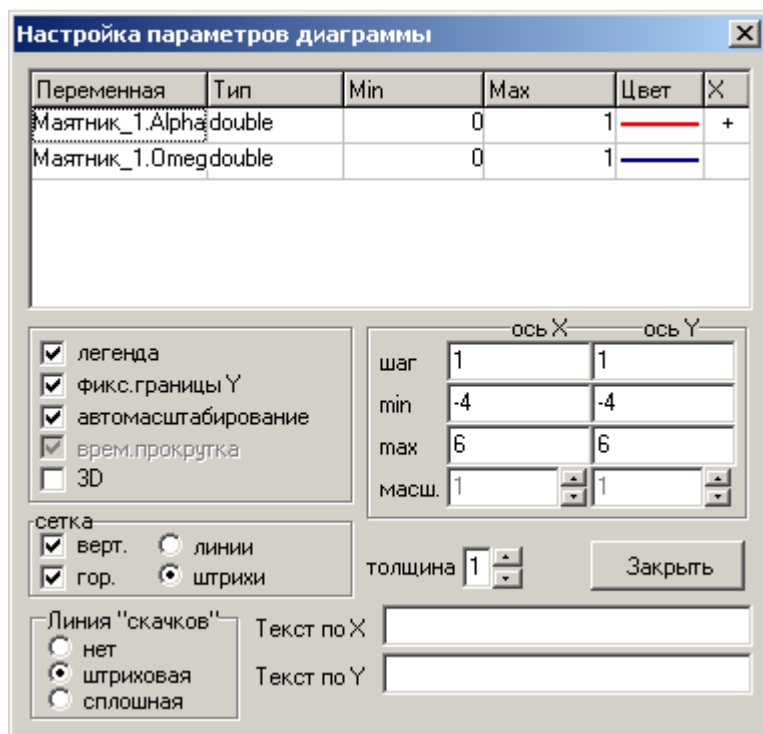
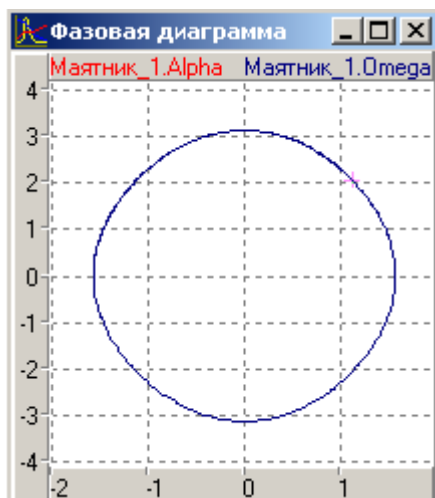


Рис 4.10

Рис.4.11



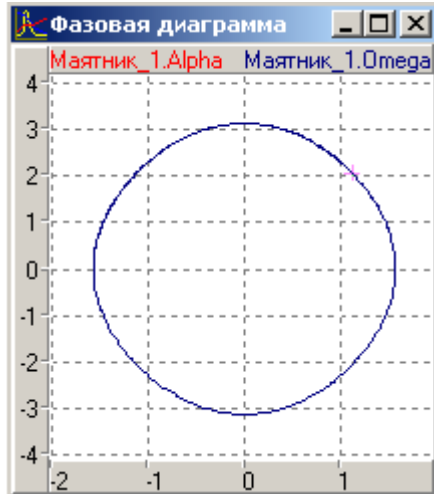


Рис 4.12

4.4.3. 3D-анімація.

Для багатьох моделей (особливо моделей механічних систем) користувач може отримати найбільше інформації з безпосереднього спостереження тривимірного динамічного зображення модельованої системи. У візуальній моделі для цього призначено вікно 3D-анімації. Створити його можна за допомогою команди головного меню «Вікна / Нова 3D-анімація».

Вікно 3D-анімації дозволяє будувати динамічні тривимірні зображення, у вигляді сукупності тривимірних примітивів (лінія, куля, циліндр, конус і т.д.), деякі параметри яких (координата, радіус, колір і т.п.) Однозначно пов'язані зі значенням відповідних змінних моделі ..

За допомогою подвійного клацання миші в області вікна або команди «Властивості» спливаючого меню виклинемо діалог редагування властивостей 3D-анімації. У даній моделі нам знадобиться лише два стандартні об'єкта - відрізок і сфера (див.Рис 4.14) .. Один кінець лінії повинен завжди знаходитися на початку координат, а координати другого кінця (параметри x_2 , y_2) повинні змінюватися

відповідно до значення змінних X і Y маятника. Для завдання цієї відповідності тягнемо методом «drag-and-drop» відповідні змінні з вікна змінних і кидаємо їх у колонці «Змінна» відповідних параметрів відрізка (Рис 4.14). Аналогічним чином цим же змінним X, Y ми співставляємо координати центра сфери (параметри x1, y1).

Після цього достатньо запустити модель і ви побачите хитний маятник (Рис 4.16). У будь-який момент ви можете змінити точку спостереження, натиснувши ліву клавiшу миші і переміщуючи її з притиснутою клавiшею. Таким чином, ви можете розглядати коливання маятника зверху, знизу і т.д.

За допомогою натиснення лівої клавiшу миші при натиснутій клавiші <Control> Ви можете видаляти точку спостереження від початку системи координат, утримуйте <Shift> наближати її до початку координат.

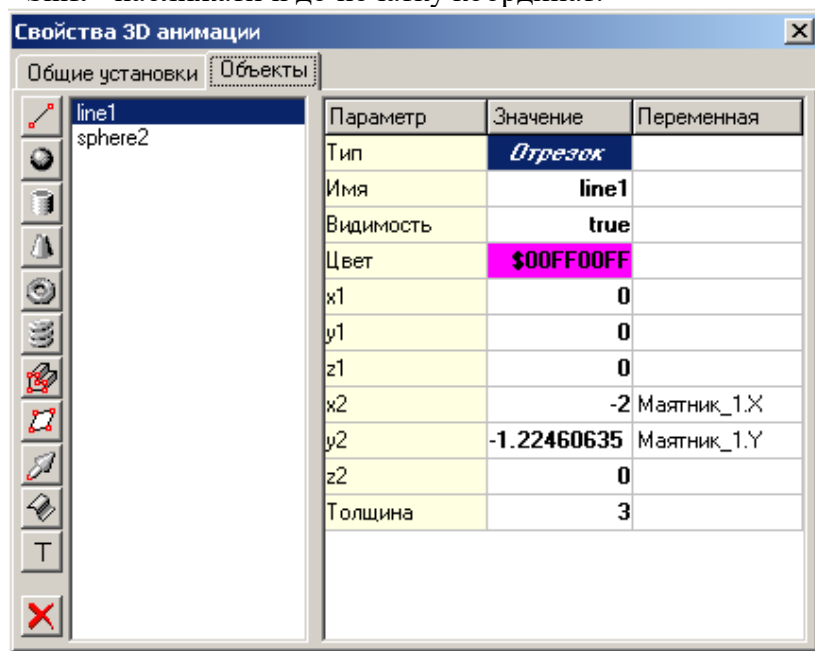


Рис 4.14

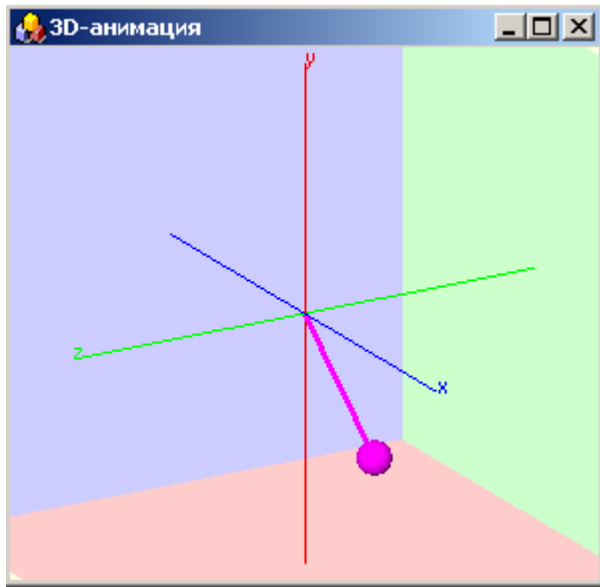


Рис 4.16

4.4.4. Інтерактивне втручання.

Концепція віртуального стенду передбачає можливість активного втручання користувача у функціонування моделі. Зокрема, користувач може змінювати значення змінних під час прогону моделі.

Наприклад, нехай ви хочете дослідити поведінку вашого маятника при зміні прискорення сили тяжіння g . Для цього потрібно, по-перше, трансформувати g з константи в змінну.

Зауваження. Константа - це величина, яка ніколи не змінюється. Параметр - це величина, яка може змінюватися для різних екземплярів класу і для даного екземпляра тільки одноразово на початку прогону. Змінна може змінюватися під час прогону моделі.

Таку трансформацію дуже легко виконати методом «drag-and-drop», перетягнувши g з секції констант в секцію змінних у визначенні класу (Рис 4.18).

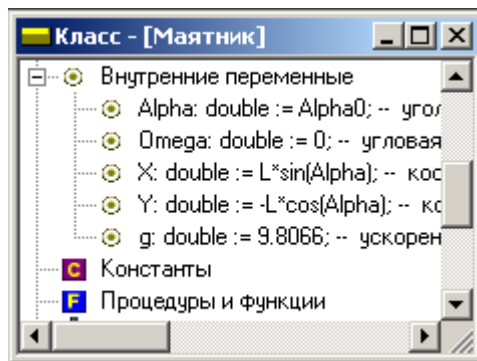


Рис 4.18

Нехай ми бажасмо, наприклад, в кінці першого періоду коливань зробити значення g рівним 50. Клацніть двічі мишею на рядку цієї змінної у вікні змінних) модель можна і не зупиняти) і в діалозі, що з'явився (Рис 4.19) задайте нове значення (можна вводити не тільки числові значення, але і будь-які формули, що включають зумовлені функції і змінні моделі, наприклад $5 * g * \sin(\text{Time})$).

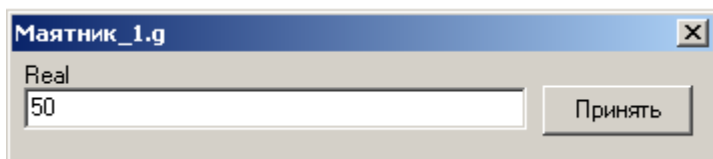


Рис 4.19

Результат зміни видно на Рис 4.20: зі зростанням g частота коливань збільшується.

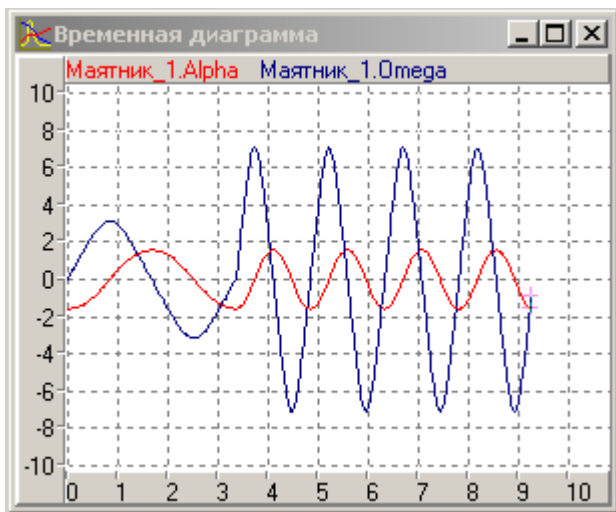



Рис 4.20

Для того, щоб змінювати змінну плавно, необхідна 2D-анімації.

Зауваження. Візуальні інструменти моделювання дають величезні можливості, але користуватися ними треба з оглядкою на істота математичної моделі. Наприклад, якщо ви захочете стрибком змінити довжину маятника L , вам потрібно буде не забути одночасно змінити і кутову швидкість Ω , щоб не порушити закон збереження моменту імпульсу, в іншому випадку ви отримаєте невірний результат. Якщо ж ви захочете змінювати L плавно, то це взагалі не можна робити: треба вводити інші рівняння. На жаль, пакет моделювання не в змозі контролювати семантику завдання і дозволить вам плавно міняти L , залишаючи на вашій совісті абсолютно фізично безглуздий отриманий результат.

4.4.5.2D-анімації.

Незважаючи на все багатство можливостей 3D-анімації, двовимірна анімація не втрачає свого значення для пакетів візуального моделювання. Це насамперед стосується різного роду динамічних блок-схем і панелей управління, які за своєю природою є двовимірними.

За допомогою команди «Вікна / Нова 2D-анімація» створимо анімаційне вікно. Далі відкриємо за допомогою кнопки  або команди «Сервіс / Стандартні 2D-компоненти» панель 2D-компонент (Рис 4.21).

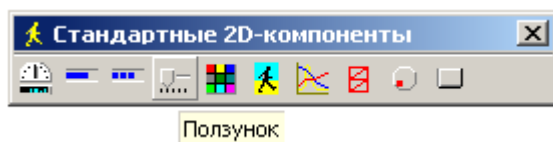


Рис 4.21

Виберемо на цій панелі компоненту «Повзунок» і методом «drag-and-drop» перетягнемо її в ліву частину вікна анімації. Клацнемо на який з'явився зображенні повзунка правою кнопкою миші і за допомогою команд спливаючого меню встановимо вертикальне розташування повзунка, режим показу числового значення і інтервал значень 0 .. 100 (Рис 4.22).

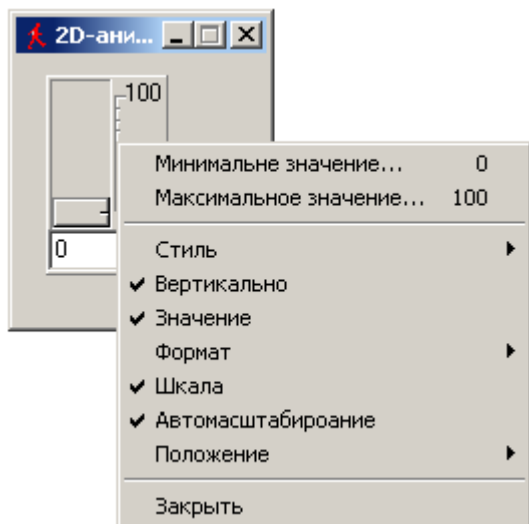


Рис 4.22

За допомогою миші можна переміщати 2D-компоненти в межах вікна анімації, а також змінювати розмір компоненти. Клацнемо правою кнопкою миші зверху від компоненти і, виконавши команду «Додати напис» спливаючого меню, створимо пояснювальну напис для повзунка (Рис 4.23). Напис також можна переміщати за допомогою миші з притиснутою клавішею. Тепер потрібно зробити найважливіше - зв'язати повзунок зі змінною моделі. Робиться це аналогічно 3D-анімації: методом «drag-and-drop» переміщаємо змінну g з вікна змінних і кидаємо її на зображенні повзунка. Тепер повзунок буде відображати поточне значення змінної, а при переміщенні повзунка за допомогою миші присвоювати відповідне значення цієї змінної. Крім повзунка інтерактивними 2D-компонентами є ще «Круговий регулятор» і «Кнопка». Інші компоненти дозволяють тільки відображати поточне значення змінної. Використовуємо два компоненти «Лінійний індикатор суцільний» для індикації значень змінних α і ω (Рис 4.23).

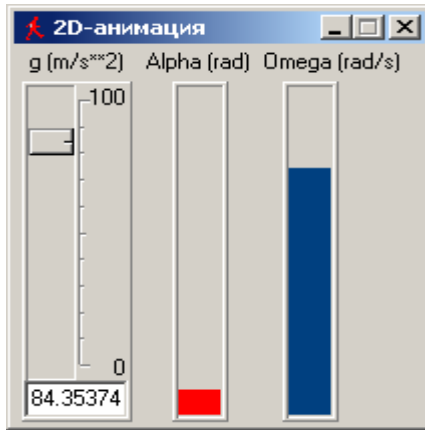


Рис 4.23

Результат плавної зміни прискорення сили тяжіння показаний на Рис 4.24. Зверніть увагу, якщо різко зменшити величину g в момент проходження нижнього вертикального положення (максимум кінетичної енергії), то маятник можна закрутити навколо осі обертання. Це ілюстрація важливої властивості інтерактивної анімації: через користувача замикається зворотній зв'язок, ви бачите на екрані динаміку системи і відповідно негайно реагуєте.

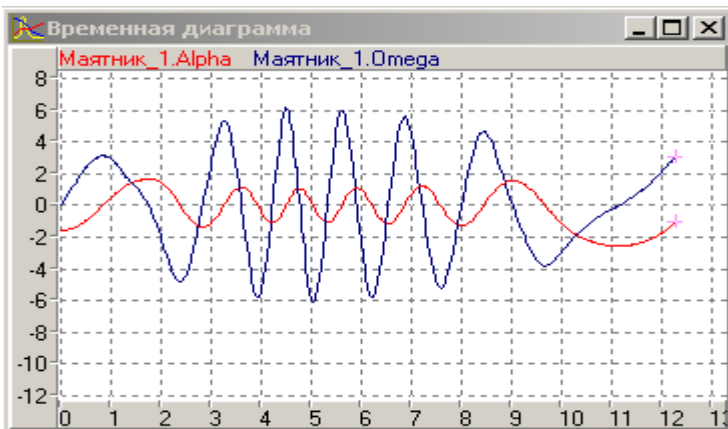


Рис 4.24

4.6. Зупинка за умовою.

При проведенні експериментів з моделлю часто виникає необхідність зупинитися в деяких характерних точках фазової траєкторії (наприклад, ми міняли величину g «в кінці першого періоду коливань»). Точно визначити цей момент «на око», наприклад, спостерігаючи графіки, досить складно. Тому в пакеті передбачена можливість зупинки по умові. Цим умовою може бути істинність деякого логічного виразу (предиката), спрацьовування визначеного або будь-якого переходу, а також вхід в певний вузол карти станів. У даному випадку нам знадобиться останов по предикату.

За допомогою кнопки або команди «Сервіс / Умови зупину» відкриємо діалог «Умови зупинки», виберемо сторінку «За предикату», клацнемо на ній правою кнопкою миші і, виконавши команду «Додати» спливаючого меню, перейдемо в діалог редагування логічного предиката (Рис 4.25).

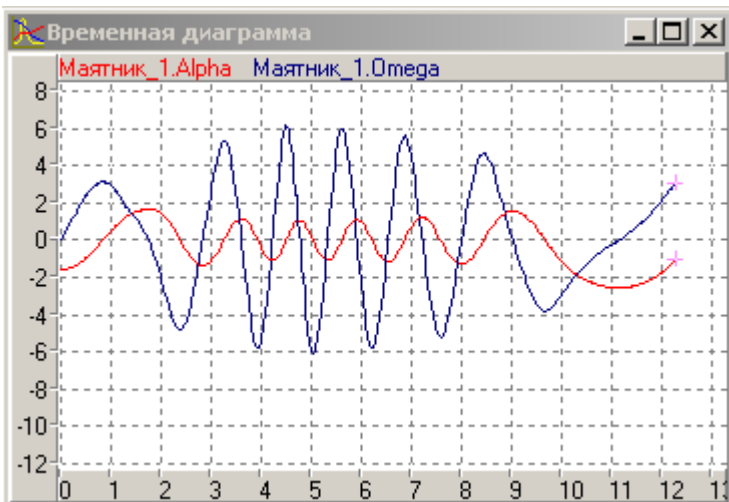


Рис 4.25

Новий предикат додається до списку «чекають» предикатів (Рис 4.26). Галочка близько предиката показує, що він активний, тобто по ньому може відбутися зупинка моделі. Предикат можна пасивізувати і знову активізувати за допомогою миші.

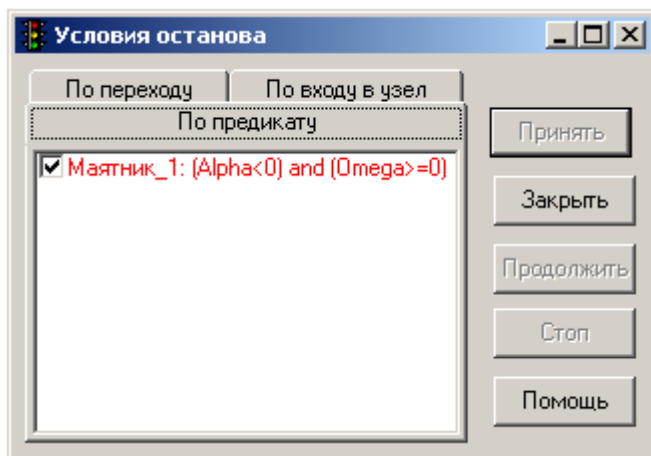


Рис 4.26

Зупинка за що чекає предикату відбувається в момент, коли його значення змінюється з FALSE на TRUE.

Спрацював предикат виділяється червоним кольором.

У нашому випадку перший останов відбудеться в початковий момент. Натиснемо кнопку «Продовжити» і дочекаємося другого останову - це і буде початок другого періоду коливань. Момент зупинки (як і будь-якого іншого перемикавання в гібридній одягли) визначається з точністю до часової похибки, заданої в установках чисельних методів (за замовчуванням 1e-6).

4.4.7. Формульний калькулятор.

Іноді в ході експерименту виникає необхідність разових розрахунків за певними формулами. Наприклад, якщо ви захочете інтерактивно змінити довжину маятника L , то, як було зазначено вище, необхідно одночасно змінити значення кутової швидкості, виходячи зі співвідношення. Для цього зручно використовувати формульний калькулятор.

У потрібний момент часу за допомогою кнопки або команди «Сервіс / Калькулятор» відкриємо вікно формульного калькулятора. Потім введемо потрібний вираз і натиснемо клавішу «Enter» (Рис 4.27).

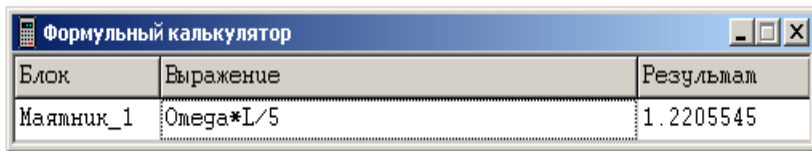


Рис 4.27

Вирази в калькуляторі можуть включати в себе будь-які зумовлені функції (див. опис вхідної мови), а також імена змінних моделі (береться їх поточне значення). Детальніше робота з калькулятором описана у вбудованій довідковій системі.

4.4.8. План.

Часто буває зручно запрограмувати певну послідовність впливів на модель і відтворити її автоматично в кожному прогоні. Для цього служить план прогону моделі.

Редактор плану викликається кнопкою чи командою «Сервіс / План» головного меню. На Рис 4.28 показаний

план прогону, в якому на 4-й одиниці часу величина g приймає значення 50, а на 12-й одиниці прогін завершується. Детальніше робота з планом описана у вбудованій довідковій системі.

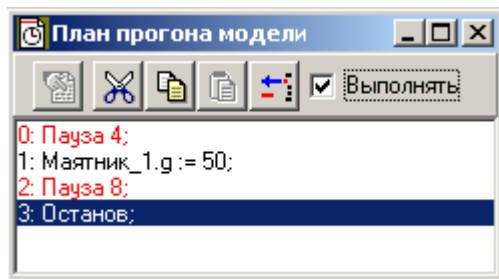


Рис 4.28

Лекція 5. Створюємо гібридну модель.

В якості типової гібридної моделі розглянемо модель відриву маятника. Новий проект ми будемо створювати на основі розглянутого вище проекту «Маятник.mvb». Для цього за допомогою команди «Проект / Зберегти як» збережемо цей проект як «Маятник що відривається.mvb» і далі будемо редагувати вже існуючий клас «Маятник». Трохи пізніше буде розглянуто як створити для відриву маятника новий клас у тому ж проекті, успадкувавши потрібні властивості від простого маятника.

Шлях до готового прикладу «Моделі \ Маятник \ Маятнк, що відривається.mvb».

Це модель ізольованої системи з декількома якісними станами. Тому клас, що створюється за замовчуванням при відкритті нового проекту, для неї підходить не цілком. Нагадаємо, що при відкритті нового проекту в створюваному за замовчуванням класі автоматично створюється «Система_уравнений_1», яка приписується початкового стану Init в головній карті

поведінки.«Система_уравнений_1» нам ще стане в нагоді, а ось приписування їй даму станом в даному випадку не потрібно. Виділимо вузол Init в головній карті поведінки (Рис 5.1) і за допомогою кнопки або команди «Встановити пусте локальне поведінка» спливаючого меню зробимо вузол Init порожнім.

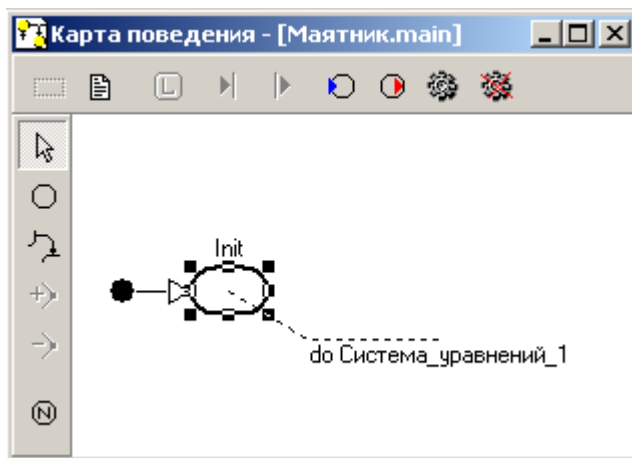


Рис 5.1

Враховуючи, що вузол Init успадкований від батьківського класу CDevice і ні видалити, ні перейменувати в даній версії пакету його не можна, можна рекомендувати дві «стратегії» побудови головної карти поведінки гібридної моделі:



Рис 5.2

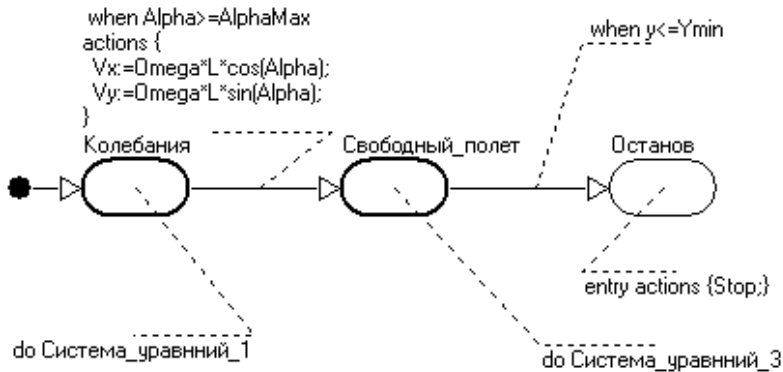


Рис 5.3

1) стоїться нова «Карта_поведення_1» (Рис 5.3), яка приписується вузлу Init (Рис 5.2).

2) Вузол Init як і раніше використовується як початковий з безумовним переходом у вузол, відповідний перший змістовному станом моделі - в даному випадку це стан «Коливання» (Рис 5.4). Цей підхід зручний у тому випадку, коли обчислення узгоджених початкових значень змінних є нетривіальною процедурою, пов'язаної, наприклад, з одноразовим рішенням системи алгебраїчних рівнянь. У цьому випадку початкову послідовність формул зручно поміщати у вхідних діях початкового вузла, а початкову систему алгебраїчних рівнянь приписувати початкового вузла як безперервне поведінку. Важливо усвідомити, що навіть якщо тривалість перебування в будь-якому стані дорівнює нулю в безперервному часу) вхід в стан і вихід зі стану відбуваються в одній «тимчасовій щілині»), то алгебраїчна складова системи рівнянь, приписаної цього стану, все одно буде вирішена! На Рис 5.4 показано обчислення початкових значень в початковому вузлі Init.

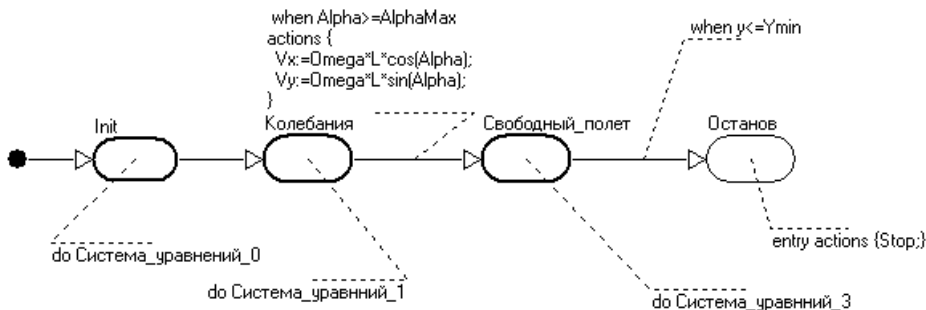


Рис 5.4

У будь-якому випадку для опису гібридного поведінки вам доведеться створювати нову карту поведінки або модифіковані головну.

Далі в цьому прикладі ми будемо використовувати підхід, показаний на Рис 5.2, Рис 5.3.

5.1. Створення карти поведінки.

Нову карту поведінки можна створити двома способами:

- 1) перемістити мишу на секцію «Локальні карти поведінки» у вікні класу, викликати по правій кнопці миші спливаюче меню і виконати команду «Додати». Надалі вам доведеться приписати нову карту поведінки якомусь вузлу іншу картку поведінки більш високого рівня ієрархії методом «drag-and-drop»;
- 2) виділити вузол, якому ви хочете приписати нову карту поведінки, викликати правою кнопкою миші спливаюче меню і виконати команду «Створити нову карту поведінки». В результаті нова карта поведінки буде не тільки створено (вона з'явиться в секції «Локальні карти поведінки» вікна класу), а й буде приписана даного вузла.

5.2. Редагування карти поведінки.

У загальному випадку карта поведінки являє собою граф, вузли якого відповідають якісним станам модельованої системи, а дуги - переходах з одного стану в інший (Рис 5.5). Вузлів - станам відповідають певні поведінки:

- Пусте поведінка, за яким значення змінних не змінюються, відповідний вузол зображується тонкою лінією (на Рис 5.5 це вузол Node_2);
- Безперервне поведінка, заданий системою рівнянь, відповідний вузол зображується жирною лінією (на Рис 5.5 це Node_3);
- Дискретне чи гібридне поведінку, задане іншою картою поведінки, відповідний вузол зображується подвійний тонкою лінією (на Рис 5.5 це Node_1).

Переходи зображуються ламаною лінією зі стрілкою, що вказує напрямок переходу. Перехід, у якого відсутня початковий або кінцевий вузол (або обидва вузла), є незавершеним. Незавершені переходи зображуються пунктирними лініями і відсутні у виконуваний моделі. Один з переходів є початковим (замість вихідного вузла зображується жирна крапка). Він спрацьовує відразу при ініціалізації примірника карти поведінки і таким чином вказує на початковий вузол карти станів (на Рис 5.5 це вузол Node_2).

Крім того, у вікні карт поведінки зображуються:

- Імена вузлів. Якщо з малюнка не очевидно, яке ім'я у даного вузла, необхідно виділити цей вузол і натиснути кнопку на лівій інструментальній панелі. Перейти в режим редагування імені можна подвійним клацанням миші на його зображенні або за допомогою команди «Редагувати / Редагувати як текст (F2)» головного меню;
- Зноски, що вказують на приписані поведінки, вхідні і вихідні дії у вузлі, умови та дії переходу. Виноски можуть не зображуватися (у разі перенасиченості графа) при

скиданні перемикача «Показувати виводи на картах стану» на сторінці «Вид» діалогу «Установки інтегрованого середовища», який можна відкрити за командою «Сервіс / Налаштування» головного меню.

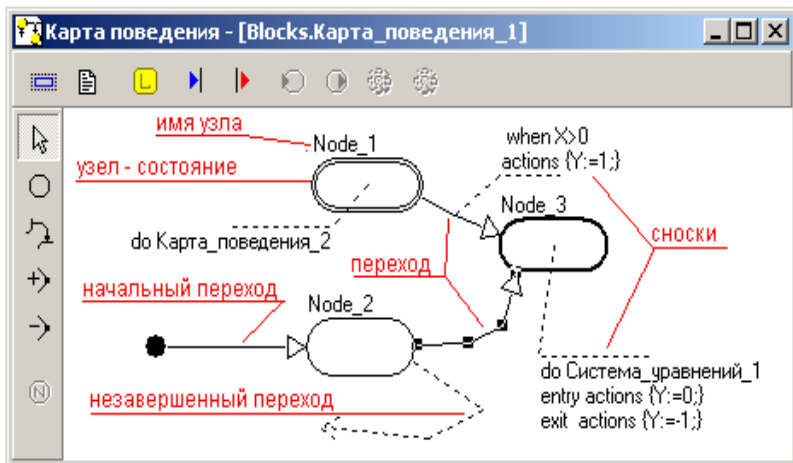



Рис 5.5

Один з елементів карти поведінки може бути виділений натисканням лівої кнопки миші на зображенні цього елемента. Виділення зображується прямокутниками на місці опорних точок (на Рис 5.5 виділений перехід з Node_2 в Node_3) або кольором (для імен вузлів). Виділений елемент можна переміщати або за допомогою миші з притиснутою лівою кнопкою, або за допомогою клавіш переміщення курсора.

Натисканням лівої кнопки миші при притиснутою клавіші «Shift» можна виділити групу елементів. Групу елементів можна перемістити за допомогою миші з притиснутою лівою кнопкою, або використовувати для вирівнювання за допомогою команди «Редагувати / Вирівняти» головного меню.

Будь-яка зміна в карті поведінки може бути скасовано за допомогою кнопки  на панелі кнопок головного вікна або команди «Редагувати / Скасувати зміни (Alt + Backspace)» головного меню. Кількість відмінюються змін задається в настройках інтегрованого середовища на сторінці «Основні» (команда «Сервіс / Налаштування» головного меню).

Вікно редактора карти поведінки має дві панелі кнопок: ліву і верхню. Ліва панель містить кнопки з фіксацією, пов'язані з різними режимами роботи графічного редактора карти поведінки (Рис 5.6). Стандартним є режим виділення, переходом у цей режим як правило закінчуються всі операції.

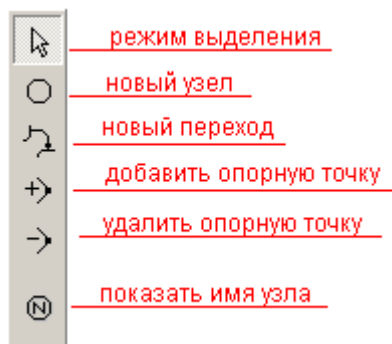











Рис 5.6

Верхня панель містить кнопки, пов'язані з редагуванням властивостей виділеного елемента або карти поведінки в цілому:

- Кнопка  призначена для переходу в діалог редагування імені карти поведінки;
- Кнопка  призначена для переходу в діалог редагування пояснювального тексту для карти поведінки;

- Кнопка  призначена для редагування локальних змінних, функцій і процедур карти поведінки;
- Кнопка  призначена для переходу в діалог редагування умови спрацьовування виділеного переходу;
- Кнопка  призначена для переходу в діалог редагування послідовності миттєвих дій у виділеному переході;
- Кнопка  призначена для переходу в діалог редагування послідовності вхідних дій виділеного вузла;
- Кнопка  призначена для переходу в діалог редагування послідовності вихідних дій виділеного вузла;
- Кнопка  призначена для переходу у вікно редагування поведінки, приписаного виділеного вузла;
- Кнопка  призначена для приписування виділеного вузла порожнього поведінки.

5.2.1. Редагування вузлів.

Щоб створити новий вузол потрібно:

- За допомогою кнопки або команди «Створити новий вузол» спливаючого меню перевести редактор у режим створення нового вузла (ознакою цього режиму є поява хрестоподібного курсору миші);
- Поставити хрестоподібний курсор у місце, де повинен розташовуватися верхній лівий кут нового вузла, натиснути ліву кнопку миші і, не відпускаючи її, переміщати мишу вправо і вниз. Разом з мишею змінюється зображення вузла. Положення курсору відповідає правому нижньому куті сайту. У потрібному положенні слід відпустити кнопку миші.

Новий вузол буде за замовчуванням мати ім'я «Node_1, 2, ...».

Якщо новий вузол був першим в карті поведінки, то йому буде автоматично сопоставлен початковий перехід, тобто

перший вузол за замовчуванням вважається початковим. Надалі ви можете зробити початковим інший вузол, перемістивши початковий перехід так, щоб його стрілка опинилася на зображенні цього вузла.

Щоб видалити вузол, виділіть його і виконайте або команду «Видалити вузол» спливаючого меню, або команду «Редагувати / Видалити» головного меню.

Щоб перемістити вузол, натисніть на його зображенні ліву кнопку миші і, переміщаючи мишу з притиснутою кнопкою, відбуксують в потрібне місце після чого відпустіть кнопку. Для більш точного переміщення виділеного вузла в певному напрямку можна використовувати клавіші переміщення курсору на клавіатурі або спеціальні кнопки в правій частині панелі кнопок головного вікна. Крім того, ви можете точно задати положення вузла за допомогою команди «Положення» спливаючого меню. Разом з вузлом автоматично переміщуються останні сегменти пов'язаних з цим вузлом переходів.


Щоб змінити розмір вузла, виділіть його, схопите мишею одну з опорних точок, показаних квадратами, і тягніть її до потрібного положення.

Щоб перемістити ім'я вузла перетягніть його методом «drag-and-drop».

Щоб змінити ім'я вузла клацніть на ньому двічі мишею або виділіть його і виконайте команду «Редагувати / Редагувати як текст» головного меню (функціональна клавіша F2). На місці імені з'явиться однорядковий текстовий редактор, в якому ви можете відредагувати ім'я вузла. Щоб скасувати редагування натисніть «Esc», щоб завершити редагування натисніть «Enter» або клацніть мишею за межами редактора рядка.

Щоб помістити у вузол вже існуюче локальне поведінка (систему рівнянь або карту поведінки), виділіть відповідну поведінку у вікні класу і методом «drag-and-drop» перетягніть на зображення вузла.

Щоб помістити у вузол нове локальне поведінка, виконайте команду «Створити нову систему рівнянь» або команду «Створити нову карту поведінки» спливаючого меню

Щоб видалити з вузла приписане йому локальне поведінка, виділіть вузол і натисніть кнопку  або виконайте команду «Встановити пусту локальну поведінку» спливаючого меню.

Керуючись цими правилами, створимо в класі «Маятник» нову карту поведінки «Карта_поведенки_1», клікнувши двічі мишею на її назві, перейдемо у вікно редактора карт поведінки, введемо в ній три нових вузла і замінимо їх імена на «Коливання», «Свободний_полет» і «Зупинка» (Рис 5.7).

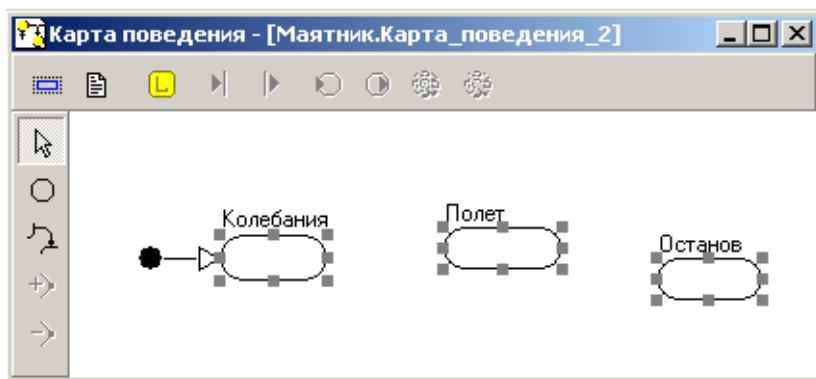



Рис 5.7

З метою поліпшення зовнішнього вигляду карти стану виділимо всі три вузли як групу (нагадаємо, це робиться лівою кнопкою миші при притиснутою клавіші «Shift») і виконаємо команду «Редагувати / Вирівняти» головного меню. У діалозі (Рис 5.8) виберемо вирівнювання по верхніх сторонам. Натиснемо «ОК» і виконаємо цю ж команду знову, вибравши на цей раз вирівнювання по ширині і висоті. У результаті ми отримаємо три абсолютно однакових за розмірами вузла, розташовані на одній горизонтальній лінії.

Нагадаємо, що будь-яка зміна в карті поведінки може бути скасовано за допомогою кнопки  на панелі кнопок головного вікна або команди «Редагувати / Скасувати зміни (Alt + Backspace)» головного меню.

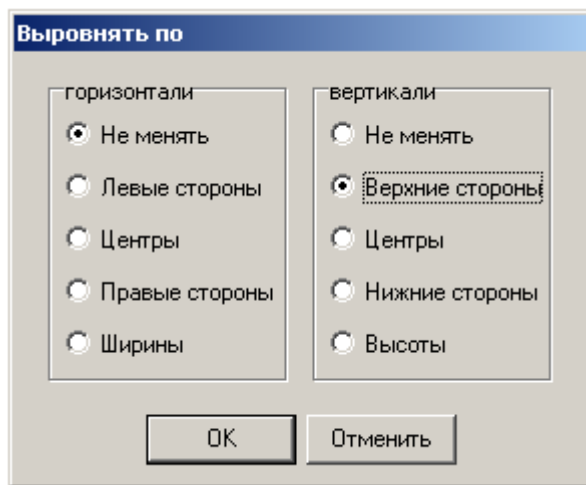


Рис 5.8

Всі створені вузли карти поведінки за замовчуванням містять порожні локальні поведінки, тобто при знаходженні моделі в цьому стані в ній нічого не буде

змінюватися ..«Система_уравнений_1», що описує коливання маятника, вже існує в класі «Маятник» і ми просто методом «drag-and-drop» перетягнемо її з вікна класу на вузол «Колівання». Систему рівнянь, що описує вільний політ («Система_уравнений_3»), необхідно створити (Рис 5.9), а потім присвоїти вузла «Свободний_полет». Для цієї системи рівнянь нам знадобляться ще дві змінні:

Vx: double: = 0;

Vy: double: = 0;

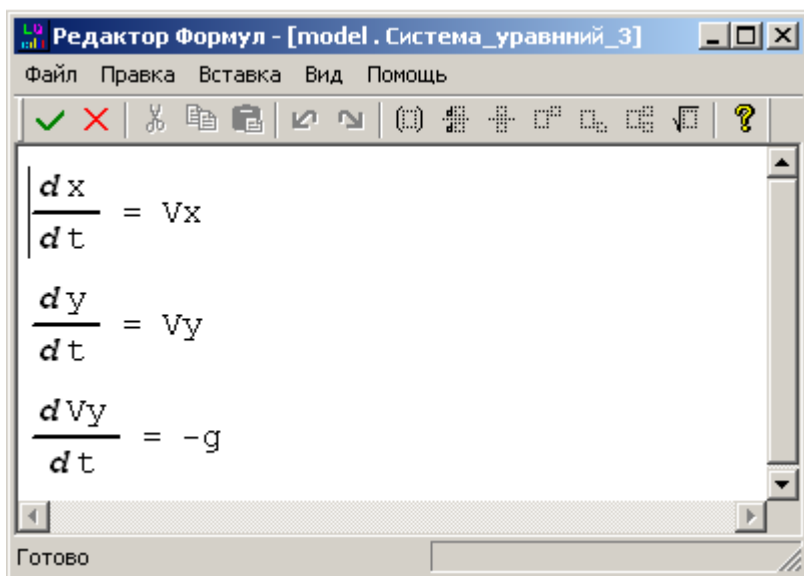



Рис 5.9

Вузол «Зупинка» повинен містити в своїх вхідних діях виклик визначеної процедури «Stop». Виділимо цей вузол і потім за допомогою кнопки  або команди «Вхідні дії у вузлі» спливаючого меню перейдемо до вікна редагування послідовності вхідних дій (Рис 5.10).

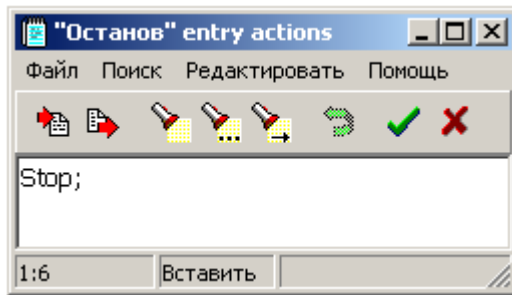



Рис 5.10

Послідовність миттєвих дій (вхідних дій у вузлі, вихідних дій у вузлі чи дій в переході) - це послідовність алгоритмічних операторів, задана в текстовій формі. Набір допустимих алгоритмічних операторів (невелике підмножина мови програмування ADA) описано в розділі «Вхідна мова / Алгоритмічні оператори» довідкової системи. У даному випадку послідовність включає один єдиний оператор - звернення до процедури.

Тепер можна переходити до створення переходів.

5.2.2. Редагування переходів.

Щоб створити завершений перехід, потрібно:

- За допомогою кнопки  або команди «Створити новий перехід» спливаючого меню перевести редактор у режим створення нового переходу (ознакою цього режиму є поява хрестоподібного курсору миші);
- Підвести курсор миші на зображення вихідного вузла (наприклад, «Коливання»). Курсор при цьому зміниться на зображення хреста в колі. Потім натиснути ліву кнопку миші і, не відпускаючи її, перемістити мишу на зображення кінцевого вузла (наприклад, «Свободний_полет»). Коли курсор при цьому зміниться на

зображення хреста в колі, відпустити кнопку. Початковий і кінцевий вузол може бути один і той же.

Отримуємо перехід з вузла «Коливання» у вузол «Свободний_полет» Точно таким же чином створюємо перехід з вузла «Свободний_полет» у вузол «Зупинка» (Рис 5.11).

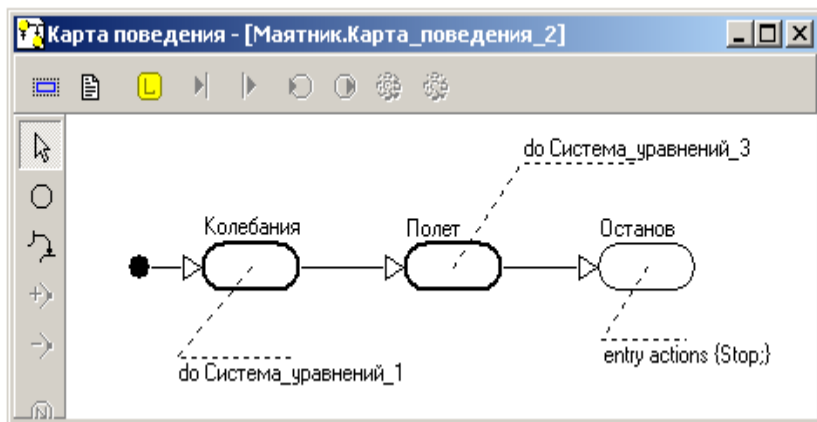


Рис 5.11


Щоб змінити лінію переходу, потрібно виділити перехід, натиснути ліву кнопку миші на квадраті, що зображає одну з опорних точок (не крайніх), потім, не відпускаючи кнопки, переміщати мишу і в необхідному положенні відпустити кнопку. Для точної вказівки координат опорних точок необхідно за допомогою команди «Координати точок» спливаючого меню викликати діалог редагування координат (на Рис 5.12 показаний цей діалог стосовно до виділеного переходу на Рис 5.5).


Зауваження. Не забувайте, що у вікнах MS Windows вісь X спрямована вправо, а вісь Y вниз.

Точки	X	Y
1: Node_2	206	113
2:	237	123
3:	257	114
4: Node_3	265	86

Рис 5.12

Щоб змінити вихідний або кінцевий вузол переходу, потрібно перетягнути мишею відповідну крайню опорну точку на зображення іншого вузла. Зокрема, якщо перетягнути її на вільне поле, можна зробити перехід незавершеним.

Щоб додати нову опорну точку на лінію переходу, потрібно виділити перехід і за допомогою кнопки  або команди «Додати опорну точку» спливаючого меню перевести редактор у режим додавання опорної точки (ознакою цього режиму є поява хрестоподібного курсору миші). Потім слід підвести курсор до потрібної точки лінії переходу (курсор зміниться на хрест з квадратом в центрі) і клацнути лівою кнопкою миші.

Щоб видалити опорну точку з лінії переходу, потрібно виділити перехід і за допомогою кнопки  або команди «Видалити опорну точку» спливаючого меню перевести редактор у режим видалення опорної точки (ознакою цього режиму є поява хрестоподібного курсору миші). Потім слід підвести курсор до опорної точки, яку слід видалити

(курсор зміниться на зображення руки з витягнутим вказівним пальцем) і клацнути лівою кнопкою миші.

Щоб створити незавершений перехід, потрібно в режимі створення нового переходу натискати кнопку або відпускати кнопку поза зображення будь-якого вузла, коли курсор має форму хреста. Завершений перехід можна зробити незавершеним, якщо «відтягнути» крайню точку лінії переходу від зображення вузла і кинути її на вільному полі.

Зауваження. Виникає питання: а навіщо, власне, потрібні незавершені переходи, якщо вони все одно не потрапляють у виконувану модель? Справа в тому що, по-перше, незавершеними переходи можуть ставати тимчасово, коли або знищується вихідний або кінцевий вузол, або перехід «відривається» від свого вихідного чи кінцевого вузла. По-друге, іноді буває зручно використовувати незавершений перехід як проміжну форму в процесі редагування. Наприклад, перехід з «Коливання» в «Свободний_полет» можна створити в три етапи: спочатку створити незавершений перехід (Рис 5.13а), потім, перетягнувши його початкову опорну точку на вузол «Коливання», зробити його вихідним (Рис 5.13б) і потім, перетягнувши останню опорну точку на вузол «Свободний_полет», зробити цей вузол кінцевим (Рис 5.13в).

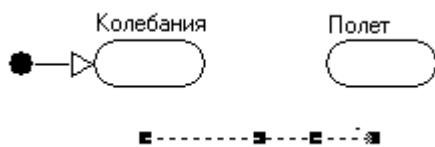


Рис 5.13 а)



Рис. 5.13.б)



Рис 5.13.в)

Всі введені нами переходи за замовчуванням є безумовними, тобто вони спрацюють негайно, як тільки вихідний вузол стане поточним. Таким чином, робота моделі, що задається картою поведінки, показаної на Рис 5.11, почнеться і закінчиться «в одну мить» під «тимчасової щілини».

Щоб цього не сталося, для переходів повинні бути визначені умови спрацьовування. Виділимо перехід «Коливання» □ «Свободний_полет» і за допомогою кнопки або команди «Умова спрацьовування переходу» спливаючого меню перейдемо в діалог редагування умови спрацьовування (Рис 5.14), в якому введемо необхідна умова (докладніше про переходах та умови їх спрацьовування в розділі «Вхідна мова / Карта поведінки» довідкової системи). Попередньо додамо параметр AlphaMax: double: = pi / 4;

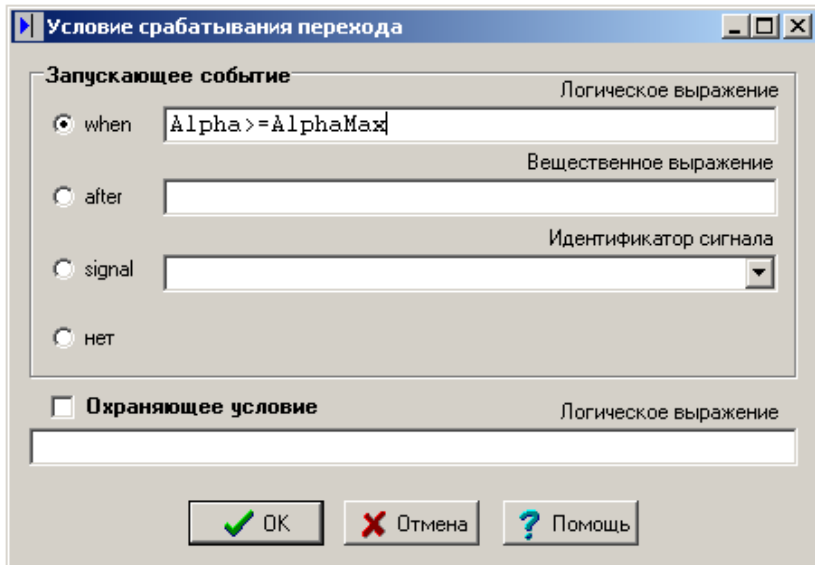


Рис 5.14

Крім умови в цьому переході є ще миттєві дії, що виконуються при спрацьовуванні переходу - розрахунок початкових значень складових швидкості при відриві маятника. Виділимо перехід і за допомогою кнопки або команди «Дії переходу» спливаючого меню викличемо редактор послідовності дій (Рис 5.15). Зауважимо, що в будь-яку послідовність миттєвих дій як і в систему рівнянь можуть бути вставлені необхідні коментарі.

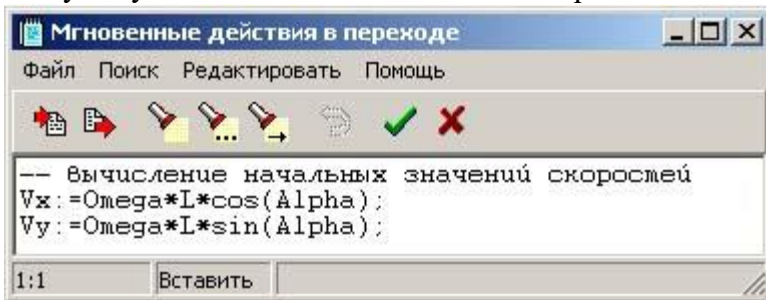


Рис 5.15

Аналогічним чином введемо умова спрацьовування «when $y \leq Y_{min}$ » для переходу «Свободний_полет» □ «Зупинка», додавши попередньо параметр $Y_{min}: \text{double} = -3;$

Помістимо тепер нашу карту стану (Рис 5.3) у вузол Init головної карти поведінки (Рис 5.2) і наша модель готова.

5.3. Візуальна модель відривається маятника.

Запустивши модель, ми бачимо по фазовій діаграмі, що маятник начебто дійсно відривається (Рис 5.16).

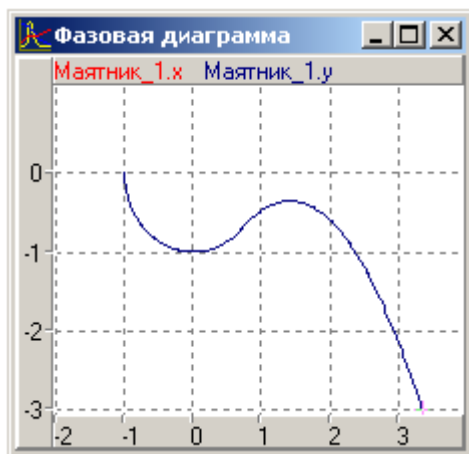


Рис 5.16

Однак, хотілося б спостерігати в динаміці зміну якісних станів моделі на карті поведінки. Для цього у вікні віртуального стенду клацнемо правою кнопкою миші на зображенні примірника маятника «Маятник_1» і виконаємо команду «Поведінка» спливаючого меню (Рис 5.17).

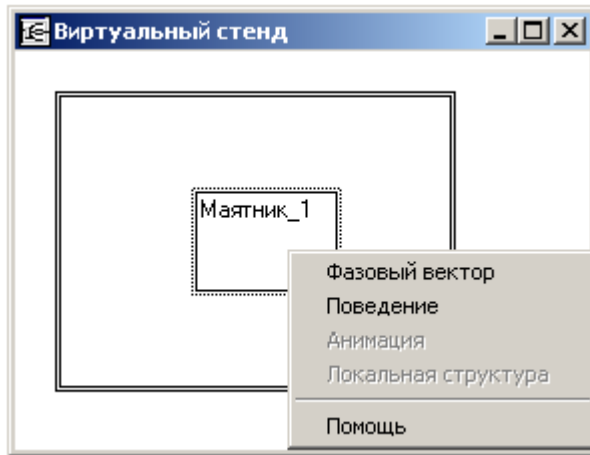


Рис 5.17

В результаті виконання цієї команди відкриється вікно динамічної візуалізації головної карти поведінки даного блоку. Якщо (як в даному випадку) ця карта виродилася, тобто є тільки один вузол, якому приписана інша карта стану, то вона відкриється в якості головної (Рис 5.18).

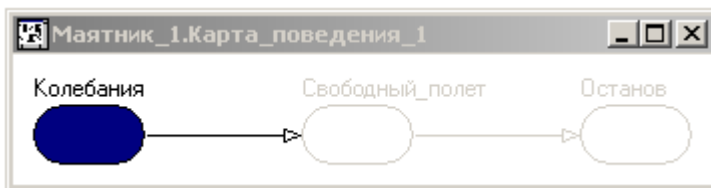



Рис 5.18

При динамічній візуалізації карти поведінки використовуються наступні угоди: поточний вузол зображений суцільний зафарбуванням (на Рис 5.18 це вузол «Коливання»), активні переходи (вихідні з поточного вузла) показані чорною лінією (на Рис 5.18 це перехід з вузла «Коливання» у вузол «Свободний полет»), неактивні вузли та переходи - сірими лініями (на Рис 5.18 це вузол

«Зупинка» і перехід до нього). Лінія спрацьовує переходу підсвічується на інтервал часу, достатній для фіксації оком (200 мс). Всі ці кольори можна налаштувати по-своєму на сторінці «Кольори» в установках виконуваної моделі (перехід до редагування установок по кнопці  або команді «Установки / Модель»).

Якщо в діалозі редагування умов зупинки на сторінці «За переходу» поставити прапорець «Будь-який перехід», то при спрацьовуванні кожного переходу модель буде зупинятися і спрацьовує перехід буде виділятися мерехтінням у вікні своєї карти поведінки (якщо вікно не було відкрито, воно відкриється автоматично).

Якщо локальним поведінкою у вузлі є інша карта поведінки, то відповідна їй вікно відкривається подвійним клацанням миші на цьому вузлі.

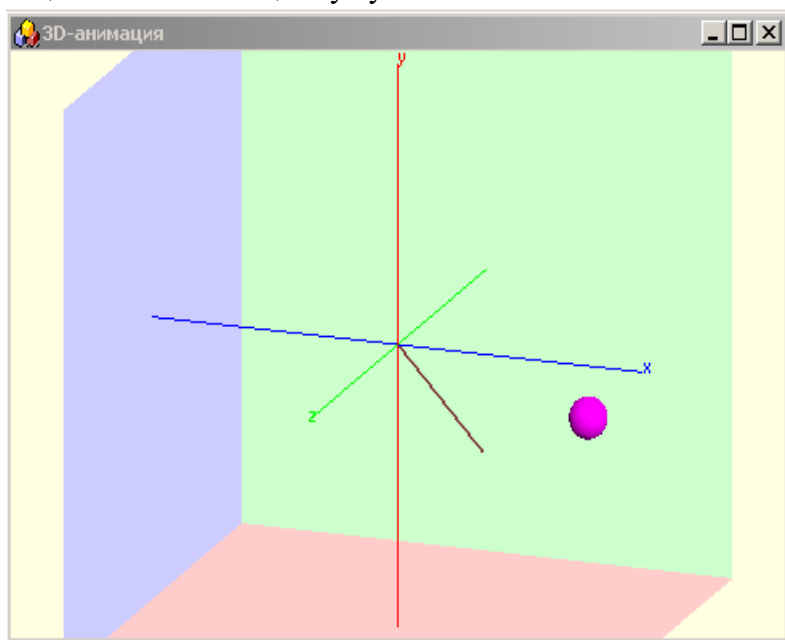


Рис 5.19

У нас залишається ще одна невелика неладність: У вікні анімації видно, що після моменту відриву стрижень починає розтягуватися, не відриваючись від кульки. Це відбувається тому, що в якості координат вільного кінця стержня в установках анімації задані координати матеріальної точки. Тому нам доведеться повернутися в інтегровану оболонку, ввести в класі «Маятник» ще дві змінних

```
xL: double: = 0;
```

```
yL: double: = 0;
```

і додати в «Сістема_уравнений_1» ще дві формули

```
xL = x;
```

```
yL = y;
```

Знову запустимо нову візуальну модель, в установках анімації координати вільного кінця стержня можна порівняти змінним xL , yL і отримаємо нарешті правильну картинку (Рис 5.19).

Ця невелика доробка є гарною ілюстрацією того, що модель іноді доводиться спеціально змінювати виключно на догоду анімації.

Лекція 6. Створюємо компонентну модель.

Створення компонентної моделі ми продемонструємо на прикладі вимірювальної системи швидкості та шляхи маятника (див.«Моделювання в прикладах»). Відкриємо проект «Маятник.mvb» і за допомогою команди "Файл / Зберегти як» скопіюємо його у файл «ІзмерительнаяСістема.mvb».

Шлях до готового прикладом «Моделі \ Маятник \ ИзмерительнаяСістема.mvb».

6.1. Зовнішні змінні.

Клас «Маятник» буде використовуватися практично без змін: потрібно тільки зробити внутрішні змінні зовнішніми видимими ззовні виходами. Зробити це дуже просто. У вікні класу «Маятник» хапаємо мишкою зображення змінної в секції «Внутрішні змінні», тягнемо його з притиснутою кнопкою миші в розділ «Виходи» і там кидаємо (знову все той же метод drag-and-drop!). Після всіх цих маніпуляцій у класі «Маятник» зникне внутрішня змінна і з'явиться вихід. На відміну від внутрішньої змінної в описі зовнішньої змінної має бути присутня додаткова інформація про її розташуванні на умовному зображенні блоку. Тому при додаванні в класі «Маятник» нової зовнішньої змінної негайно відкривається вікно редагування структури класу «Маятник» (Рис 6.1а). За замовчуванням всі входи розташовуються на лівій стороні прямокутника, що зображує на структурній схемі блок, а всі виходи на правій. Ви можете задати будь-яке розташування зовнішніх змінних, перетягуючи мишкою зображення входу або виходу. на потрібну сторону. Наприклад, щоб перемістити вихід на верхню сторону, підведіть курсор миші на стрілку, що зображає вихід (курсор зміниться, показуючи можливість переміщення), натисніть ліву кнопку миші і, не відпускаючи її, переміщайте мишу на верхню сторону прямокутника (курсор зміниться на хрестоподібний). При попаданні на сторони прямокутника в центрі хреста курсору з'явиться жирна крапка, що сигналізує про можливість кидання. Відпустіть кнопку миші і вихід буде зображуватися на верхній стороні блоку. За замовчуванням зовнішні змінні розташовуються на сторонах рівномірно, тобто розмістяться точно на середині сторони.

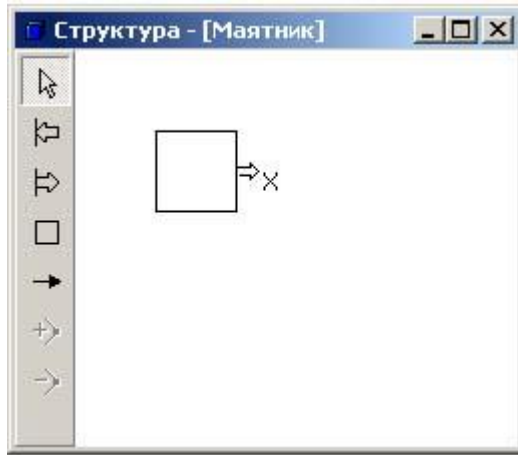


Рис 6.1 а)

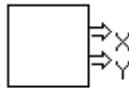




Рис 6.2 б)

Аналогічним чином зробимо виходом і внутрішню змінну. Виходи розмістяться на правій стороні блоку на відстані однієї третини сторони один від одного і від краю (Рис 6.1б). Якщо ви захочете поміняти їх місцями, наприклад, розташувати нижче, перетягніть нижче на ту ж праву сторону і киньте. Після цього змінні знову автоматично розмістяться рівномірно, але в новому порядку.

У вікні структури ви можете додавати нові зовнішні змінні, редагувати властивості існуючих зовнішніх змінних, а також видаляти ці змінні. Нагадаємо, що ті ж дії ви можете виконати і у вікні класу.

Щоб додати новий вхід або вихід натисніть відповідно на кнопки  або  на лівій панелі інструментів у вікні структури (Рис 6.1) або виконайте команди «Додати новий вхід» або «Додати новий вихід» спливаючого меню (з'являється при натисканні правої кнопки миші у вікні структури). Далі підведіть курсор миші до сторони блоку, на якій ви хочете розташувати нову зовнішню змінну (в центрі хреста курсору з'явиться жирна крапка), і натисніть ліву кнопку миші. Після цього виникне діалог редагування імені, типу та початкового значення змінної, такий же, що і для внутрішньої змінної (Рис 4.1).

Щоб увійти в діалог редагування властивостей зовнішньої змінної двічі клацніть мишею на її зображенні.

Щоб видалити зовнішню змінну виділіть її та виконайте команду «Видалити виділений елемент» спливаючого меню або команду «Редагувати / Видалити» головного меню.

6.2. Структурна схема.

Наступним кроком має бути створення класу «ІзмерительVS». Ми не будемо детально розглядати цей етап, оскільки він практично нічим не відрізняється від створення класу «Маятник» («Створюємо безперервну модель.»). Новий блок матиме два входи, на які надходять поточні координати, і два виходи - поточні значення повній швидкості і шляху. Рівняння блоку наведено в документі «Моделювання в прикладах».

Далі нам потрібно перейти до вікна «Віртуальний стенд» і зібрати схему їх маятника і вимірювача. Ця схема повинна містити один екземпляр класу «Маятник» і один екземпляр класу «ІзмерительVS». Примірник класу «Маятник» у нас уже є - це блок «Маятник_1» (Рис 3.1). На майбутньої схеми він повинен знаходитись зліва. Тому ми виділимо

блок і трохи перемістимо його вліво. Це можна зробити або просто мишкою з притиснутою лівою кнопкою, або з допомогою кнопок точного переміщення в правій частині інструментальної панелі головного вікна, або за допомогою клавіш переміщення курсору на клавіатурі. Крім того, ми ще трохи розширимо вправо прямокутник, що означає кордону віртуального стенду. Для цього потрібно виділити його, клацнувши мишею де-небудь на вільному місці всередині цього прямокутника і потягнути мишею одну з опорних точок на периметрі, позначених чорними квадратами (напрямок можливих переміщень вказується формою курсору).


Нагадаємо, що «Віртуальний стенд» - це специфічний ізольований клас-контейнер, в який поміщаються екземпляри блоків, що моделюється. Примірник віртуального стенду - це і є власне діюча модель. Цей примірник створюється виконуючою системою пакета моделювання щоразу при початку чергового прогону моделі та знищується після закінчення прогону. Треба сказати, що редагування схеми віртуального стенду рівно нічим не відрізняється від редагування будь-якої іншої структурної схеми, так що все сказане нижче відноситься і до будь-якого блоку-контейнера взагалі.

6.2.1. Додавання локального блоку.

Тепер потрібно помістити в структурну схему віртуального стенду примірник вимірювача. Це можна зробити двома способами:

1) методом «drag-and-drop»: у вікна управління проектом хапається мишкою необхідний клас, переноситься у вікно структурної схеми і впадає в потрібному місці. Це місце кидання буде верхнім лівим кутом для зображення локального блоку - примірника даного класу. Розміри

зображення блоку в цьому випадку приймаються стандартними.

2) шляхом безпосереднього завдання прямокутника, що зображує локальний блок. Натисніть кнопку  на лівій панелі інструментів або виконайте команду «Додати новий локальний блок». У діалозі виберіть клас нового локального блоку і натисніть кнопку «ОК». Поставте хрестоподібний курсор миші в позицію, де повинен розташовуватися верхній лівий кут зображення нового локального блоку, натисніть ліву клавішу миші і, не відпускаючи її, рухайте мишу вправо і вниз, розтягуючи прямокутник, що зображає блок. У потрібний момент відпустіть кнопку миші.

Тим чи іншим способом ми отримуємо стан вікна віртуального стенду, зображене на Рис 6.2.

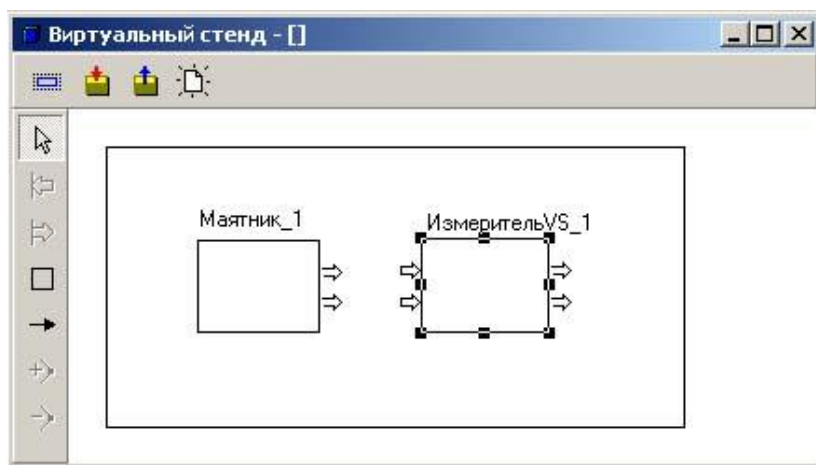


Рис 6.2

За замовчуванням новий локальний блок отримує ім'я, складене з імені класу і номера примірника даного класу у даній схемі. Так, якщо ми додамо ще один маятник, він отримає ім'я «Маятник_2» і т.д.

За замовчуванням імена зовнішніх змінних локальних блоків на структурній схемі не показуються. Цим

властивістю можна управляти на сторінці «Вид» діалогу налаштувань інтегрованого середовища (команда «Сервіс / Налаштування» головного мню). Однак навіть за відсутності імен при підведенні мишки до зовнішньої змінної її ім'я спливе як підказка.

Аналогічно при підведенні мишки до блоку ім'я його класу спливе як підказка.

6.2.2. Редагування локального блоку.

Для того, щоб видалити локальний блок виділіть його, натисніть на його зображенні праву клавішу миші і виконайте команду «Видалити блок» спливаючого меню. Щоб перемістити зображення локального блоку, виділіть його і або перетягніть методом «drag-and-drop», або використовуйте клавіші переміщення курсору. За допомогою команди «Положення» спливаючого меню можна задати точне положення блоку.

Щоб змінити розмір блоку, виділіть його, схопите мишею одну з опорних точок, показаних квадратами, і тягніть її до потрібного положення

Щоб перемістити ім'я блоку перетягніть його методом «drag-and-drop».

Щоб змінити ім'я блоку клацніть на ньому двічі мишею або виділіть його і виконайте команду «Редагувати / Редагувати як текст» головного меню (функціональна клавіша F2). На місці імені з'явиться однорядковий текстовий редактор, в якому ви можете відредагувати ім'я блоку. Щоб скасувати редагування натисніть «Esc», щоб завершити редагування натисніть «Enter» або клацніть мишею за межами редактора рядка.

Зовнішні змінні на зображенні нового локального блоку за замовчуванням розташована так само, як у вікні структури відповідного класу. Таке розташування може бути незручним в конкретній структурній схемі: наприклад, бажано розташувати всі входи на верхній

стороні, а не на лівій. Для цього виділіть зовнішню змінну на структурній схемі і перемістіть її або методом «drag-and-drop», або з допомогою клавіш переміщення курсора. За допомогою миші можна перемістити ім'я змінної, якщо воно виводиться на схемі.

За замовчуванням значення параметрів нового блоку рівні їх значенням у визначенні класу цього блоку. Щоб задати інші значення параметрів (наприклад, зробити початкове відхилення маятника рівним, $-\frac{\pi}{4}$ а не $-\frac{\pi}{2}$), виділіть блок, правою кнопкою миші викличте спливаюче меню і виконайте команду «Параметри». У діалозі змініть значення параметра (змнені значення виводяться жирним шрифтом) (Рис 6.3). Після закриття діалогу значення параметрів блоку зміняться. Для відновлення значення параметра за замовчуванням викличте діалог редагування параметрів, виділіть потрібний параметр і виконайте команду «Відновити значення за замовчуванням» спливаючого меню.

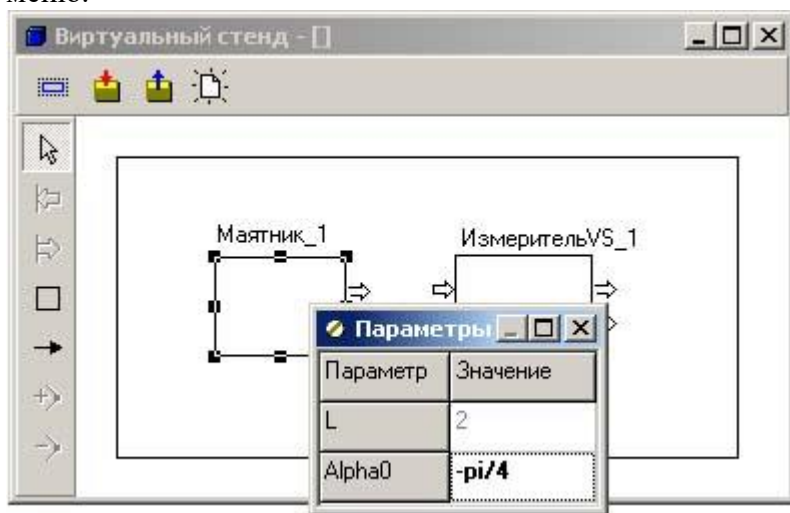



Рис 6.3

Для поліпшення зовнішнього вигляду структурної схеми (на Рис 6.2 видно, що блоки мають трохи різні розміри) виділимо обидва блоки як групу (нагадаємо, це робиться лівою кнопкою миші при притиснутою клавіші «Shift») і виконаємо команду «Редагувати / Вирівняти» головного меню. У діалозі (Рис 5.8) виберемо вирівнювання по верхніх сторонам. Натиснемо «ОК» і виконаємо цю ж команду знову, вибравши на цей раз вирівнювання по ширині і висоті. У результаті ми отримаємо два абсолютно однакових прямокутника, розташовані на одній горизонтальній лінії.


Нагадаємо, що будь-яка зміна в карті поведінки може бути скасовано за допомогою кнопки  на панелі кнопок головного вікна або команди «Редагувати / Скасувати зміни (Alt + Backspace)» головного меню.

Якщо блок сам має внутрішню структуру (на схемі такі блоки показуються подвійною лінією), то для переходу до його внутрішньої схеми потрібно двічі клацнути мишкою на його зображенні або виконати команду «Структура» спливаючого меню.

Для того щоб перейти до визначення класу блоку виконайте команду «Клас» спливаючого меню.

6.2.3. Додавання зв'язку.

Тепер потрібно поєднати ці два локальних блоку зв'язками. Для того щоб додати новий зв'язок, натисніть

кнопку  на лівій панелі інструментів або виконайте команду «Додати новий зв'язок» спливаючого меню.

Підведіть курсор миші до першої зовнішньої змінної, що бере участь у новій зв'язку. Ознакою можливості участі зовнішньої змінної у зв'язку є зміна курсору миші на хрест в колі (наприклад, вхід може брати участь тільки в одній зв'язку) .. Натисніть ліву кнопку миші і, не відпускаючи її, рухайте мишу по напрямку до другої змінної. За курсором

миші буде тягнутися пунктирне зображення зв'язку (Рис 6.4а).

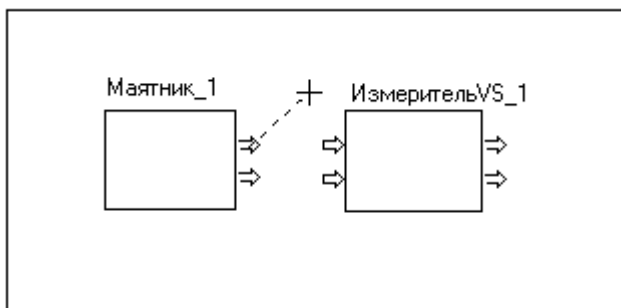


Рис 6.4 а)

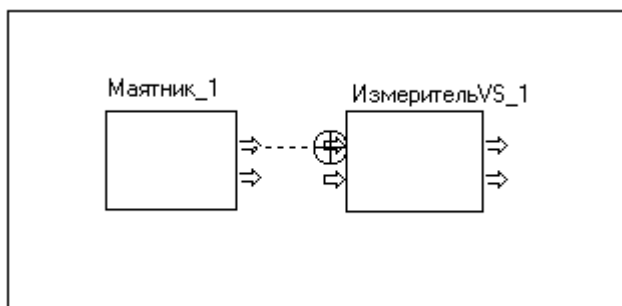


Рис 6.4 б)

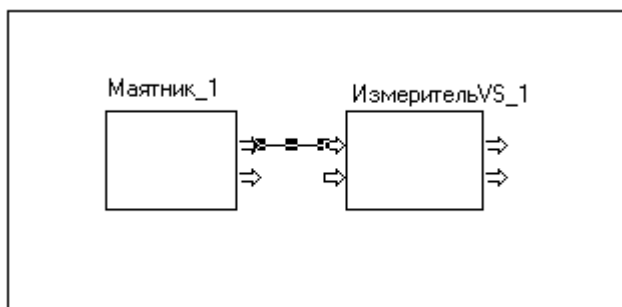


Рис 6.4 в)

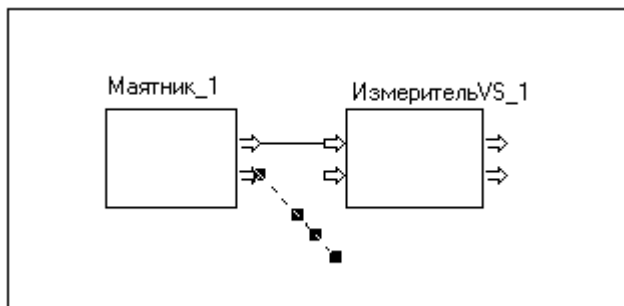


Рис 6.4 г)

Підведіть хрестоподібний курсор миші до зображення другої змінної. У випадку, якщо вона може брати участь у цьому зв'язку, курсор миші зміниться на зображення хреста в колі (Рис 6.4б). Друга змінна не може брати участь у цьому зв'язку, якщо:

- Її спрямованість не відповідає першій змінній (наприклад, ви намагаєтеся з'єднати два входи);
- Це вхід і він вже бере участь у зв'язку;
- Її тип не відповідає типу першої змінної (наприклад, ви намагаєтеся з'єднати вихід типу boolean з входом типу double). Зауважимо, що вектор розмірності [1] або матриця розмірності [1,1] сумісні як один з одним, так і з типом double.

Після цього відпустіть кнопку миші і нова зв'язок готова (Рис 6.4в). Якщо ви відпустіть кнопку миші не на зображенні зовнішньої змінної, то буде створена незавершена зв'язок, яка зображується на структурній схемі пунктирною лінією (Рис 6.4г). Цю зв'язок ви можете зробити завершеною, перемістивши крайню опорну точку на відповідну зовнішню змінну.

6.2.4. Редагування зв'язку.


Щоб видалити зв'язок, виділіть її та виконайте команду «Видалити зв'язок» спливаючого меню.

Нова зв'язок зображується у вигляді прямої лінії, що з'єднує відповідні зовнішні змінні. У даному випадку це цілком підходить, але часто потрібно, щоб зображення зв'язку мало більш складну форму. У загальному випадку зв'язок зображується ламаною лінією, що з'єднує набір опорних точок. Тому для зміни зображення лінії зв'язку необхідно перемістити потрібні опорні точки, додати нові опорні точки або видалити непотрібні.


Щоб змінити лінію зв'язку, потрібно виділити зв'язок, натиснути ліву кнопку миші на квадраті, що зображає одну з опорних точок (не крайніх), потім, не відпускаючи кнопки, переміщати мишу і в необхідному положенні відпустити кнопку. Для точної вказівки координат опорних точок необхідно за допомогою команди «Координати точок» спливаючого меню викликати діалог редагування координат (на Рис 5.12 показаний цей діалог стосовно до виділеного переходу на Рис 5.5).

Зауваження. Не забувайте, що у вікнах MS Windows вісь X спрямована вправо, а вісь Y вниз.

Щоб змінити першу або другу зовнішню змінну лінії зв'язку, потрібно перетягнути мишею відповідну крайню опорну точку на зображення іншої зовнішньої змінної. Зокрема, якщо перетягнути її на вільне поле, можна зробити зв'язок незавершеною.

Щоб додати нову опорну точку на лінію зв'язку, потрібно виділити зв'язок і за допомогою кнопки  або команди «Додати опорну точку» спливаючого меню перевести редактор у режим додавання опорної точки (ознакою цього режиму є поява хрестоподібного курсору миші). Потім слід підвести курсор до потрібної точки лінії зв'язку (курсор

зміниться на хрест з квадратом в центрі) і клацнути лівою кнопкою миші.

Щоб видалити опорну точку з лінії зв'язку, потрібно виділити зв'язок і за допомогою кнопки  або команди «Видалити опорну точку» спливаючого меню перевести редактор у режим видалення опорної точки (ознакою цього режиму є поява хрестоподібного курсору миші). Потім слід підвести курсор до опорної точки, яку слід видалити (курсор зміниться на зображення руки з витягнутим вказівним пальцем) і клацнути лівою кнопкою миші.

6.3. Експеримент з компонентної моделлю.

З'єднуємо зв'язками виходи маятника зі входами вимірвача. Запускаємо модель і отримуємо графіки, показані на Рис 6.5. Багатокомпонентна модель працює.

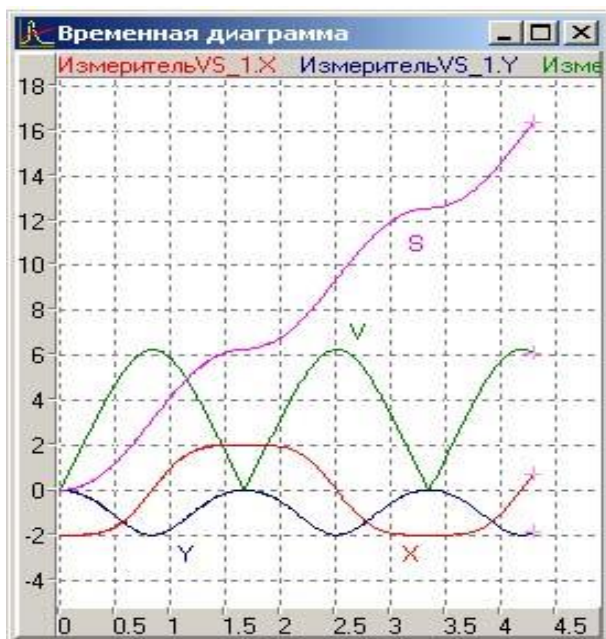


Рис 6.5

Лекція 7. Експорт та імпорт класів.

Припустимо, що ви почали робити нову модель (назвемо цей проект «МодельN») і вам знову знадобився блок обчислення повній швидкості та шляхи по координатах. Подумавши, ви згадуєте, що такий блок вже розроблений у проекті «ІзмерительнаяСистема» і було б непогано запозичити його опис з цього проекту. Може виявитися, що деякий набір блоків постійно використовується в моделях і доцільно створити бібліотеку класів, тобто проект, який не пов'язаний ні з якою моделюється системою (вікно «Віртуальний стенд» порожнє), а лише містить визначення корисних класів, які використовуються в конкретних моделях.

7.1. Експорт класу в інший проект.

Для того, щоб експортувати клас «ІзмерительVS» з проекту «ІзмерительнаяСистема» у проект "МодельN» відкрійте проект «ІзмерительнаяСистема.mvb», виділіте у вікні проекту клас «ІзмерительVS», виконайте команду «Експортувати» спливаючого меню і виберіть у діалозі проект «МодельN . mvb »(Мал. 7.1).

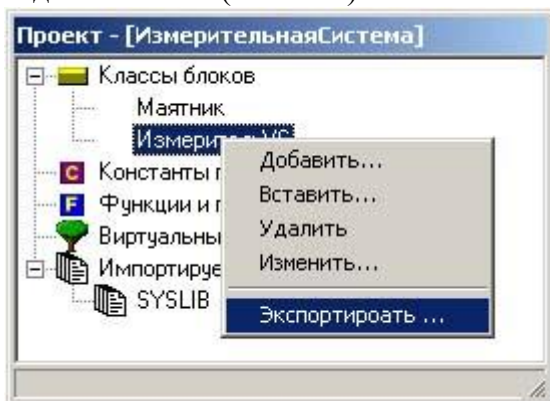


Рис. 7.1

Після цього опис класу «ІзмерительVS» з'явиться в проєкті «МодельN.mvb». Слід зазначити, що операція експорту класу є разовою. Якщо ви надалі змініть визначення класу «ІзмерительVS» у проєкті «ІзмерительнаяСистема.mvb », то це ніяк не відіб'ється на визначенні класу« ІзмерительVS »у проєкті« МодельN.mvb ».

У випадку, якщо ви хочете експортувати клас до бібліотеки класів, підключену до вашого проєкту (перераховані в секції «Імпортовані бібліотеки класів») все ще простіше: перетягніть назва класу на назву бібліотеки і киньте.

7.2. Імпорт класу з бібліотеки класів.

У принципі будь-який проєкт можна використовувати як бібліотеку класів. Для цього його потрібно лише приєднати до вашого проєкту за допомогою команди «Додати» спливаючого меню в секції «Імпортовані бібліотеки класів» вікна проєкту. За замовчуванням до будь-якого проєкту завжди приєднана стандартна бібліотека SysLib, містить найбільш поширені стандартні блоки (інтегрування, диференціювання, насичення і Т.Л.). Перелік класів, що містяться в бібліотеці, можна отримати за допомогою подвійного клацання мишею на її назві у вікні проєкту або команди «Відкрити» спливаючого меню (Мал. 7.2).

За допомогою подвійного клацання мишею на назві класу бібліотеки можна викликати вікно довідки для цього класу. Бібліотечні класи можна використовувати при побудові структурних схем зовсім так само, як і класи, визначені в проєкті, тобто хапати їх мишкою у вікні бібліотеки, тягнути в вікно структурної схеми і там кидати в потрібному місці ..Наприклад, якщо нам потрібно в проєкті «Ізмерительная Система» отримати значення повного прискорення, ми можемо використовувати стандартний клас CDifferentiator з бібліотеки SysLib (Мал. 7.3).

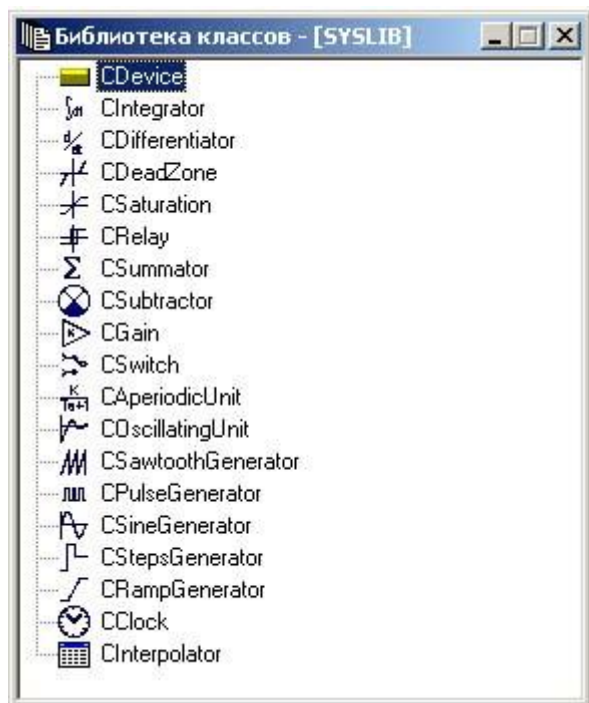


Рис. 7.2

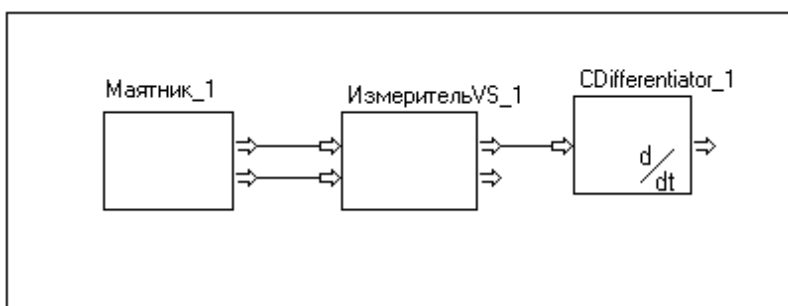


Рис. 7.3

На відміну від експорту, імпорт класу встановлює постійний зв'язок даного проекту з бібліотекою: якщо при завантаженні вашого проекту буде виявлено, що якась бібліотека, використовувана в проекті, оновлена, вам буде запропоновано оновити визначення імпортованих з неї класів.

7.3. Створення власної бібліотеки класів.

Як вже зазначалося вище, будь-який проект може розглядатися як бібліотека класів. У «правильній» бібліотеці повинен бути порожнім віртуальний стенд і повинна бути підготовлена довідкова система. Ви можете також створити для вашої бібліотеки Help-систему, підготувавши файли довідника і, якщо потрібно, змісту за допомогою звичайних засобів створення Help для Windows. Ці файли повинні знаходитися в тій же директорії, що і ваша бібліотека, і мати такі ж імена файлів (наприклад, для бібліотеки `syslib.mvb` поруч повинні знаходитися файли `syslib.hlp` і `syslib.cnt`). Кожному класу у вашій бібліотеці повинен відповідати розділ (`topic`) у файлі `*.HLP`, який повинен мати той же ідентифікатор, що й клас (наприклад, класу `CIntegrator` відповідає розділ `CINTEGRATOR`). У цьому випадку користувач зможе за допомогою подвійного натискання кнопки миші на виділеному класі у вікні експортованих класів вашої бібліотеки викликати відповідний розділ Help. Якщо файл `*.HLP` відсутній, виводиться текст коментаря даного класу (див. вікно менеджера класу).

ПРАКТИЧНІ РОБОТИ

8.Моделювання складних динамічних систем у прикладах.

У даній роботі робиться спроба проілюструвати основні поняття моделювання складних динамічних систем на спеціально підібраних характерних прикладах. Спочатку розглядаються моделі ізольованих систем і вся увага концентрується на описі поведінки. Послідовно обговорюються моделі безперервних, дискретних і безперервно-дискретних - гібридних - систем. Потім розглядаються компонентні моделі, що складаються з окремих блоків. Одночасно «поволі» розвивається ідея об'єктно-орієнтованого моделювання.

Практична робота 1

8.1. Моделі ізольованих систем.

Ізольованою називається система, що не взаємодіє з зовнішнім оточенням. Таким чином, модель ізольованої системи повинні включати в себе як опис системи, що вивчається, так і опис зовнішнього оточення.

8.1.1. Моделі математичного маятника.

У цьому розділі ми розглянемо декілька послідовно ускладнюються моделей математичного маятника: простий маятник, маятник з підштовхує силою, маятник з пружиною, що відриваються маятник.

Простий математичний маятник.

Дана модель є прикладом моделі чисто безперервної ізолюваної системи.

Модельована система являє собою матеріальну точку (ми будемо представляти її як кулька досить малого розміру), прикріплену до нерозтяжного і невагомому стрижня завдовжки, інший кінець якого шарнірно закріплений на початку системи координат (див. Рис 1).

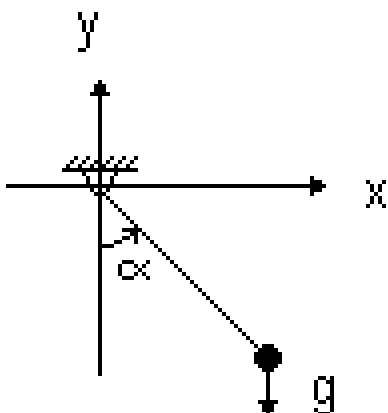


Рис 1

Стан маятника повністю визначається значенням двох змінних: кута відхилення і кутовий швидкістю.

Динаміка маятника визначається двома диференціальними рівняннями (1).

$$\begin{cases} \frac{d\alpha}{dt} = \omega \\ \frac{d\omega}{dt} = \frac{-g \cdot \sin \alpha}{L} \end{cases} \quad (1),$$

де

$$\alpha_0 = -\frac{\pi}{2} \quad \omega_0 = 0.$$

При чисельному моделюванні ми будемо вважати.

$$\alpha(0) = \alpha_0, \omega(0) = \omega_0.$$

Однак, для анімації руху маятника будуть потрібні додаткові змінні - координати та матеріальної точки, що задаються двома формулами (1а)

(1а)

$$\begin{cases} x = L \cdot \sin \alpha \\ y = -L \cdot \cos \alpha \end{cases}$$

На Рис 2б показана траєкторія руху маятника у координатах. Залежності показані на Рис 2а. Моделювана система здійснює незгасаючі коливання. На Рис 2в показана фазова траєкторія маятника в системі координат.



Рис 2 а)

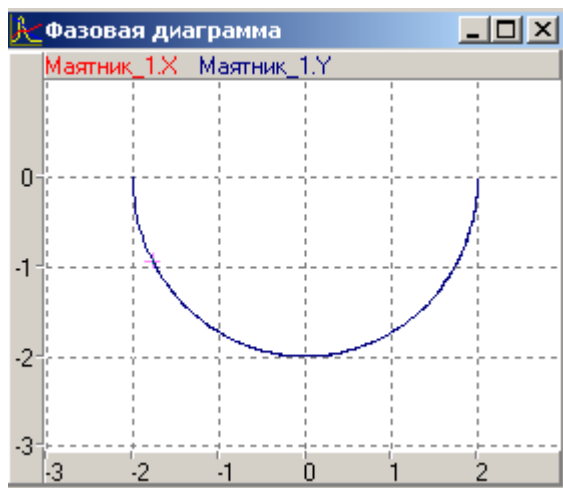


Рис 2 б)

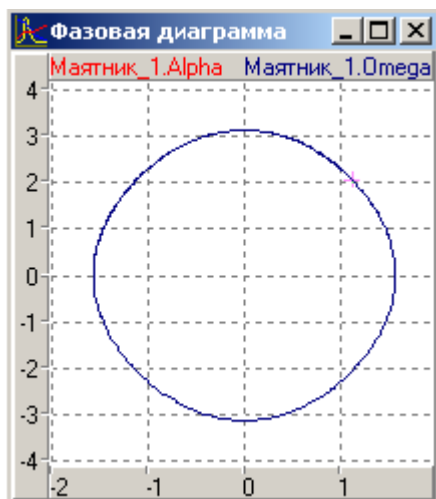


Рис 2 в)

Сукупність змінних, що визначають стан динамічної (тобто змінюється в часі) системи, називають фазовим вектором, а область його зміни - фазовим простором. Набір початкових значень визначає початкову точку, що відповідає моменту часу. При зміні кінець фазовий вектор визначає послідовність точок, звану фазовою траєкторією. Фазовий простір з додатковою координатою - часом - називають розширеним фазовим простором. Графіки, показані на Рис 2, є різними двовимірними проекціями траєкторії системи в розширеному фазовому просторі.

Об'єктний підхід.

Функціонування маятника визначається сукупністю системи рівнянь (1) і визначень параметрів L, g, α_0, ω_0 і змінних. Однак, це опис задає не один якийсь конкретний маятник, а будь-який, абсолютно довільний, тобто задає цілий клас маятників. Будь-який конкретний маятник є екземпляром класу «Маятник». Ніщо не заважає вам побудувати модель, що включає, скажімо, три маятника. Опис їх буде абсолютно однаковим, проте, значення змінних в якийсь момент можуть бути різними в залежності від конкретних значень параметрів. L, g, α_0, ω_0 Об'єктом прийнято називати деяку сутність, що інкапсулює у собі дані і методи як єдине ціле і взаємодіє із зовнішнім оточенням через певний інтерфейс. З поняттям об'єкта тісно пов'язане ставлення подвійності - «клас-екземпляр». Примірник об'єкта може мати унікальні значення параметрів, які встановлюються тільки один раз при створенні об'єкта (можна сказати, що параметр - це константа, яка може мати різне значення для різних. Примірників). Ясно, що наша динамічна система цілком вписується у визначення об'єкта.

Розрізняють об'єкти пасивні та активні. Пасивні об'єкти (велика частина об'єктів в програмах) тільки «відгукуються» на виклики методів та повідомлень ззовні,

але самі нічого не роблять, тобто не можуть змінювати значення своїх даних з власної ініціативи. Активні об'єкти (наприклад, екземпляри класу Thread в мові Java) мають свою власну «нитку управління» і функціонують незалежно від інших об'єктів і паралельно з ними. У UML пропонується задавати функціонування активного об'єкта з допомогою карти станів, у якій вузлам відповідають деякі діяльності, протяжні в часі, а переходах - миттєві реакції на зовнішні і внутрішні події. Динамічна система безумовно є активним об'єктом, але особливим об'єктом, оскільки вони змінюють значення своїх змінних безперервно, в той час як в UML діяльність все ж передбачає наявність потоку керування, який виконує деяку послідовність дій (швидше за все циклічну) паралельно і незалежно від інших потоків. Будемо називати такі об'єкти активними динамічними.

Практична робота 2.

Математичний маятник з підштовхуючою силою.

Дана модель є прикладом моделі ізольованої гібридної системи з гібридним поведінкою найпростішого типу - миттєві скачки значень змінних при незмінних системі рівнянь і наборі змінних.

Припустимо, що наш маятник отримує миттєве збільшення кутової швидкості з відповідним знаком щоразу при проходженні через нижнє вертикальне положення. Для опису такого маятника системи рівнянь (1) недостатньо: необхідно крім вирішення цієї системи рівнянь ще ловити дискретне подія - перехід ліворуч або праворуч нижнього вертикального положення - і з цієї події виконувати миттєве дискретне дія - стрибком змінювати значення змінної на величину M . Мовою гібридних карт станів це означає, що ми повинні виконувати рішення системи

рівнянь (1) в стані «Коливання», а за вказаною дискретною події виконати перехід в той же самий стан (Мал. 1). У діях цього переходу буде вироблено миттєве збільшення кутової швидкості і далі буде продовжено рішення системи рівнянь (1) з новими початковими умовами.

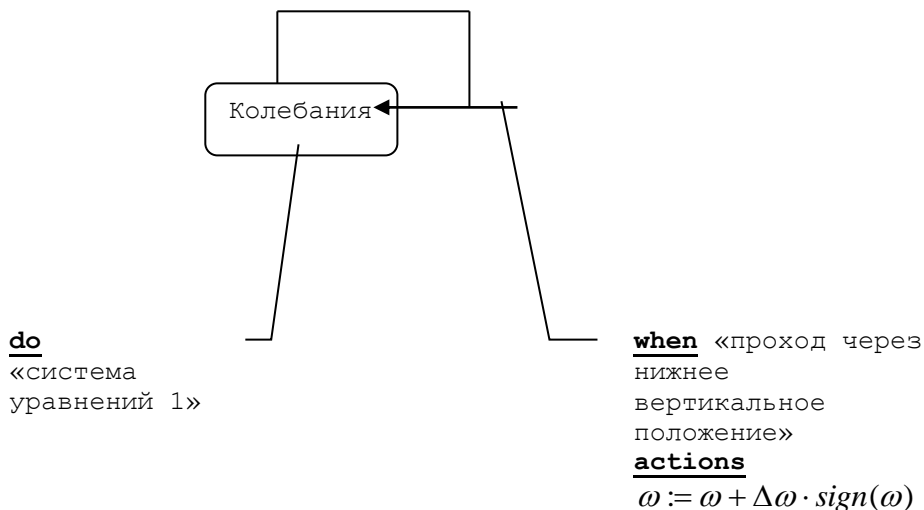
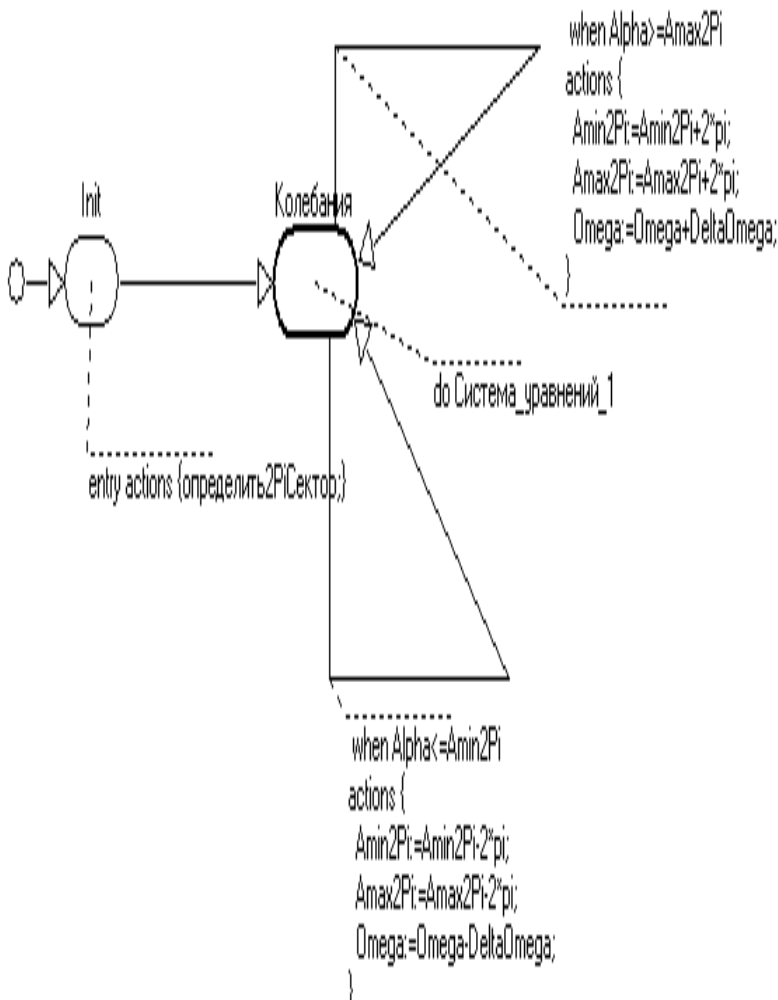


Рис. 1

Тепер виникає питання: як поставити подію «прохід через нижнє вертикальне положення»? Перше, що спадає на думку, це задати його логічним предикатом, $\alpha = 2 \cdot \pi \cdot n$, де n - ціле число. Математично це абсолютно правильне визначення, проте не цілком придатне для чисельного моделювання. Події, задані строгими равенствами речових виразів - це дуже неприємний сюрприз для алгоритмів пошуку точок перемикання в гібридних моделях. Тому ми



використовуємо інше, більш конструктивне визначення події «прохід через нижнє вертикальне положення». Розіб'ємо всі безліч значень кута α на сектори розміром 2π . Для будь-якого поточного значення α ми можемо вказати сектор, якому належить це значення, а також нижню і верхню межі цього сектора $A_{\min} 2\pi$ і $A_{\max} 2\pi$. Тоді виконання умови $\alpha \geq A_{\max} 2\pi$

Рис 2

буде означати перехід в наступний сектор, а виконання умови $\alpha \leq A \min 2Pi$ - повернення в попередній сектор. В результаті ми отримуємо гібридну карту станів, показану на Рис. 2.

При переході з одного 2π -сектору в інший відповідно зміщуються нижня і верхня межі. Початкове визначення поточного 2π -сектору проводиться у вхідних діях стану Init шляхом виклику процедури «определіть2PiСектор».

```

procedure определіть2PiСектор is
N_2Pi: integer;
begin
N_2Pi: = abs (trunc (Alpha / (2 * pi)));
if Alpha > 0 then
Amin2Pi: = 2 * pi * N_2Pi;
Amax2Pi: = 2 * pi * (N_2Pi + 1);
else
Amax2Pi: = - 2 * pi * N_2Pi;
Amin2Pi: = - 2 * pi * (N_2Pi + 1);
end if;
end определіть2PiСектор;

```

Залежності $\alpha(t)$, $\omega(t)$ показані на Рис 3а. Внаслідок впливу підштовхує сили амплітуда коливань збільшується і на другому циклі маятник «перевалює» через верхнее вертикальне положення після чого починає крутитися проти годинникової стрілки зі зростаючою кутовий швидкістю. Траєкторія руху показана на Рис 3б. Фазова траєкторія (α, ω) в координатах показана на Рис 3в.

Розглянемо уважно збільшений фрагмент графіка $\omega(t)$ на Рис 3г. На інтервалі $1 [0, t^*]$ моделєма система веде себе як класична безперервна динамічна система, поведінка якої визначається системою рівнянь (1) і значення змінної ω змінюється безупинно. На правому кінці цього інтервалу в момент $t^* = 0.837$ $\omega = \omega_1^*$. У момент t^* спрацьовує перехід і

змінна ω стрибком набуває значення ω_2^* . Далі на інтервалі $2 [t^*, \dots]$ система знову веде себе як безперервна. Місце «склеювання» двох сусідніх безперервних інтервалів в гібридному моделюванні називають «тимчасової щілиною». Всередині «тимчасової щілини» безперервний час зупиняється і починає просуватися локальне дискретний час. Моменти дискретного часу зазвичай співвідносять з натуральним рядом чисел (тобто просто нумеруються $1, 2, \dots, n$). Один момент дискретного часу відокремлюється від іншого дискретним подією - спрацьовуванням якого-небудь переходу. У даному випадку ми маємо два значення дискретного часу - момент 1 до спрацьовування переходу і момент 2 після спрацьовування переходу. Всі моменти локального дискретного часу «тимчасової щілини» одночасні в безперервному часу. Пару (t, n) можна розглядати як гібридне час. Таким чином, якщо в безперервному часі функція є багатозначною в точці, t^* то в гібридному часі вона однозначна: $\omega(t^*, 1) = \omega_1^*$ і $\omega(t^*, 2) = \omega_2^*$. У більш складних моделях під «тимчасової щілини» може послідовно миттєво спрацьовувати більше одного переходу.

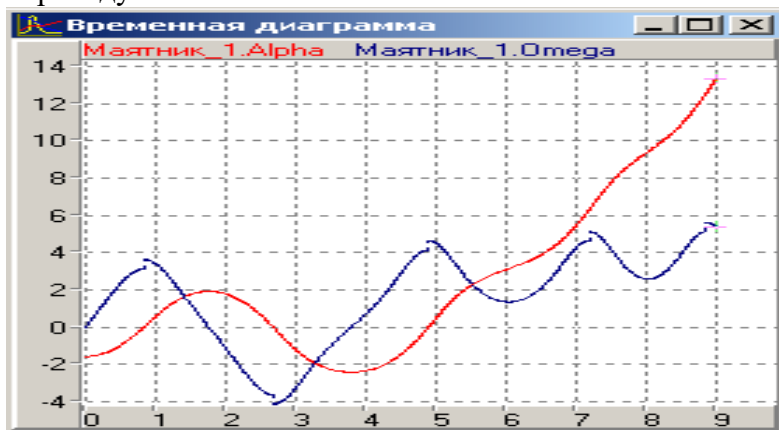


Рис 3 а)

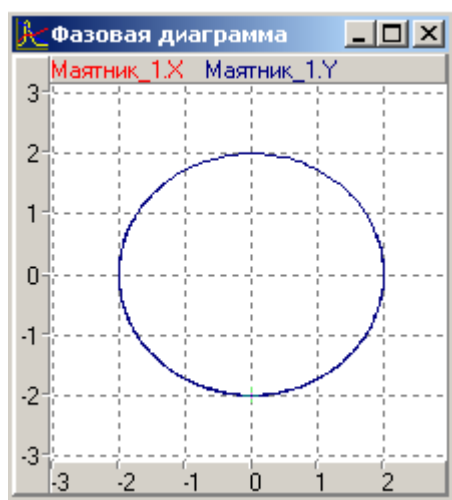


Рис 3 б)



Рис 3 в)

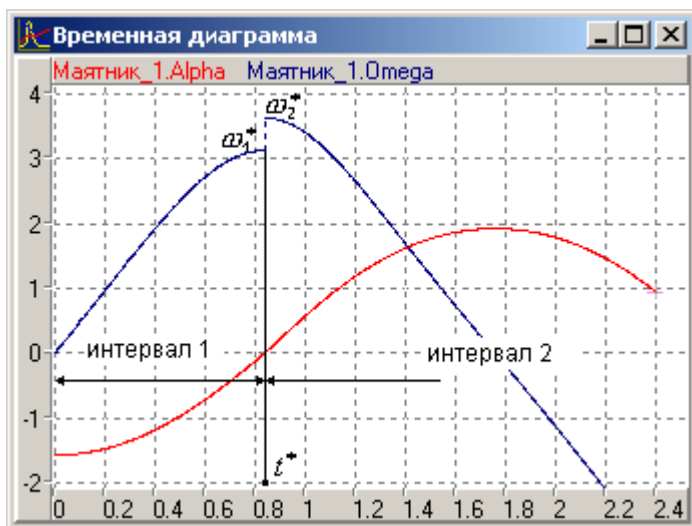


Рис 3 г)

Об'єктний підхід.

Давайте розглянемо більш уважно, як ми зробили цю модель. Ми «взяли за основу» вже наявну модель простого маятника і модифікували її. Модифікація полягала в наступному:

- Додані нові змінні;
- У головній карті поведінки доданий вузол «Коливання» і нові переходи;
- «Система_уравнений_1" вилучено з вузла «Init» і поміщена у вузол «Коливання»;
- У вузол «Init» додано вхідні дії.

Всі ці дії цілком укладаються в традиційну схему спадкоємства в об'єктно-орієнтованому підході: клас об'єкта, який береться за основу, називається базовим класом (батьківським класом, класом-предком,

суперкласом), клас, який виходить в результаті модифікації цієї основи, називається похідним класом (класом-спадкоємцем, класом-нащадком, підкласом). При цьому модифікація зводиться до додавання нових елементів опису і перевизначення існуючих елементів (в існуючому сайті «Init» порожня послідовність вхідних дій заміщена на непустих і, навпаки, непорожнє локальне поведінка замінено на порожнє).

Таким чином ми можемо побудувати цю модель і так: на основі класу «Маятник» побудувати похідний клас «МаятникСПодталкуванням». Від способу «копіювання і модифікація» цей спосіб відрізняється тим, що в силу відносини спадкування будь-яка зміна в базовому класі (наприклад, модифікація системи рівнянь) буде негайно і автоматично відобразитися в похідному класі (до речі, це може і породжувати конфлікти). Таким чином, якщо в опис найпростішого маятника вводиться уточнення (наприклад, облік опору повітря), то це уточнення автоматично пошириться на всі маятники - спадкоємці найпростішого.

Практична робота 3.

Математичний маятник з односторонньою пружиною.

Дана модель є прикладом моделі ізольованої гібридної системи з гібридним поведінкою другого, більш складного типу - змінюються праві частини рівнянь при незмінних наборі рівнянь і наборі змінних

Припустимо, що маятник постачений пружиною, яка працює тільки в області негативних значень кута. Тоді рівняння руху маятника приймуть вигляд

$$(2) \begin{cases} \frac{d\alpha}{dt} = \omega \\ \frac{d\omega}{dt} = \frac{-g \cdot \sin \alpha}{L} + T \\ T = \text{if } \alpha \geq 0 \text{ then } 0 \text{ else } -K \cdot \alpha \end{cases}$$

Рівняння, що містять вирази виду, називаються умовними або гібридними. Таке рівняння неявно визначає два дискретних події, пов'язаних з перемиканням гілок умовного вираження. Будь-які гібридні рівняння можна представити у вигляді гібридної карти стану (але не навпаки). Наприклад, гібридну систему рівнянь (2) можна представити картою станів, зображеної на Рис 4.

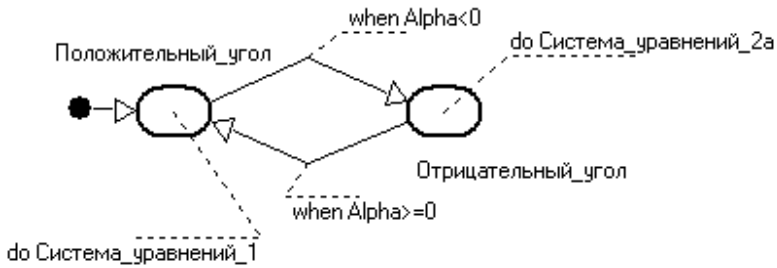


Рис 4

де

$$(2a) \begin{cases} \frac{d\alpha}{dt} = \omega \\ \frac{d\omega}{dt} = \frac{-g \cdot \sin \alpha}{L} + T \\ T = \text{if } \alpha \geq 0 \text{ then } 0 \text{ else } -K \cdot \alpha \end{cases}$$

Звичайно користувачеві буває простіше і природніше написати систему рівнянь (2) ніж малювати еквівалентну карту станів. Тому пакети гібридного моделювання

звичайно включають умовні рівняння як допустимих конструкцій вхідної мови і визначають моменти відповідних дискретних подій (точки перемикання) автоматично.

Зауваження. Користувачі, які не користуються спеціальними інструментами моделювання, а безпосередньо розробляють свою модель як прикладну програму мовою програмування з використанням готових чисельних бібліотек, часто кодують системи рівнянь типу (2) прямо в умовні оператори, не піклуючись ні про яких точках перемикання. Результат зазвичай виходить правдоподібним. Однак, чисельні методи інтегрування звичайно припускають гладкість шуканої функції і її похідних. Тому, хоча більшість чисельних методів справляються з розривами значень похідних, точність результату може не відповідати заданій.

Залежності $\alpha(t)$, $\omega(t)$ показані на Рис 5а. Траєкторія руху показана на Рис 5б. Траєкторія несиметрична, оскільки в початковому положенні пружина стиснута і потенційна енергія маятника більше, ніж у моделі 1. Фазова траєкторія в координатах показана на Рис 5в.

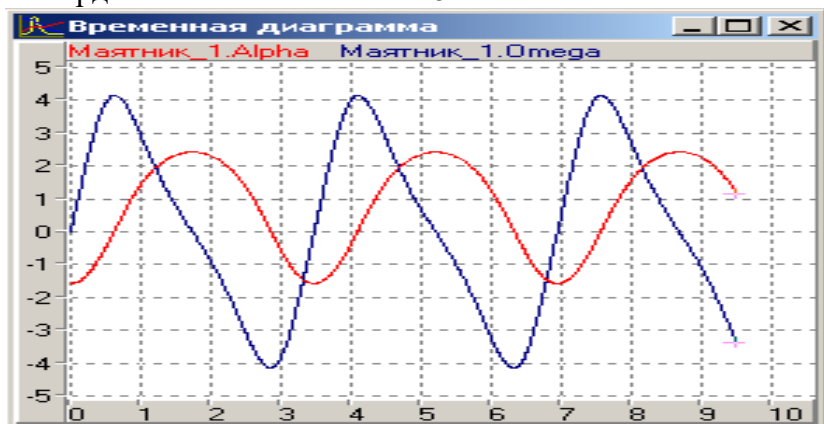


Рис 5 а)

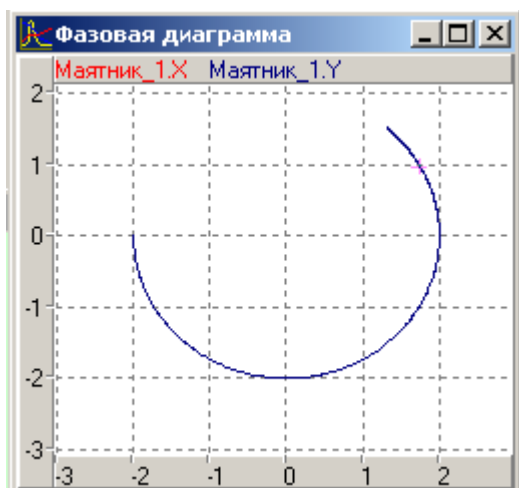


Рис 5 б)

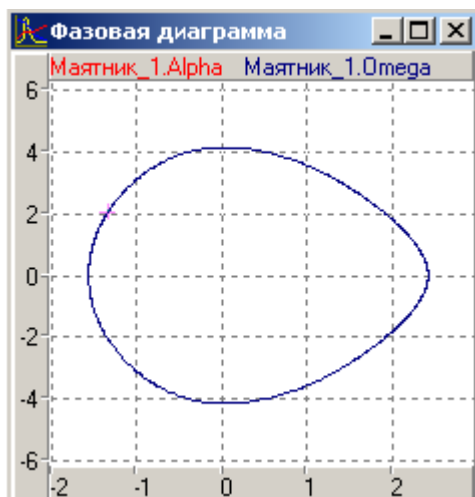


Рис 5 в)

Практична робота 4.

8.1.2.

Відривається маятник.

Дана модель є прикладом моделі ізольованої гібридної системи з гібридним поведінкою третього, найскладнішого типу - змінюються м система рівнянь, і набір змінних.

Нехай в деякий момент t^* (наприклад, визначається умовою $\alpha \geq \alpha_{\max}$) кріплення кульки до стрижня руйнується й далі кулька продовжує свою незалежну від стрижня рух. Рух кульки після відриву задається системою рівнянь

$$(3) \begin{cases} \frac{dX}{dt} = V_x \\ \frac{dY}{dt} = V_y \\ \frac{dV_y}{dt} = -g \end{cases}$$

Гібридна карта станів для відривається маятника показана на Рис 6. Система має два основних якісних стани: «Коливання» і «Свободний_полет». Крім того, передбачена зупинка експерименту, коли кулька відлетить нижче деякого y_{\min} - перехід в кінцевий стан «Зупинка». У стані «Коливання» розв'язується система рівнянь (1) та визначаються значення змінних. (α, ω, x, y) У момент відриву відбувається перехід в стан «Свободний_полет», в якому розв'язується система рівнянь (3) та визначаються значення змінних (x, y, V_x, V_y) . Початковими значеннями змінних x і y в системі рівнянь (3) стають їх поточні значення на момент часу t^* , а початкові значення лінійних

швидкостей V_x і V_y , які не входили в систему рівнянь (1), визначаються у миттєвих діях переходу.

У вхідних діях стану «Зупинка» викликається зумовлена процедура, *Stop* яка припиняє просування модельного часу. Якби ця процедура не викликала, то після переходу в стан «Зупинка», оскільки цього стану не приписано ніякої системи рівнянь або карти станів, значення всіх змінних будуть залишатися незмінними, а змінюватися буде тільки значення модельного часу.

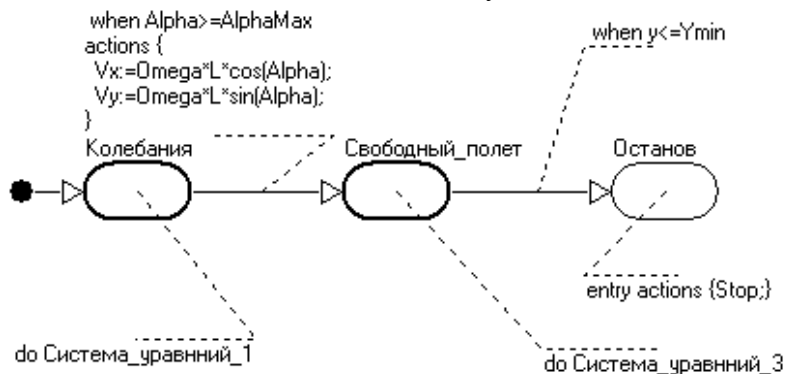


Рис 6

На Рис 7 показана траекторія руху маятника при.

$$\alpha_{max} = \frac{\pi}{4} \quad y_{min} = -3$$

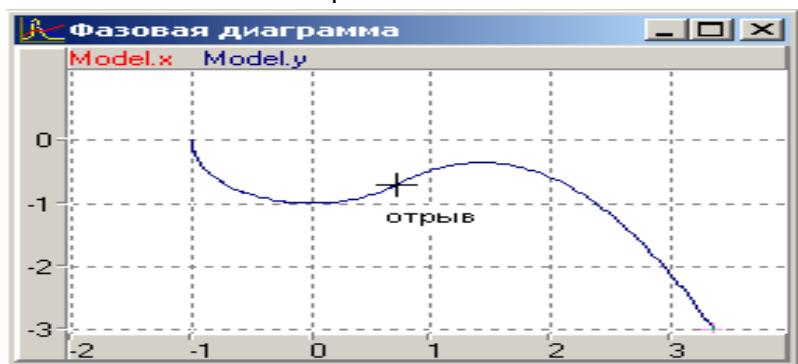


Рис 7

Практична робота 5.

8.1.3. Стрибаючий м'ячик.

Ця модель розглядається практично завжди, коли мова заходить про гібридні системи, тому ми не будемо виділятися і поступимо як усі.

Абсолютно пружний м'ячик (матеріальна точка) відпускається на висоті H над абсолютно твердою горизонтальною площиною.

Рух м'ячика в полі тяжіння.

Це безперервна ізольована система. Рух описується системою рівнянь (1)

$$(1) \begin{cases} \frac{dy}{dt} = V_y \\ \frac{dV_y}{dt} = -g \end{cases} \cdot y(0) = H, \quad V_y(0) = 0$$

де y - вертикальна координата м'ячика відносно площини, V_y - вертикальна швидкість м'ячика.

На Рис 8 показані тимчасова (а) і фазова (б) діаграми (модельовання зупинено за умовою) $y \leq 0$.

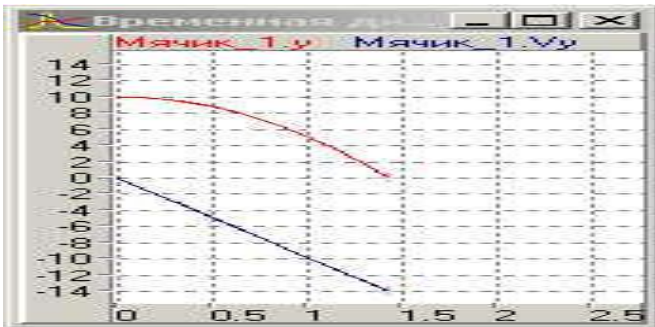


Рис 8 а)

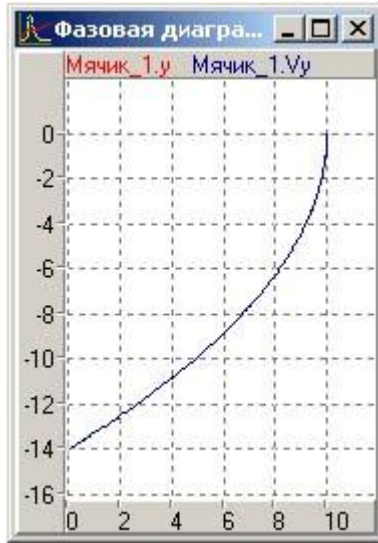


Рис 8 б)

Практична робота 6

Відскокує м'ячик.

При миттєвому абсолютно пружному ударі швидкість V_y стрибком змінює знак. Відповідна карта поведінки показана на Рис 9. Після відскоку продовжує вирішуватися

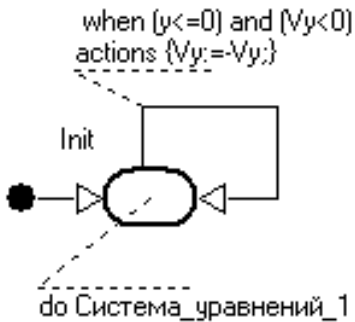


Рис 9

та ж сама система рівнянь (1) з новими початковим умовами. Тимчасова (а) і фазова (б) діаграми показані на Рис 10

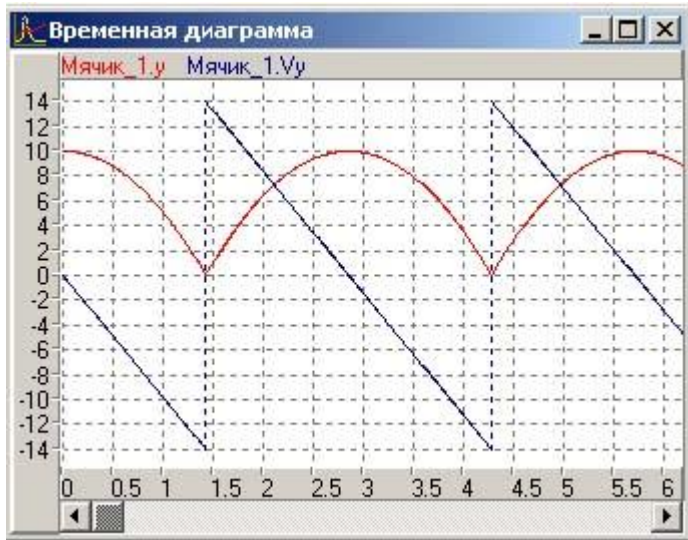


Рис 10 а)

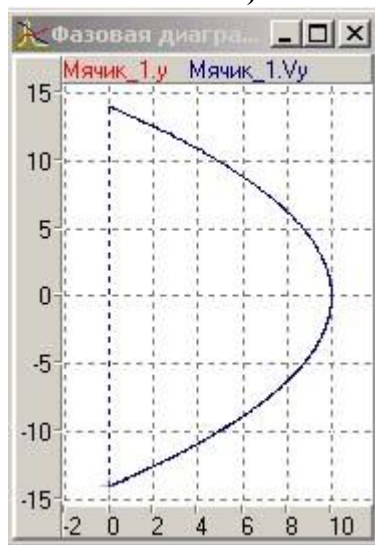


Рис 10 б)

Практична робота 7. М'ячик, що падає на пружину.

Нехай тепер м'ячик падає на вільний кінець пружини довжиною H_s , закріпленої вертикально на площині (Рис 11).

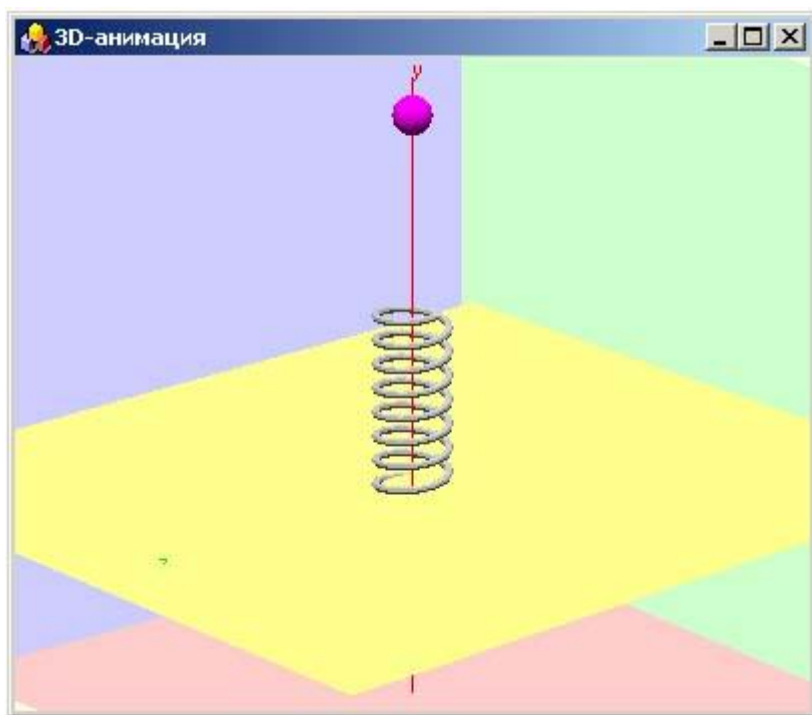


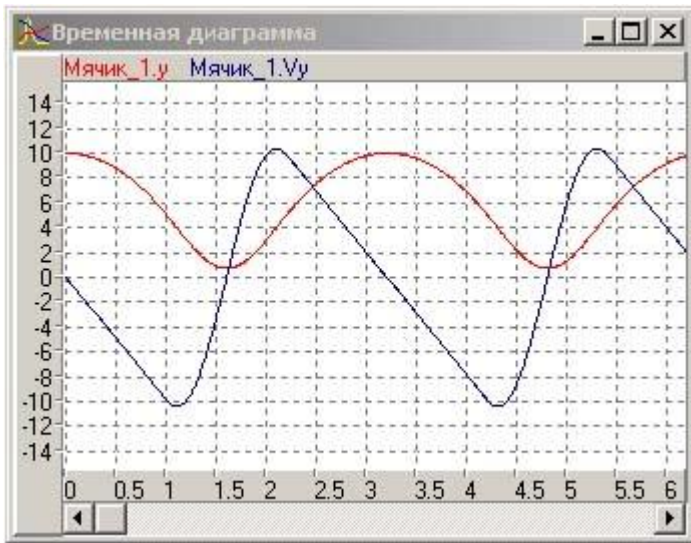
Рис 11

Система рівнянь (1) трансформується в систему рівнянь

$$(1a) \begin{cases} \frac{dy}{dt} = V_y \\ \frac{dV_y}{dt} = \text{if } y > H_s \text{ then } -g \text{ else } K \cdot (H_s - y) - g \\ y_s = \text{if } y > H_s \text{ then } H_s \text{ else } y \end{cases}$$

де y_s - координата вільного кінця пружини (необхідна виключно для анімації руху).

У залежності від коефіцієнта жорсткості пружини удар м'ячика про площину може не відбуватися



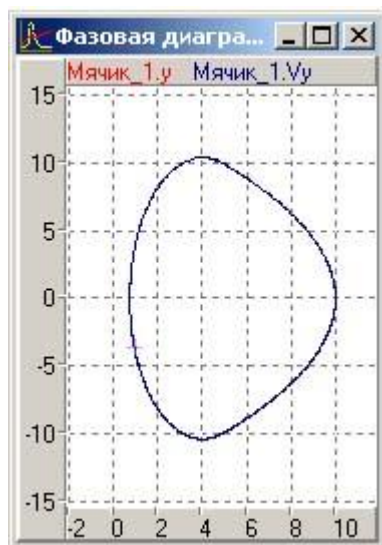


Рис 12

або відбуватися (Рис 13).



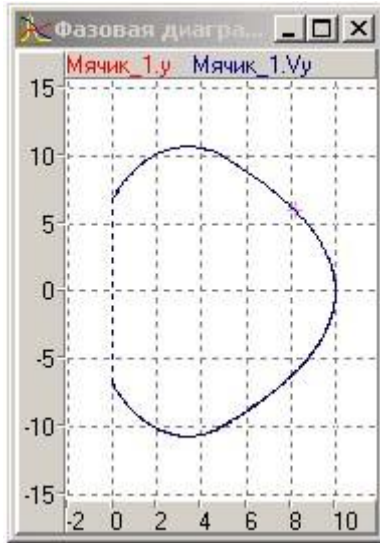


Рис 13

Практична робота 8.

8.1.4. Генератор прямокутних імпульсів.

Дана модель є прикладом суто дискретної ізольованої системи.

Нехай речова змінна Y приймає значення $+1$ в протягом частки періоду T і значення -1 в долю періоду $1 - K_p$, що залишився. Такий закон зміни можна реалізувати за допомогою карти станів, наведеної на Рис 14.

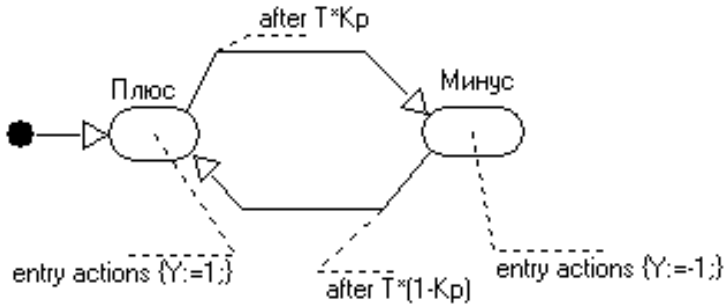


Рис 14

Відповідні привласнення змінній Y знаходяться у вхідних діях станів «Плюс» і «Мінус». Ці миттєві дії виконуються щоразу, коли даний стан (вузол графа) стає поточним. Переходи з одного стану в інший спрацьовують через відповідні інтервали часу (величина K_p може бути і змінною!).

На Рис 15а показана залежність $Y(t)$ при $K_p = 0.5$, а на Рис 15б - при $K_p = 0.75$.



Рис 15 а)

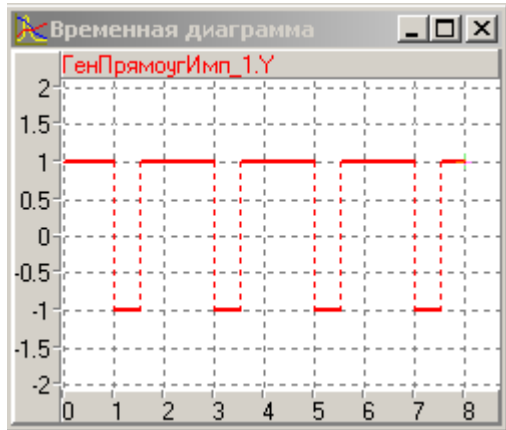


Рис 15 б)

Практична робота 9. 8.2. Компонентні моделі.

Представлення модельованої системи як ізольованою не завжди зручно. По-перше, розмірність сукупної системи рівнянь або карти поведінки може бути дуже великий і її ручне складання може бути процесом дуже трудомістким і чреваті помилками. По-друге, складна технічна або природна система як правило має свою природну структуру, яку було б надзвичайно корисно зберегти в моделі для полегшення її розуміння. Нарешті, окремі фрагменти поведінки складної системи можуть бути вже давно відпрацьовані на моделях більш простих систем і вельми хотілося б використовувати ці готові напрацювання. Тому моделі складних систем практично завжди є компонентними.

Компонентна модель представляється у вигляді сукупності незалежно і паралельно функціонують блоків або пристроїв, які взаємодіють між собою через функціональні

зв'язки. Зв'язки можуть з'єднувати між собою зовнішні змінні блоків (сполучаються змінні, звичайно, повинні збігатися за типом значення). Внутрішні змінні блоку недоступні ззовні. Ізольована система є блоком, у якого всі змінні внутрішні.

Ми будемо розглядати в цьому розділі тільки орієнтовані блоки і зв'язку. Орієнтованим блоком є блок, всі зовнішні змінні якого можна розбити на входи і виходи. Значення входу не може бути змінено всередині блоку, а значення виходу, навпаки, може бути змінена тільки всередині блоку. Орієнтована зв'язок може з'єднувати вихід одного блоку і вхід іншого чи блоку. Один вихід може бути з'єднаний з будь-якою кількістю входів.

8.2.1. Система визначення лінійної швидкості та шляхи маятника.

Припустимо, що нам чогось знадобилася інформація про значеннях лінійної швидкості маятника і повному пройденім шляхом. Якщо це робити безпосередньо в описі маятника, то знадобляться всього лише додаткові рівняння

$$\begin{cases} V = L \cdot \omega \\ \frac{dS}{dt} = V \end{cases}$$

, де V - лінійна швидкість, S - повний шлях.

Проте, може виявитися, що такий визначник швидкості та шляху корисний не тільки для маятників, а для будь-яких рухомих в площині об'єктів. Тому ми створимо новий клас «Измеритель VS», функціонування якого визначається системою рівнянь

$$\left\{ \begin{array}{l} V = \sqrt{V_x^2 + V_y^2} \\ V_x = \text{deriv}(x) \\ V_y = \text{deriv}(y) \\ \frac{dS}{dt} = V \end{array} \right.$$

де x, y - поточні координати вимірюваного об'єкта, deriv - функція диференціювання аргументу. Таким чином, новий блок також є чисто безперервним.

Однак, він не може бути ізольованою системою, точно так само, як і вимірюваний об'єкт - поточні координати цього об'єкта як-то повинні передаватися на блок виміру. Тому змінні x, y і в блоці «ИзмерительVS», і в блоці «Маятник» повинні бути оголошені як зовнішні і між ними повинні бути створені зв'язки. У даному випадку спрямованість зв'язків не викликає сумніву: змінні повинні бути оголошені як виходи в блоці «Маятник» (Рис 16а) і як входи в блоці «ИзмерительVS» (Рис 16б). Модельована система включає в себе по одному примірнику того й іншого блоків, з'єднані двома зв'язками (Рис 16в).

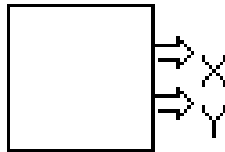


Рис 16 а)

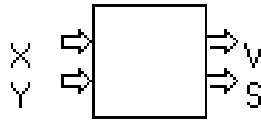


Рис 16 б)

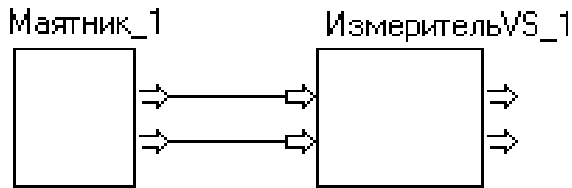


Рис 16.в)

Тимчасова діаграма для швидкостей показана на Рис 17.

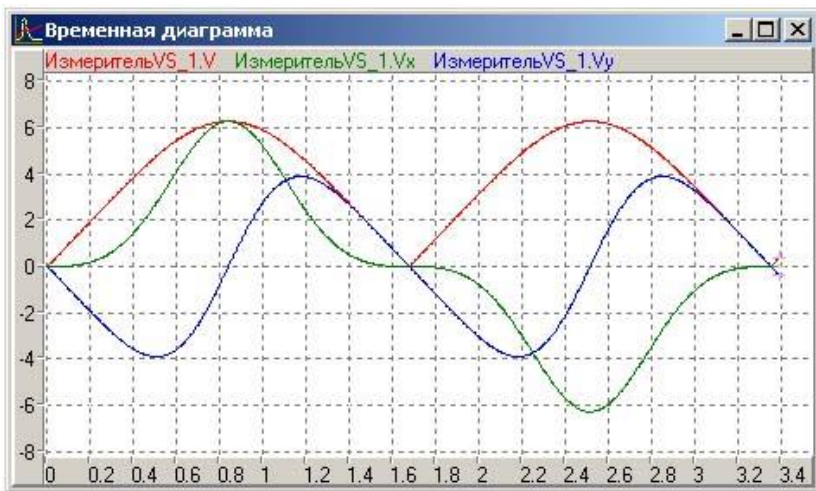


Рис 17

Яким же чином функціонує компонентна модель? Блоки функціонують паралельно в модельному часі. Це означає, що їхні головні карти станів виконуються як паралельні дискретні процеси, а поточні безперервні поведінки - системи рівнянь - об'єднуються в одну глобальну систему рівнянь з урахуванням зв'язків, які розглядаються як додаткові рівняння. У даному випадку глобальна система рівнянь виходить простим механічним складанням системи рівнянь руху маятника і системи рівнянь вимірювача, до яких додаються два рівняння зв'язків:

$$\text{Измеритель}VS_1.X = \text{Маятник_1}.X$$

$$\text{Измеритель}VS_1.Y = \text{Маятник_1}.Y$$

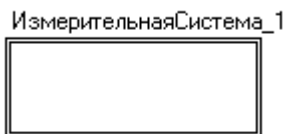


Рис 18 а)

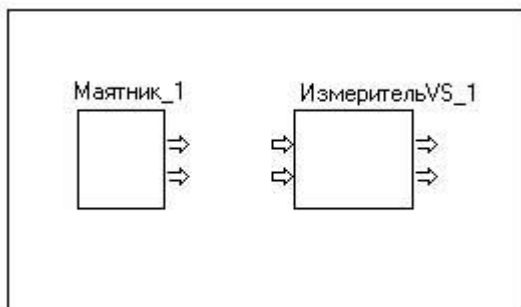


Рис 18. б)

*Измеритель*VS_1.X = *Маятник*_1.X

*Измеритель*VS_1.Y = *Маятник*_1.Y

Слід зазначити, що функціональні зв'язку - це всього лише зручний і наочний спосіб запису рівнянь взаємодії блоків. Замість схеми на Рис 16в можна створити блок-контейнер класу «Измерительная_система» (Рис 18а), всередину нього помістити як локальні блоки маятник і вимірювач без жодних зв'язків (Рис 18б), а в якості безперервного поведінки блоку-контейнера помістити ті ж рівняння зв'язку

Це можливо, оскільки зовнішні змінні локальних блоків видимі в описі охоплює блоку-контейнера. Таким чином, локальні блоки можуть взаємодіяти і крім явних зв'язків «по повітрю» - через рівняння блоку-контейнера. Іноді це може бути зручно. Наприклад, в задачі про взаємодію N заряджених частинок кожна частка пов'язана з кожною, тобто число зв'язків пропорційно N^2 . При досить великому N малювання функціональних зв'язків стає скрутним (ми вважаємо, що N постійна, тобто розглядається система з статичною структурою). У даному випадку набагато зручніше пов'язати з охоплює блоком-контейнером сумарне електричне поле, напруженість якого в будь-якій точці і буде обчислюватися в залежності від сукупного положення частинок в системі рівнянь блоку-контейнера.

Зауваження.

Слід пам'ятати, що глобальна система рівнянь виходить простим механічним складанням системи рівнянь компонент тільки для орієнтованих блоків. Для неорієнтованих блоків отримання глобальної системи рівнянь є в загальному випадку непростим завданням.

Об'єктний підхід.

Погляньмо ще раз на Рис 18. В описі класу «ІзмерительнаяСистема» з'явилися екземпляри класів «Маятник» і «ИзмерительVS» з іменами «Маятник_1» і «ИзмерительVS_1» відповідно. Їх можна розглядати як особливі внутрішні змінні типу «активний динамічний об'єкт». В описі класу «Измерительная система» будуть видимі всі зовнішні змінні цих об'єктів.

8.2.2. RS-тригер.

Дана модель як раз ілюструє принцип «поділу складного на частини»: не так-то просто правильно написати закон функціонування тригера, в той же час відома його схема, що складається з двох схем Ілїнь і двох фільтруючих повторювачів. Кожен елемент цієї схеми є досить простим і зрозумілим пристроєм, а їх сукупна поведінка відтворюється автоматично. Ця нескладна модель ілюструє і ще один важливий момент: у схемі знаходяться по два примірники одного і того ж класу (два примірники класу Ілїнь і два примірники класу ФП), причому екземпляри повторювачів розрізняються значенням параметра «Інтервал_фільтрації».

На цьому ж прикладі ми проілюструємо використання блокового підходу при обладнанні віртуального стенду: для «випробувань» створеного нами тригера ми створимо генератор тестових сигналів і з'єднаємо його виходи зі входами випробуваного тригера ..

Фільтруючий повторювач.

Цей пристрій має вхід X і вихід Y типу boolean, а також параметр τ типу double. Пристрій функціонує наступним чином. При зміні входу X , якщо нове значення безперервно тримається протягом інтервалу фільтрації τ , вихід Y

приймає значення X , у протилежному випадку Y не змінюється, тобто пристрій не пропускає коротких змін X тривалістю менше τ . Карта поведінки фільтруючого повторювача наведена на Рис 19.

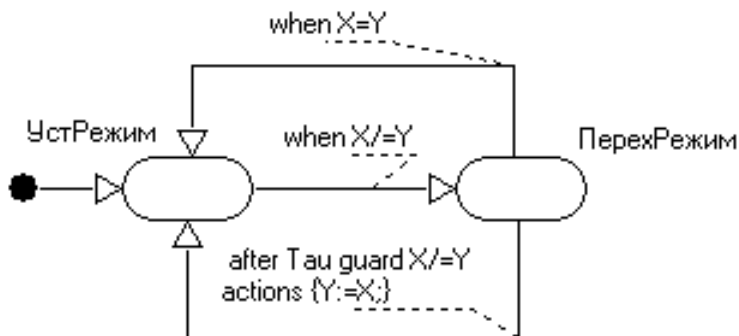


Рис 19

Пристрій знаходиться в сталому режимі (стан «УстРежим») до тих пір, поки значення входу перестане збігатися зі значенням виходу. Пристрій переходить в перехідний режим (стан «ПерехРежим»), в якому буде перебувати протягом інтервалу фільтрації τ . Якщо до закінчення цього інтервалу значення входу зміниться, то пристрій повернеться в початковий стан (верхній перехід на Рис 19). Після закінчення інтервалу фільтрації змінюється значення виходу і пристрій повертається в початковий стан (нижній перехід на Рис 19). У умови цього переходу використано охороняє умова (або охороняє предикат) «guard». Перехід спрацьовує, якщо в момент появи запускає події (у даному випадку це закінчення інтервалу τ) значення умови істинно. У даному конкретному випадку цей охороняє предикат потрібен лише для одного окремого випадку, коли X змінюється точно в момент закінчення інтервалу τ . У цьому випадку в карті поведінки виявляється два готові до спрацьовування переходу. Така

карта поведінки є недетермінованою. В одних пакетах моделювання (наприклад в AnyLogic) це вважається допустимим і спрацьовує перехід вибирається випадковим чином, в інших (у Stateflow, ModelVision) ця ситуація вважається помилкою. Краще таких ситуацій уникати.

Схема Іліне.

Цей пристрій має входи X1, X2 і вихід Y типу boolean. Функціонування пристрою визначається формулою

$$Y = \text{not} (X1 \text{ or } X2)$$

Зверніть увагу, в систему рівнянь, задану безперервну поведінку, можуть входити рівняння, що визначають значення змінних, що мають тип, відмінний від речового (в даному випадку булевський). Такі рівняння повинні бути обов'язково дозволені щодо шуканої змінної (тобто бути формулами), оскільки чисельне рішення алгебраїчних рівнянь нематеріального типу є скрутним.

Зауваження.

Користуватися такими формулами слід з обережністю. Будь-яка зміна нематеріальній змінної (булевою, цілою чи перераховуваній) за своєю суттю є дискретним подією. Момент ж настання дискретного події визначається точно, якщо воно задано явно у формі переходу або умовного рівняння. У даному прикладі формула задає статичну характеристику пристрою і, оскільки вона буде автоматично обчислюватися після кожного дискретного події в моделі і все буде правильно працювати. Інша річ, якщо булевський формула буде явно містити час, наприклад, якщо ми захочемо задати генерацію сходинок формулою

$$Y = (\text{Time} \geq 10)$$

Якщо величина Y буде використовуватися десь в умові переходу або в умовному рівнянні, то все буде правильно, однак буде потрібно процедура чисельного пошуку точки перемикавання. Краще за все поставити сходинок переходом з запуском подією «after 10» з одним і тим самим вихідним і кінцевим вузлом.

Тригер.

Тепер ми можемо зібрати з заготовлених класів схему тригера (Рис 20). Це своєрідний блок, який не має власної поведінки - його поведінка складається із сукупного поведінки локальних блоків з урахуванням зв'язків.

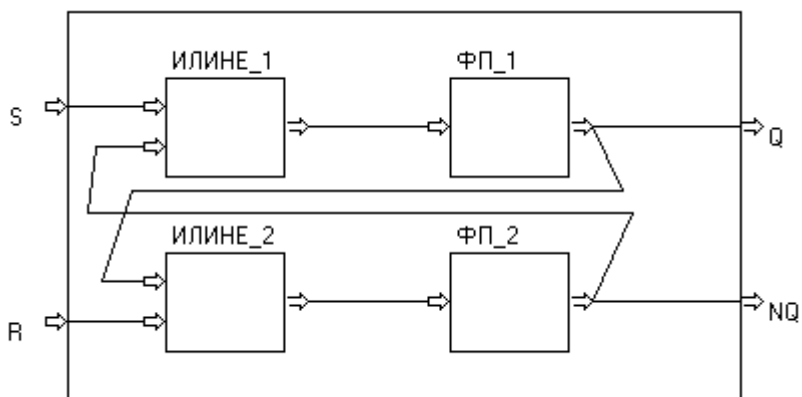


Рис 20

Випробувальний стенд.

Для того, щоб перевірити функціонування нашого чудового тригера, нам потрібно подати на його входи тестові послідовності сигналів. В інтерактивній моделі це можна зробити вручну, лінійну послідовність змін значень змінних можна задати в плані експерименту, але для більш складних алгоритмів потрібно створити генератор тестових сигналів (Рис 21).

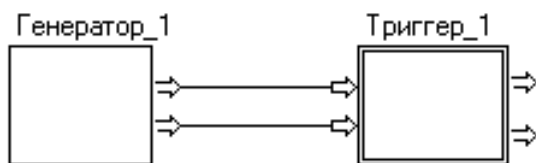


Рис 21

Головна карта поведінки генератора показана на Рис 22: виконується вкладена карта поведінки «ТестоваяПоследовательность» поки не закінчиться час експерименту.

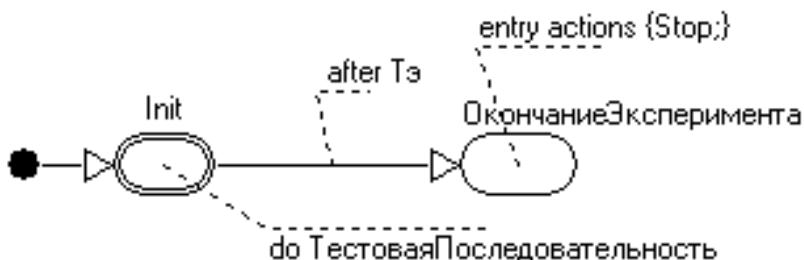


Рис 22

Локальна карта поведінки «ТестоваяПоследовательность» є простою картою станів (Рис 23), яка задає програму подачі тестових сигналів: після деякого початкового інтервалу сигналам установки і скидання присвоюється коректна комбінація, яка далі інвертується в циклі через певний інтервал часу.

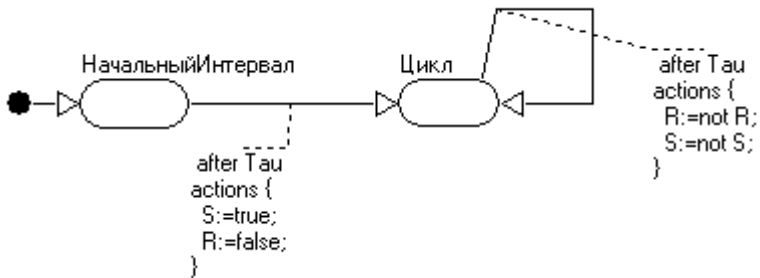


Рис 23

На Рис 24а показана тимчасова діаграма вхідних і вихідних сигналів тригера при значеннях

$$\text{Генератор_1.Tau} = 4$$

$$\Phi\Pi_1.Tau = \Phi\Pi_2.Tau = 1$$

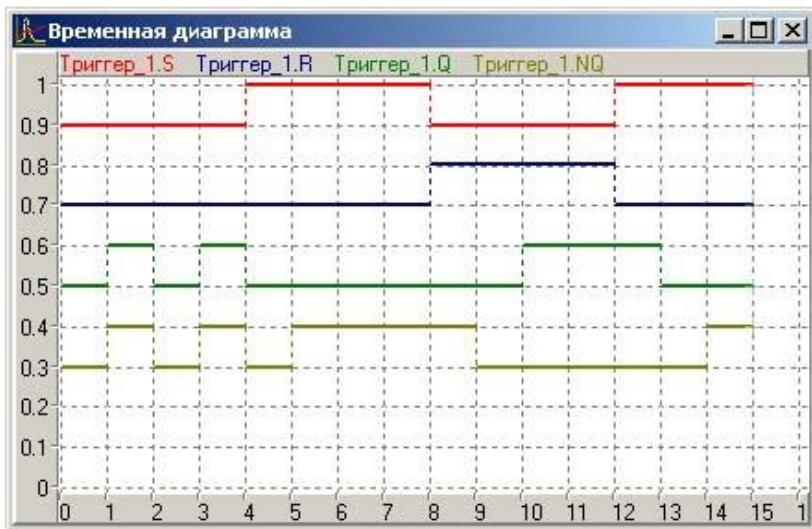


Рис 24 а)

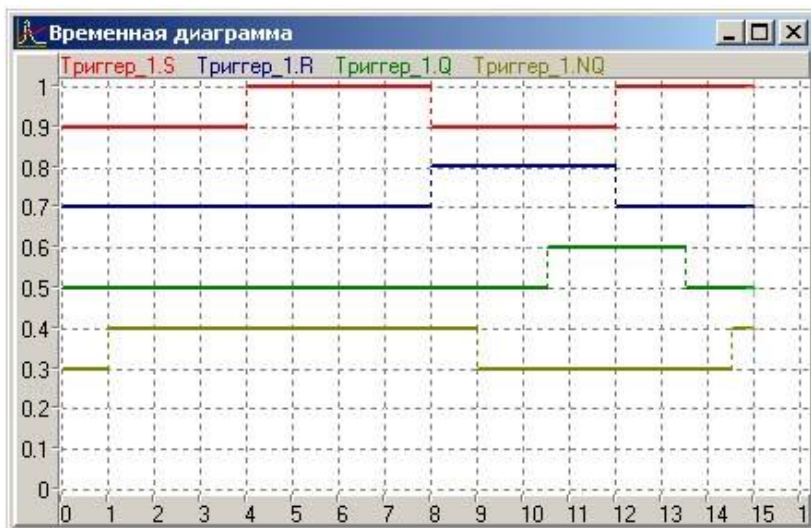


Рис 24 б)

Зверніть увагу, що при некоректному поєднанні початкових значень тестових сигналів тригер входить в режим автоколивань. Цей ефект зникає, якщо зробити затримки у фільтруючих повторювачів різними, наприклад, покласти (Рис 24б).

На закінчення звернемо вашу увагу, що для проведення цього експерименту зовсім не обов'язково було створювати спеціальний блок генератора. Досить було приписати поведінка, показане на Рис 22, Рис 23, самому віртуальному стенду. У самому справі, графічна мова карт стану сам по собі дуже зручний для наочного завдання послідовності дій, що виконується в безперервному часу. Його розширення - мова гібридних карт станів - дозволяє доповнити разові дискретні дії ще й безперервними діями (наприклад, з 1-ї по 10-у секунду подаємо на вхід синусоїдальний сигнал, а далі прямокутні

імпульси,одночасно з певною частотою знімаючи з виходу і заносючи в таблицю результати для обробки в кінці експерименту). Тому ми можемо дозволити собі сказати: гібридний автомат - це ще і візуальний мова управління обчислювальним експериментом!

..... ЛІТЕРАТУРА

1. <http://www.exponenta.ru/soft/others/mvs/mvs.asp> ,

а в упакованном виде:

2.

http://www.exponenta.ru/soft/others/mvs/archive/MVS_R_FREE_3.1.5.zip

3. Кострікин А.І. Введення в алгебру. - М., Наука, 1977. - 495 с.

4. Кук Д.,Бейз Д. Компьютерная математика: Учебник.пер. с англ.-М.: Наука, 1990.-384 с.

5. Компьютерная алгебра: Символьные и алгебраические вычисления: Учебник.Перев. с англ./Под ред..Б.Бухбергера и др..-М.:Мир, 1986.-392 с.,илл.

6. Курош а.Г. Теория групп. - М., Наука, 1967. - 648 с.

7. Літнарівич Р.М. Ілюстрований самовчитель по MAPLE. Ч.15. Пакети лінійної алгебри і функціональних систем. МEGУ, Рівне, 2009,- 46 с.

8. Половко А.М.,Бутусов П.Н. MATLAB для студента:учебник СПб.:БХВ-Петербург, 2005.-320 с.

9. Якимчук А.Й. Побудова і дослідження математичної моделі пункту GPS спостережень методом статистичних випробувань Монте Карло. Множинний регресійний аналіз . Модель ДА – 50. МEGУ, Рівне, 2010, -112 с.

Додаткова література

1. Літнарівич Р.М. Конструювання і дослідження математичних моделей. Онтодидактика поліноміальної апроксимації. Частина 3. МEGУ, Рівне, 2009,-32 с.

Руслан Миколайович Літнарівч
кандидат технічних наук, доцент
Юрій Георгієвич Лотюк,
кандидат педагогічних наук, доцент

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ
НАВЧАЛЬНО-МЕТОДИЧНИЙ ПОСІБНИК

КНИГА 1

ФАКУЛЬТЕТ КІБЕРНЕТИКИ

КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

**Комп'ютерний набір, верстка, редагування
і макетування та дизайн в редакторі
Microsoft® Office® Word 2003**

Р.М.Літнарівч, Ю.Г.Лотюк

Відповідальний редактор Й.В. Джунь

Підп.до друку 11. 12. 2010 р.

Формат 60x84/16. Папір офсетн.№1.

Гарнітура Times New Roman.

Друк різнограф. Тираж 300 пр.

Редакційно-видавничий центр «Тетіс»

Міжнародного економіко-гуманітарного університету

Імені академіка Степана Дем'янчука

33027 Рівне , Україна

Вул..С.Дем'янчука, 4, корпус 1

Телефон : (+00380) 362 23 – 73 – 09

Факс :(+00380) 362 23 – 01 – 86

E-mail:mail@regi.rovno.ua