

10.3 Алгоритми генерування тестових послідовностей

Розглянемо детальніше найскладніший з етапів – генерування випадкових чисел із заданим законом розподілу ймовірностей і кореляційними властивостями. Цю задачу розв’язують в декілька етапів. Спочатку отримують послідовність рівномірно розподілених на інтервалі $[0,1]$ випадкових чисел, а з неї – послідовність випадкових чисел із заданим законом розподілення.

Виділяють три основних методи формування таких послідовностей.

1. Пряме перетворення числа x_i , яке є реалізацією випадкової величини X , що рівномірно розподілена на інтервалі $[0,1]$, за допомогою деякої функції W_N у число y_i , яке може бути розглянуте як реалізація випадкової величини Y , що має заданий закон розподілення. Функція необхідного нелінійного перетворення може бути визначена з рівняння (1.94), в якому у даному випадку f_x – рівномірний розподіл, f_y – необхідний нам розподіл, $N(x)$ – шукане нелінійне перетворення.

2. Відсіювання чисел із початкової послідовності рівномірно розподілених на інтервалі $[0,1]$ випадкових чисел таким чином, щоб числа, які залишились, були розподілені за заданим законом.

3. Моделювання умов відповідних граничних теорем теорії ймовірності. Зокрема, для отримання послідовності з нормальним законом розподілу достатньо усереднювати декілька (не менше 6) даних, розподілених за рівномірним законом.

На практиці доцільно використовувати всі три методи формування. Вибір оптимального методу визначається, виходячи з мети й задач моделювання та їх особливостей.

Складнішим є генерування *випадкових числових послідовностей* із заданими законами розподілу ймовірностей та кореляційними функціями.

Відомий метод отримання числових послідовностей з допомогою *сортування* вихідних послідовностей. Метод ґрунтується на тому, що коефіцієнт кореляції випадкових чисел залежить більше від порядку їх розташування, ніж від величини. Саме тому дві випадкових послідовності, що відповідають двом різним розподіленням, у випадку, якщо вони впорядковані приблизно однаково, будуть мати приблизно рівні коефіцієнти кореляції.

Відповідно до методу сортування генерується випадкова послідовність $X(k)$ із заданою *кореляційною функцією*, але довільним розподілом. Ставиться їй у відповідність послідовність цілих чисел $I(k)=k$ (тобто члени послідовності нумеруються). Потім обидві послідовності попарно сортуються. При цьому величини $X(k)$ розташовуються за зростанням, а масив $I(k)$ «запам’ятовує» їх попереднє положення (місце в несортованому масиві $X(k)$). Таким чином, масив цілих чисел $I(k)$ відображає кореляцію між елементами масиву $X(k)$. Після впорядкування масив $X(k)$ цікавості не викликає, оскільки вся інформація про кореляційну функцію тепер знаходиться в масиві $I(k)$.

Потім генерується випадкова послідовність $Y(k)$ із заданим *законом розподілу ймовірностей* і нульовою кореляцією і записується на місце масиву $X(k)$. Після цього вона сортується порядку зростання. Далі масиви $I(k)$ і $Y(k)$ попарно сортуються, причому масив $I(k)$ розташовується у порядку зростання. Схема алгоритму наведена на рис. 10.3.

Алгоритм сортування доцільно використовувати в тих випадках, коли для моделювання достатньо невеликої кількості статистичних даних, що не потребує оперування числовими масивами великої розмірності. При великій розмірності цих масивів суттєво збільшується час моделювання.

Відомий алгоритм *фільтрації*, що потребує менших витрат машинного часу для отримання випадкового процесу із заданою кореляцією і законом розподілу ймовірностей, який базується на використанні нормального стаціонарного випадкового процесу $X(t)$ як початкового. Завжди існує таке нелінійне безінерційне перетворення $Y=W_N(X)$, яке перетворює нормальну функцію щільності $f_X(x)$ процесу $X(t)$ на задану $f_Y(y)$. Якщо вихідний процес $X(t)$ має кореляційну функцію $R_{XX}(\tau)$, то перетворений процес $Y(t)$ буде мати кореляційну функцію $R_{YY}(\tau)$, що відрізняється від функції $R_{XX}(\tau)$ і пов'язана з нею деякою залежністю $R_{YY}=J(R_{XX})$. Вигляд цієї залежності визначає перетворення $Y=W_N(X)$. Для того щоб, кореляційна функція перетвореного процесу була заданою, необхідно вибрати кореляційну функцію вихідного процесу:

$$R_{XX}(\tau) = J^{-1}[R_{YY}(\tau)]. \quad (10.9)$$

При використанні цього методу підготовча робота складається з таких етапів.

1. Знаходження функції перетворення $Y=W_N(X)$ за заданою функцією щільності $f_Y(y)$.
2. Отримання за знайденою функцією $Y=W_N(X)$ залежності $R_{YY}=J(R_{XX})$.

3. Розв'язання рівняння $R_{YY}=J(R_{XX})$ відносно R_{XX} , тобто визначення кореляційної функції $R_{XX}(\tau)$ нормального вихідного процесу $X(t)$.

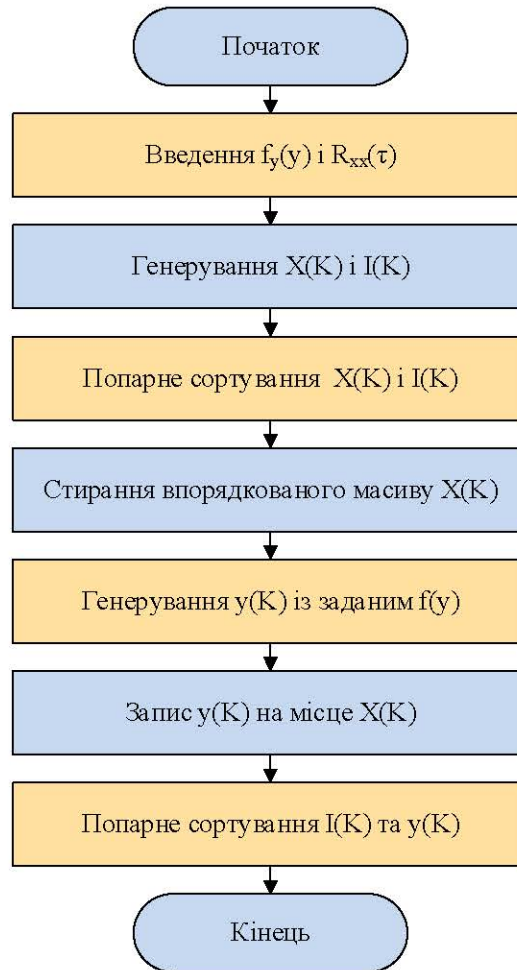


Рисунок 10.3 – Алгоритм генерування випадкової числової послідовності із заданими статистичними характеристиками методом сортування

Після закінчення підготовчої роботи моделювання випадкового процесу із заданими характеристиками зводиться до формування дискретних реалізацій $X[n]$ нормального випадкового процесу $X(t)$ і перетворення цих реалізацій за формулою

$$Y[n] = W_N \{X[n]\}. \quad (10.10)$$

Описаний алгоритм потребує менших витрат часу, ніж алгоритм сортування, не потребує накопичення і зберігання у пам'яті числових масивів великої розмірності. Недоліком методу є його складність для програмування і великий обсяг підготовчої роботи.

При використанні обох описаних алгоритмів виникає задача генерування на комп'ютері випадкових числових послідовностей із заданими законами розподілення і нульовою кореляцією та випадкових числових послідовностей із заданою кореляційною функцією та довільним розподіленням.

Задача дещо спрощується при генеруванні нормальної послідовності із заданою кореляційною функцією, оскільки отримання заданої кореляційної функції виконується шляхом лінійної фільтрації, яка не призводить до втрати нормальності розподілу послідовності даних.

Методика генерування нормальної послідовності із заданою кореляційною функцією передбачає використання початкової послідовності типу «нормальний білий шум» – таку послідовність звичайно генерують стандартні генератори. Спектральна щільність потужності такої послідовності є константою

$G_{xx} = D_x / 2\pi$. Методика складається з кроків:

1. Підготовчий етап:

1.1. Знаходження спектральної щільності необхідної послідовності як перетворення Фур'є заданої кореляційної функції $G'_{xx}(j\omega) = \frac{1}{2\pi} \int_0^{\infty} R_{xx}(\tau) e^{-j\omega\tau} d\tau$;

1.2. Знаходження амплітудно-частотної характеристики необхідного фільтра $A(\omega) = |W(j\omega)| = \frac{|G'_{xx}(j\omega)|}{D_x / 2\pi}$;

1.3. Знаходження відповідного дискретного перетворення за формулою (5.41);

2. Власне моделювання:

2.1. Генерування початкової послідовності типу нормального білого шуму з заданою дисперсією;

2.2. Перетворення її за знайденою формулою дискретної фільтрації – отримання вхідної послідовності для імітаційної моделі;

2.3. Імітація перетворення вхідних даних системою;

3. Обробка результатів імітаційного моделювання.

Окремим випадком є генерування числової послідовності з нормальним розподілом ймовірностей і експоненціальною автокореляційною функцією. Цей вид випадкової послідовності є найпоширенішим і найлегшим для генерування. Генерування здійснюється у два етапи:

1. Генерування некорельованої рівномірно розподіленої послідовності $\{x_i\}$ за допомогою будь-якої програми-генератора, які наразі є в усіх мовах програмування;

2. Перетворення на задану послідовність $\{x'_j\}$ методом підрахунку ковзного середнього

$$x'_j = \frac{1}{m} \sum_{i=j}^{j+m-1} x_i.$$

При $m \geq 6$ через дію умов центральної граничної теореми теорії ймовірностей розподіл послідовності $\{x'_j\}$ буде близьким до нормального. Крім того, елементи цієї послідовності при утворенні мають різну кількість спільних доданків, наприклад:

$$\text{при } m = 6 \text{ маємо } x'_1 = x_1 + x_2 + \dots + x_6,$$

$$x'_2 = x_2 + \dots + x_6 + x_7,$$

$$x'_3 = x_3 + \dots + x_6 + x_7 + x_8,$$

тобто x'_1 має 5 спільних доданків з x'_2 , 4 спільних доданки з x'_3 і т. д., що приводить до поступового зменшення зв'язку між елементами отриманої послідовності. Це приблизно відповідає експоненціальному характеру кореляційної функції. Очевидно, інтервал кореляції цієї послідовності буде $m - 1$.

10.3.1 Обробка результатів імітаційного моделювання

Обробка результатів імітаційного моделювання здійснюється за тими ж правилами і методиками, що й обробка результатів реальних натурних експериментів. Зокрема, якщо моделями вхідних сигналів є випадкові числові послідовності, то для обробки результатів використовуються методи дисперсійного, кореляційного і регресійного аналізу, розглянуті у підрозділі 6.1.

10.3.2 Оцінювання необхідного обсягу тестів та трудомісткості імітаційного моделювання

Основою для визначення необхідного обсягу тестових даних є оцінка похибки імітаційного моделювання. Залежно від вигляду статистичних характеристик вихідних сигналів імітаційної моделі, які необхідно оцінити у конкретній задачі, похибки оцінки обчислюються за формулами:

– при оцінюванні розподілу ймовірностей способом підрахування кількості даних, які потрапляють у кожен інтервал розбиття діапазону можливих значень (спосіб побудови гістограми) відносна похибка розраховується за формулою

$$\varepsilon_p^* = \sqrt{(1 - P_A) / N}, \quad (10.11)$$

де P_A – ймовірність потраплення даного у окремий інтервал; N – обсяг вибірки.

При оцінюванні розподілу ймовірностей на основі визначення відносного часу перебування у заданому інтервалі

$$\varepsilon_p^* = \sqrt{\frac{2(1-P_A)\tau}{TP_A}}, \quad (10.12)$$

де T – загальний час спостереження; τ – інтервал надходження дискретних даних;

– при оцінюванні середнього значення усієї множини результатів моделювання і відсутності кореляції між окремими результатами

$$\varepsilon_m^* = \frac{1}{m_x} \sqrt{\frac{D_x}{N}} = \frac{\sigma_x}{m_x \sqrt{N}}, \quad (10.13)$$

де m_x і σ_x – математичне сподівання і с.к.в. результатів.

При оцінюванні поточного середнього значення методом ковзного середнього на основі рекурентної формули $m_x = \frac{m_x(n-1) + x}{n}$ і відсутності кореляції між окремими результатами

$$\varepsilon_m^* = \sqrt{\frac{1}{n} \left(\frac{m_x^2}{m_x^2} - 1 \right)} = \frac{\sigma_x}{m_x \sqrt{n}}; \quad (10.14)$$

– при оцінюванні дисперсії нормально розподіленої послідовності і відсутності кореляції між окремими результатами

$$\varepsilon_D = \frac{\sigma_D}{D_x} = \sqrt{\frac{2}{N-1}}; \quad (10.15)$$

– при оцінюванні коефіцієнта кореляції на основі зіставлення часу перевищення деякого рівня U у двох корельованих послідовностях

$$\varepsilon_{\rho}^2 = \frac{D[\rho_x^*(\tau)]}{U^2 \rho_x^2(\tau)} = \frac{1}{(\beta_{\rho}^*)} = \frac{[1 - \rho_x^2(\tau)]}{N \rho_x^2(\tau)}, \quad (10.16)$$

де $\beta_{\rho}^* = U/\sigma_{\rho}^*(\tau)$.

З наведених співвідношень видно, що в усіх випадках похибка прямо чи опосередковано (через час проведення моделювання) залежить від кількості отриманих даних. Тому необхідна кількість даних може бути знайдена шляхом розв'язування нерівності

$$\varepsilon(N) \leq \varepsilon_{\max}, \quad (10.17)$$

де ε_{\max} – максимально допустима похибка моделювання.

Наприклад, для оцінювання закону розподілу ймовірностей шляхом побудови гістограми скористаємось нерівністю на основі формули (10.11)

$$\sqrt{(1-P)/N} \leq \varepsilon_{\max}. \quad (10.18)$$

Якщо гістограма будується на 10 інтервалах розбиття діапазону можливих значень результатів і попередній гіпотезі, що цей закон близький до трикутного, розрахуємо мінімальну ймовірність (з формули видно, що кількість даних обернена до ймовірності)

$$P_{\min} = \frac{1}{2} \cdot (0,1D_0) \cdot (0,2h),$$

де D_0 – діапазон; h – висота трикутного розподілу. Враховуючи, що $\frac{1}{2}D_0h = 1$, отримуємо $P_{\min} = 0,02$.

Розв'язуючи нерівність (10.18) відносно N , знаходимо, що для отримання відносної статистичної похибки не гірше 0,1 (тобто 10% – досить неточний експеримент!) на кожному інтервалі необхідно мати $N \geq (1-0,02)/0,1^2$, тобто не менше 98 даних, а усього для 10 інтервалів імітаційна вибірка повинна містити не менше 980 даних.

При попередній гіпотезі про нормальність закону розподілу результатів кількість даних повинна бути значно більшою, оскільки на кінцях діапазону значення ймовірності для нормального закону менше, ніж для трикутного. Очевидно, найменша кількість даних необхідна за умови гіпотези про рівномірний характер розподілу результатів.

Оскільки заздалегідь невідомо, виправдається висунута гіпотеза, чи ні, то найчастіше імітаційний експеримент проводять у два етапи: спочатку генерують мінімальну кількість даних, проводять експеримент, попередньо визначають тип розподілу ймовірностей результатів, а потім уточнюють розрахунок необхідної кількості даних і проводять додаткове моделювання.

10.3.3 Програмні пакети для імітаційного моделювання

Інтерактивне середовище моделювання SIMULINK

Програма SIMULINK є додатком до пакета MATLAB. В певному сенсі SIMULINK можна розглядати як самостійний продукт фірми MathWorks, однак він працює лише за наявності ядра MATLAB і використовує функції, які входять до його складу.

Додаток SIMULINK є інтерактивним інструментом для моделювання, імітації та аналізу динамічних систем, охоплюючи дискретні, неперервні, гібридні, нелінійні та розривні системи, за допомогою якого можна поєднувати блоки, що відповідають окремим елементам динамічної системи в єдине ціле й вивчати їх поведінку в часі.

Розробка моделей засобами SIMULINK (S-моделі) основана на технології drag-and-drop («перетягни і залиш»). Для побудови S-моделі використовуються модулі (або блоки), які зберігаються в бібліотеці SIMULINK. Компоненти моде-

лей, які містяться в ряді бібліотек і за допомогою миші можуть переноситися в основне вікно та з'єднуватися один з одним необхідними зв'язками. До складу моделей можуть входити джерела сигналів різного виду, віртуальні реєструвальні прилади, графічні засоби анімації. Подвійне клацання мишею на блоці моделі виводить вікно зі списком його параметрів, які користувач може змінювати. Запуск імітації забезпечує математичне моделювання побудованої моделі з наочним візуальним поданням результатів.

Пакет заснований на побудові блокових схем шляхом перенесення блоків з бібліотеки компонентів у вікно редагування створюваної користувачем моделі. Блоки можуть бути пов'язані одні з одним як за інформацією, так і за управлінням. Вид зв'язку залежить від типу блока та логіки роботи моделі. Дані, якими обмінюються блоки, можуть бути скалярними величинами, векторами або матрицями довільної розмірності.

Будь-яка S-модель може мати ієрархічну структуру, тобто складатись з моделей більш низького рівня, причому кількість рівнів ієрархії практично не обмежена. Разом з іншими параметрами моделювання користувач може задавати спосіб зміни модельного часу (з постійним чи змінним кроком), а також умови завершення моделювання.

В ході моделювання надається можливість спостерігати за процесами, які відбуваються в системі. Для цього використовуються спеціальні «оглядові вікна», які входять до складу бібліотеки SIMULINK. Характеристики, які цікавлять користувача, можуть бути подані як в числовій, так і в графічній формах.

Застосування принципів структурного і модульного програмування дозволяє подавати різноманітні алгоритми у вигляді набору уніфікованих програмних модулів, що покращує оглядовість програми, полегшує її відлагодження, і в кінцевому результаті, зменшує загальний обсяг програмного забезпечення, яке необхідно розробити. При цьому, на відміну від класичних способів моделювання, користувач не має необхідності досконало вивчати мову програмування і чисельні методи математики. Йому достатньо загальних знань, необхідних при роботі за комп'ютером, і знань тієї предметної області, в якій він працює. До того ж склад бібліотеки SIMULINK може бути доповнений користувачем за рахунок розробки власних блоків.

Варто відмітити, що у складі MATLAB є велика кількість інших додатків, оснований на методах графічного (візуального) програмування, які припускають можливість сумісної роботи з додатком SIMULINK.

Aerospace Blockset – містить спеціальні інструменти для моделювання авіаційних, космічних, реактивних і турбореактивних систем.

DSP Blockset – призначений для проектування систем і моделювання задач цифрової обробки сигналів (DSP).

Дані бібліотеки містять в собі такі ключові операції, як класична, багатоступенева і адаптивна фільтрації, перетворення, матричні операції і лінійна алгебра, статистика і спектральний аналіз.

Nonlinear Control Design Blockset – надає користувачу графічний інтерфейс для налаштування параметрів динамічних об'єктів

SimPowerSystems – призначений для моделювання електротехнічних і електроенергетичних пристроїв і систем.

SimMechanics – дозволяє моделювати системи управління за допомогою ненаправлених сигнальних графів, об'єднувати їх з фізичними моделями і моделями з інших бібліотек.

Моделювання за допомогою SIMULINK складається з таких кроків:

- вибір або створення блоків моделі.
- організація зв'язків між блоками моделі.
- встановлення параметрів окремих блоків.
- організація виведення результатів моделювання.
- запуск моделі і отримання результату.

На рис. 10.4 наведений елементарний приклад подібного моделювання. Ви в інтерактивному режимі складаєте з блоків необхідну модель і бачите на екрані віртуального осцилографа результат моделювання.

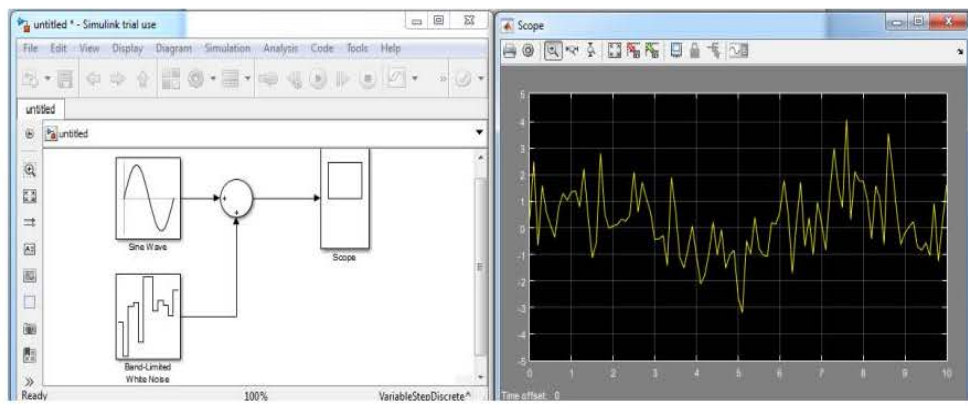


Рисунок 10.4 – Принцип візуального програмування в SIMULINK

На рис. 10.5 показаний процес моделювання простої системи – гідравлічного циліндра. Контроль здійснюється за допомогою віртуальних осцилографів – на рис. 10.6 видно екрани двох таких осцилографів і вікно простої підсистеми моделі. Можливе моделювання складних систем, що складаються з безлічі підсистем. Simulink складає і вирішує рівняння стану моделі і дозволяє підключати в потрібні точки різноманітні віртуальні вимірювальні прилади.

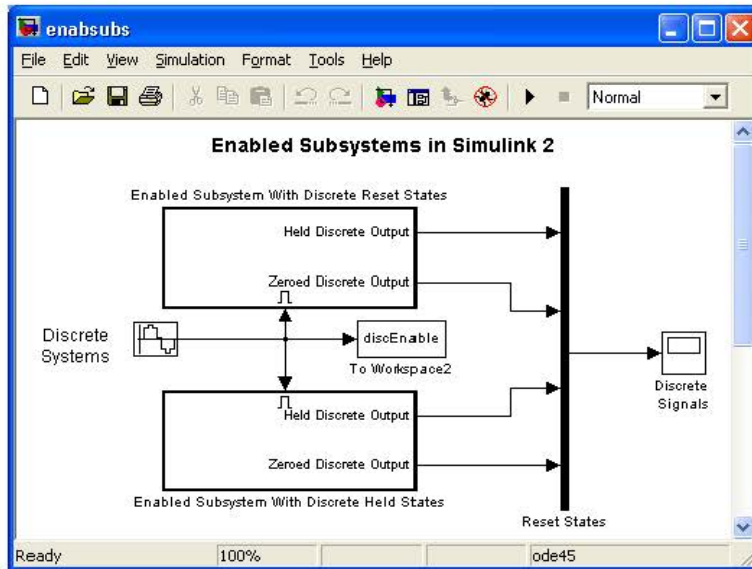


Рисунок 10.5 – Процес моделювання гідравлічного циліндра

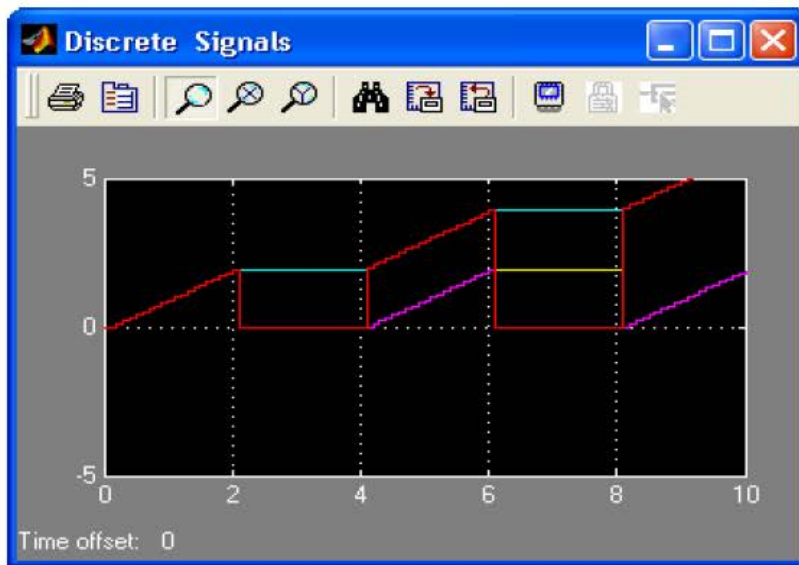


Рисунок 10.6 – Приклад моделювання системи гідравлічного циліндра за допомогою розширення Simulink

Принципи моделювання в SIMULINK

Для запуску програми необхідно попередньо запустити пакет MATLAB. Після відкриття основного вікна програми MATLAB (рис. 10.7) необхідно запусити програму SIMULINK.

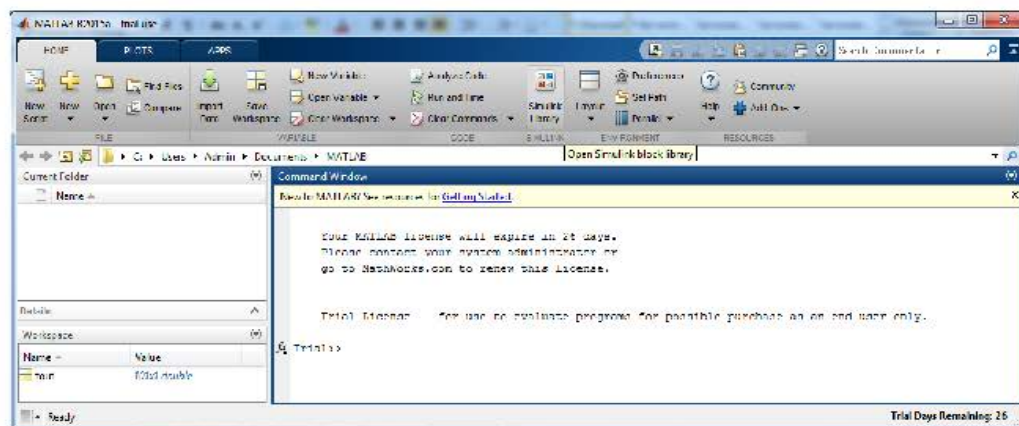


Рисунок 10.7 – Основне вікно програми MATLAB

При запуску відкривається вікно оглядача розділів бібліотеки SIMULINK (рис. 10.8).

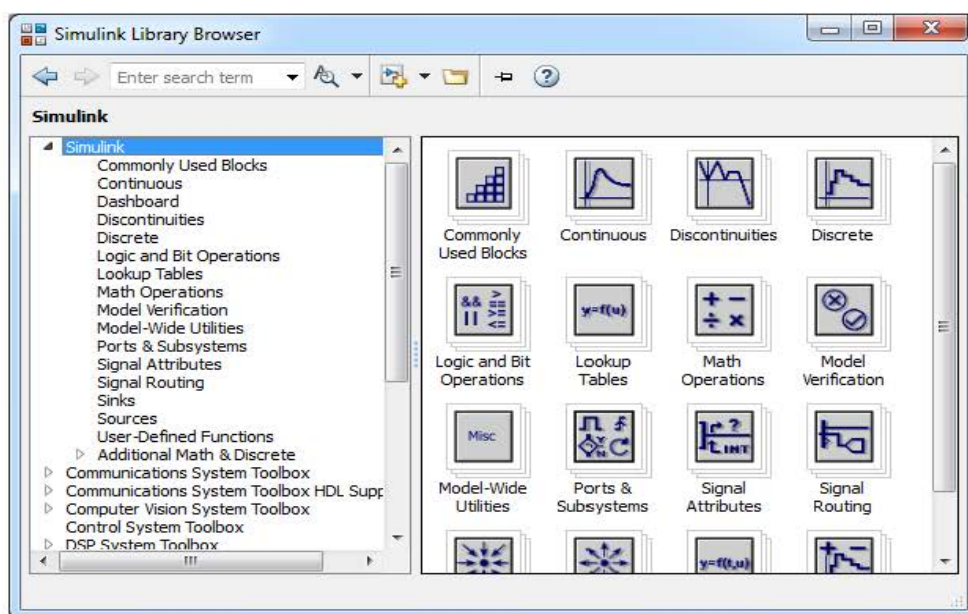


Рисунок 10.8 – Вікно оглядача розділів бібліотеки SIMULINK

Бібліотека SIMULINK містить такі основні розділи:

1. *Commonly Used Blocks* – блоки, які використовуються найчастіше;
2. *Continuous* – компоненти з неперервними характеристиками;
3. *Dashboard* – панель приладів;
4. *Discontinuities* – компоненти з розривними характеристиками;
5. *Discrete* – дискретні компоненти;
6. *Logic and Bit Operations* – логічні і бітові компоненти;
7. *Lookup Tables* – довідкові таблиці;
8. *Math Operations* – математичні компоненти;
9. *Model Verification* – верифікація моделей;
10. *Model-Wide Utilities* – утиліти;
11. *Ports & Subsystems* – порти і підсистеми;
12. *Signal Attributes* – атрибути сигналів;
13. *Signal Routing* – маршрутизація сигналів;
14. *Sinks* – пристрої реєстрації;
15. *Sources* – джерела сигналів і впливів;
16. *User-Defined Functions* – функції, які задаються користувачем.

Для створення моделі в середовищі SIMULINK необхідно:

1. Створити новий файл моделі. Для цього потрібно використати кнопку New Model на панелі інструментів оглядача розділів бібліотеки SIMULINK (рис. 10.9).

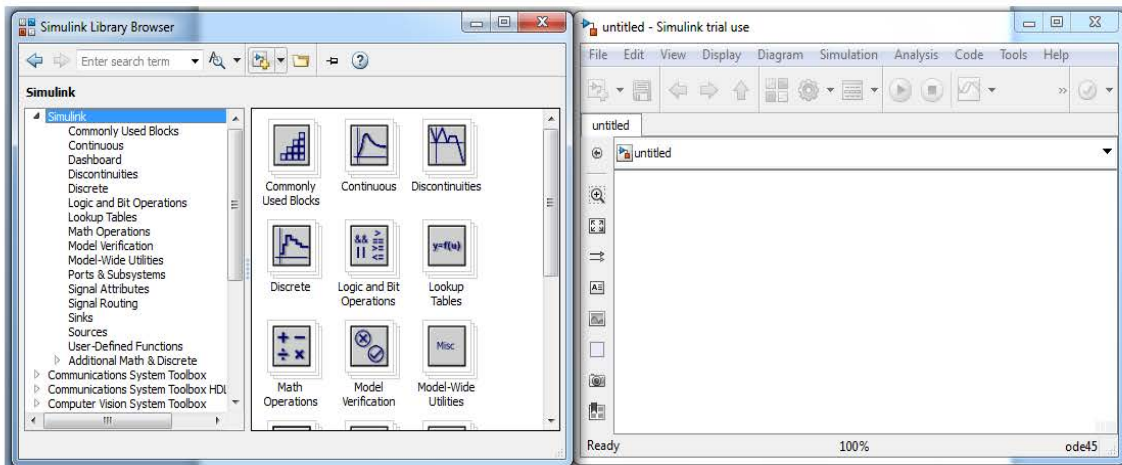


Рисунок 10.9 – Створення нового файлу моделі і результуюче вікно моделі

2. Розташувати блоки у вікні моделі (рис 10.10). Для цього необхідно відкрити відповідний розділ бібліотеки (наприклад, Sources – Джерела). Далі, вказавши курсором на потрібний блок і натиснувши ліву кнопку миші, “перетягнути” блок у створене вікно. Параметри блоків можна змінювати шляхом відкриття вікна окремого блока за допомогою подвійного кліку по обраному блоку мишкою.

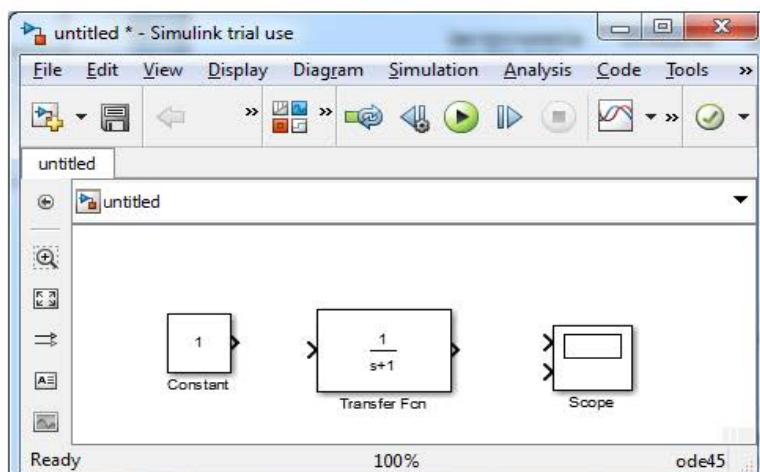


Рисунок 10.10 – Вікно моделі, яке містить блоки

Для видалення блока необхідно обрати блок і натиснути клавішу Delete на клавіатурі.

3. Після встановлення на схемі всіх блоків з потрібних бібліотек необхідно виконати з'єднання елементів схеми (рис. 10.11).

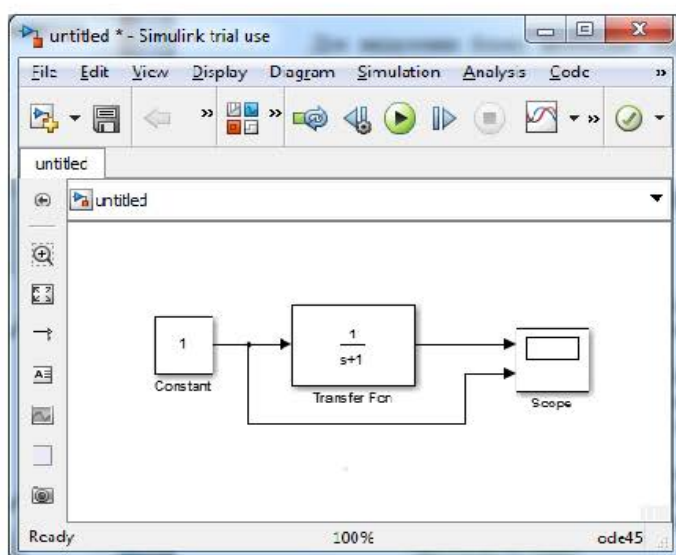


Рисунок 10.11 – Схема моделі

Для з'єднання блоків потрібно вказати курсором на "вихід" блока, а потім натиснути і, не відпускаючи ліву кнопку миші, провести лінію до "входу" іншого блока.

Для створення точки розгалуження в з'єднувальній лінії необхідно навести курсором на місце, де потрібно встановити вузол, і натиснувши праву кнопку миші, протягнути лінію.

Для видалення лінії потрібно обрати лінію і натиснути клавішу Delete на клавіатурі.

4. Після побудови схеми необхідно зберегти її у вигляді файлу на диску, обравши пункт меню File/Save As... у вікні схеми і вказавши ім'я файлу.

5. Виконання розрахунку. Виконання розрахунку здійснюється за допомогою пункту меню Simulation/Run або відповідної кнопки на панелі інструментів.

6. Завершення роботи. Для цього необхідно зберегти модель у файлі, закрити вікно моделі, вікно оглядача бібліотек, а також основне вікно пакету MATLAB.

Приклади застосування Simulink для вирішення практичних задач

Приклад 1. Розглянемо приклад побудови моделі сигналу виду $x(t) = 2 \sin \pi t + 0,2t^2$ на інтервалі $[0; 10]$, і відобразимо його на віртуальному осцилографі.

Сигнал $2 \sin \pi t$ задається в параметрах блока Sin Wave (рис. 10.12). Для сигналу $0,2t^2$ використовується три блоки – блок лінійного сигналу, масштабний блок і блок математичної функції, де була обрана функція піднесення до квадрата. Інтервал моделювання заданий в межах від 0 до 10 у налаштуваннях вікна віртуального осцилографа.

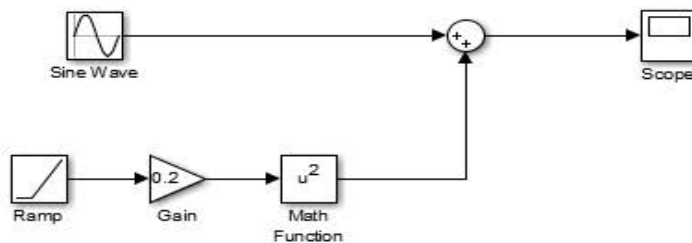


Рисунок 10.12 – Модель сигналу $x(t) = 2 \sin \pi t + 0,2t^2$

Результати роботи виведені на екран віртуального осцилографа (рис. 10.13).

Приклад 2. В аналоговій обчислювальній техніці існує декілька способів моделювання об'єктів, які описуються системами алгебраїчних рівнянь. Один з них, найбільш поширений, зводиться до розв'язання системи звичайних диференціальних рівнянь, розв'язок якої і дає вирішення алгебраїчної задачі.

Нехай безінерційний об'єкт описується системою рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_n \end{cases}$$

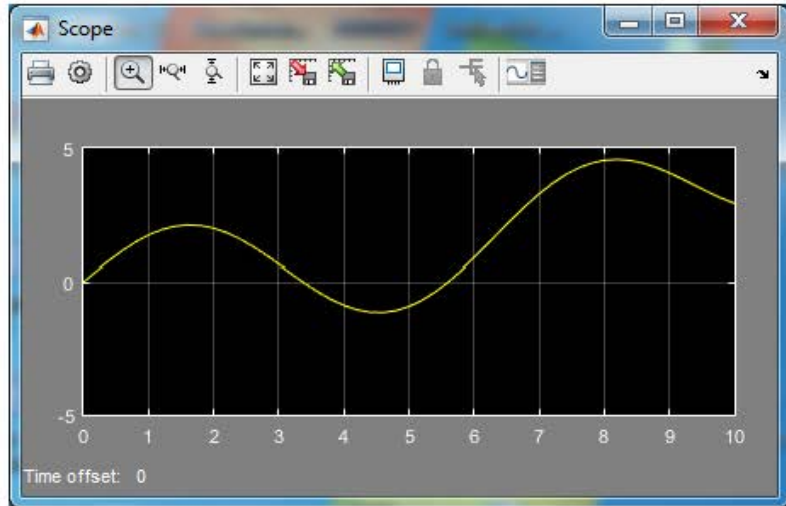


Рисунок 10.13 – Сигнал $x(t) = 2 \sin \pi t + 0,2t^2$ на інтервалі $[0;10]$

Для побудови моделі даного об'єкта використаємо метод зведення до еквівалентної системи диференціальних рівнянь.

Введемо таку систему диференціальних рівнянь:

$$\begin{cases} \frac{dx_1}{dt} + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - b_1 = 0 \\ \frac{dx_2}{dt} + a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - b_2 = 0 \\ \frac{dx_n}{dt} + a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n - b_n = 0 \end{cases}$$

Як тільки всі похідні набудуть згасаючого характеру, тобто $\frac{dx_i}{dt} = 0$, отримаємо розв'язок $\{x_1, x_2, \dots, x_n\}$.

Для еквівалентності цих двох систем рівнянь має забезпечуватись згасаючий розв'язок системи диференціальних рівнянь. Достатньою умовою, яка забезпечує згасаючий розв'язок, є додатно визначена матриця коефіцієнтів лінійної системи рівнянь.

Для визначення того є матриця додатно або від'ємно визначеною, застосовують Критерій Сильвестра. Згідно з ним квадратна матриця вважається додатно визначеною в тому випадку, якщо всі визначники матриць меншого порядку приймають значення ≥ 0 .

Нехай необхідно знайти розв'язок системи:

$$\begin{cases} 5x_1 + 2x_2 + x_3 = 2,5 \\ 3x_1 + 7x_2 + 2x_3 = -1,5 \\ 4x_1 + 0,5x_2 + 5x_3 = 11,5 \end{cases} .$$

Перейдемо до еквівалентної системи диференціальних рівнянь:

$$\begin{cases} \frac{dx_1}{dt} = 2,5 - 5x_1 - 2x_2 - x_3 \\ \frac{dx_2}{dt} = -1,5 - 3x_1 - 7x_2 - 2x_3 \\ \frac{dx_3}{dt} = 11,5 - 4x_1 - 0,5x_2 - 5x_3 \end{cases} .$$

Модель системи і результати її роботи наведені на рис. 10.14 і рис. 10.15 відповідно.

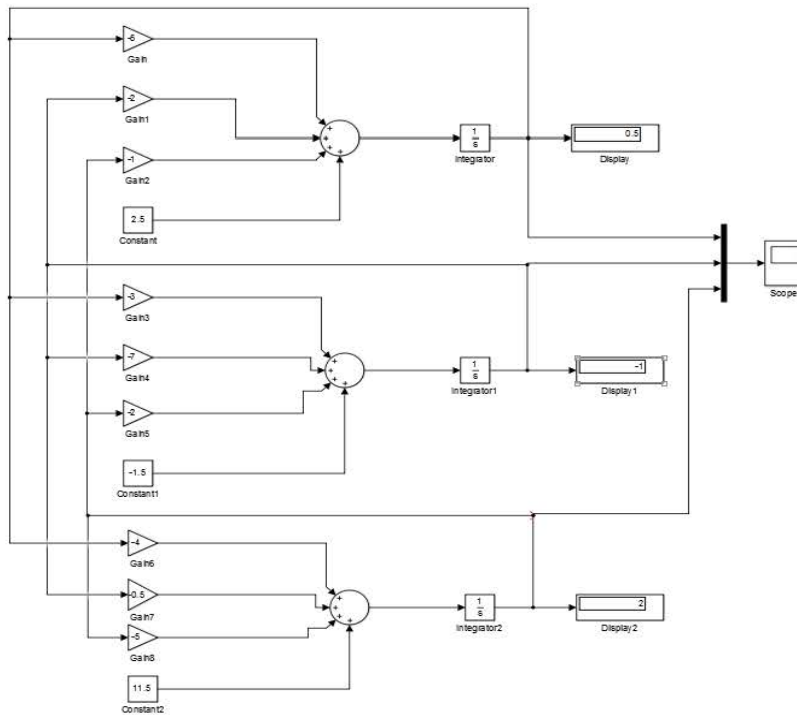


Рисунок 10.14 – Модель системи лінійних алгебраїчних рівнянь третього порядку

Розв'язок системи рівнянь, відображений на цифрових реєстраторах, становить:

$$x_1 = 0,5;$$

$$x_2 = -1;$$

$$x_3 = 2.$$

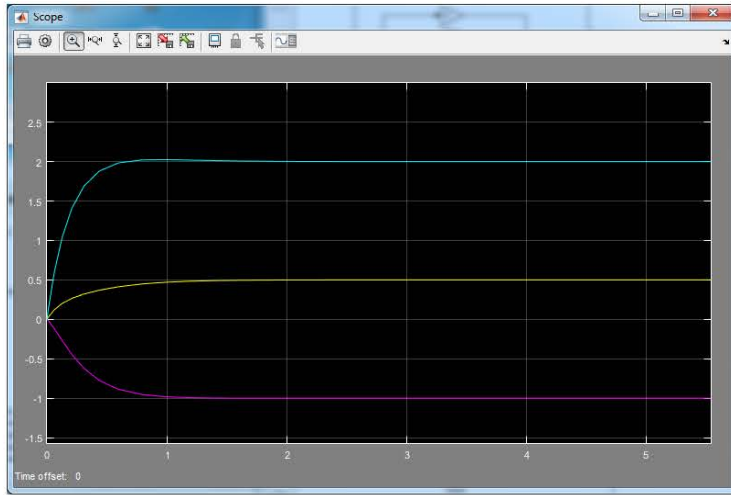


Рисунок 10.15 – Перехідний процес встановлення розв’язку системи рівнянь

Приклад 3. Поширеним способом опису поведінки динамічної системи є система диференціальних або інтегрально-диференціальних рівнянь.

Розглянемо реалізацію такої математичної моделі в SIMULINK.

Нехай є модель динамічної системи, яка описується диференціальним рівнянням 3-го порядку.

Задане диференціальне рівняння:

$$\frac{d^3 y(t)}{dt^3} + 2,5 \frac{d^2 y(t)}{dt^2} + 6 \frac{dy(t)}{dt} + 2,5 y(t) = e^{-t};$$

$$y(0) = 1;$$

$$y'(0) = -1;$$

$$y''(0) = 2.$$

Для зручності моделювання подамо дане диференціальне рівняння у формі Коші. Введемо нові змінні:

$$\begin{cases} x_1 = y \\ x_2 = \frac{dy}{dt} \\ x_3 = \frac{d^2 y}{dt^2} \end{cases}.$$

Подаємо рівняння за допомогою введених змінних:

$$\begin{cases} \frac{dx_1}{dt} = \frac{dy}{dt} = x_2 \\ \frac{dx_2}{dt} = \frac{d^2 y}{dt^2} = x_3 \\ \frac{dx_3}{dt} = \frac{d^3 y}{dt^3} = -2,5x_1 - 6x_2 - 2,5x_3 + e^{-t} \end{cases}$$

Оскільки операція диференціювання збільшує похибку (див. розділ 9), то за допомогою інтегрування система диференціальних рівнянь перетворюється на систему інтегральних рівнянь:

$$\begin{cases} x_1 = \int_0^t x_2 dt \\ x_2 = \int_0^t x_3 dt \\ x_3 = -2,5 \int_0^t x_1 dt - 6 \int_0^t x_2 dt - 2,5 \int_0^t x_3 dt - e^{-t} \end{cases}$$

Структурна схема моделі динамічної системи, побудована за звичайними правилами аналогової обчислювальної техніки, наведена на рис. 10.16.

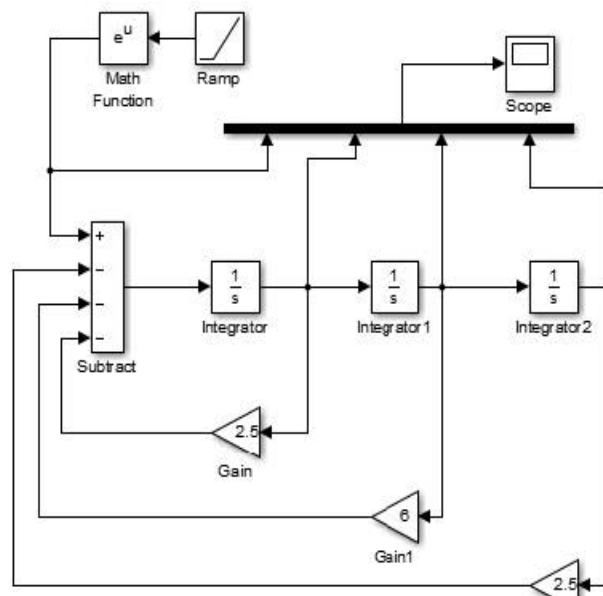


Рисунок 10.16 – Структурна схема моделі динамічної системи третього порядку

На виході першого інтегратора маємо сигнал x_3 , на виході другого – сигнал x_2 , на виході третього – сигнал x_1 . Коефіцієнти рівняння встановлюються у вікнах параметрів масштабних блоків Gain. Сигнали з відповідними коефіцієнтами надходять на суматор з від’ємними знаками, що відповідають рівнянню Коші. Початкові умови для функцій і похідних задаються у вікнах параметрів інтеграторів. Права частина диференціального рівняння, яка сформована за допомогою блоків лінійного сигналу Ramp (генератор аргументу $-t$) і блока функції Math Function, налаштованого на реалізацію експоненти, також надходить на суматор з додатним знаком. Візуалізація перехідного процесу наведена на екрані віртуального осцилографа (рис. 10.17).

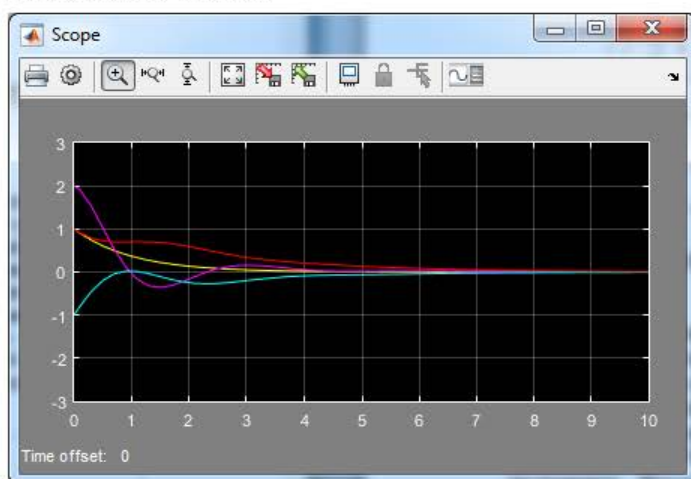


Рисунок 10.17 – Перехідний процес системи 3-го порядку

Приклад 4. Дуже поширеними зараз є задачі моделювання систем автоматичного управління. Розглянемо на прикладі. Нехай у нас вже є комбінована система, що містить релейний елемент, змодельована у середовищі SIMULINK (рис. 10.18).

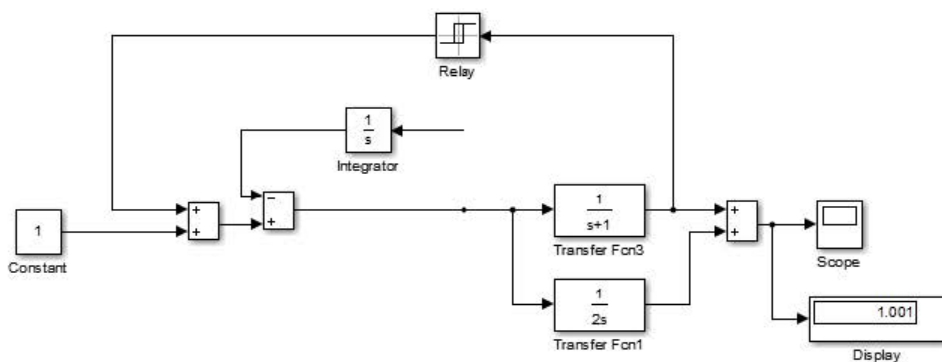


Рисунок 10.18 – Система автоматичного управління

Припустимо, що нам необхідно знайти сигнал на виході системи. Зробити це без застосування певних програмних додатків дуже важко. Крім того, виконання обчислень займає багато часу. Зокрема, для того, щоб знайти вихідний сигнал, необхідно спочатку знайти передатну функцію системи. Наступним кроком буде знаходження спектральної щільності потужності вхідного сигналу. Знаючи передатну функцію системи і спектральну щільність потужності вхідного сигналу, шляхом їх множення знайдемо спектральну щільність потужності вихідного сигналу. І нарешті, виконавши обернене перетворення Лапласа, отримаємо сам вихідний сигнал. Зрозуміло, що обрахування інтегралів Лапласа є досить трудомістким, звідки й випливає актуальність використання спеціалізованих програмних засобів. У нашому випадку дослідити вищенаведену систему можна лише шляхом імітаційного моделювання, оскільки вона не є лінійною.

Змоделювавши схему у SIMULINK, ми можемо як вивести чисельний результат, так і оцінити результат візуально за допомогою графіка на віртуальному осцилографі (рис. 10.19).

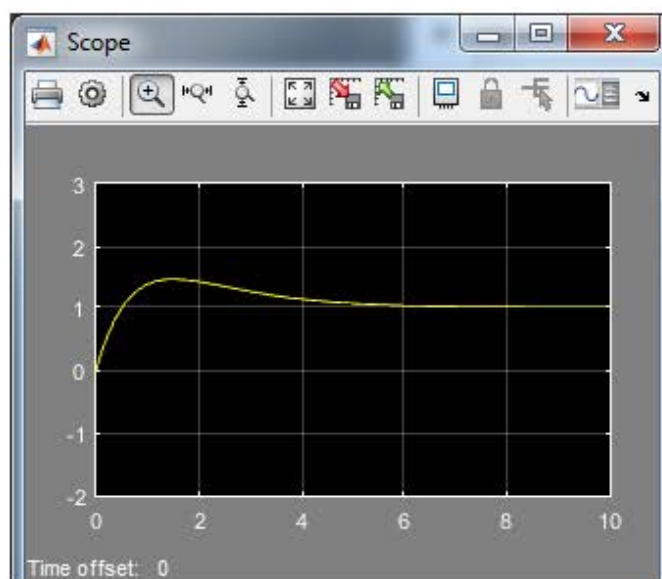



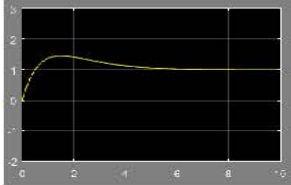

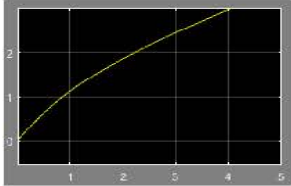

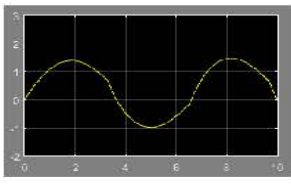

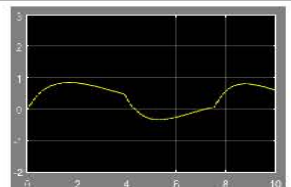

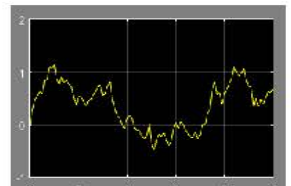

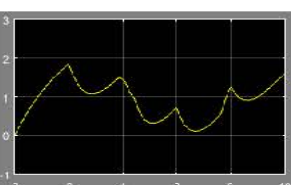
Рисунок 10.19 – Результат моделювання системи автоматичного управління

Розглянемо як змінюється сигнал на виході вищенаведеної системи залежно від різних видів вхідних впливів. Порівняльна характеристика залежностей вхідних і вихідних сигналів наведена у таблиці 10.1. В таблиці також зазначено параметри вхідних впливів.

Головною перевагою чисельних експериментів перед натурними експериментами є їх доступність та відносно низька вартість. Вони дозволяють без витрат на створення системи перевірити її характеристики в процесі майбутньої

експлуатації, обрати оптимальні параметри, прогнозувати поведінку системи і передбачати можливість виникнення критичних ситуацій.

Таблиця 10.1 – Результати моделювання для різних вхідних сигналів

| № | Вхідний сигнал | Вихідний сигнал |
|---|---|--|
| 1 |  (константа) Параметри: Constant value: 1 |  |
| 2 |  (лінійний сигнал) Параметри: Slope: 1 |  |
| 3 |  (синусоїда) Параметри: Amplitude: 1, Frequency: 1 |  |
| 4 |  (синусоїда) Параметри: Amplitude: 0.2, Frequency: 1 |  |
| 5 |  (випадкове число) Параметри: Mean: 0 Variance: 1 |  |
| 6 |  (повторювана послідовність) Параметри: Time Values: [0 2] Output Values: [0 2] |  |

Спеціалізована мова моделювання GPSS

GPSS World (GPSSW, General Purpose System Simulation World – Світова загальноцільова система моделювання), розроблена для ОС Windows.

GPSS була розроблена співробітником компанії IBM Джефрі Гордоном на початку 1960-х років. Не зважаючи на те, що система є досить старою, вона залишається актуальною і на сьогоднішній день, надаючи можливість використовувати одну з найпопулярніших мов моделювання у світі.

Цей програмний продукт увібрав у себе весь арсенал новітніх інформаційних технологій. Він містить розвинені графічні оболонки для створення моделей та інтерпретації вихідних результатів моделювання, засоби мультимедіа та відео, об'єктно-орієнтоване програмування та ін. В основу системи GPSS World покладена мова імітаційного моделювання GPSS (General Purpose System Simulation – Загальноцільова система моделювання).

Для студентів доступна безкоштовна версія програми GPSSW, яку можна скачати за посиланням: <http://gpss-world-student-version.software.informer.com/download/>.

Програма GPSS World має досить простий та зрозумілий інтерфейс користувача. При відкритті програми можна побачити пусту робочу область (рис. 10.20). Більшість функцій, що передбачені у даному додатку, поки не доступні. Для їх активації необхідно створити модель, натиснувши на іконку New або File→New, чи відкрити уже існуючий файл, натиснувши на іконку Open або обравши у меню пункт File→Open. При створенні нового файлу користувачу пропонується обрати створення моделі або текстового документа (як у елементарному текстовому редакторі).

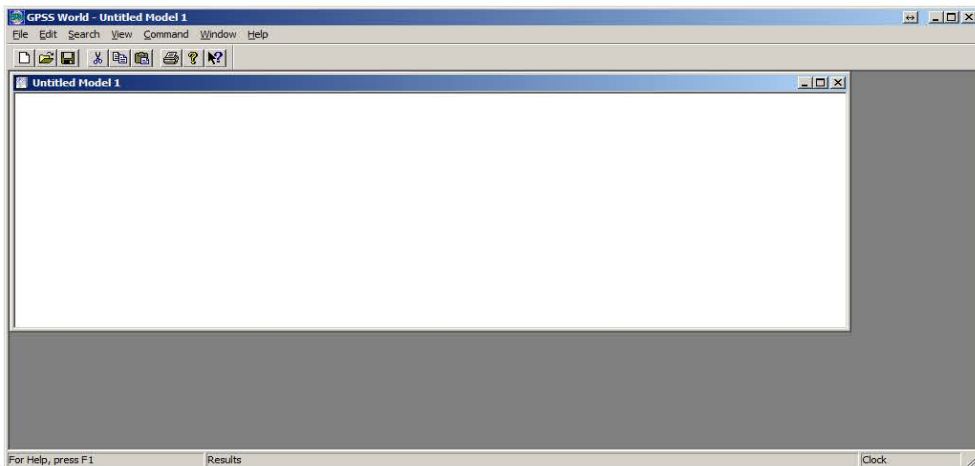


Рисунок 10.20 – Головне вікно середовища програмування GPSS

Пункт меню *Edit* крім стандартних операцій редагування (Undo, Cut, Copy, Paste, Insert line, Delete line, Font) містить наступні підпункти:

- Insert GPSS blocks – вставка блоків GPSS (рис. 10.21);

| | | |
|------------|----------|-------------|
| ADOPT | ASSEMBLE | ALTER |
| ADVANCE | CLOSE | COUNT |
| ASSIGN | GATE | DISPLACE |
| BUFFER | JOIN | EXAMINE |
| DEPART | LINK | EXECUTE |
| ENTER | LOGIC | FAVAIL |
| GENERATE | LOOP | FUNAVAIL |
| LEAVE | MATCH | GATHER |
| MARK | OPEN | INDEX |
| MSAVEVALUE | PREEMPT | INTEGRATION |
| PLUS | PRIORITY | SAVAIL |
| QUEUE | READ | SCAN |
| RELEASE | REMOVE | SELECT |
| SAVEVALUE | RETURN | SUNAVAIL |
| SEIZE | SEEK | TABULATE |
| SPLIT | TEST | TRACE |
| TERMINATE | UNLINK | UNTRACE |
| TRANSFER | WRITE | |

Рисунок 10.21 – Блоки GPSS

- Insert experiment – вставлення експерименту (рис. 10.22):
 - Screening – відображуваного;

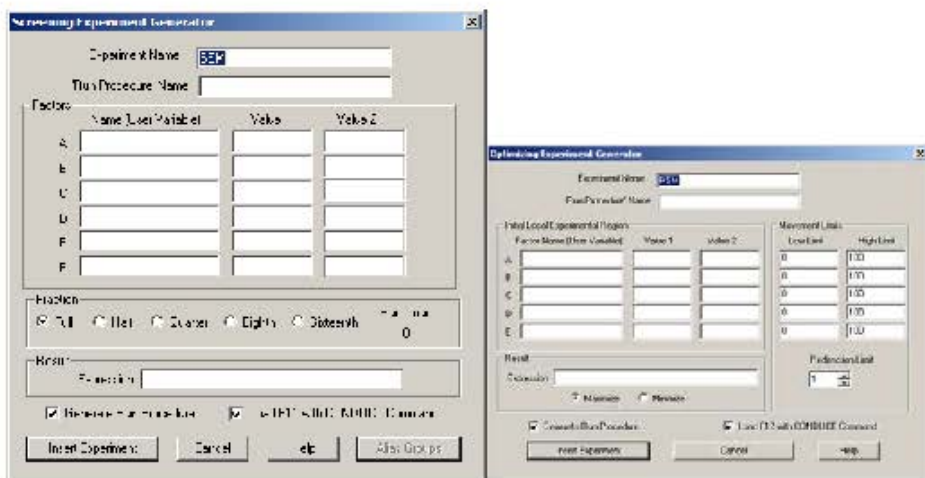


Рисунок 10.22 – Вікна експериментів

- Optimizing – з оптимізації;
 - Settings – налаштування роботи GPSS Word. Містить вкладки для налаштування параметрів моделювання, звітів, функціональних ключів, виразів тощо (рис. 10.23)

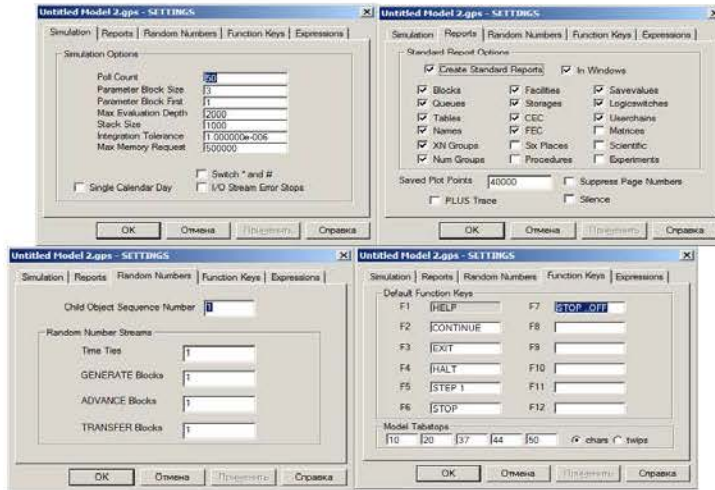


Рисунок 10.23 – Налаштування GPSS World

Пункт Search містить дії пов'язані зпошуком конкретних частин тексту, зокрема крім стандартних операцій (Find/Replace, Go to line, Next bookmark, Mark, Unmark, Unmark all, Select to bookmark) містить засоби пошуку помилок:

- Next error – наступна помилка (доступно у режимі компіляції);
- Previous error – попередня помилка;

Пункт View:

- Toolbar – визначає чи відобразити панель інструментів;
- Entity details – визначає чи відобразити параметри елементів програми;
- Simulation clock – визначає чи відобразити час моделювання.

Пункт Command містить у собі основні команди для керування плином програми:

- Create simulation (Ctrl + Alt + S) – створити модель – запускає роботу написаної програми;
- Retranslate (Ctrl + Alt + R) – ретранслювати програму;
- Repeat last command (Ctrl + Alt + L) – повторити останню команду;
- CONDUCT – почати експеримент. Це – миттєва команда, що посилається призупиненому процесу моделювання;
- START – почати моделювання;
- STEP 1 (Ctrl + Alt + 1) – перейти до першого кроку;
- HALT (Ctrl + Alt + H) – перервати виконання програми. Ця команда не лише зупиняє моделювання, але і видаляє усі команди, що розміщені у черзі команд моделювання;
- CONTINUE (Ctrl + Alt + C) – продовжити виконання програми;
- CLEAR – очистити статистику (дана команда повертає модель до початкового стану);
- RESET – скидання статистики;
- SHOW – розрахувати та відобразити вираз;

– Custom – введення деякої команди.

Пункт Window – дозволяє використати різні варіанти подання моделі, зокрема

– Simulation window – надає доступ до великого різноманіття форм представлення результатів моделювання, які користувач може обрати (вікна текстового та модельного редактора, журнал подій, вікно блоків, вікно відображення значень виразів, динаміки роботи модельованих пристроїв, логістичних перемикачів, вікна для відображення матриць, графіків, черг, збережених значень, накопичувачів і таблиць);

– Simulation snapshots – вікна знімків моделювання – відображає стан моделі у деякі моменти часу (знімки відображення транзакцій моделювання, стану ланцюга поточних дій, ланцюга майбутніх дій, стан числових груп моделі, ланцюги сутностей користувачів).

Для кращого розуміння моделювання мовою GPSS необхідно мати уявлення про категорії об'єктів, що використовуються при створенні моделей. Існує 7 категорій об'єктів у мові GPSS:

– *Динамічна категорія*. Ця категорія містить у собі транзакти. Це – об'єкти, що являють собою аналог потоків із певними характеристиками. Вони створюються у деяких точках моделі та просуваються за допомогою планувальника блоками та, згодом, знищуються;

– *Операційна категорія*. Ця категорія містить у собі блоки. Вони визначають логіку функціонування моделі системи і визначають шляхи руху транзактів між об'єктами апаратної категорії. Існують блоки для створення та знищення транзактів, керування числовими атрибутами об'єкта, затримку транзакту на деякий період часу, зміна маршруту переміщення транзакту у моделі, організації введення та виведення інформації, групування і т. д.;

– *Апаратна категорія*. Її об'єкти являють собою складові модельованої системи. Вони мають деякі характеристики і можуть бути змінені під дією транзактів. Апаратна категорія містить в собі логічні ключі, одно- та багатоканальні пристрої;

– *Обчислювальна категорія* містить змінні, функції та генератори випадкових чисел;

– *Статистична категорія* містить черги та таблиці;

– *Запам'ятовувальна категорія* включає у себе комірки та матриці комірок;

– *Групуюча категорія* – числові групи, групи транзактів та списки.

Опишемо деякі стандартні блоки, які необхідні для створення нескладної моделі. Деякі із них згадувалися раніше при описі пункту меню Command.

Блок START. Відповідає за початок моделювання та присвоєння лічильнику Termination Count деякого значення, заданого параметром.

Блок GENERATE. Даний блок створює та вводить у модель транзакти (групами або по одному) через задані інтервали часу. Кожна модель повинна містити хоча б 1 блок GENERATE. Обмежень на їх кількість немає.

Блок TERMINATE. Даний блок видаляє транзакти з моделі. Даний блок має необов'язковий операнд, що визначає необхідність зменшити лічильник Termination Count.

Блок STEP намагається здійснити обмежену кількість входів у блоки.

Блок STORAGE. Даний блок визначає сутність Storage. Його перший параметр визначає ім'я деякого одноканального пристрою, а другий – кількість приборів даного пристрою або місткість об'єкта.

Блок ADVANCE затримує транзакт на час, визначений операндом. Причому, можна задати приблизний час затримки із деяким відхиленням, що визначається іншим операндом.

Блок SEIZE моделює зайняття транзактом одноканального пристрою. Час же затримки визначається блоком ADVANCE, описаним вище. Єдиний параметр блока SEIZE визначає назву пристрою, який буде зайнято.

Блок RELEASE дозволяє звільнити пристрій, ім'я якого задане єдиним операндом блока, що був зайнятий при використанні блока SEIZE.

Блок QUEUE поміщає транзакт у кінець черги. Причому, його перший параметр являє собою ім'я черги, а другий – кількість елементів, що додаються у чергу (за замовчуванням дане число становить 1).

Блок DEPART (LEAVE) дозволяє транзакту покинути чергу. Перший параметр визначає чергу, яку покидає транзакт. Другий параметр – це кількість одиниць, на яку зменшується черга (за замовчуванням це значення становить 1).

Блок INITIAL ініціалізує чи модифікує сутності Logicswitch, Savevalue чи Matrix.

Блок CLEAR обнулює усю статистику та звільняє транзакти.

Блок CONDUCT запускає до виконання експеримент.

Блок CONTINUE дозволяє продовжити виконання призупиненого експерименту.

Блок EQU присвоює деяке значення змінній, створеній користувачем.

Блок EXIT завершує сеанс роботи у системі GPSS World.

Блок HALT перериває процес моделювання та видаляє усі команди у черзі.

Блок SHOW обробляє та виводить на екран значення виразу.

Блок INCLUDE дає змогу зчитати та транслювати допоміжний файл моделі.

Блок RESET повністю обнуляє статистику моделі.

Блок REPORT надає запит на виведення поточного звіту.

Також використовуються блоки TABLE, B VARIABLE (булева змінна), FUNCTION, MATRIX та інші, що використовуються для визначення різних типів об'єктів (як матриці та змінні) і режимів їх використання.

Розглянемо *приклад* моделювання процесу обслуговування запитів в СМО на наступному прикладі (рис. 10.24). Нехай на вхід надходять запити кожних 5 ± 1 секунди. Черга працює за принципом FIFO. Обробка кожного запиту займає 3 ± 1 секунди. Змоделюємо роботу даної системи протягом 400 секунд та виведемо статистику її роботи.

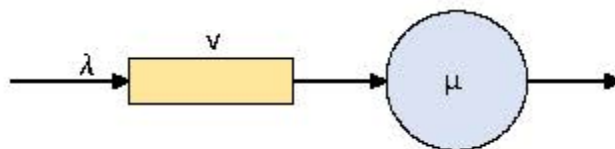


Рисунок 10.24 – Приклад одноканальної СМО з чергою

Для виконання описаного вище завдання створимо імітаційну модель. Її код наведено нижче:

```

GENERATE 5, 2 ; Моделюємо надходження запитів
QUEUE SYSQ ; Запити потрапляють у чергу названу
                SYSQ, де вони очікують обробки
SEIZE SYSPR ; Запити надходять на обробку процесором
                SYSPR
DEPART SYSQ ; При цьому вони виходять із черги
ADVANCE 3, 1 ; Моделюємо обробку запиту процесором
RELEASE SYSPR ; Звільняємо його, щоб дозволити обро-
                бку наступних запитів
TERMINATE ; Дозволяємо транзакту закінчити роботу у
                даному блоці програми
GENERATE 400 ; визначаємо час моделювання. Таким чи-
                ном транзакт прийде у момент часу 400
TERMINATE 1 ; Завершення моделювання

```

Синтаксис даної мови дозволяє писати коментарі у рядку після символу «;».

Оберемо пункт меню Command→Create Simulation для створення моделі та Command→START 1, щоб запустити виконання моделі з першої секунди. У результаті ми побачимо зображене на рис. 10.25.

За замовчуванням виводиться журнал, в якому виконується реєстрація подій, пов'язаних із моделюванням, та звіт, що описує статистику роботи моделі.

Для нас важливі також такі значення, отримані у процесі моделювання:

- START TIME – час початку моделювання;
- END TIME – час закінчення моделювання;
- BLOCKS – саме стільки блоків було використано нами для моделювання даного процесу;
- FACILITY – перелік працюючого обладнання, що обслуговує користувачів (у нашому випадку це один обслуговуючий процесор FACILITIES = 1);
- DELAY = 0 – усі запити, що надійшли до моменту закінчення моделювання, були оброблені;

- ENTRIES = 80 – кількість входжень запитів для обслуговування до пристрою;
- UTIL = 0.593 – коефіцієнт використання пристрою – пристрій працює 59,3% усього часу;
- AVE. TIME = 2.964 – показує, скільки часу у середньому пішло на обслуговування одного запиту;
- AVAIL = 1 – доступність процесора.
- QUEUE – черга (STAND):
- MAX = 1 – максимальна місткість черги за час роботи;
- CONT. = 0 – означає, що на момент закінчення моделювання у черзі не залишилося жодного запиту;
- ENTRY = 80 – кількість входжень у чергу;
- ENTRY(0) = 80 – кількість входжень без очікування у черзі;
- AVE. TIME = 0.000 – середній час очікування у черзі.

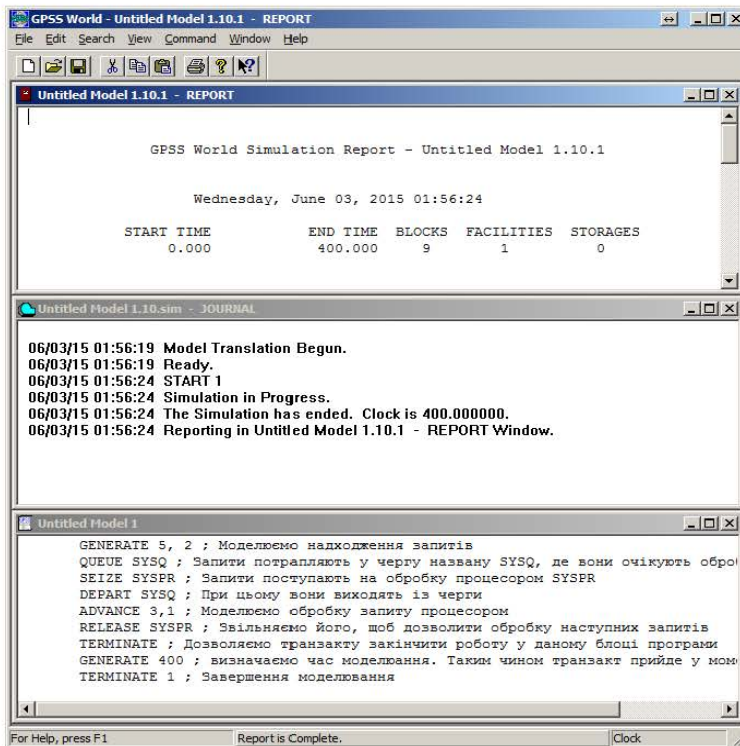


Рисунок 10.25 – Результати моделювання

З цих результатів ми можемо зробити висновок про те, що запити стабільно та швидко обробляються без необхідності простоювання у черзі на обробку. Процесор простоює $(100-59,3) = 40,7\%$ часу своєї роботи, отже, можна підвищувати навантаження на нього. Якщо ж зменшити час надходження запитів у систему до 1 ± 2 секунд, а час їх обробки збільшити до 13, то результати уже будуть

гіршими. Кількість запитів, що потраплять на обробку, зменшиться аж до 31, коефіцієнт використання пристрою обслуговування зросте до 99,5%. Максимальна кількість запитів, що простоюватимуть у черзі, становитиме 367, причому, середній час очікування перевищуватиме 183 секунди, що є неприпустимим.

Також можна більш докладно розглянути результати моделювання, використовуючи різні типи вікон, що пропонуються системою GPSS World (рис. 10.26).

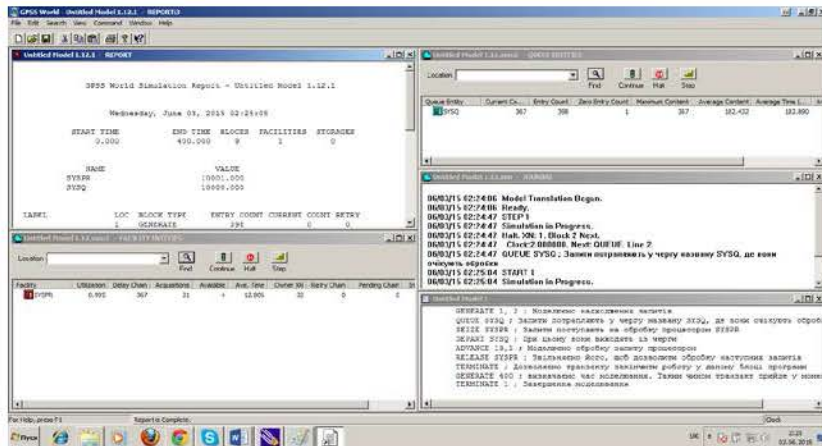


Рисунок 10.26 – Вікна перегляду програми, даних тощо

Таким чином система GPSS World дозволяє моделювати процеси у системах масового обслуговування та вираховувати можливий вплив зміни їх окремих параметрів на діяльність систем в цілому.

Система GPSS World надає можливість виводити у графічному вигляді дані за великою кількістю параметрів. Кожен тип об'єктів GPSS має власний набір стандартних числових атрибутів (СЧА):

- Системні об'єкти:
 - C1 – поточне значення умовного часу;
 - AC1 – поточне значення абсолютного часу;
- Транзакти:
 - P – значення параметра активного транзакту;
 - PR – пріоритет активного транзакту;
 - M1 – час перебування у моделі активного транзакту. Обчислюється як різниця поточного значення абсолютного часу та часу народження активного транзакту;
 - MP – транзитний час перебування у моделі активного транзакту. Розраховується як різниця поточного значення абсолютного часу та вміщуваного параметра активного транзакту;
 - XN1 – номер активного транзакту.
- Блоки:

- N – загальна кількість транзактів, що увійшли в блок;
- W – загальна кількість транзактів у блоці.
- Прилади:
 - F – Стан приладу (0 – вільний, 1 – усі інші випадки);
 - FV – прапорець готовності приладу до використання (1 – готовий, 0 – ні);
 - FR – коефіцієнт використання приладу;
 - FC – загальна кількість моментів заняття приладу;
 - FT – середній час використання приладу однією задачею;
- Об'єкти пам'яті.
- Ключі.
- Черги:
 - Q – поточна довжина черги;
 - QA – зважена за часом середня довжина черги;
 - QM – максимальна довжина черги;
 - QC – загальна кількість входжень у чергу;
 - QZ – число нульових входів у чергу;
 - QT – середній час перебування транзактів у черзі (враховуючи нульові входи);
 - QX – середній час перебування транзактів у черзі (без нульових входів);
- Таблиці.
- Генератори випадкових чисел – RN – число, вираховане генератором.
- Списки користувачів.

За умов попереднього прикладу запити надходили на вхід системи приблизно кожних 5 секунд, а обробка 1 запиту займала всього 3 секунди. Таким чином, запитам не доводилося простоювати у черзі. Це видно із нижченаведеного графіка рис. 10.27.

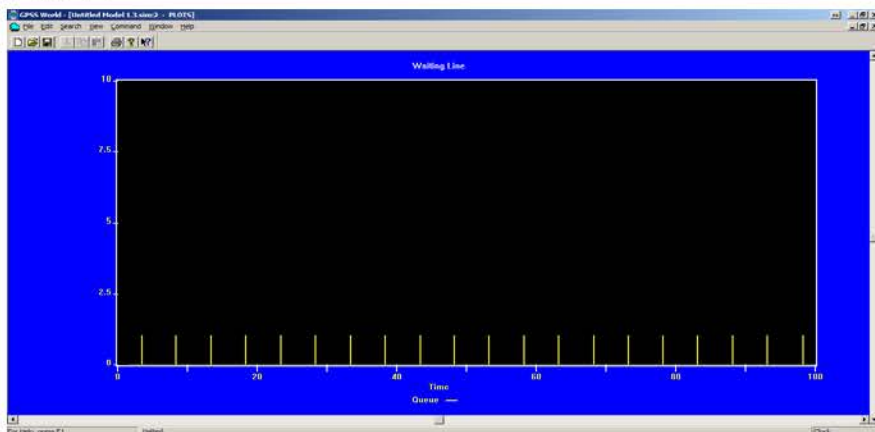


Рисунок 10.27 – Приклад графічного подання результатів

Запит потрапляв у чергу і одразу ж із неї виходив та обслуговувався. Внесемо деякі зміни у програму. Змінимо значення 2 рядків:

```
GENERATE 4, 2 ; Тепер запити приходитимуть частіше,
приблизно, раз у 4 секунди
QUEUE SYSQ
SEIZE SYSPR
DEPART SYSQ
ADVANCE 6,1 ; Збільшимо час обробки запитів до, при-
близно, 6 секунд
RELEASE SYSPR
TERMINATE
GENERATE 400
TERMINATE 1
```

Як ми бачимо з графіка рис. 10.28, незважаючи на те, що запити виходять із черги на обслуговування, вони недостатньо швидко обробляються. У результаті ми можемо побачити постійне накопичення запитів.

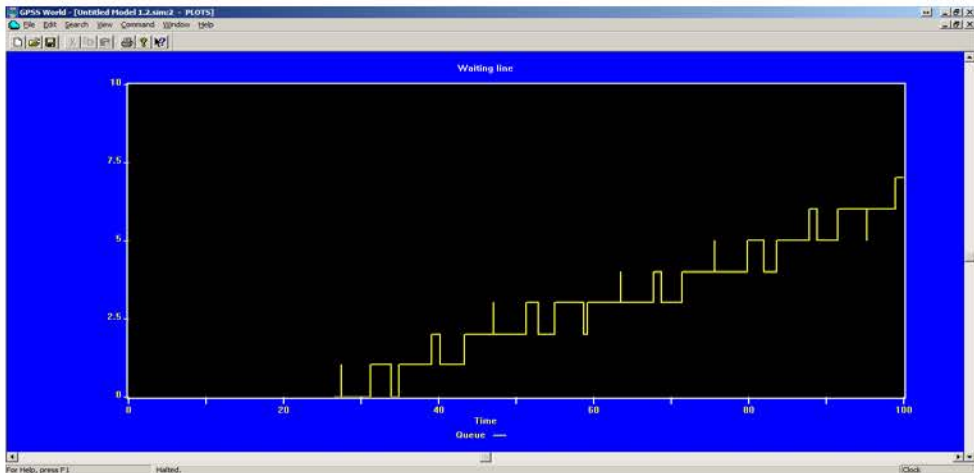


Рисунок 10.28 – Приклад графічного подання результа-