

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

**Лабораторна робота № 2**

з «Методи моделювання та оптимізації безпечних комп'ютерних систем»

(назва дисципліни)

на тему: «Імітаційне моделювання складських приміщень, розподільчих центрів з використанням системної динаміки в NetLogo»

Виконав: студент 5 курсу групи № 555 ім  
напряму підготовки (спеціальності)

125 Кібербезпека та захист  
інформації

(шифр і назва напряму підготовки (спеціальності))

Орлов Станіслав Валерійович

(прізвище й ініціали студента)

Прийняв: д.т.н., професор

Морозова Ольга Ігорівна

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

**Тема:** Імітаційне моделювання складських приміщень, розподільчих центрів з використанням системної динаміки в NetLogo

**Мета роботи:** дослідити динаміку процесів шляхом імітаційного моделювання.

**Постановка завдання:** побудувати імітаційну модель (виробництво, склад, магазин, покупець), яка має певну кількість рівнів. Обсяг потоку вибрати самостійно. За допомогою інструменту Plot отримати динаміку значень стоків.

N – номер варіанта завдання,  $N = 22 (20 + 2)$

2	Ремонт техніки: Виробництво → Запчастини → Склад → Замовлення	3
---	--	---

Відкриємо середовище NetLogo та бачимо наступне вікно розміщення об'єктів.

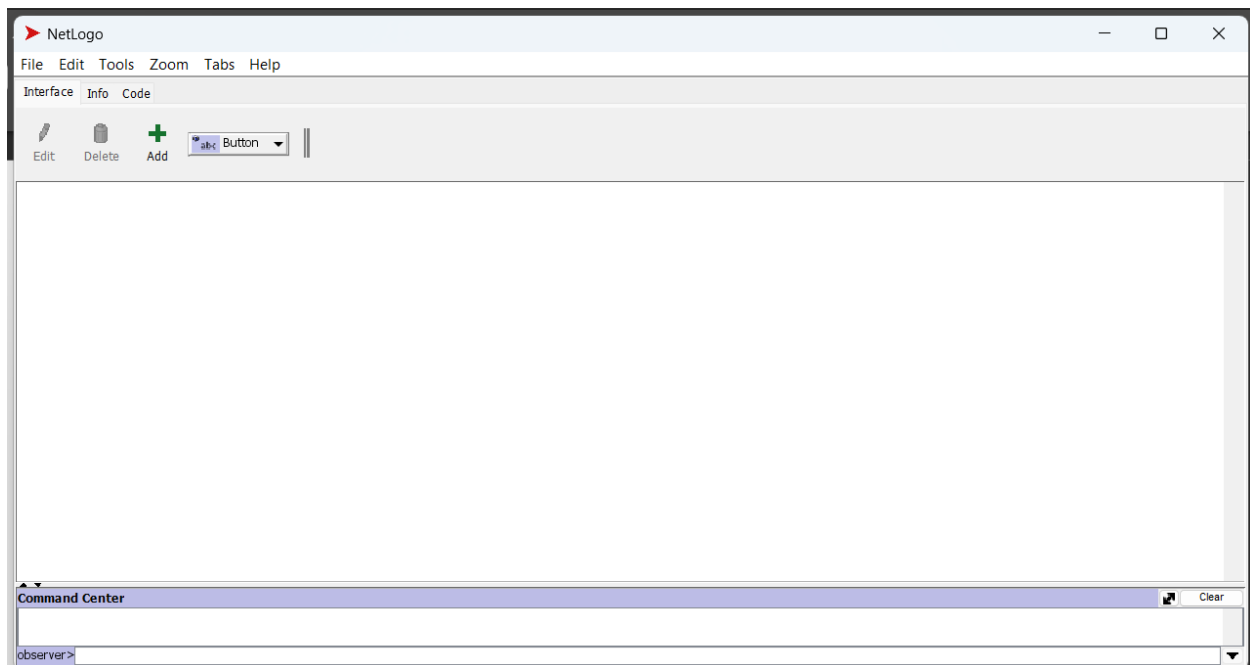


Рисунок 1 – основне вікно середовища NetLogo

Тепер відкриємо вікно моделювання системної динаміки для розміщення об'єктів.

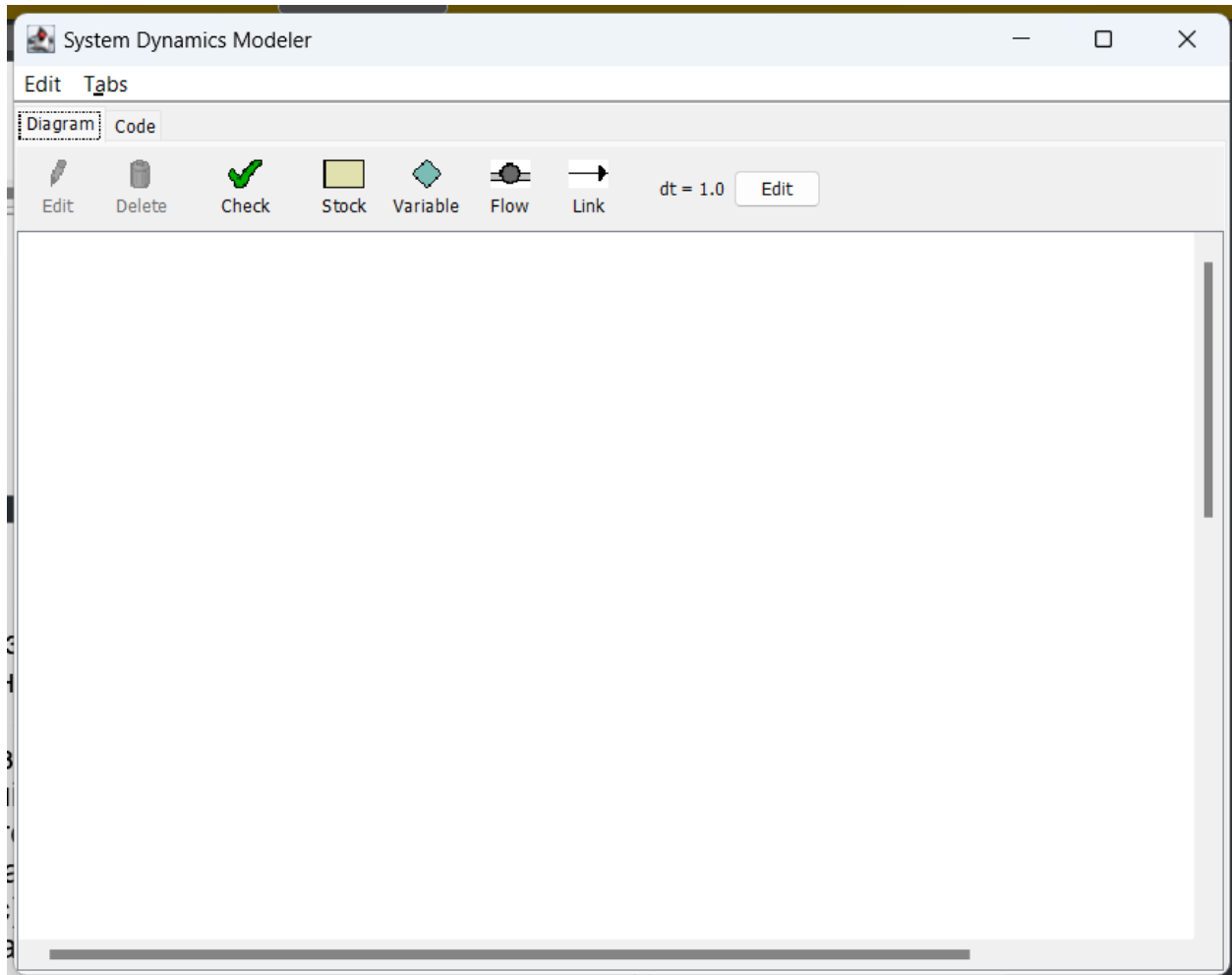


Рисунок 2 – вікно моделювання середовища NetLogo

Згідно поставленого завдання побудуємо модель третього рівня переміщення продукції за наступним маршрутом:

Виробництво → Запчастини → Склад → Замовлення

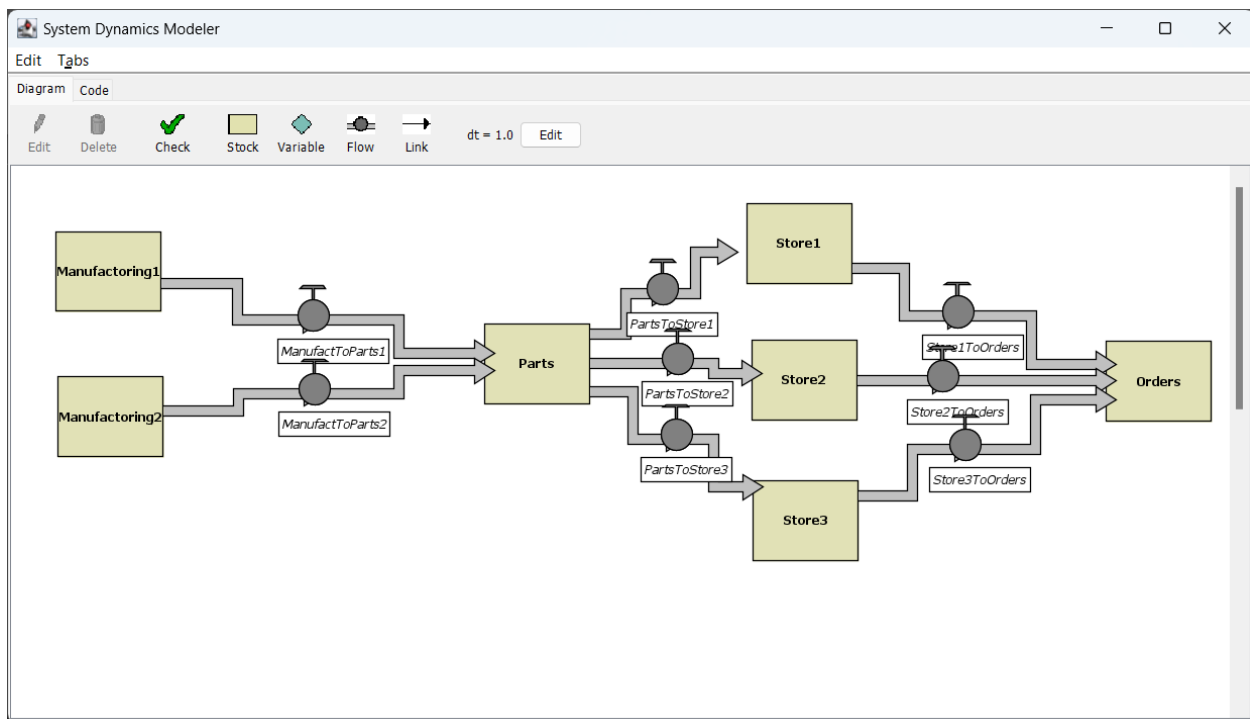


Рисунок 3 – побудована модель 3-го рівня у системі NetLogo

```

System Dynamics Modeler
Edit  Tabs
Diagram  Code

) * dt
end

;; Plot the current state of the system dynamics model's stocks
;; Call this procedure in your plot's update commands.
to system-dynamics-do-plot
  if plot-pen-exists? "Manufacturing1" [
    set-current-plot-pen "Manufacturing1"
    plotxy ticks Manufacturing1
  ]
  if plot-pen-exists? "Manufacturing2" [
    set-current-plot-pen "Manufacturing2"
    plotxy ticks Manufacturing2
  ]
  if plot-pen-exists? "Orders" [
    set-current-plot-pen "Orders"
    plotxy ticks Orders
  ]
  if plot-pen-exists? "Parts" [
    set-current-plot-pen "Parts"
    plotxy ticks Parts
  ]
  if plot-pen-exists? "Store1" [
    set-current-plot-pen "Store1"
    plotxy ticks Store1
  ]
  if plot-pen-exists? "Store2" [
    set-current-plot-pen "Store2"
    plotxy ticks Store2
  ]
  if plot-pen-exists? "Store3" [
    set-current-plot-pen "Store3"
    plotxy ticks Store3
  ]
end

```

Рисунок 4 – вкладка код у системі NetLogo

```

;; System dynamics model globals
globals [
  ;; stock values
  Manufacturing1
  Manufacturing2
  Orders
  Parts
  Store1
  Store2
  Store3
  ;; size of each step, see SYSTEM-DYNAMICS-GO
  dt
]

;; Initializes the system dynamics model.
;; Call this in your model's SETUP procedure.
to system-dynamics-setup
  reset-ticks
  set dt 1.0
  ;; initialize stock values
  set Manufacturing1 15
  set Manufacturing2 25
  set Orders 0
  set Parts 0
  set Store1 0
  set Store2 0
  set Store3 0
end

;; Step through the system dynamics model by performing next iteration of Euler's
method.
;; Call this in your model's GO procedure.
to system-dynamics-go

  ;; compute variable and flow values once per step
  let local-ManufactToParts1 ManufactToParts1
  let local-ManufactToParts2 ManufactToParts2
  let local-PartsToStore1 PartsToStore1
  let local-PartsToStore2 PartsToStore2
  let local-PartsToStore3 PartsToStore3
  let local-Store1ToOrders Store1ToOrders
  let local-Store2ToOrders Store2ToOrders
  let local-Store3ToOrders Store3ToOrders

  ;; update stock values
  ;; use temporary variables so order of computation doesn't affect result.
  let new-Manufacturing1 ( Manufacturing1 - local-ManufactToParts1 )
  let new-Manufacturing2 ( Manufacturing2 - local-ManufactToParts2 )
  let new-Orders ( Orders + local-Store1ToOrders + local-Store2ToOrders + local-
Store3ToOrders )
  let new-Parts ( Parts + local-ManufactToParts1 + local-ManufactToParts2 - local-
PartsToStore1 - local-PartsToStore2 - local-PartsToStore3 )
  let new-Store1 ( Store1 - local-Store1ToOrders )
  let new-Store2 ( Store2 + local-PartsToStore2 - local-Store2ToOrders )
  let new-Store3 ( Store3 + local-PartsToStore3 - local-Store3ToOrders )
  set Manufacturing1 new-Manufacturing1
  set Manufacturing2 new-Manufacturing2
  set Orders new-Orders

```

```

    set Parts new-Parts
    set Store1 new-Store1
    set Store2 new-Store2
    set Store3 new-Store3

    tick-advance dt
end

;; Report value of flow
to-report ManufactToParts1
  report ( Manufacturing1 * 0.1
    ) * dt
end

;; Report value of flow
to-report ManufactToParts2
  report ( Manufacturing2 * 0.2
    ) * dt
end

;; Report value of flow
to-report PartsToStore1
  report ( Parts * 0.33
    ) * dt
end

;; Report value of flow
to-report PartsToStore2
  report ( Parts * 0.33
    ) * dt
end

;; Report value of flow
to-report PartsToStore3
  report ( Parts * 0.33
    ) * dt
end

;; Report value of flow
to-report Store1ToOrders
  report ( Store1 * 0.5
    ) * dt
end

;; Report value of flow
to-report Store2ToOrders
  report ( Store2 * 0.35
    ) * dt
end

;; Report value of flow
to-report Store3ToOrders
  report ( Store3 * 0.15
    ) * dt
end

;; Plot the current state of the system dynamics model's stocks
;; Call this procedure in your plot's update commands.

```

```
to system-dynamics-do-plot
  if plot-pen-exists? "Manufacturing1" [
    set-current-plot-pen "Manufacturing1"
    plotxy ticks Manufacturing1
  ]
  if plot-pen-exists? "Manufacturing2" [
    set-current-plot-pen "Manufacturing2"
    plotxy ticks Manufacturing2
  ]
  if plot-pen-exists? "Orders" [
    set-current-plot-pen "Orders"
    plotxy ticks Orders
  ]
  if plot-pen-exists? "Parts" [
    set-current-plot-pen "Parts"
    plotxy ticks Parts
  ]
  if plot-pen-exists? "Store1" [
    set-current-plot-pen "Store1"
    plotxy ticks Store1
  ]
  if plot-pen-exists? "Store2" [
    set-current-plot-pen "Store2"
    plotxy ticks Store2
  ]
  if plot-pen-exists? "Store3" [
    set-current-plot-pen "Store3"
    plotxy ticks Store3
  ]
end
```

Рисунок 5 – код побудованої моделі

Отримаємо динаміку значень стоків за допомогою інструмента Plot.

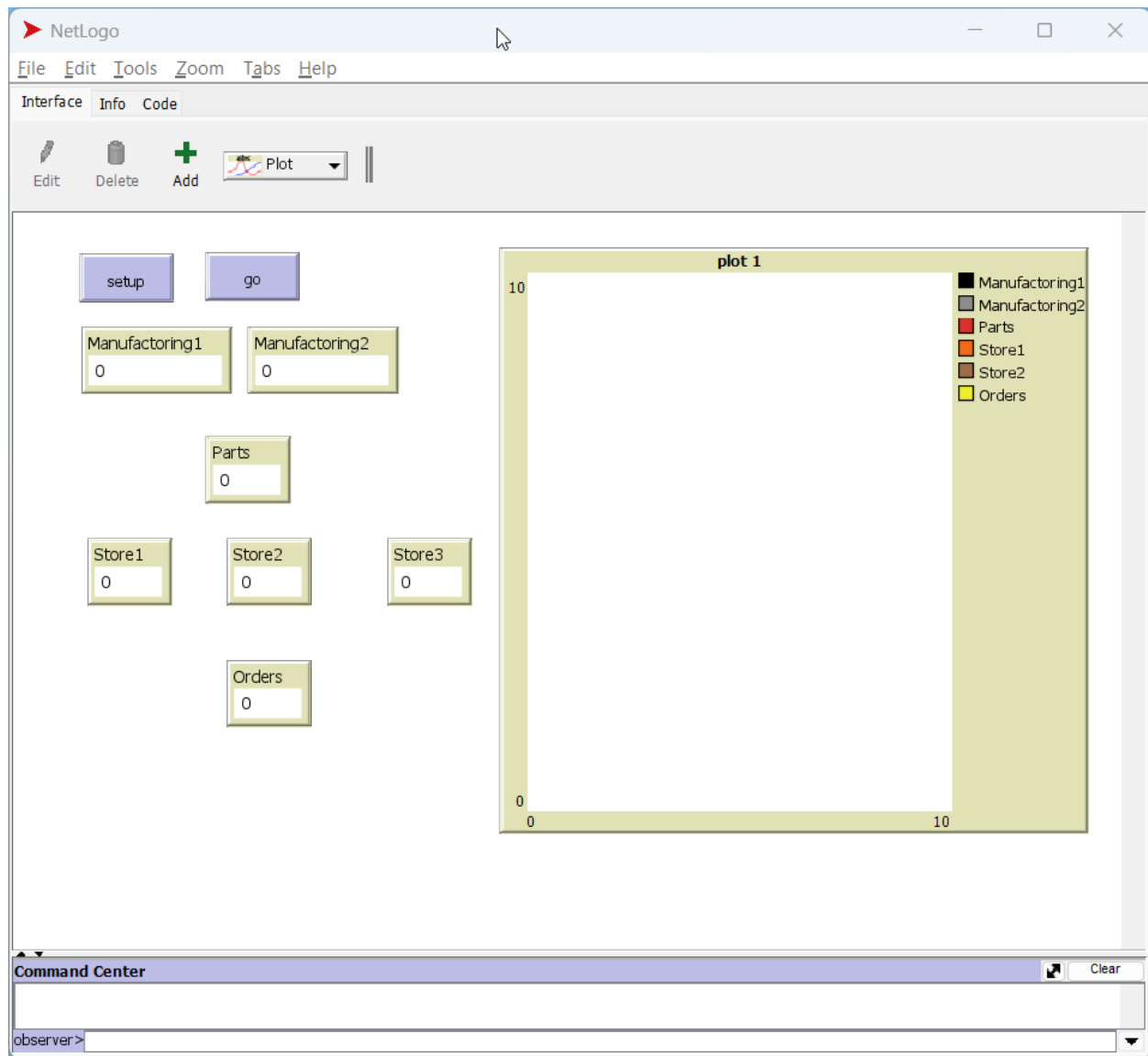


Рисунок 6 – динаміка значень стоків за допомогою інструмента Plot  
(початкова стадія)



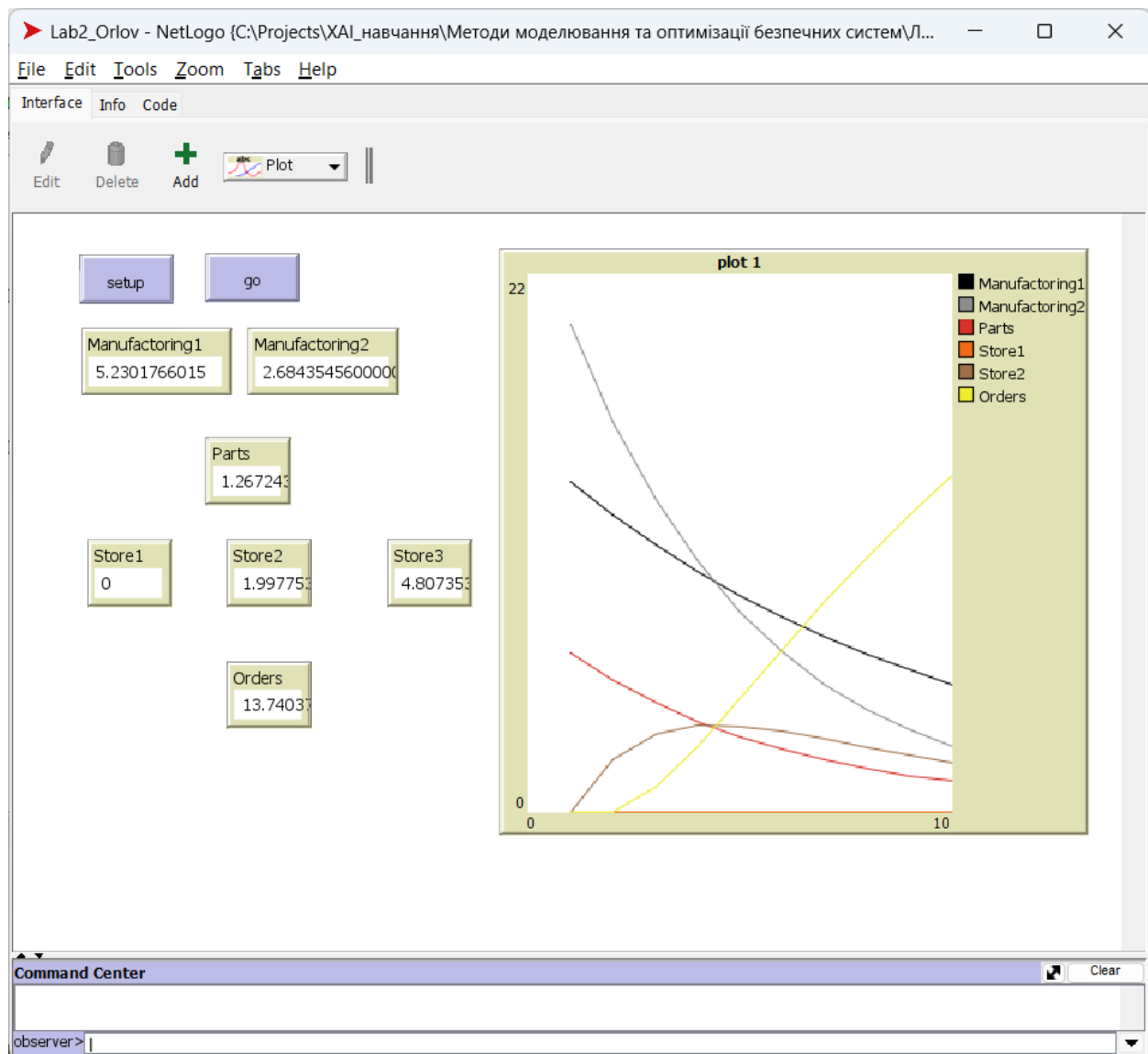


Рисунок 7 – динаміка значення стоків за допомогою інструмента Plot

**Висновок:** у ході виконання лабораторної роботи було досліджено динаміку процесів шляхом імітаційного моделювання. Було побудовано імітаційну модель, яка має певну кількість рівнів. За допомогою інструменту Plot отримано динаміку значень стоків.