Лекція 20.

Технологія використання комп'ютерних моделей

Як ми вже відзначали в лекції 01 «Поняття моделювання. Способи подання моделей», моделі будуються для рішення певних задач, тому тут ми розглянемо типи таких задач і те, як у них використаються моделі, які ми вже навчилися будувати.

Отже, модель — закономірність, що перетворить вхідні значення у вихідні: $Y = M(X)$. Під цим можна розуміти таблицю, графік, вираження з формул, закон (рівняння) і т.д. Це питання способу запису закономірності. У нашому курсі й далі в курсі «Моделі й методи штучного інтелекту» ми докладно покажемо, як переходити від одного типу запису до іншого.

Під Y у системотехніку розуміють деякого дослідника, що цікавить, або власника системи показник. Кожна система існує або створюється, щоб реалізувати певну мету. Немає систем без цілей. От ціли-те і є вихідним, останнім параметром у ланцюзі перетворень від входу до виходу, що може нас цікавити, тому що заради нього, властиво, і проделываются всі перетворення. Ті змінні, які якось не зв'язані по ланцюгах з вихідним показником, не ставляться до розглянутої системи й повинні бути відкинуті.

Представим нашу систему як граф. Це можливо, тому що система є елементи й зв'язки між ними, що відповідає вершинам і дугам графа. Подальший виклад матеріалу будемо вести на прикладі графа, зображеного на мал. 20.1.

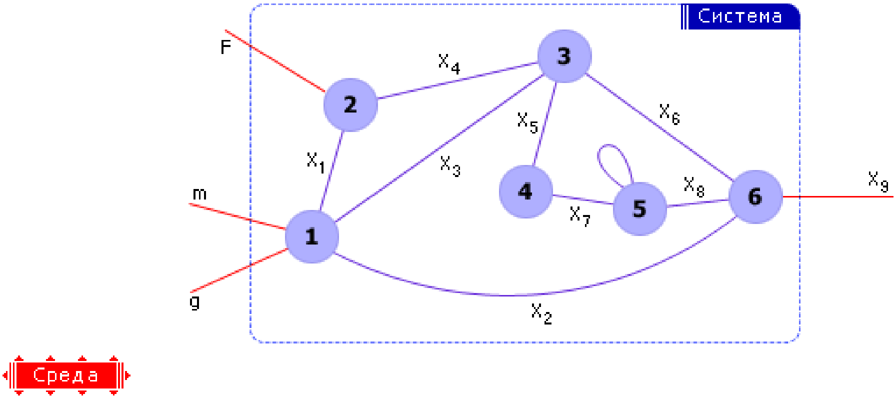

Рис. 20.1. Подання системи у вигляді графа

Елементи системи описуються законами, тобто рівняннями (змінні, операції між ними й знаки уравнивания) або системами рівнянь у загальному випадку, що відповідає вершинам графа. Якщо рівнянь у вершині трохи, то цю вершину завжди, при бажанні, знову можна буде розбити на подграф, де кожній вершині вже буде відповідати тільки одне рівняння. Зв'язку графа вказують на зв'язку елементів системи між собою, тобто зв'язок відповідає загальної для двох вершин змінної. Отже, кожна вершина асоціюється з формулою (наприклад, присвоєнням вираження або рівнянням), що зв'язує змінну вершини з іншими змінними, доступними їй по її зв'язках.

Якщо граф досить докладний, такий, що кожній вершині відповідає тільки одне рівняння, то можна асоціювати вершини зі змінними. Одна вершина — одна змінна. Перетворений у такий спосіб граф ми вже бачили в лекции 11. Перехід від однієї форми подання графа до іншої можливий завжди.

Зверніть увагу: частина зв'язків перебуває усередині графа — це внутрішні зв'язки системи. Частина зв'язків графа зв'язує **змінні системи** X із **зовнішніми змінними**, які не є частиною системи, а є частиною середовища. Ці зв'язки перетинають границі графа, границі системи.

Для моделювання дуже важливо визначити, де проходить ця границя, що буде підлягати моделюванню, а що ні. Що буде описано причинно-наслідковими зв'язками, а що ні й так і залишиться нескінченно більшим і непізнаним. Ще раз звернемося до тексту лекції 01: «…«Модель — пошук кінцевого в нескінченному» — ця думка належить Д. И. Менделєєву. Що відкидається, щоб перетворити нескінченне в кінцеве? У модель включаються тільки істотні аспекти, що представляють об'єкт, і відкидаються *всі інші* (нескінченна більшість)…»

Отже, границя відокремлює кінцеве (система) від нескінченного (середовище). Границя ця може проходити в декількох різних місцях. Це говорить про те, що модель може граничити з декількома областями, що не мають опису, які задаються стосовно системи тільки що як впливають на неї сигнали, як дані, але не як закони (див. мал. 1.12). Нескінченність може виявитися й усередині системи, тоді така система називається відкритої (див. лекцію 11).

Заважимо, що граф є сильно зв'язаним утворенням, кількість зв'язків у такому графі більше чим кількість вершин. У складних системах зв'язків набагато більше, ніж вершин. Для нормального графа кожна його вершина повинна мати зв'язок з будь-якою іншою вершиною графа через ланцюжок зв'язків. Якщо в складі графа ви виявили незв'язані між собою шматки, то модель системи некоректна або ви маєте справу із двома або декількома незалежними системами. Якщо граф великий, то має сенс розрізати його на невеликі подграфы й вивчати їх окремо (див. мал. 20.2). Логічно розрізати граф по таких лініях, щоб при цьому розривалося якнайменше зв'язків, і виходило якнайбільше окремих шматків (подграфов). Помітимо, що цими діями ми, фактично, привели граф до ієрархічної форми. Ієрархія - це спосіб боротьби зі складністю досліджуваної системи. У цьому випадку між лініями розрізу залишають одну вершину, структуру якої розшифровують окремо.

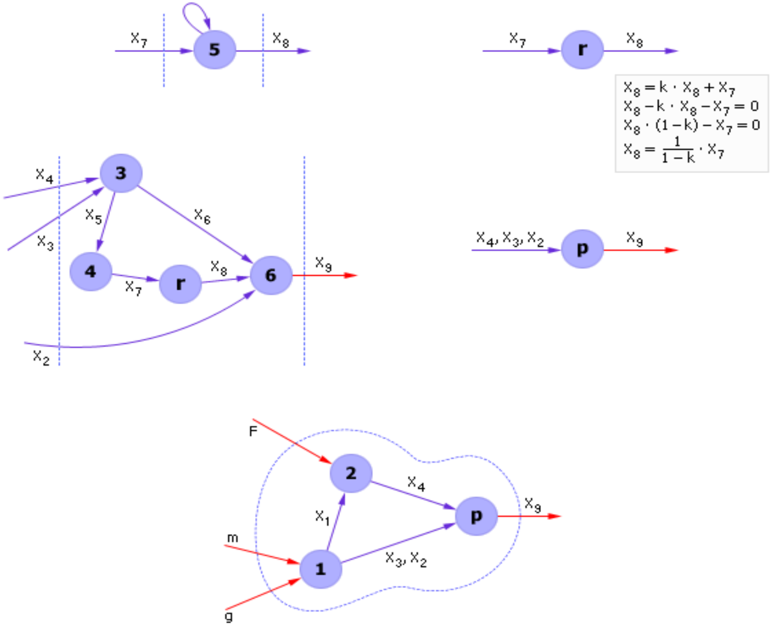

Рис. 20.2. Розбивка графа (системи) на підграфи (підсистеми)

Застосування цього прийому дуже ефективно, якщо в графі зустрічаються трохи однакових підграфів. У цьому випадку вони вивчаються окремо й один тільки раз, а результат використається багаторазово й узагальнюється на всі інші випадки. Далі в курсі «Моделі й методи штучного інтелекту» ми відзначимо цей прийом як основний у мисленні людини, що тільки й займається тим, що будує в голові моделі об'єктів навколишнього світу, згортає складні конструкції в нові поняття, щохвилини вирішує задачі на ієрархічних моделях і бореться, таким чином, зі складністю навколишнього світу.

Отже, граф задає своєю структурою модель системи, що виражена як система взаємозалежних рівнянь (див. мал. 20.3).

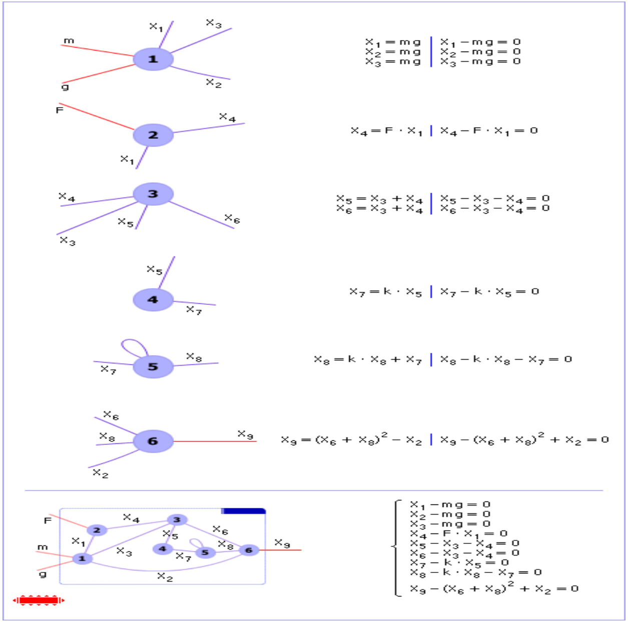

Рис. 20.3. Ілюстрація відповідності вершин графа опису підсистем великої системи

Або в самому загальному виді пишуть:

$$\overline{M}(\overline{X}) = 0, \text{ где } \overline{M}, \overline{X} \text{ — векторы}$$

Якщо серед зовнішніх змінних визначити мета й визначити керування (за рахунок чого досягається мета), то частина зовнішніх змінних буде називатися вихідними змінними (ціль), а інша — вхідними змінними. Якщо визначено вхід (керування) і вихід (ціль) на графі, то зв'язку графа стають спрямованими, від входу до виходу (див. мал. 20.4). Ці зв'язки виражають причинно-наслідкові відносини - зміни на вході ведуть до зміни значень на виході.

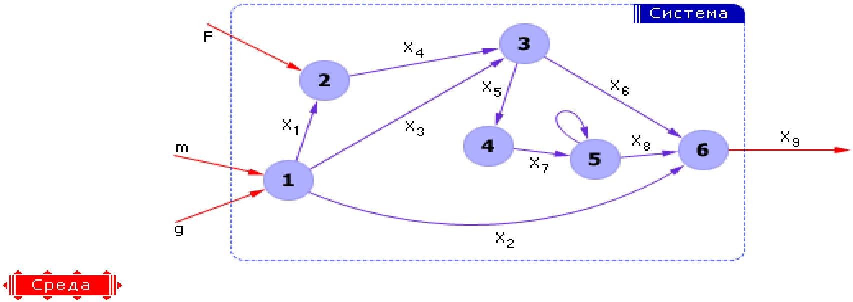

Рис. 20.4. Подання системи у вигляді орієнтованого графа, граф відповідає певній задачі, розв'язуваної на системі

У цьому виді граф відповідає задачі, розв'язуваної на графі. Задача впорядковує порядок обчислень.

Якщо тепер застосувати послідовно рівняння системи від призначеного користувачем входу до виходу, то з математичної точки зору утвориться ланцюжок виражень (див. мал. 20.5). Шукані змінні будуть виражені в підсумку по ланцюжку через вхідні змінні (див. мал. 20.6). Система рівнянь підстановкою свертывается у формулу.

$$X_9 = M_6(M_5(M_4(M_3(M_2(M_1(m, g)), F))))$$

$$X_9 \overset{6}{=} (X_6 + X_8)^2 - X_2 \overset{5}{=} \left(X_6 + \frac{1}{1-k}\cdot X_7\right)^2 - X_2 \overset{4}{=} \left(X_6 + \frac{1}{1-k}\cdot k \cdot X_5\right)^2 - X_2 \overset{3}{=} \left((X_3 + X_4)\cdot\left(1 + \frac{k}{1-k}\right)\right)^2 - X_2 \overset{2}{=}$$

$$\overset{2}{=} \left((X_3 + F\cdot X_1)\cdot\frac{1}{1-k}\right)^2 - X_2 \overset{1}{=} \left((mg + F\cdot mg)\cdot\frac{1}{1-k}\right)^2 - mg$$

Рис. 20.5. Явне рішення задачі «Керування виходом X_9 системи через вхід (m, g, F)» шляхом підстановки

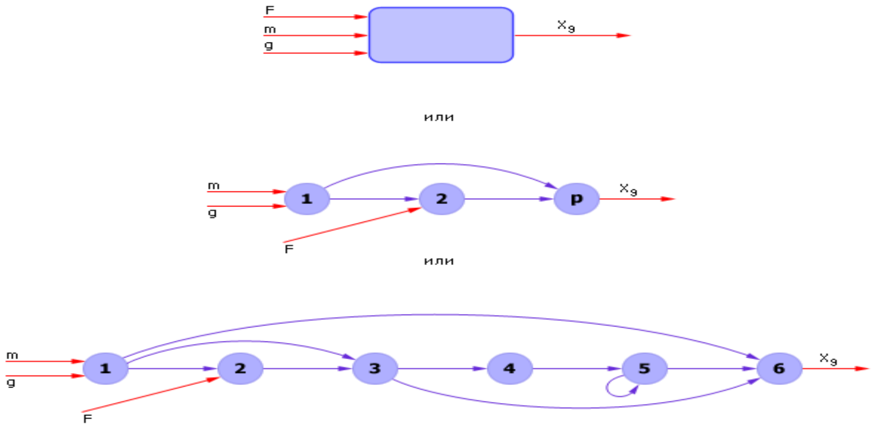

Рис. 20.6. Процедура послідовної деталізації графа (операція композиції й декомпозиції)

У загальному виді це виглядає так: $Y = M(M\ldots \ldots (M(X)) \ldots)\ldots$ Така математична структура називається композицією й задає ланцюжок (послідовність) обчислень, а значить алгоритм обчислення відповіді задачі, що у свою чергу визначає рішення системи. Рішення може бути як чисельним, так й аналітичним. Якщо задача буде інший, то модель всієї системи розгорнеться в інший ланцюжок, від інших вхідних змінних до іншого виходу. Композиція, що відповідає задачі, зміниться, але модель всієї системи залишиться незмінної.

Звичайно, не завжди ланцюжок може виразити явно залежність виходу від входу, ще частіше це відбувається, коли виражають вхід через вихід (вхід як функція виходу). Виражаючи шукане через відоме, потрібне застосування зворотних до кожної із застосованих у моделі операції. Наприклад, до $x_1 = \sin(x_2)$ застосовне зворотне перетворення $x_2 = \arcsin(x_1)$, до $x_1 = x_2^2$ потрібно застосувати зворотну операцію $x_2 = \mathrm{sqrt}(x_1)$ і так далі. А це не завжди можливо. Це залежить від того, наскільки розвинена алгебра (правила перетворень) даного виду виражень. Якщо алгебра не може визначити деякі зворотні перетворення до ряду виражень, операцій або функцій, то тоді модель залишається неявної, і доводиться застосовувати спеціальні методи розрахунку неявних рівнянь. Рішення в цьому випадку одержують чисельними методами.

До таких ускладнень приходять також моделі, що містять петлі в графі (див. мал. 20.7).

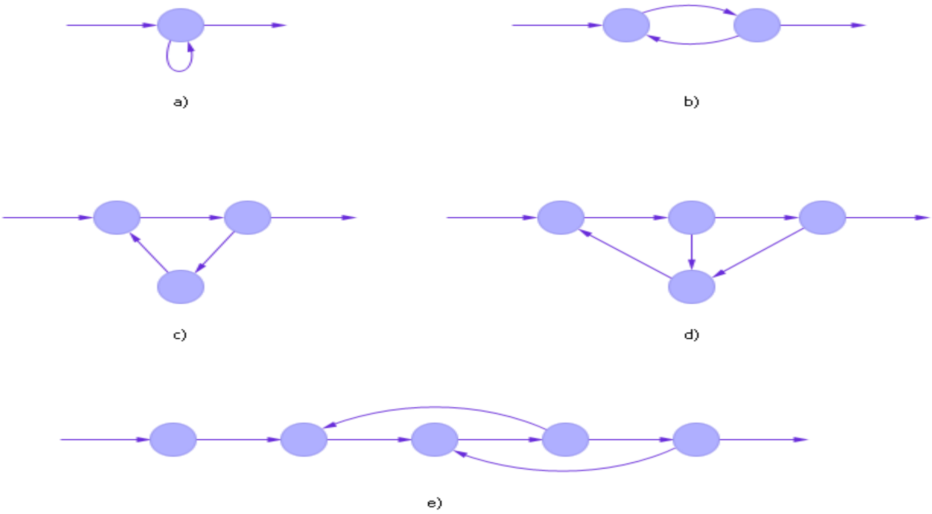

Рис. 20.7. Приклади графів різної складності, що містять петлі

Отже, якщо визначено вхід і вихід і на моделі визначена задача, граф стає орієнтованим. Задача визначає композицію моделі, спосіб обчислення відповіді. Якщо загальна формула системи вирішується, то формула явна, і алгоритм її реалізації на цифрових машинах буде лінійним, якщо аналітичного рішення ні, то формула неявна, і алгоритм буде циклічним.

Тепер прийшов час уточнити поняття вхідних змінних, оскільки їх багато й список їх досить неоднорідний. Треба мати на увазі, що вхідні змінні, які раніше ми позначали як X, можуть бути позначені з метою деталізації як X_i, U_i, P_i, Q_i.

По-перше, X може бути не однією змінною, а цілим **вектором змінних** $\{X_1, X_2, \ldots, X_n\}$, тому що складні системи, які ми моделюємо, звичайно зв'язані із середовищем множиною факторів $\{X_1, X_2, \ldots, X_n\}$... Їхні значення звичайно мало цікаві або недоступні прямо для зміни власником системи, але вони існують. Іноді це частина внутрішніх змінні системи, змінні стани, фазові змінні, пам'ять системи й так далі.

По-друге, логічно зручно розділити вектор X на вхідні змінні (властиво X) і **змінні керування** U. Тоді під X звичайно розуміють не залежні від волі власника системи фактори, а під U — фактори, якими власник системи може безпосередньо розпоряджатися по власній волі. Такі фактори прийнятий називати керованими змінними або просто керуванням. Помітимо, що звичайно значення змінних U чимсь обмежені. Справді, не можна адже відкрити водопровідний кран більше чим на 1 (кран відкритий повністю) або менше ніж на 0 (кран повністю закритий). Тому якщо розуміти під U ступінь відкриття крана, те $0 \le U \le 1$. В інших випадках пишуть більше загальний варіант $U_{\min} \le U \le U_{\max}$. У цьому змісті тут і далі ми будемо вважати, що керування, оскільки воно обмежено, це некоторый ресурс.

По-третє, P — *мало мінливі змінні*, які в цьому випадку називають параметрами системи; по своїй суті, звичайно, вони мало відрізняються від X. У прикладних задачах їх часто виносять окремо, тому що динамічно вони (на відрізку часу розгляду або існування задачі) не міняються й не міняють властивостей системи.

По-четверте, перешкоди Q. Це змінні, які діють на систему мимо волі її власника й погіршують значення бажаного показника Y. Перешкоди завжди діють на шкоду власникові системи, занижуючи бажані показники системи. Керування U — фактор, що покликаний компенсувати негативна дія перешкод Q на вихідний показник мети Y. Тобто при тому самому значенні U, при дії перешкод, на відміну від випадку їхньої відсутності, показник Y буде нижче.

Ліквідувати взагалі всі діючі на об'єкт перешкоди часто не вдається по трьох перерахованим нижче причинах.

1. Перешкода діє опережающе, швидше, ніж ми її можемо компенсувати, тому що вгадати перешкоди й підвести під це заздалегідь відповідне керування складно (хоча іноді це зробити можна спеціальними методами пророкування й випередження).
2. Перешкода звичайно діє рідко, але у великій кількості (великої амплітуди), і ресурсу керування для її компенсації відразу звичайно не вистачає. До речі, тримати великий запас керування під можливі більші перешкоди скрізь, де вони можуть виникнути, накладно, оскільки такий ресурс є омертвленим капіталом (витратами, що не приносять прибуток). Саме тому в більших системах виникають «black-out», каскадні збої й відключення, катастрофи. Керуючий ресурс звичайно зберігають в одному місці й намагаються доставити його якомога швидше до місця перешкоди або місцям, де вже встигли сформуватися відхилення, викликані ними. До того ж ефекту приводять рідкі збіги декількох перешкод, які самі по собі окремо некатастрофічні.

3. Випадок, коли перешкода діє часто (завжди) і потроху теж існує й добре компенсується керуванням (астатичні системи).

   Випадок, коли перешкода діє багато й часто, неймовірний, тому що варто визнати, що в цьому випадку система погано спроектована, і її треба попросту перепроектувати, а не управляти нею.

   Випадок, коли перешкода діє рідко й помалу, тривіальний і простий для керування або зводиться до попередніх варіантів як приватний.
4. Перешкода виникає часто не там, де її чекали й де перебуває ресурс, достатній для її компенсації.

Тому часто борються не із самими перешкодами, а з **відхиленнями** змінних *X* й *Y* від ідеальних планових їхніх значень, борються з наслідками аварій, а не з їхніми причинами.

Відійдемо на хвилину від серйозної розмови й пояснимо важливу думку на жартівному прикладі. «Щоб корова давала більше молока й менше їла, її треба більше доїти й менше годувати». Автор цієї фрази — відомий шоумен Микола Фоменко («Російське радіо»). Приклад демонструє досить розповсюджену помилку, коли інженер плутає й уважає що $X = –U$ (щоб забрати перешкоду, треба подати сигнал, що компенсує, такої ж величини й на тугіше змінну) , чого практично в складних системах не буває. Задача керування в цьому жарті вирішується надто тривіально або, точніше сказати, просто невірно поставлена, вона попросту відсутній. Причина цього полягає у відсутності моделі *M*. Тут просто не описана корова як система, як сіно перетворюється в молоко. Не враховане, що керування (сіно) не може бути використане (доставлене на вихід) як мета (молока).

Отже, за допомогою керування *U* вдається часто знизити негативна дія перешкод на цільовий показник *Y*. Дія керування на перешкоду є, але воно неявне, точніше сказати й перешкода *Q*, і керування *U* діють на показник *Y*, при цьому керування вибирається таким, щоб звести нанівець негативна дія перешкод на *Y*.

І, звичайно, варто пам'ятати, що зусилля по компенсації перешкод завжди чогось коштують власникові системи, тому що використають той самий ресурс, що ми позначили раніше як $U_{\max}$. Отже, помітьте: з керуванням завжди зв'язане поняття ресурсу. Керування черпає свої сили в ресурсі. Якщо ресурс малий, то керування зв'язане й не може впоратися із сильною перешкодою.

Якщо ресурс миттєво відновляємо, то $U_{\min} \leq U \leq U_{\max}$. Якщо ресурс має властивість аддитивности, накопичується й витрачається, не може миттєво відновитися, то

$$\int_0^t U_{\text{пр}}(t)\mathrm{d}t \leq \int_0^t U_{\text{ир}}(t)\mathrm{d}t$$

де $U_{\text{ир}}(t)$ — темп використання ресурсу, $U_{\text{пр}}(t)$ — темп поставки ресурсу.

Як відомо з математики й було вже розглянуте в лекції 01, з вираженням $Y = M(X)$ можна вирішити три види задач, які наведені в табл. 20.1.

Таблиця 20.1.

Форми запису моделі й типи розв'язуваних задач

	Відомо	**Невідомо**	**Рішення**
Пряма задача	X, M	Y	$Y = M(X)$
Зворотна задача	Y, M	X	$X = M^{-1}(Y)$
Задача настроювання моделі	X, Y	M	$M = f(X, Y)$

Задача аналізу (пряма задача)

Вивчається вплив деякого вхідного параметра U на кінцевий результат або показник Y. 1, 2, …, N — експерименти, проведені з моделлю. Якщо подавати неодноразово різні значення U на модель M (алгоритм показаний на мал. 20.10), те, вимірюючи Y, у результаті моделювання на виході моделі можна побудувати залежність $Y = M(U)$, див. мал. 20.8. Звичайно реально обмежуються деяким набором вхідних впливів $U_{\min} \le U \le U_{\max}$, проходячи значення U крапку за крапкою з певним кроком ΔU. При цьому, під час такого експерименту частина вхідних параметрів X заморожують, залишаючи їхнього значення незмінними. При необхідності можна повторити експеримент по переборі U з інтервалу $U_{\min} \le U \le U_{\max}$ при іншому значенні X. У цьому випадку виходить сімейство кривих $Y = M(U, X)$, див. мал. 20.9.

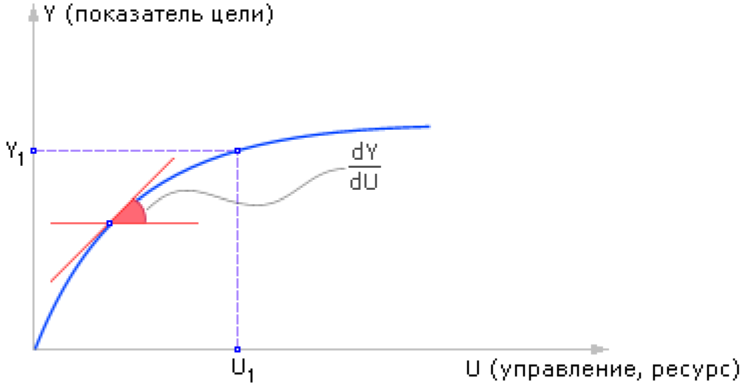

Рис. 20.8. Зразковий вид залежності мети Y від керування U, отриманий експериментально на моделі системи

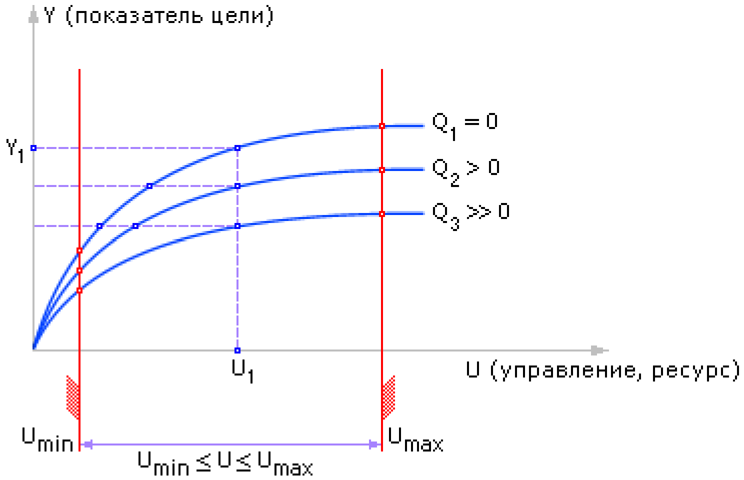

Рис. 20.9. Зразковий вид залежності мети Y від керування U при різних значеннях діючих перешкод Q

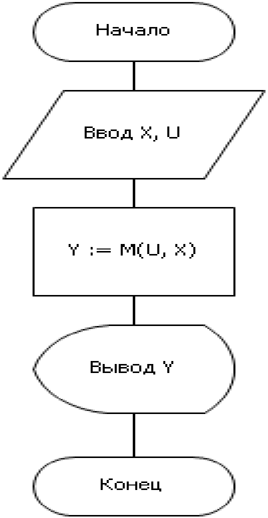

Рис. 20.10. Алгоритм, застосовуваний при рішенні прямої задачі дослідження моделі (аналіз)

Тобто, випробовуючи неодноразово модель при різних вхідних сигналах, ми можемо одержати залежність виходу від входу. Така задача називається прямій (див. лекцию 01). Результатом задачі є крива, сімейство кривих, таблиця, а коли це можливо, те формула, закон і т.д. Основне питання аналізу — пізнання властивостей об'єкта. «Впливаємо на об'єкт і дивимося, що вийде, як він реагує, робимо висновок про його властивості, можливості».

Важливо! Важливими поняттями в системотехніки є **«керованість»** й **«спостережність»**. По виду кривих $Y = M(U)$ (мал. 20.9) можна визначити, для чи всіх значень Y можливо деяке значення вхідного сигналу (U, X)? Чи будь-яке значення Y можна досягти, використовуючи змінні (U, X) з обраного діапазону. Тобто характер кривих указує, у якій області Y об'єкт є керованим. Поняття «керованість» стосується вихідний змінної.

Спостережність - можливість виміру, аналізу тієї або іншої характеристики об'єкта. Іноді через те, що деяка величина не може бути безпосередньо обмірювана в результаті експерименту, доводиться, щоб одержати про неї хоч якесь подання, задовольнятися непрямими показниками. Поняття «спостережність» стосується вихідний змінної. Проектувати системи треба так, щоб якість спостережності й керованості були забезпечені.

Відношення зміни Y до зміни U (при фіксованому X) називається чутливістю Y по U. Звичайно, тому що крива $Y = M(U)$ для складних систем нелинійна, та зміна U приймають невеликою величиною, в ідеалі ΔU –> 0. У математичному змісті, чутливість — це похідна d/d. Поняття чутливості стосується відношення виходу до входу (мал. 20.8).

Щоб скоротити кількість випробувань, вхідні впливи вибирають за певним правилом. Природне бажання одержати необхідний обсяг інформації про систему при мінімальній кількості випробувань. Таку систему випробувань планують факторним експериментом.

Задача синтезу (зворотна задача)

Ціль задачі синтезу — знаходження екстремума функції результату. Коли аналіз закінчений і побудовані функції, графіки, таблиці, коли об'єкт (його властивості й поводження) досліджений у всіх варіантах можливих вхідних впливів, має сенс знайти серед усього цього різноманіття відгуків найкращий. Звичайно вихід — ціль функціонування системи, і логічно прийняти, що ціль повинна приймати кращі із всіх можливих значень, тому має сенс знайти такі значення вхідних параметрів U, при яких вихідний показник Y прийме своє найкраще значення (екстремум). При цьому під екстремумом може матися на увазі як мінімум, так і максимум залежності $Y(U)$. Щоб знайти екстремум, модель включають у контур (див. мал. 20.11) з деяким алгоритмом A, що здійснює автоматичне керування входом U і побудованим так, що в результаті його роботи виробляється пошук такого вхідного впливу U на модель M, при якому вона видає найкращий вихідний результат.

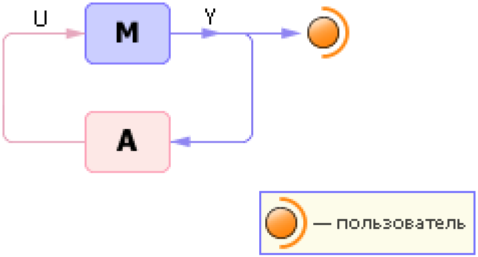

Рис. 20.11. Схема рішення зворотних задач (синтез)

Існують різні алгоритми пошуку оптимуму функції $Y = M(U)$. Згадаємо три з них (докладно ці й інші методи ви будете вивчати в дисципліні «Системний аналіз і дослідження операцій»).

1. Метод перебору. Алгоритм методу перебору представлений на мал. 20.12. Цей метод забезпечує пошук глобального екстремуму, але марнотратний до обчислювальних ресурсів, тому що переглядає всі можливі вхідні значення U з певним кроком H і вибирає найкращий серед всіх вихідних результатів Y (див. мал. 20.13). Найкраще із зустрінутих Y зберігається й уточнюється в осередку R, значення U при цьому значенні Y зберігається по ходу алгоритму в осередку Z. Крім цього є ризик пропустити потрібну крапку, «переступивши» через неї через занадто великий розмір кроку H.

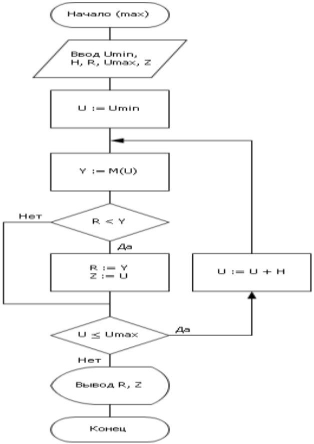

Рис. 20.12. Алгоритм перебору, застосований до рішення задачі синтезу — пошук найкращого U для максимізації Y

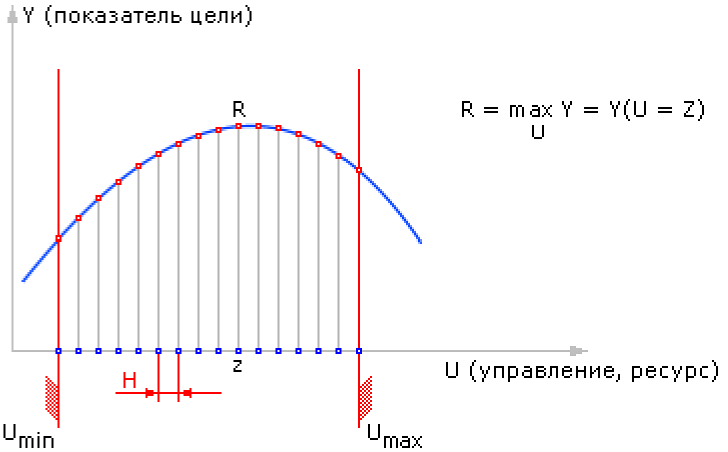

Рис. 20.13. Характерний малюнок пошуку екстремуму функції Y = M(U) методом перебору

2. **Метод ділення кроку навпіл**. Цей метод більше економічний стосовно методу перебору, тому що аналізує по ходу руху, у якому напрямку відбувається поліпшення (збільшення або зменшення) функції $Y = M(U)$ і намагається рухатися саме в цьому напрямку. Якщо при цьому попутно виявляється, що тенденція під час руху змінилася (зменшення Y змінилося на його збільшення або навпаки), то алгоритм (мал. 20.14) розвертається обернено (тобто міняє знак приросту вхідного сигналу на зворотний) і знову йде в потрібну сторону, але крок при цьому зменшується вдвічі. Це дозволяє підійти до крапки екстремуму з малим значенням кроку пошуку, що краще локалізує результат (див. мал. 20.15).

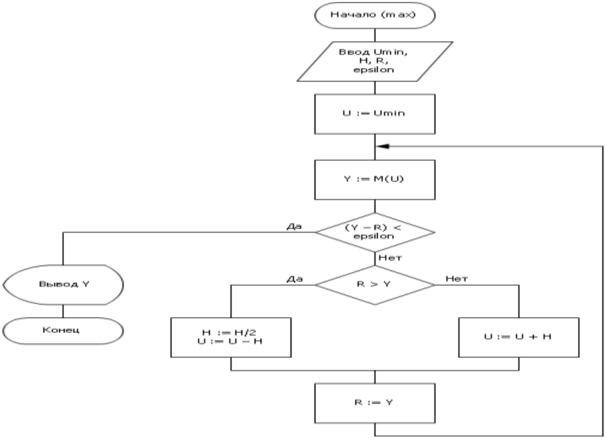

Рис. 20.14. Алгоритм ділення кроку навпіл, застосований до рішення задачі синтезу — пошук найкращого U для максимізації Y

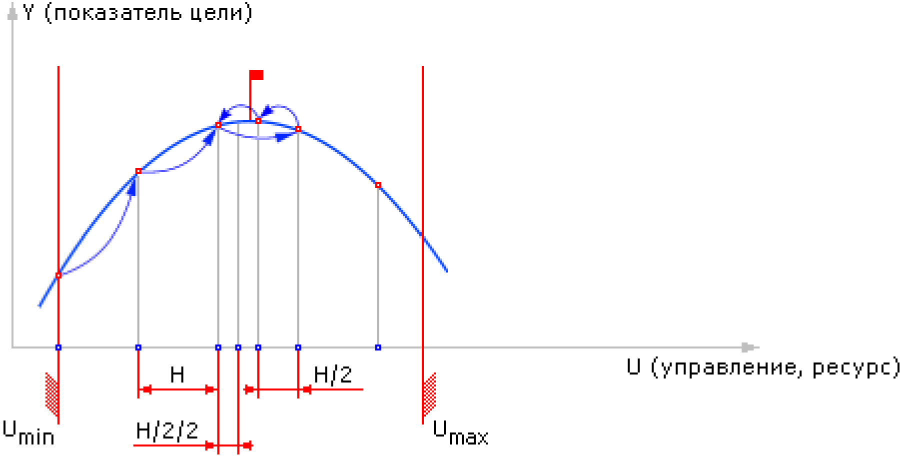

Рис. 20.15. Характерний малюнок пошуку екстремуму функції $Y = M(U)$ методом ділення кроку навпіл

3. **Метод градієнта**. Метод, алгоритмічна реалізація якого представлена на мал. 20.16, використає властивості гіпотетичній кривій $Y = M(U)$, а саме той факт, що по напрямку похідній p, обчисленої на основі двох рядом вартих крапок, можна визначити, у яку сторону зменшується (збільшується) значення мети Y і рухатися відразу в потрібну сторону (див. мал. 20.17). Така стратегія істотно скорочує час пошуку. Недоліком алгоритму є те, що похідна може перестати мінятися в області локального екстремуму, і глобальний екстремум такий алгоритм не знайде.

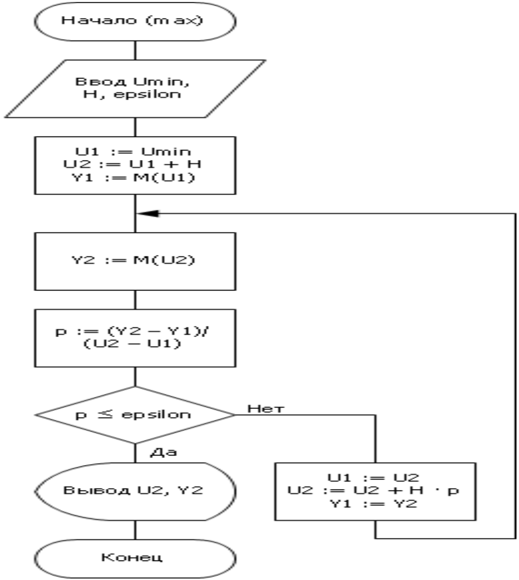

Рис. 20.16. Алгоритм пошуку екстремуму методом градієнта,
застосований до рішення задачі синтезу

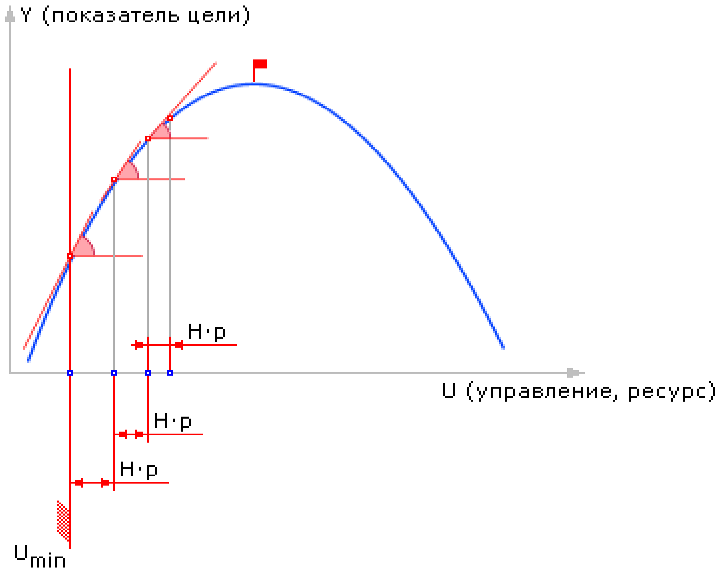

Рис. 20.17. Характерний малюнок пошуку екстремуму функції Y = M(U) методом градієнта

Задачу настроювання моделі ми вже докладно обговорювали в лекціях 02—08 (див. лекцию 02), і зупинятися на ній ми вже не будемо. Це способи побудови властиво самої моделі.

Тренажери

Цей клас задач, що використають моделі, застосовують для вироблення навичок навчання в керуючого персоналу. До тренажерів близькі комп'ютерні ігри. Керування моделлю в цьому випадку здійснює людина-оператор, що спостерігає за виходом моделі (див. мал. 20.18). Впливаючи на вхід моделі, оператор намагається домогтися потрібного вихідного результату, і в процесі цих дій одержує необхідні навички по керуванню, які потім може перенести на реальний об'єкт. На мал. 20.19 показаний зразковий вид алгоритму реалізації тренажера на базі моделі.

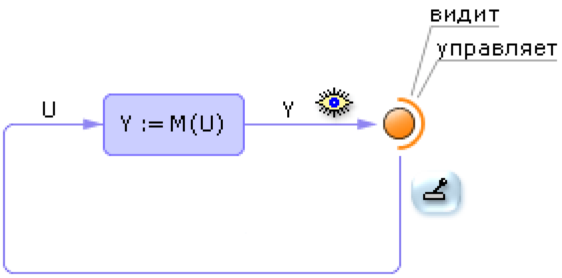

Рис. 20.18. Схема використання моделі в тренажерах

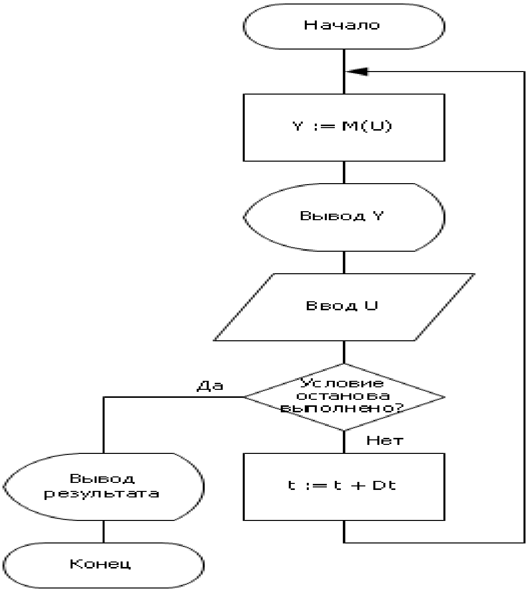

Рис. 20.19. Типовий вид алгоритму реалізації тренажера

Зрозуміло, тренажер повинен мати якості спостережності й керованість. Тобто оператор у принципі може й повинен судити про якість своїх дій, тільки спостерігаючи якісь важливі для себе результати на виході. І модель повинна бути побудована таким чином, щоб можна було досягти хоча б у принципі шуканих результатів якимись вхідними впливами на неї (керованість).

Паралельно з моделлю може функціонувати система оцінки діяльності оператора, а також блок автоматичного визначення найкращих рішень, які можуть у певних режимах (наприклад, режим навчання або підказки) допомагати операторові (див. мал. 20.20). Для цього до моделі варто підключити експертну систему, що дає рекомендації операторові в скрутні для нього випадках.

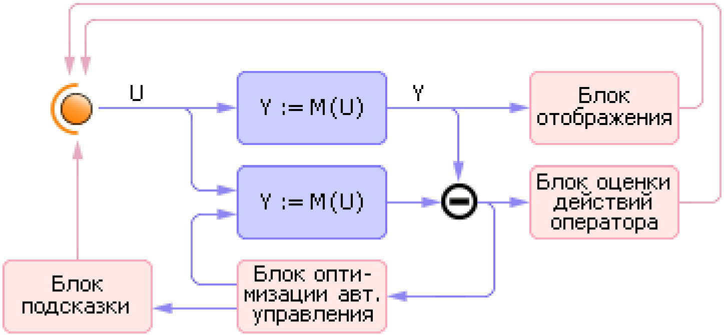

Рис. 20.20. Схема побудови тренажера
з функціями експертної системи

Для вироблення стійких навичок у персоналу в процесі тренажу в інформацію вносять додаткові перешкоди, що імітують реальні складності, що виникають на об'єкті. Можна вносити перешкоди на вході (порушується керованість), на виході (порушується спостережність) або в змінні стани моделі (див. мал. 20.21). Варто розрізняти равнодушно діючі перешкоди й цілеспрямована протидія. У першому випадку мова йде про випадковий процес, що заважає операторові досягти мета. Випадкова перешкода може, як збільшити своє значення, так і рівноймовірно зменшити його, тобто найчастіше середнє значення перешкоди на великому інтервалі часу дорівнює нулю. У другому випадку мова йде про цілеспрямовану дезінформацію оператора (середнє її дії не дорівнює нулю).

Для тренажу відіграє більшу роль середнє значення й дисперсія величини перешкод, які поступово нарощують із ростом досвіду оператора.

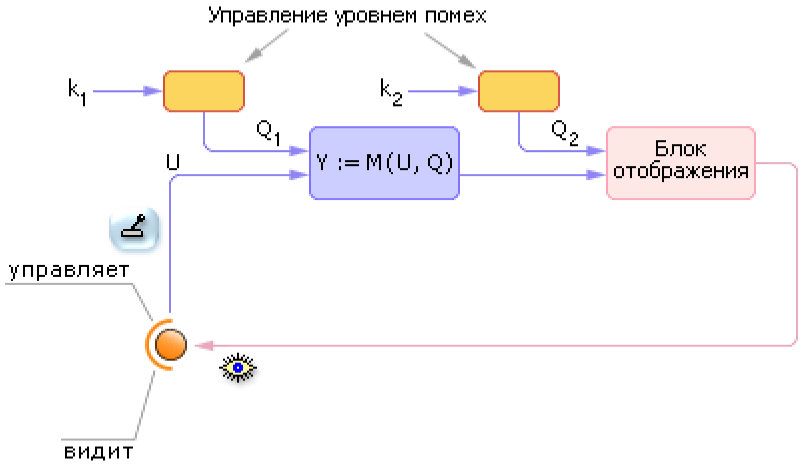

Рис. 20.21. Схема тренажера, доповненого генератором перешкод

Тепер обговоримо питання зняття й використання системних характеристик, тобто таких характеристик, які представляють властивості системи в цілому. Нагадаємо, найважливішими поняттями для системи є керування, перешкоди, мета. Системна характеристика повинна зв'язати ці поняття разом. Методика зняття характеристик така.

Виділяємо змінні U (вхід, керування) і Y (вихід, ціль) для дослідження.

Закріплюємо інші X у вигляді деякого фіксованого значення. Для кожного U з діапазону припустимих значень $U_{\min} \le U \le U_{\max}$ спостерігаємо й фіксуємо в табл. 20.2 результат Y.

Таблиця 20.2.
Залежність результату від керування при відсутності дії перешкоди. Крапки залежності знятий як результат роботи імітаційної моделі

Q (перешкоди)	**U (керування)**	**Y (результат)**
0	0	0
0	1	10
0	2	15
0	3	17
0	4	18

На графіку (див. мал. 20.22) будуємо крапку з координатами (U, Y). У результаті ряду експериментів виходить крива $Y(U)$, що показує залежність виходу від входу, мети від керування.

Звичайно, якщо ми маємо справу моделлю, що відбиває складну систему досить реально, з великим ступенем адекватності, залежність $Y(U)$ повинна мати приблизно таку залежність, як показано на мал. 20.22.

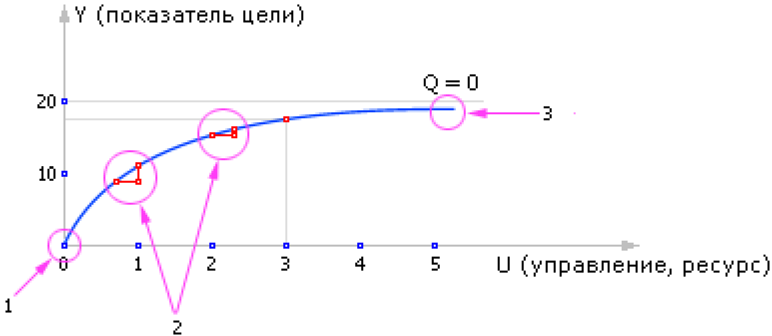

Рис. 20.22. Зразковий вид залежності показника мети від керування (ресурсу), характерний для складних систем

УВАГА! Тут наведені найбільш загальні міркування, вид кривої може бути досить різним!!!! На мал. 20.22 чітко видні наступні закономірності.

1. Крива виходить із нуля. Якщо не управляти, не вкладати ресурс, не намагатися, сама по собі система навряд чи видасть той результат, що потрібний вам, що вам корисний.
2. Крива нелінійна. Спочатку (при малих значеннях U) додаток навіть невеликих зусиль веде до гарного приросту показника мети. Далі (при більших значеннях U) кожен черговий успіх ΔY дається все більшим зусиллям ΔU. Говорячи математичною мовою, спочатку при малих U, похідна $\Delta Y/\Delta U$ має велике значення, з ростом U її значення зменшується.
3. Крива має загасаючий характер, прагне до насичення. Дійсно, представте, якщо ви вкладете величезні значення ресурсу (наприклад, затратите на рекламу своєї автозаправки в Пермі мільярд рублів), те навряд чи дістанете відповідний прибуток. Прибуток Y на цій ділянці U буде лімітована вже іншими факторами (наявність бажаючих заправитися буде не більше, ніж число автоволодарів у місті, наприклад). Реальна система рано або пізно, але виходить на певен «межу», на «упор», насичення.

Виділите на об'єкті змінну перешкода Q. За змістом ця змінна повинна заважати досягати мета й не залежати від волі власника, управляти нею він не може.

Далі варто перемінити значення Q (раніше ми вважали, що воно дорівнює 0) і провести всі описані вище дії знову (див. табл. 20.3).

Таблиця 20.3.
Залежність результату від керування
при підвищеному рівні дії перешкоди.
Крапки залежності знятий як результат
роботи імітаційної моделі

Q (перешкоди)	**U (керування)**	**Y (результат)**
1	0	0
1	1	8
1	2	13
1	3	15
1	4	16

І знову побудувати по таблиці експериментів графік (див. мал. 20.23). Очевидно, що при дії збурювань *Q* графік 2 пройде нижче, ніж графік 1, оскільки наявність перешкоди означає, що для досягнення того ж ефекту *Y* варто прикласти більше керуючих зусиль *U*. Помітимо, що міняти *Q* під час зміни *U* не треба, щоб чітко бачити зв'язок *Y* саме від *U*.

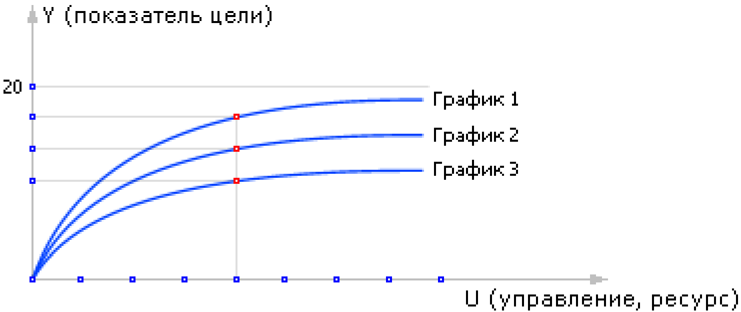

Рис. 20.23. Зразковий вид залежності показника мети від керування (ресурсу) і збурювання, характерний для складних систем

Важливо!!! Якщо перешкода все-таки під час зняття кривій $Y(U)$ міняється мимовільно, а це буває в тому випадку, коли перешкода носить випадковий характер, то треба для нанесення однієї крапки на графік спочатку провести кілька експериментів при тому самому U, а потім усреднить результат Y. Середня величина більше достовірна, чим одна з випадкових реалізацій. Скільки треба провести експериментів для усереднення, щоб забезпечити задану точність відповіді, ми обговоримо з вами пізніше в лекції 21 і лекції 34.

Знова збільште Q і знову проведіть експерименти, і знову одержите нову таблицю (див. табл. 20.4) і новий графік (див. мал. 20.23) — $Y(U)$. У результаті ви одержите сімейство кривих 1-2-3, що відбивають залежність мети, як від керування, так і від перешкоди.

Таблиця 20.4.
Залежність результату від керування
при високому рівні дії перешкоди.
Крапки залежності знятий як результат
роботи імітаційної моделі

Q (перешкоди)	U (керування)	Y (результат)
2	0	0
2	1	5
2	2	10
2	3	12
2	4	13

Зняття експериментальних даних закінчено. Тепер у будь-який момент при заданих Q й Y, використовуючи графіки 1, 2, 3, ви можете пророчити результат — необхідний для досягнення мети Y рівень керування U. Така задача, нагадаємо, називається зворотної.

Тепер, використовуючи зняті залежності, корисно знайти найкращі рішення серед множини можливих. Для цього на кожному графіку 1, 2, 3 додатково побудуємо лінію витрат S (spending), тому що керування завжди чогось коштує, і чим більше ви використаєте цей ресурс, тим більше доводиться за це платити. Нахил цієї лінії вказує на ціну ресурсу (див. мал. 20.24).

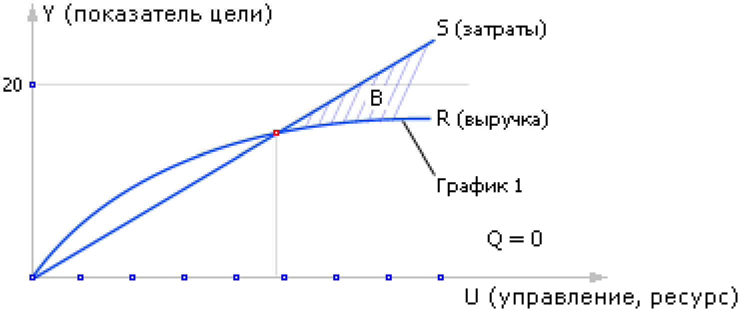

Рис. 20.24. Сполучені графіки виторгу від реалізації мети (виторг) і витрат на її досягнення

Приймемо для приклада, що ціна на ресурс незмінна й не залежить від того, скільки ви його використаєте (хоча, помітимо, що бувають оптові знижки).

Допустимо, ми повинні максимізувати мета Y. Тоді крива R (receipts), виражена у вартісних одиницях, символізує виторг, а лінія S (spending), виражена в тих же вартісних одиницях символізує витрати. Якщо відняти з виторгу витрати, тобто відняти по крапках один графік з іншого ($R - S$), те одержимо в підсумку прибуток P (profit): $P = R - S$. А саме, те, як прибуток залежить від керування (див. мал. 20.25).

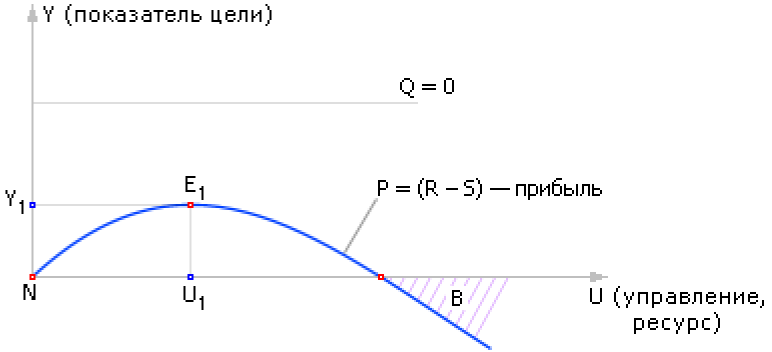

Рис. 20.25. Сумарний графік прибутку, отриманої (виторг мінус витрати) залежно від величини керування (ресурсу) U. Найкраще рішення — максимум прибутку — крапка E_1, найкраще керування — U_1

Очевидно, що зона B (bankrupt) — зона банкрутства, крапка N (null) — крапка «нічого не роби й нема чого не май» і крапка E_1 (extremum) — зона найбільшого прибутку. Одержати прибутки більше, ніж Y_1 при цьому рівні, перешкод Q не вдасться. Ця крапка символізує той простий факт, що результат, досягнутий за всяку ціну, не окупає надмірних зусиль по його досягненню, «всі добре в міру». Будь-які керуючі впливи, навіть більші, ніж U_1, дають **гірший результат**.

Аналогічно знайдемо крапку E на інших графіках 2, 3 — E_2, E_3.

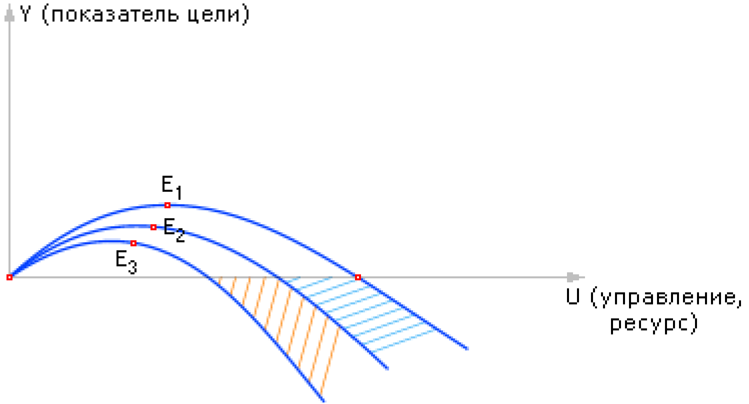

Рис. 20.26. Графіки прибутку, залежно від величини керування (ресурсу) U і збурювання Q. Крапки найкращих рішень E_i — максимум прибутку. Відповідні ним найкращі керування — U_i

Зведемо всі крапки *E* із всіх трьох графіків на один новий графік (див. мал. 20.27).

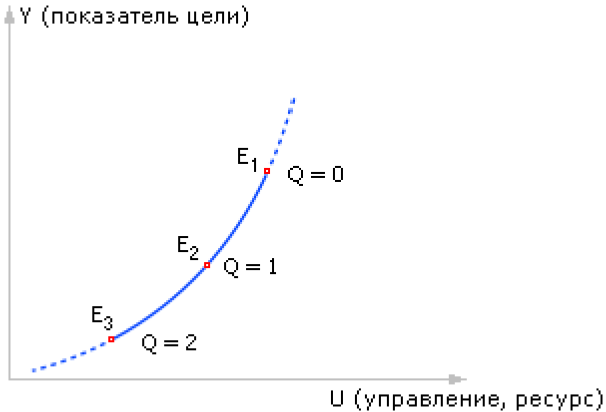

Рис. 20.27. Підсумковий графік найкращих рішень E за критерієм прибутку Y залежно від величини керування (ресурсу) U і збурювань Q

Ми одержали чудову залежність «крива оптимальних значень мети Y залежно від найкращих рішень U при заданому рівні перешкод Q», по якій можна довідатися оптимальні прикладені керуючі зусилля, необхідні для того, щоб досягти найкращого в цих умовах результату. Назвемо цю криву «взаємозалежність мети, керування й перешкод».

На графіку видно, що найбільший досяжний можливий прибуток зменшується від крапки до крапки зі збільшенням величини перешкоди — крапка E зміщається. Наприклад, можливий випадок, якщо перешкоди дуже сильні, а ресурс має фіксовану ціну, те, можливо, що краще нічого не робити.

Урахувавши вищесказане й повертаючись до лекції 01 (мал. 1.9—1.10), ще раз оборотний увага, що побудова й використання моделей у складі програмних продуктів — перспективний новий напрямок у проектуванні програмного забезпечення. Вивчення оптимальних варіантів дій по керуванню підприємством повинне бути забезпечене інструментами моделювання.