УДК 681.3, 621.3

О.М. Колодчак
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Розглянуто сучасні системи виявлення вторгнень, а також сучасні методи виявлення аномалій в комп’ютерних системах. Запропоновано альтернативні шляхи вирішення проблеми.

Ключові слова: системи виявлення вторгнень, системи виявлення зловживань, системи виявлення аномалій, параметри доступних реєстрацій, нейронні мережі зворотного поширення помилки.

Analyzed overviewed intrusion detection systems. Attention is given to recent methods of anomaly detection in computer systems. The article provides possible solutions for the problem.

Key words: intrusion detection systems, misuse detection systems, anomaly detection systems, instance-based learning, backpropagation neural network.

Вступ

Останнім часом в світі переконались, що навіть найнадійніші системи захисту не здатні захистити від атак комп’ютерні системи державних і комерційних установ. Одна з причин – у тому, що в більшості систем безпеки застосовують стандартні механізми захисту: ідентифікацію та аутентифікацію, механізми обмеження доступу до інформації згідно з правами суб’єкта і криптографічні механізми. Такий традиційний підхід має певні недоліки, а саме: незахищеність від власних користувачів – зловмисників, розмитість поділу суб’єктів системи на “своїх” і “чужих” через глобалізацію інформаційних ресурсів, порівняна легкість підбору паролів внаслідок використання їхнього змістового різновиду, зниження продуктивності й ускладнення інформаційних комунікацій внаслідок обмеження доступу до ресурсів організації. Тому виникла потреба в механізмах, які би доповнювали традиційні та давали можливість виявити спроби несанкціонованого доступу й інформували про це відповідальних за безпеку або реагували у відповідь. Важливим фактором є те, щоб такі системи могли протистояти атакам, навіть якщо зловмисник вже був аутентифікований та авторизований і з формального погляду додержання прав доступу мав необхідні повноваження на свої дії. Такі функції і виконують системи виявлення вторгнень IDS (intrusion detection systems). Оскільки передбачити всі сценарії розгортання подій в системі з активним “чужим” суб’єктом неможливо, потрібно або якомога детальніше описати можливі “зловмисні” сценарії або ж, навпаки, – “нормальні” і прийняти, що всяка активність, на яку не поширюється прийняте розуміння “нормальності”, є небезпечною. Системи IDS поділяються на системи, що реагують на відомі атаки, – системи виявлення зловживань MDS (misuse detection systems) і системи виявлення аномалій ADS (anomaly detection systems), які реєструють відхилення розвитку системи від нормального перебігу.

Стан проблеми

Відомо, що атака є багатокроковим процесом, його здійснення потребує високої кваліфікації зловмисника. Тому найпростішим способом злому або приведення системи в недієздатний стан є застосування “експлойтів”, тобто вже написаних модулів, що реалізують необхідні етапи атаки. Величезна кількість експлойтів доступна через Internet, що привело до поширення саме такого виду атаки. Як наслідок, часто послідовність аудит-подій, що реалізують атаку, фіксована.

Робота MDS базується на створенні шаблонів таких атак. Їх рівень абстракції може бути простим, таким як наявність певних значень у заголовку мережевого пакета або послідовності команд у файлі аудиту, або складним, таким як проходження траєкторії системи у просторі станів

через певні небезпечні стани. Системи захисту такого типу ефективні, коли схема атак відома, однак у випадках невідомої атаки або відхилень перебігу атаки від шаблону виникають проблеми, тому слід підтримувати велику базу даних для кожної атаки та її варіації і організовувати безперервне поповнення баз шаблонів.

Основним припущенням ADS є те, що дії зловмисника (події в атакованій системі) обов'язково відрізняються від поведінки звичайного користувача (від подій в нормальному стані), тобто є аномаліями. Тому такі системи здатні реєструвати і невідомі атаки. Роботі ADS передує період накопичення інформації, коли складається концепція нормальної активності системи, процесу чи користувача. Вона стає еталоном, відносно якого оцінюють наступні дані. Тут визначається оптимальна кількість факторів, за якими вестимуться спостереження. Їх сукупність не повинна бути надто великою, оскільки це знизить загальну продуктивність роботи. Вона також не повинна бути надто обмеженою, оскільки за недостатньо вичерпними характеристиками неможливо буде побудувати профіль нормальної поведінки. Можливі два загальні види помилок:

а) нормальна поведінка системи або користувача помилково приймається за зловмисну (false positives);

б) спроба зловмисного проникнення в систему приймається за нормальну активність (false negatives).

Хоча жодна з цих ситуацій небажана, але друга все ж таки небезпечніша, і тому одним з основних завдань побудови ADS є чітке визначення умов, за яких ситуація сприймається як аномальна, так, щоб жодна з перелічених ситуацій не виникала занадто часто [1].

2. Постановка задачі

Уже зроблено чимало спроб побудови ADS[2]. Більшість із них – концептуальні моделі, мета яких – перевірити можливість застосування математичної моделі чи підходу. Комерційних продуктів у царині IDS дуже мало, а ті, що є, майже ніколи не виходять за межі MDS. Практично всі описані в літературі методи для виявлення аномалій можна розділити на:

а) ті, що базуються на зберіганні прикладів поведінки;

б) частотні;

в) нейромережеві;

г) ті, що будують скінченні автомати;

д) інші спеціальні.

Основним завданням є пошук найоптимальніших методів побудови ADS у системах IDS.

Аналіз методів виявлення аномалій в комп'ютерних системах

Методи, основані на зберіганні прикладів поведінки.

Найпростішим підходом є пряме запам'ятовування прикладів дій, послідовностей команд користувачів чи взагалі будь-яких параметрів, доступних реєстрації (instance-based learning). Попри неможливість застосування цього підходу в інших випадках моделювання людської поведінки, у задачах виявлення вторгнень він є доволі дієвим, що зумовлено обмеженою кількістю можливих дій суб'єктів комп'ютерної системи, значною детермінованістю задач, що можна виконати, і самою структурою операційної системи. Реакцією комп'ютерної системи на аномальну поведінку процесу є його примусове уповільнення. Отже, процеси, що демонструють жваву аномальну поведінку, будуть майже зупинені й автоматично знищені системою як такі, що не реагують на запити. Зафіксовані раніше підпослідовності системних викликів, що входили в тренувальну множину, запам'ятовуються і під час роботи перевіряється їх наявність в поточній сесії. Оскільки спостереження ведеться за системними викликами програми, то через високу їх регулярність (послідовність викликів і їх типи значно детерміновані вихідним кодом програми) розміри бази підпослідовностей не будуть великими [4]. Підпослідовність з поточної сесії, якої не було серед тренувальних, вважається аномальною. Підхід потребує дописування ядра операційної системи, що нелегко і не завжди можливо. Крім того, постійна наявність такого моніторингового компонента призводить до загального уповільнення роботи всієї системи, яке становить 4 % – 50 % [3]. Іншим зразком instance-based системи є [5]. Вводиться спеціальна метрика схожості символьних послідовностей. Через необхідність збереження великої кількості даних на кожного користувача виникає потреба у застосуванні спеціальних методів зменшення об'єму баз послідовностей. Автори

розглядають два таких методи, а саме: вибіркової селекції прикладів, коли зберігаються не всі приклади послідовностей, а тільки останні n , або всі, крім n з найменшими ймовірностями, та перетворення бази на базу представників класів елементів. Через високі вимоги до ресурсів instance-based системи можуть застосовуватися лише в задачах виявлення аномалій з незначною кількістю можливих сигналів та переважно із статичною поведінкою суб'єктів спостереження.

Метод на базі частотної моделі

Розвиненням ідей instance-based систем є врахування частотного розподілу параметрів системи. Вже в [2] пропонувалося зберігати інформацію про суб'єктів у шаблонах активності – виражених у статистичних термінах наборах характеристик поведінки суб'єкта відносно певного об'єкта, таких як: входження до системи, запуски програм, доступи до файлів і пристроїв, з метою реєстрації відхилень. Потім перевіряється, чи попадає відносна кількість певних подій у заданий експертом інтервал. Модифікацією частотного підходу є робота [6], де пропонується метод, оснований на так званих структурних нулях. Він полягає у використанні інформації про команди, які використовують дуже рідко або зовсім не використовують, – відповідні їм комірки в таблиці ймовірностей дорівнюють нулю, тобто є структурними нулями. Вводиться індекс унікальності, що обчислюється для кожної сесії та кожного користувача і. Цей індекс отримує додатний приріст для частих команд у межах поточної сесії, який тим менший, чим частіше ця команда використовується взагалі. Так, широке використання рідких команд спричинить великі значення індексу унікальності. В разі виникнення команд, не властивих користувачеві, індекс зменшується. Припустивши, що значення індексу є стабільним для певного суб'єкта, автори намагаються розрізняти їх за значеннями індексу. Поширеними недоліками частотних методів є неадаптивність, оскільки часто еталонні значення частот визначаються одноразово, за тренувальною множиною або за експертними даними, і невраховування послідовності виконання команд.

Метод на базі нейромережевої моделі

Застосування нейронних мереж зумовлене самою неформальною постановкою задачі – виявити аномальну поведінку. Ідея полягає в тому, щоб, отримавши деяку "тренувальну" множину параметрів вхід-вихід, що характеризують поведінку системи, дати мережі "звикнути" до них. Виходом може бути деякий коефіцієнт "нормальності" поведінки або один із параметрів системи. Якщо вхідні дані мають закономірності, то роблять припущення, що мережа здатна "навчитися" на них. Якщо в процесі роботи запропонований нейронною мережею вихід є деяким коефіцієнтом, попадає в небезпечну область або відрізняється від наявного в реальній системі за умови, що це один із параметрів системи, то робиться висновок, що в системі наявна аномалія. Для побудови шаблону поведінки користувача використовуються такі параметри: години, коли він зазвичай працює, набір вузлів, з яких він починає робочу сесію, характеристики використання ресурсів системи [7]. Ці параметри оцифровуються і є вхідними для нейронної мережі зворотного поширення помилки (backpropagation neural network, BPNN), а вихідним є коефіцієнт, що дорівнює нулю для користувача з нормальною поведінкою й одиниці – з аномальною. Тобто мережа тренується на парах типу ("нормальні" параметри, 0) та ("аномальні" параметри, 1). Оскільки для отримання "ненормальної" поведінки треба було б змусити користувача поводитися не так, як він звик, то аномальні дані генеруються випадково, що ускладнює інтерпретацію результатів відносно роботи з реальними даними. В роботі [8] запропонована система ідентифікує користувачів на основі обмеженої кількості команд – 100. Враховується кількість запусків кожної команди, а не їхня послідовність. Ці певним способом закодовані кількості стають вхідними даними до BPNN, на яких вона тренується з учителем. Вихідні дані – ідентифікатор користувача (параметр, який, хоч і має в UNIX числове значення, але створює штучну близькість користувачів з близькими ідентифікаторами). Якщо користувача ідентифіковано неправильно – на це звертається увага адміністратора. Отримано непогані результати, але лише з 10 користувачами. Такі умови є "тепличними", оскільки в реальних системах кількість користувачів досягає кількох тисяч, причому більшість виконує однотипні дії, що ускладнює їх розрізнення таким способом. У роботі [9] теж запропоновано підхід на основі BPNN. На відміну від [7, 8], тут не обмежувалися вузьким колом користувачів, кількість команд була більшою (до 512 команд) і не використовували штучних даних. Мережа застосовувалася для передбачення наступної (n+1)-ї команди в сесії користувача,

використовуючи попередні n команд. Через невпорядкований характер множини команд було запропоновано компромісне кодування команд: кожній команді ставився у відповідність вектор – двійковий запис номера команди після довільного перенумерування всіх команд. Тим самим автори частково позбувалися проблеми штучної впорядкованості, коли командам надавали скалярних значень і одержали досить економний метод кодування (на відміну від методу, коли окремій команді присвоюється вектор, який має одну одиницю у фіксованій позиції). Після тренування на входах, що відповідали контексту попередніх декількох команд, які потрапляли у вікно ширини n і з виходом – наступною командою, мережа повинна була на реальних даних за наявним контекстом визначати наступну. Якщо відносна кількість правильно передбачених команд у сесії не перевищувала заданого рівня, це сигналізувало про аномалію. Для забезпечення адаптивності нейронна мережа після кожної сесії дотреновувалася. Результати свідчать про можливість застосування такого підходу і про ефективність вибраного методу кодування команд, проте нерегулярність поведінки користувачів значно підвищує рівень помилок типу false positives. Очікується, що кращі результати можна отримати на даних від регулярніших джерел, таких як системні процеси. Недослідженою залишається можливість застосування нейронних мереж неперсептронного типу, наприклад, ансамблевих. Недоліком багатьох нейронних мереж є їхня погана пристосованість для роботи з невпорядкованими величинами. Введення штучного порядку на множині значень елементів тільки спотворить картину, оскільки нейронна мережа враховуватиме близькість числових величин.

Метод, що базується на моделях, які будують скінченні автомати

В цьому методі досягається більша моделювальна здатність, ніж у разі використання тривіальних частотних та instance-based методів. Вхідні дані розглядаються як потік дискретних подій, наприклад, системних викликів або ідентифікаторів процесів. Мета – отримати автомат, який моделює вказану послідовність подій. Для багатьох послідовностей характерно, що ймовірність наступного символу, елемента або сигналу залежить від попередніх. Часто вони залежать лише від невеликої кількості попередніх. Це наштовхує на думку моделювати їх за допомогою марковських ланцюгів [2, 4, 13–15]. Проте у разі зростання порядку ланцюгів, що може суттєво збільшити точність моделі, кількість станів відповідного автомата поводить себе як $O(\Sigma L)$, де Σ – розмір алфавіту символів; L – порядок ланцюга. Це ставить великі вимоги до ресурсів і збільшує час обробки. За вхідними даними будується матриця переходів ланцюга першого порядку і ймовірність сесії визначається як добуток імовірностей переходу між станами, що відповідають елементарним подіям у файлі аудиту. Потужнішою моделлю є приховані марковські моделі (HMM). Від марковських ланцюгів вони відрізняються тим, що вихідні символи автомата не детерміновані його станами, а залежать від них стохастично. Є теоретичні результати, які свідчать про складність навчання HMM[13,14]. Крім того, часто необхідний підбір кількості прихованих станів, який звичайно проводиться вручну і після тренування реально задіяними (такими, що мають відмінні від нуля ймовірності переходів) виявляються лише деякі з них. Для забезпечення адаптивності необхідно періодично перенастроювати HMM на нових даних, що є значним недоліком моделі. Скінченний автомат будується за n-грамами подій з аудит-файла, який не містить аномальних даних. У першому варіанті побудови кожному стану відповідає одна або кілька n-грам. Кілька n-грам можуть бути присвоєні одному стану, якщо при послідовній обробці даних, вікно за вікном, маємо ситуацію, коли наступна n-грама ще не має свого стану й одночасно не існує ребра, що виходить з поточної n-грами в наступну. Тоді наступна n-грама присвоюється тому самому станові, що й поточна. Побудований у такий спосіб недермінований автомат може знаходити раніше не бачені події, але він не бере до уваги статистичних параметрів послідовності. За другим способом будується імовірнісний автомат з множиною входів – подіями з аудит-потоку і множиною виходів – послідовністю чисел, що вказують на ступінь аномальності вхідних подій. Ці числа показують, як імовірність наявного переходу з поточного стану в наступний співвідноситься з вектором перехідних імовірностей з цього стану. Попри деяку необґрунтованість вибору саме такої міри аномальності, підхід на експерименті виявився ефективним. Удосконалюючи модель для виявлення аномалій за системними викликами (згадаємо фіксованість вихідного коду програми і, через це, певну статичність картини системних генерованих викликів), можна не навчатися на прикладах, а

спробувати, використовуючи вихідний код, задати статичну граматику, що описує деяку мову викликів. Таку граматику задано за допомогою аналізу вихідних текстів програм. Запропоновано три варіанти моделей: модель графу викликів (недетермінований скінченний автомат), стекову модель (контекстно вільну граматику) і модель, схожу на описану в роботі [4] (для заданої k-грами перевіряється, чи належить вона до мови, породженої певною граматикою). Перевагою такого підходу є відсутність помилок типу false positives через залежність побудованих моделей безпосередньо від вихідного коду програми. Тобто несумісна з моделлю послідовність викликів відразу сигналізує про відхилення роботи програми від її вихідного коду, і тільки такі відхилення сигналізуватимуть про аномалію, а все, що передбачено текстом програми, буде прийнято. Основний недолік такого підходу – складний процес побудови моделі за аналізом текстів програми. Крім того, є обмеження на види програм, які можна описати статичними граматиками. Існує модель, якій не властиві недоліки HMM та марковських ланцюгів з фіксованим порядком. Її основна ідея – використовувати модель марковських ланцюгів змінного порядку, оскільки наступний символ або сигнал у послідовностях, що генеруються в комп'ютерних системах, рідко визначається попереднім контекстом постійної довжини. Врахування лише істотних контекстів дає змогу обійти експоненційне зростання вимог до ресурсів, не погіршуючи точності моделі. Крім того, більшість попередніх підходів до моделювання поведінки є неадаптивними. Найкраще, що пропонується, – періодичне перенастроювання моделі. Звідси втрати часу і зменшення надійності з часом, внаслідок чергового перетренування. Для підтримки реальної адаптивності ми застосували певну процедуру зміни параметрів нашої моделі так, що остання здобута інформація стає вагомішою. При цьому зберігаються апроксимуючі властивості вихідної моделі. Єдиною відомою нам роботою, де застосований схожий на наш метод адаптивного настроювання ймовірностей, є [13]. Проте там за основу взято найпростішу марковську модель першого порядку, а тому не враховується контекст, довжина якого більша за одну команду.

Інші методи

Крім описаних вище методів, для виявлення аномалій використовуються байєсівські мережі – графічні моделі представлення у власній структурі залежностей між об'єктами та розподілів імовірностей. За множиною тренувальних даних оцінюються кореляції між станами і будується мережа зі з'єднаними, залежними між собою, вузлами, ймовірності переходів між станами, які заповнюються, і зв'язані з вузлами таблиці розподілів імовірностей за даними станами вузлів-предків. Через жорстку залежність побудованої структури від наданих тренувальних даних отримана модель не є адаптивною.

Теоретико-інформаційний підхід поєднує два методи вибору оптимальної ширини вікна для виявлення аномалій у послідовностях програмних системних викликів. Перший – обчисленням умовної ентропії $H(X|Y)$ наступного символу $X \in \Sigma$ за умови попереднього контексту $Y \in \Sigma n$. Значення n, окреме для кожної програми, за якого значення ентропії мінімальне, вибирається за ширину вікна. Другий метод – використанням ансамблів імовірнісних дерев. Кожне враховується з певною вагою при обчисленні ймовірності наступного системного виклику. Якщо якесь дерево правильно "вгадало" наступний виклик, його вага в ансамблі збільшується, чим досягається адаптивність.

У роботі [14] описано цікавий підхід, який, на відміну від традиційних методик виявлення аномалій після навчання за неаномальними даними, призначений для виявлення аномалій без попереднього тренування на прикладах нормальної активності. Для цього сукупність послідовних даних подано як таку, що була згенерована "змішаною" стохастичною моделлю. Тобто з ймовірністю λ елемент послідовності є аномальним і з ймовірністю $(1-\lambda)$ – нормальним. Розподіл A аномальних елементів за відсутності якихось апріорних даних про нього вважається рівномірним. Розподіл нормальних елементів M може оцінюватися будь-яким методом з арсеналу технік машинного навчання. Отже, маємо генеруючий розподіл-суміш. Далі, вважаючи, що відповідні розподіли відомі, можна обчислити функцію правдоподібності $L(D)$ для заданої послідовності елементів. Тоді, якщо поточний елемент x_t перемістити з нормального розподілу M в аномальний A і при цьому правдоподібність збільшиться, то елемент там і залишається, якщо ні – нічого не змінюється. Застосування цього підходу дає змогу не залежати від "чистоти" тренувальних даних, які можуть містити й аномальні включення (за умови їх незначної кількості).

Є спроби використовувати для задачі виявлення аномалій генетичні алгоритми – алгоритми пошуку оптимуму, що базуються на аналогіях з природним добором у природі. За "популяцію" береться множина "особин" – бінарних рядків фіксованої довжини. Кількість рядків у множині також фіксована. У процесі еволюції, за певним критерієм – "пристосованістю до навколишнього середовища" вибираються рядки, з яких генерується наступне покоління рядків. Генерація наступного покоління відбувається з використанням трьох основних операторів над рядками: селекція (вибір найпристосованішого представника), репродукція (обмін частинами рядків між собою) і мутація (випадкові зміни бітів у рядках для запобігання незворотній втраті інформації). Генетичні алгоритми використовуються для знаходження песимістичного сценарію в гібридній задачі виявлення зловживань і аномалій. З одного боку, виявляються тільки відомі атаки, з іншого – застосування генетичних алгоритмів і несуворих обмежень на гіпотези може дозволити реєструвати також і варіації цих атак. Підхід перевірено на штучних даних зі змодельованими простими атаками.

Альтернативні шляхи вирішення проблеми

Вище розглянуто основні методи, що застосовуються в ADS, проте жодний з них не гарантує виявлення всіх атак. Отже, в реальних IDS, спираючись на аналіз вищезгаданих методів, можна рекомендувати реалізувати комбінацію різних методів і згідно з нею робити остаточний висновок про наявність чи відсутність вторгнень та їх характер.

Моделі в цих методах не повинні залежати від конкретного типу аудит-даних, тоді їх можна застосувати до будь-яких аудит-послідовностей. Так, такими даними, окрім команд, що безпосередньо виконує користувач, можуть стати послідовності системних викликів, характерні для програми, значення інтервалів між натисканнями клавіш на клавіатурі, інтервали між робочими сесіями, послідовні значення координат положення курсора миші на екрані, послідовності значень полів заголовка мережевих пакетів усіх рівнів або величини, що в них не містяться, але від них залежать. Тобто будь-яка послідовність сигналів, команд, елементів або хронологічних чи інших значень, характерна для користувача, процесу, системи або мережевого сегмента, може бути використана в ADS.

Зрозуміло, що користувачам властиво змінювати свою поведінку (через зміну задач, набуття нових звичок), тому моделі методів обов'язково повинні бути адаптивними. Велика частина атак – це атаки проти системних програм, наприклад, набуття прав суперкористувача шляхом використання вразливостей у SUID програм. Після цього програма, як правило, демонструє кардинально відмінний характер системних викликів від тих, що спостерігалися до атаки. Тому слід звернути увагу на виявлення аномалій на рівні системних викликів, що стає дедалі популярнішим, оскільки дає змогу абстрагуватися від нерегулярної людської поведінки. Одночасно не можна відмовлятися від аналізу аудит-даних, що надходять від користувачів, оскільки тільки аналіз на цьому рівні дає можливість виявити вторгнення, які на рівні системних викликів не проявляються (наприклад, використання вкраденого пароля).

Повноцінна IDS повинна містити також компоненту з виявлення зловживань, оскільки сьогодні атаки з використанням експлойтів залишаються одними з основних видів атак.

Висновки

В роботі розглянуто проблему виявлення атак у комп'ютерних системах. Розглянуто відомі методи для систем виявлення аномалій. Проаналізовано переваги та недоліки кожного методу. Наведено основні переваги на недоліки для кожної моделі методів. Запропоновано альтернативні шляхи для створення ефективної системи виявлення аномалій в комп'ютерних системах.

1. Axelsson S.. The base-rate fallacy and its implications for the difficulty of intrusion detection // In ACM Conference on Compute r and Communications Security, pages 1-7, 1999. 2. Denning D.E. An intrusion-detection model // In Proc. IEEE Symposium on Security and Privacy, pages 118-131, 1986. 3. Somayaji A. Automated response using system-call delays // In Proc. of USENIX Security Symposium 2000, pages 185-197, 2000. 4. Forrest S., Hofmeyr S.A., Somayaji A., and Longstaff T.A. A sense of self for Unix processes // In Proc. of the 1996 IEEE Symposium on Research in Security and Privacy, pages 120-128. IEEE Computer Society Press, 1996. 5. Lane T. and Brodley C.E.. Temporal sequence learning

and data reduction for anomaly detection // ACM Transactions on Information and System Security, 2(3): 295-331, 1999. 6. Theus M. and Schonlau M. Intrusion detection based on structural zeroes // Statistical Computing and Graphics Newsletter, 9(1): 12-17, 1998. 7. Tan K. The application of neural networks to unix computer security // In Proc. of the IEEE International Conference on Neural Networks, volume 1, pages 476–481, Perth, Australia, 1995. 8. Ryan J., Lin M., and Miikkulainen R. Intrusion detection with neural networks // Advances in Neural Information Processing Systems, pages 254-272, 1998. 9. Резник А.М., Куссуль Н.Н., Соколов А.М. Нейросетевая идентификация поведения пользователей компьютерных систем // Кибернетика и вычислительная техника, 1999. – Вып. 123. – С. 70–79. 10. Endler D. Intrusion detection: Applying machine learning to solaris audit data. In Proc. Annual Computer Security Applications Conference (ACSAC'98), pages 268-279, Los Alamitos, CA, De c. 1998. IEEE Computer Society Press. Scottsdale , AZ. 11. Ghosh, J. Wanken, and F. Charron. Detecting anomalous and unknown intrusions against programs. In Proc. of the 1998 Annual Computer Security Applications Conference (ACSAC'98), Dec. 1998. http://rstcorp.com/~anup/ACSAC98.pdf 12. Ghosh K., Schwartzbard A., and Schatz M. Learning program behavior profiles for intrusion detection // In Proc. 1-st USENIX Workshop on Intrusion Detection and Network Monitoring, pages 51-62, Santa Clara, California, Apr. 1999. 13. Davison D. and Hirsh H.. Predicting sequences of user actions // In Predicting the Future: AI Approaches to Time-Series Problems, pages 5-12, Madison, WI, July 1998. AAAI Press. Proc. of AAAI-98/ICML-98 Workshop. 14. Eskin E.. Anomaly detection over noisy data using learned probability distributions. In Proc. 17th International Conf. on Machine Learning, pages 255-262. Morgan Kaufmann, San Francisco, CA, 2010.

УДК 621.317.7

Р.В. Кочан
Національний університет "Львівська політехніка",
кафедра спеціалізованих комп'ютерних систем

ДОСЛІДЖЕННЯ ІНТЕГРАЛЬНОЇ НЕЛІНІЙНОСТІ СИГМА-ДЕЛЬТА МОДУЛЯТОРА ДРУГОГО ПОРЯДКУ

Досліджено вплив інтегральної нелінійності функції перетворення інтеграторів сигма-дельта модулятора другого порядку на інтегральну нелінійність функції перетворення модулятора загалом. За результатами досліджень визначено вигляд функції нелінійності модулятора та коефіцієнти впливу кожного інтегратора.

Ключові слова: сигма-дельта модулятор, інтегральна нелінійність, імітаційне моделювання.

There is investigated influence of integrators' integral nonlinearity on integral nonlinearity of single bit second order sigma-delta modulator. The obtained results allow us to define the view of nonlinearity of modulator and the influence coefficients of all integrators.

Key words: sigma-delta modulator, integral nonlinearity, simulation.

Вступ

Поширення засобів обчислювальної техніки та алгоритмів цифрового опрацювання сигналів привело до того, що аналого-цифрові перетворювачі (АЦП) стали необхідним компонентом практично всіх вимірювальних систем, а їх метрологічні характеристики визначають характеристики систем загалом, тому покращення параметрів АЦП є актуальним завданням.

Огляд літературних джерел

Нішу прецизійних АЦП напруги постійного струму займають перетворювачі на базі сигма-дельта модулятора (СДМ) [1, 2]. Їх високу точність забезпечують структурно-алгоритмічні методи – встановлення нуля та калібрування, що дає змогу усунути вплив на результат перетворення адитивної та мультиплікативної складових похибки перетворення. Тоді похибка результату