ДОДАТОК А

Апбробація результатів досліджень

Міністерство освіти і науки України

NURE

Харківський національний університет радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2023

(Випуск 2)

[електронне видання]

http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam

http://itez.zntu.edu.ua/

http://kafea.kdu.edu.ua

Харків 2023

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2023

(Випуск 2)

[електронне видання]

Харків 2023

УДК 658.51: 004.4.22

СУЧАСНЕ ВИРОБНИЦТВО З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

О.С. Пащенко, К.О. Зозуля
Харківський національний університет радіоелектроніки
Україна, 61166, Харків, пр. Науки 14
E-mail: oleksandr.pashchenko@nure.ua, kyrylo.zozulia@nure.ua

Анотація: Розглянуто сучасне виробництво з комп'ютерною системою управління і широким застосуванням інформаційних технологій, що забезпечує можливість пов'язування окремих процесів, функцій і завдань у єдину систему для підвищення ефективності та якості виробництва.

Ключові слова: комп'ютерно-інтегроване виробництво, інформаційні технології, система управління, верстат, виконавчи пристрої

MODERN PRODUCTION WITH THE USE OF COMPUTER CONTROL AND INFORMATION TECHNOLOGY

O.S Pashchenko, K.O. Zozulya
Kharkiv National University of Radio Electronics
Ukraine, 61166, Kharkiv, Nauky av.,14
E-mail: oleksandr.pashchenko@nure.ua, kyrylo.zozulia@nure.ua

Annotation: The article considers modern production with a computerized control system and widespread use of information technology, which makes it possible to link individual processes, functions and tasks into a single system to improve the efficiency and quality of production.

Keywords: computer-integrated production, information technology, control system, machine tool, actuators

Вступ. Принциповою особливістю інтегрованих виробництв є наявність нової компоненти – комп'ютерної системи управління, а також широкого застосування інформаційних технологій, що забезпечують можливість ув'язування окремих процесів, функцій і завдань в єдину систему для підвищення ефективності виробництва. Подальший розвиток робіт в даному напрямку призвело до появи поняття комп'ютерн-інтегрованого виробництва (КІВ). Концепція КІВ мала на увазі новий підхід до організації та управління виробництвом, новизна якого полягала не тільки в застосуванні комп'ютерних технологій для автоматизації технологічних процесів і операцій, але в створенні інтегрованої інформаційної системи управління виробничою діяльністю підприємства.

1. Комп'ютерно-інтегроване виробництво. Реалізація комплексу проектів по створенню автоматизованих виробництв (АВ) почалася з виробництва металорізальних верстатів, що представляють собою спробу практичної реалізації концепції КІВ. Було виконано попереднє проектування АВ, виготовлені дослідні зразки нового обладнання, створено випробувальний полігон, створені основні компоненти інтегрованої автоматизованої системи управління.

Ряд подібних проектів було здійснено також і за кордоном. Одним з перших став проект АВ, реалізований в Японії фірмою Mazak, для виробництва пристроїв металорізальних верстатів. Завод включав в себе: комплекс гнучких виробничих модулів (ГПМ) і ГВС, автоматизовані склади, робокарную транспортну систему. Передбачалося використання комп'ютерних мереж для сервісної та технічної підтримки філій, а також взаємодії з

підприемствами комплектуючих виробів. В цілому за період 1985-1995 рр. в різних країнах було створено близько 20 КІВ з різним рівнем автоматизації, з яких вісім АВ випускали металорізальне обладнання, чотири – вироби для аерокосмічної промисловості (США), інші КІВ були орієнтовані на випуск різних агрегатів широкої номенклатури, включаючи компоненти обчислювальної техніки і електричних машин. Від впровадження КІВ очікувалося: зменшення розмірів підприємств, збільшення коефіцієнта використання устаткування і зниження накладних витрат, значне зменшення обсягу незавершеного виробництва, скорочення витрат на робочу силу в результаті організації «безлюдного» виробництва, прискорення змінюваності моделей продукції, що випускається відповідно до вимог ринку, скорочення термінів поставок продукції і підвищення її якості.

Проте, не дивлячись на те, що дослідженням в області застосування інформаційних технологій в гнучкому виробництві було присвячено значну кількість наукових досліджень, а на створення експериментальних комп'ютерно-інтегрованих виробництв витрачені значні фінансові кошти, – досягти поставленої науково-практичної задачі загальної/глобальної інтеграції виробничого процесу за допомогою автоматизації управління не вдалося. З ряду об'єктивних причин, а також через допущені методологічних помилок системотехнического характеру, проекти не були реалізовані в повному обсязі, а науково-технічні напрацювання, отриманий в результаті проведених НДДКР, використаний в проектах меншого масштабу. Невдачі в реалізації ідей і принципів КІВ, так само як і багатьох проектів з розробки автоматизованих систем управління виробничими процесами (АСУВП) були обумовлені тим, що в їх концепцію спочатку закладений принцип максимально можливій мірі автоматизації управління, практично повністю виключає участь людей-операторів (осіб приймають рішення) в управлінні виробництвом.

Прихильники заміни природного інтелекту людини його штучним подобою для вирішення різних практичних завдань спочатку обмежилися вивченням закономірностей поведінки об'єктів управління (через їхню складність) на основі порівняння вхідних впливів і вихідних результатів, ігноруючи при цьому їх внутрішній устрій.

Даний методологічний підхід, відомий в кібернетичної науки як принцип «чорного ящика», отримав широкий розвиток в теорії автоматичного управління і забезпечив задовільні результати в створенні технічних пристроїв. Однак, спроби розширити межі його застосування до рівня управління складними соціально-економічними системами з активними елементами (людьми) виявилися безперспективними.

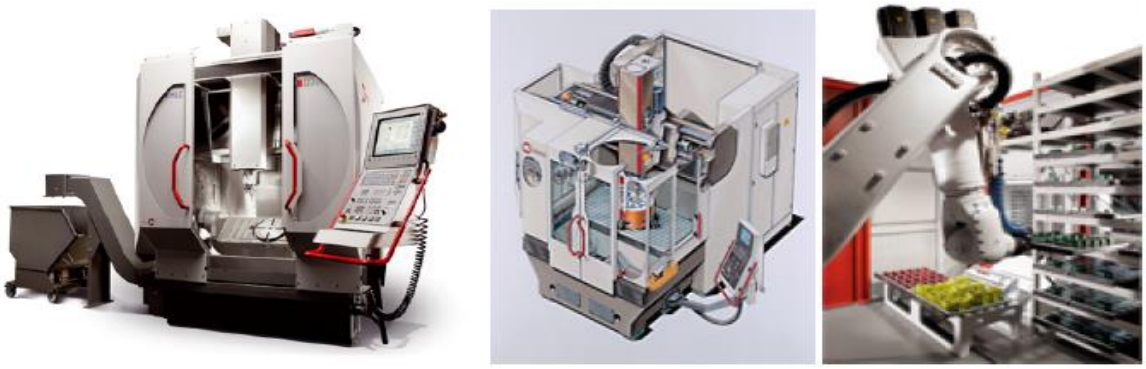
З цієї причини, незважаючи на істотний внесок цієї категорії досліджень в інженерію знань, реалізація багатьох проектів автоматизованих систем управління в XX століття закінчилася негативним результатом. Неухильне дотримання ідеї заміни людини машиною поставило перед дослідниками і розробниками КІВ складну методологічну проблему адекватного уявлення в машинної середовищі знань про реальні об'єкти і процеси, вирішити яку за допомогою апарату теорії автоматичного управління і формалізованих методів класичної математики об'єктивно не можливо в переважній більшості випадків. Спроби рішення важко формалізованих і неформалізованих задач управління, що переважають у складних виробничо-економічних системах, за допомогою обмеженого інтелекту ЕОМ з класичною архітектурою приводили до грубих помилок в оперативному регулюванні складного динамічного процесу виробництва. Нерозв'язні в рамках цього напрямку методологічні проблеми не дозволили розробникам створити в машинної середовищі автономну (тобто не вимагає додаткового звернення до інтелекту людини) модель управління виробництвом.

В кінцевому підсумку концепція КІВ вичерпала свої можливості обмежившись завданнями автоматизації технологічних операцій (процесів) в багатофункціональних обробних центрах, що дозволяють виготовляти деталі на одному робочому місці в одну установку. Створювані на

їх основі гнучкі роботизовані виробництва з надлишковою функціональністю (ГВС/FMS) здатні в напівавтоматичному (умовно-автономному) режимі здійснювати паралельну обробку пристроїв багатопредметного виробництва виробів в межах заданої номенклатури без необхідності переривання на переналагодження і передачі предметів праці на інші обробні центри. Пристосування до випуску нових виробів здійснюється за рахунок зміни робочого стану (переналадки технічних засобів / інструменту,

Основу здебільшого адаптивних/гнучких роботизованих виробничих систем складають багатофункціональні обробні центри на базі вертикальних токарних верстатів (наприклад, багатофункціональний обробний центр серій C50U виробництва німецької верстатобудівної фірми Maschinenfabrik Berthold Hermle AG, представлений на рис. 1, а), які здатні виконувати практично будь-які технологічні операції в одну установку заготовки/деталі (токарна обробка, фрезерування, свердління, розгортання, шліфування, різьбонарізання, зубофрезерування, лазерне зварювання).

В результаті їх об'єднання з роботизованою системою завантаження/розвантаження пристроїв і великовантажним стелажем може бути створена високопродуктивна адаптивна виробнича система (як приклад на рис. 1, б приведена роботизована система RS4 тієї ж верстатобудівної фірми).



а) б)

Рисунок 1 – Гнучка роботизована виробнича система, реалізована на базі багатофункціональних обробних центрів (а) і роботизованої системи завантаження/розвантаження пристроїв зі стелажем (б)

Така система може включати до трьох вбудованих обробних центрів із можливістю її розблокування щодо того чи іншого обробного центру. Це дозволяє обслуговувати як роботизовану систему, так і обробний центр вручну, не перериваючи автоматичний процес роботи іншого модуля. У такій системі можна використовувати змінні великовантажні стелажі самої різної конструкції. За допомогою автоматично замінних одинарних/подвійних захоплень можливе завантаження і вивантаження заготовок, оброблюваних пристроїв і палет.

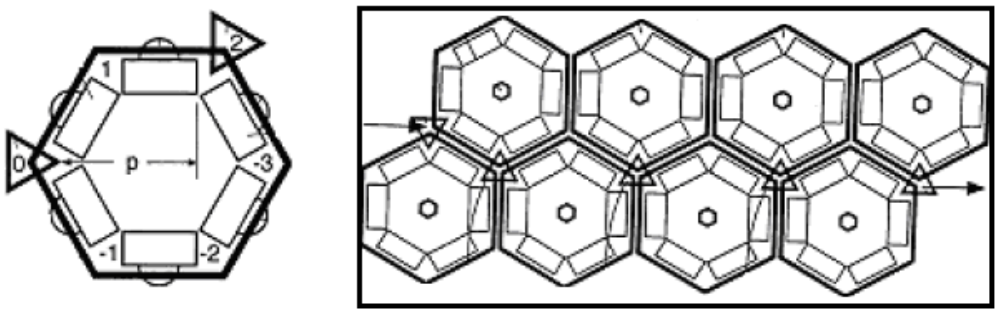
Перспективні розробки в напрямку збільшення функціональної надмірності виробничих ділянок механічної обробки, з переважним використанням машин і обладнання автоматичної дії, спрямовані на розвиток нової концепції мехатронних обробних центрів/модулів, які мають можливістю автоматичної зміни їх інструментальної компонування і просторової конфігурації в реальному режимі роботи преривнопоточного виробництва.

Таким чином, подолання згаданих в роботі протиріч між технологічною базою індустріального (масового і серійного) виробництва, в тому числі і в приладобудуванні, розрахованої на постійний випуск одномодельної продукції, і новими вимогами

конкурентного ринку здійснюється в напрямку заміни верстатів і агрегатів з жорсткою функціональною структурою і компонуванням на гнучкі виробничі модулі/системи (ГВС/FMS) з подальшим переходом в майбутньому на реконфігуровані виробничі системи (РВС/RMS), що володіють можливістю зміни/адаптації просторово-часової організації (архітектури) виробничої системи до змін ринкового попиту на продукцію в залежності від застосовуваних методів організації виробництва. Що розвивається за кордоном концепція RMS, розглядається як альтернатива гнучким виробничим системам.

Її реалізація почалася за кордоном в США і потім в Німеччині, Японії (Koren У., U1soy AG, Mehrabi MG).

З цією метою було створено і активно діє науково-дослідний центр в складі Мічиганського університету, який по суті є національним центром розвитку і реалізації концепції RMS в США (Engineering manufacturing center for Reconfigurable manufacturing systems, University of Michigan), яка вписана в концепцію національного розвитку машинобудівного виробництва до 2030 року (Visionary manufacturing challenges for 2030), як одне з основних науково – дослідних напрямків по створенню реконфігурованих виробництв і підприємств (Reconfigurable enterprises). При цьому центрі утворений і діє експериментальний завод для проведення і реалізації досліджень. Як видно з рис. 2, модулі/осередку RMS працюють одночасно і сконструйовані так, що при виконанні одного робочого циклу одноразово здійснюється кілька операцій різних видів. Останнє виключає необхідність переміщення і очікування пристроїв в черзі між операціями обробки, знижують рівень матеріально виробничих запасів і кількість робочих забезпечуючи мінімізацію витрат і збільшення оборотності ресурсів.



а) б)

Рисунок 2 – Планування ділянки реконфігурованих виробництва на основі стільникових виробничих осередків/модулів

Блочно-модульна/матрична структура реконфігурованою виробничої системи дає можливість компонувати багатовимірні віртуальні технологічні ланцюжки з різною пространственновременной конфігурацією. Це дозволяє по черзі включати технологічні осередки в роботу при послідовному чергуванні подібних технологічних процесів / партій предметів праці на багатопредметних групових/змінно-потокових лініях із суцільним запуском. Організація виробництва на основі подібного роду функціонально надлишкових модулів у вигляді розподіленої виробничо-технологічного середовища дозволяє одночасно (паралельно) виконувати більшу частину технологічних операцій з обробки предметів праці, що є найбільш ефективним щодо продуктивності.

На більшості вітчизняних підприємств машинобудування, незалежно від використовуваних ними моделей організації виробництва (традиційної або гнучкою), етапи виробничого процесу найчастіше виконується послідовно. Це призводить до того, що час проходження вироби через

технологічний процес включає тривалість всіх послідовно здійснюваних етапів виробництва, а також непродуктивні втрати часу на транспортування пристроїв і очікування між операціями по їх обробці.

У свою чергу, посилення конкурентної боротьби виробників на товарних ринках вимагає, крім забезпечення високої якості та оптимальної вартості продукції, – швидкості реакції на запити споживачів. На сьогоднішній день основною конкурентною перевагою стає висока швидкість виконання замовлень в умовах нестабільної кон'юнктури ринку. Дана перевага може бути забезпечено скороченням виробничого циклу (підвищенням потужності виробничої системи) за рахунок організації паралельних технологічних процесів на основі застосування комплексу багатофункціональних обробних центрів зі зосередженим виконанням деталеопераций.

При точкової формі організації виробництва (виконання різних деталеопераций) робота повністю виконується на одному робочому місці. Виріб виготовляється там, де знаходиться його основна частина (аналогом є збірка виробів/будівництво будівлі з переміщенням робочих навколо нього). Організація точкового виробництва має ряд переваг: забезпечується можливість частих змін конструкції виробів і послідовності обробки, виготовлення виробів різноманітної номенклатури в кількості, що визначається потребами ринку; знижуються витрати, пов'язані зі зміною розташування обладнання, підвищується гнучкість виробництва.

Проведення подальших дослідних і дослідно-конструкторських робіт в цьому напрямку передбачає переважне використання машин і устаткування автоматичної дії, і пов'язані з розвитком нової концепції мехатронних обробних центрів/модулів, які мають можливістю автоматичної зміни їх інструментальної компонування і просторової конфігурації в реальному режимі роботи преривнопоточного виробництва. На рисунку 3 представлена компонуюча схема ділянки реконфігурованих виробництва.

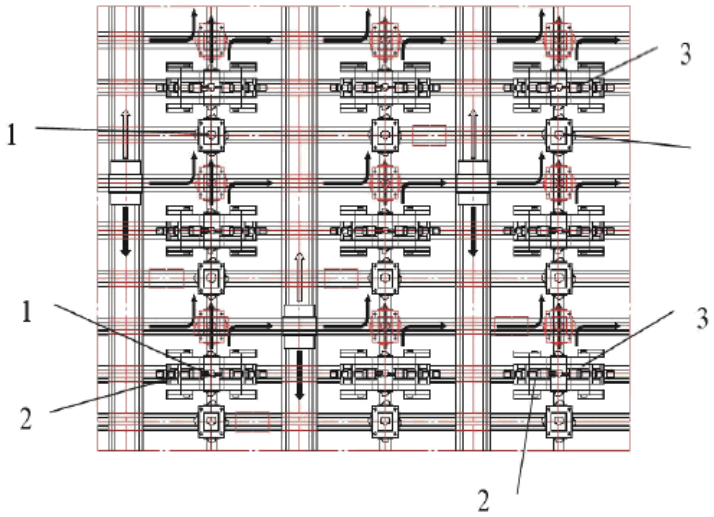


Рисунок 3 – Основні параметри ділянки реконфігурованого виробництва, реалізованого на базі компонуючих виробничих систем/модулів

Останній являє собою технологічно однорідне середовище (матричну площину), що включає перекомпонуемие блочно-модульні елементи основного і допоміжного обладнання, що несуть опорні і будівельні конструкції, а також розміщені між ними шляху переміщень технологічних / транспортних потоків і комунікації обслуговування. Вузол 1, показаний на схемі, є носієм оброблюваних пристроїв або виконавчих механізмів і пристроїв і дозволяє здійснювати одночасну обробку всіх пристроїв, закріплених на бічних гранях (за допомогою

плит з Т-образними пазами і поворотноделітельних столів). При загальній кількості граней, що дорівнює 6, на корпусі носія 1 одночасно на всіх гранях деталі піддаються багатосторонній обробці з різних сторін багато інструментальних вузлами 2 і 3.

Сучасна концепція компонуючих систем реконфігурованих виробництва, що є наслідком інтеграції різних типів виробництва, надає широкі можливості для підвищення потужності/інтенсивності та адаптації виробництва, але вимагає точної координації (синхронізації) паралельних технологічних процесів/операцій в складі єдиної виробничої системи з метою її безперервної роботи. Для того щоб паралельні операції виконувалися в загальному операційному процесі узгоджено, забезпечуючи тим самим безперервну обробку і переміщення предметів праці від операції до операції подібно до потоку, необхідне дотримання двох основних умов відомих в теорії організації як принципи пропорційності (рівномірності) і тимчасової узгодженості (ритмічності) спільних дій.

Реалізація принципу пропорційності здійснюється в складній операційній системі шляхом дискретизації (рівномірного, кратного поділу) підлягає обробці обсягу матеріальних (інформаційних) ресурсів або робіт на рівні частки, які мають загальної для операційної системи розмірністю (наприклад, планово-облікова одиниця в машинобудуванні, або уніфікована одиниця вимірювання кількості інформації в обчислювальній системі – «біт» тощо).

Реалізація принципу узгодженості операцій в часі полягає в дискретизації операційного циклу на рівні за часом відрізки, які мають єдиної загальносистемної розмірністю званої тактом операційної системи. Таким чином, в складній операційній системі, що допускає використання декількох паралельно працюючих процесорів.

Рівномірний і ритмічне здійснення спільних дій (паралельних операцій) в організаційній науці (зокрема в організаційній системотехніці) прийнято називати вирівнюванням руху потоків матеріальних (інформаційних) ресурсів/ходу робіт або синхронізацією, а показник продуктивності операційної системи, вимірюваний як обсяг операцій, що виконується за один такт роботи системи системоквантами (в англомовній термінології «pitch» – пітч).

Розраховується системоквантами/пітч виходячи з числа виробів, що розміщуються в одному транспортному контейнері готових виробів, або в декількох цілих контейнерах або їх частинах. Системоквантами/пітч – це твір часу виготовлення одного виробу на заданому ритм ділянці на число готових виробів в транспортному контейнері. Дане число є основною плановоучетной одиницею при складанні виробничого плану випуску виробів в TPS.

Синхронізація є найбільш ефективним методом системної організації паралельної безперервної роботи декількох виконавчих пристроїв в загальному процесі функціонування операційної системи, а також його адаптації до мінливих умов зовнішнього середовища. У свою чергу це вимагає застосування більш ефективних способів (наприклад, інтелектуальних методів і розподілених/асоціативних моделей) управління комплексіруемимі виконавчими пристроями, паралельно працюючими в єдиній операційній системі, з метою чіткої координації їх спільної діяльності.

Висновки. Описані в роботі методологія і науковий інструментарій системних досліджень складають основу динаміки функціонування складних об'єктів і надають можливість організації ефективного управління високотехнологічним інтегрованим виробництвом, за допомогою автоматизації проектування структурно-компонувальних перетворень адаптивної виробничої системи і параметричного синтезу базових компонентів бізнес-процесів, що реалізуються на її гнучкою технологічній платформі.

Представлені принципи інтелектуалізації та управління роботизованим виробництвом можуть виявитися корисними для вдосконалення теорії та методології управління основною діяльністю на підприємствах, що використовують традиційні типи і методи поточного,

серійного й одиничного виробництва.

В умовах постійного вдосконалення комп'ютерних технологій і подальшого впровадження їх у виробничий процес є перспективнє питання сьогодення.

ЛІТЕРАТУРА

1. Невлюдов І.Ш. Комп’ютерно-інтегровані технології виробництва технічних засобів автоматизації. Частина 1: Підручник – Харків: ФОП Панов А.М., 2021.– 604 с.

2. Невлюдов І.Ш. Виробничі процеси та обладнання об’єктів автоматизації: Підручник для студентів вищих навчальних закладів. – Кривий Ріг: Криворізький коледж НАУ, 2017. – 444 с.

3. Невлюдов І.Ш. Основи виробництва електронних апаратів. – Харків: Компанія СМІТ, 2005. – 592 с.

4. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.

5. Романенко В.Д. Методи автоматизації прогресивних технологій: Навч. посібник. – К .: Вища школа, 1995. – 517 с.

***Науковий керівник**: Аллахверанов Рауф Юсіфович, доцент, кандидат технічних наук, доцент кафедри КІТАР, Харківського національного університета радіоелектроніки.*

ДОДАТОК Б

Лістинг програми 1

```
// Файл 1.cpp


#include <afxwin.h>
#include <cxcore.h>
#include <cv.h>
#include <cvcam.h>
#include <highgui.h>
#include "1.h"
#include <string.h>
#include <assert.h>
#include <math.h>
#include <float.h>
#include <limits.h>
#include <time.h>
#include <ctype.h>

void mycallback(IplImage *img);
IplImage *image1,*src2,*dst,*dst2,*dst3,*dst4,*gray,*dst5,*dst6,*dst7,*dst8;
int contNum=0;
int harrist=0;
int cannyt=0;
int thresh=0;
int thresh2=0;
CvRect camSpace;
CvMoments moments;
CvPoint centre_of_mass;
CvSeq *contour;
CvMemStorage *storage,*storage2;
bool bCreate=true;
double m=2;
CvPoint2D32f center;
static CvHaarClassifierCascade* cascade = 0;
const char* cascade_name = "haarcascade_frontalface_alt2.xml";
int Haart= 0;
BOOL CApp::InitInstance()
{
        m_pMainWnd=new CMainWin;
        m_pMainWnd->ShowWindow(m_nCmdShow);
        m_pMainWnd->UpdateWindow();
        return TRUE;
```

```
}
BEGIN_MESSAGE_MAP(CMainWin,CFrameWnd)
ON_WM_CLOSE()
END_MESSAGE_MAP()
CApp App;
CMainWin::CMainWin()
{ Create (NULL, "OpenCV");
HWND w= this->GetSafeHwnd();
int ncams=cvcamGetCamerasCount();
if (ncams)
{       bCreate=true;
VidFormat vidFmt={800,600,20.0};
cvcamSetProperty(0,CVCAM_PROP_ENABLE,CVCAMTRUE);
cvcamSetProperty(0,CVCAM_PROP_CALLBACK,mycallback);
cvcamSetProperty(0,CVCAM_PROP_WINDOW,&w);
cvcamSetProperty(0,CVCAM_PROP_SETFORMAT,&vidFmt);
cvNamedWindow(cvGetWindowName(w),CV_WINDOW_AUTOSIZE);
cvNamedWindow( "Canny", 1 );
cvCreateTrackbar("CannyTrack","Canny",&cannyt,200,NULL);
cvResizeWindow("Canny",320,200);
cvNamedWindow( "Contour", 3 );
cvNamedWindow("Sobel",2);
cvResizeWindow("Sobel",320,200);
cvNamedWindow("CornerDetect",9);
cvNamedWindow("Harris",5);
cvResizeWindow("Harris",320,200);
cvCreateTrackbar("HarrisTrack","Harris",&harrist,200,NULL);
cvNamedWindow( "threshold", 7 );
cvCreateTrackbar("threshold","threshold",&thresh,200,NULL);
cvNamedWindow( "Adaptive", 8 );
cvCreateTrackbar("Adaptive","Adaptive",&thresh2,200,NULL);
cascade = (CvHaarClassifierCascade*)cvLoad( cascade_name, 0, 0, 0 );
if(!cvcamInit())    MessageBox("Error");
        else    cvcamStart();
}
else MessageBox ("Camera not found");
}
void mycallback(IplImage *src)
{       image1 = src;
        if (bCreate)
        {                       storage=cvCreateMemStorage();
        src2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,3);
        dst=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,3);
        dst2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,3);
```

```
        dst3=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
        dst4=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
        dst5=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,1);
        dst6=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,1);
        dst7=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
        dst8=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,1);
        gray=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
                bCreate=false;
        }
        for (int k=0;k<image1->height;k+=600)
        for(int j=(image1->widthStep)*k;j<(image1->widthStep)*(k+1);j+=image1-
>nChannels)
                {       image1->imageData[j]=(char) 255;
                        image1->imageData[j+1]=0;
                        image1->imageData[j+2]=0;
                }
                        cvFlip(src,src2);
                        cvSobel(src2,dst,1,3,7);
                        cvShowImage("Sobel",dst);
                        cvCvtColor(src2,gray,CV_RGB2GRAY);
                        cvCanny(gray,dst3,25,100+cannyt,3);
                        cvShowImage("Canny",dst3);
CvScalar
color=CV_RGB(rand()&255,rand()&255,rand()&255);contNum=cvFindContours(
dst3,storage,&contour,sizeof(CvContour),CV_RETR_CCOMP,CV_CHAIN_APP
ROX_SIMPLE);
                        for(;contour!=0;contour=contour->h_next)
                { cvDrawContours(dst7,contour,color,color,-1,1,8);
cvMoments(contour,&moments,0);
centre_of_mass.x=moments.m10/moments.m00;
centre_of_mass.y=moments.m01/moments.m00;
cvCircle(dst7,centre_of_mass,5,CV_RGB(255,255,255),1,8,0);}
cvShowImage("Contour",dst7);
cvCornerHarris(gray,dst8,3,7,harrist/1000.0);
cvShowImage("Harris",dst8);
static CvScalar colors[] =
   {            {{0,0,255}},
                {{0,128,255}},
                {{0,255,255}},
                {{0,255,0}},
                {{255,128,0}},
                {{255,255,0}},
                {{255,0,0}},
                {{255,0,255}}
```

```
    };
    double scale = 1.3;
    IplImage* gray2 = cvCreateImage( cvSize(src->width,src->height), 8, 1 );
    IplImage* small_img = cvCreateImage( cvSize( cvRound (src->width/scale),
    cvRound (src->height/scale)),8, 1 );    int i;
    cvCvtColor( src, gray2, CV_BGR2GRAY );
    cvResize( gray2, small_img, CV_INTER_LINEAR );
    cvEqualizeHist( small_img, small_img );
    cvClearMemStorage( storage );
    if( cascade )
    {  double t = (double)cvGetTickCount();
     CvSeq* face = cvHaarDetectObjects( small_img, cascade,
storage,1.1,1,0,cvSize(30,30));
     t = (double)cvGetTickCount() - t;
     printf( "detection time = %gms\n", t/((double)cvGetTickFrequency()*1000.) );
       for( i = 0; i < (face ? face->total : 0); i++ )
       {  CvRect* r = (CvRect*)cvGetSeqElem( face, i );
          CvPoint center;
          int radius;
          center.x = cvRound((r->x + r->width*0.5)*scale);
          center.y = cvRound((r->y + r->height*0.5)*scale);
          radius = cvRound((r->width + r->height)*0.5*scale);
          cvCircle( src, center, radius, colors[i%8], 3, 8, 0 );
       }
    }      cvPreCornerDetect(dst3,dst5,5);
           cvShowImage( "CornerDetect", dst5 );
           cvLogPolar( dst, src2, cvPoint2D32f(src->width/2,src->height/2), 40,
CV_INTER_LINEAR+CV_WARP_FILL_OUTLIERS+CV_WARP_INVERSE_
MAP );
           cvShowImage( "log-polar", dst );
           cvShowImage( "inverse log-polar", src2 );
           cvWaitKey();
           cvThreshold(src2,dst,2,10+thresh,CV_THRESH_BINARY_INV);
           cvShowImage( "threshold", dst );
cvAdaptiveThreshold(dst3,dst4,5+thresh2,CV_ADAPTIVE_THRESH_MEAN_C,
CV_THRESH_BINARY,3,5);
cvShowImage( "Adaptive", dst4 );
cvShowImage( "result", src );
cvReleaseImage( &gray2 );
cvReleaseImage( &small_img );
cvZero(dst);
cvZero(gray);;
cvZero(dst2);
cvZero(dst3);
```

```
cvZero(dst6);
cvZero(dst7);
cvZero(dst8);
cvZero(dst5);
}
void CMainWin::OnClose()
{ cvcamStop();
cvDestroyAllWindows();
cvcamExit();
if(!bCreate)cvReleaseImage(&dst);
DestroyWindow();
}

// Файл 1.h

#if _MSC_VER > 1000
#pragma once
class CMainWin : public CFrameWnd
{
public:
                CMainWin();
                void OnClose();
                DECLARE_MESSAGE_MAP();
};
class CApp : public CWinApp
{
public:
        BOOL InitInstance();
};
```

ДОДАТОК В

Лістинг програми 2

```
#include "cv.h"
        #include "cxcore.h"
        #include "highgui.h"
        #include "math.h"
        #include <iostream>
        #include <stdio.h>
        #include <string.h>
        #include <conio.h>
        #include <sstream>

        using namespace std;
        double angle( CvPoint* pt1, CvPoint* pt2, CvPoint* pt0 )
        {
         double dx1 = pt1->x - pt0->x;
         double dy1 = pt1->y - pt0->y;
         double dx2 = pt2->x - pt0->x;
         double dy2 = pt2->y - pt0->y;
 return (dx1*dx2 + dy1*dy2)/sqrt((dx1*dx1 + dy1*dy1)*(dx2*dx2 + dy2*dy2) +
1e-10);}
        CvSeq* findSquares4(  IplImage* img,  CvMemStorage* storage )
        {  double s = 0,  t = 0;
           CvSeq* contours =  NULL;
           CvSeq* result =  NULL;
CvSeq* squares  =  cvCreateSeq(  0,  sizeof( CvSeq),  sizeof( CvPoint),  storage );
cvFindContours(  img,  storage,  &contours,  sizeof(  CvContour),
CV_RETR_LIST,  CV_CHAIN_APPROX_SIMPLE,  cvPoint( 0, 0) );
       while(  contours )
          {       result  =  cvApproxPoly(  contours,  sizeof( CvContour),  storage,
CV_POLY_APPROX_DP,  cvContourPerimeter( contours)*0.02,  0 );
       if(  result->total  ==  4  &&  fabs(  cvContourArea(  result,
CV_WHOLE_SEQ))  >  1000  &&  fabs(  cvContourArea(  result,
CV_WHOLE_SEQ))  <(  img->height  *  img->width/2  )  &&
cvCheckContourConvexity( result) )
             {          s  =  0;
                for(  int i  =  0; i < 5; i++ )
                {       if(  i >=  2 )    {    t  =  fabs( angle(  (
CvPoint*)cvGetSeqElem(  result,  i ),  ( CvPoint*)cvGetSeqElem(  result,  i-2 ),  (
CvPoint*)cvGetSeqElem(  result,  i-1 )));
        s  =  s > t ? s : t;             }          }
```

```
if( s < 0.5 )for( int i = 0; i < 4; i++ )cvSeqPush( squares,
(CvPoint*)cvGetSeqElem( result, i )); }
contours = contours->h_next; }
cout<<">>> rectangles: "<< squares->total/4 <<endl;
return squares;}
void drawSquares(IplImage *img, CvSeq* squares )
{ CvFont font;
cvInitFont( &font, CV_FONT_HERSHEY_SIMPLEX, 0.4f, 0.4f, 0,1, 8 );
int i,j = 0;
CvSeqReader reader;
cvStartReadSeq( squares, &reader, 0 );
for( i = 0; i < squares->total; i += 4 )
        {       j++;
                CvPoint pt[4], *rect = pt;
                int count = 4;
                // read 4 vertices
                CV_READ_SEQ_ELEM( pt[0], reader );
                CV_READ_SEQ_ELEM( pt[1], reader );
                CV_READ_SEQ_ELEM( pt[2], reader );
                CV_READ_SEQ_ELEM( pt[3], reader );
                cvCircle( img, pt[1], 2, CV_RGB(200,0,0),1, 8, 0 );
                cvCircle( img, pt[2], 4, CV_RGB(200,0,0),1, 8, 0 );
                cvLine( img, pt[1], pt[2], CV_RGB(255,255,255),1, 8, 0 )
double angle = abs(pt[1].y-pt[2].y)/sqrt((pt[1].x-pt[2].x)*(pt[1].x-pt[2].x)+(pt[1].y-
pt[2].y)*(pt[1].y-pt[2].y)+0.00001);   char st[255];   sprintf(st, "%2f", angle);
cvLine( img, cvPoint(0,img->height/2), cvPoint(img->width,img->height/2),
CV_RGB(200,200,200),1, 8, 0 );
cvLine( img, cvPoint(img->width/3,0), cvPoint(img->width/3,img->height),
CV_RGB(200,200,200),1, 8, 0 );
cvLine( img, cvPoint(img->width/3*2,0), cvPoint(img->width/3*2,img->height),
CV_RGB(200,200,200),1, 8, 0 );
cvPutText( img, st, pt[1], &font, CV_RGB(200,0,0));
cvPolyLine( img, &rect, &count, 1, 1, CV_RGB(255,255,255), 1, CV_AA, 0 );}}
int main()  {  int c = 0;
CvCapture* capture = cvCaptureFromCAM(0);
f(!cvQueryFrame(capture)){cout<<"Video capture failed, please check the
camera."<<endl;}else{cout<<"Video camera capture status: OK"<<endl;};
        CvSize sz = cvGetSize(cvQueryFrame(capture));
        cvNamedWindow( "out", 0);
        cvNamedWindow( "src", 1);
        IplImage* src = cvCreateImage( sz, 8, 3 );
        IplImage* gray = cvCreateImage( sz, 8, 1 );
        IplImage* pyr = cvCreateImage( cvSize(sz.width/2, sz.height/2), 8, 3 );
        IplImage* tgray = cvCreateImage( sz, 8, 1 );
```

```
        CvMemStorage* storage = cvCreateMemStorage(0);
        CvSeq* contours = NULL;
        CvMemStorage* mainStorage = cvCreateMemStorage(0);
        CvMemStorage* cstorage = cvCreateMemStorage( 0);
        CvSeq* circles = NULL;
        while(c != 27)    {          IplImage* img = NULL;
                                     IplImage* out = NULL;
        src = cvQueryFrame( capture);
        img = cvCloneImage( src); out = cvCloneImage( src);
        cvPyrDown( img, pyr, 7 ); cvPyrUp( pyr, img, 7 );
        cvSetImageCOI( img, 2 );
        cvCopy( img, tgray, 0 );
        cvThreshold( tgray, gray, 100, 255, CV_THRESH_BINARY );
        cvShowImage( "src", gray);
        cvFindContours( gray, storage, &contours, sizeof(CvContour),
CV_RETR_LIST, CV_CHAIN_APPROX_SIMPLE, cvPoint(0,0) );
        circles =  cvHoughCircles( gray, cstorage, CV_HOUGH_GRADIENT, 1,
gray->height/16, 8, 10, 4, 50 );
        cout <<"Total circles: "<< circles->total <<endl;
        for(  int i  =  0; i < circles-> total; i++ )
                {       float* p  =  ( float*)cvGetSeqElem(  circles,  i );
        cvCircle(  out,  cvPoint( cvRound( p[0]), cvRound( p[1])),  2,  CV_RGB(
200, 0, 0),  -1,  8,  0 );
        cvCircle(  out,  cvPoint( cvRound( p[0]), cvRound( p[1])),  cvRound( p[2]),
CV_RGB( 200, 0, 0),  1,  8,  0 );        }
        drawSquares(out,findSquares4(gray,mainStorage));
        cvReleaseImage( &img);
        cvShowImage( "out", out);
        cvReleaseImage( &out);
         c = cvWaitKey(10);    }
           cvReleaseCapture( &capture );
           cvDestroyAllWindows();}
```

ДОДАТОК Г

ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

Формат	Позиція.	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮИК. 466.459.022 ПЗ	Пояснювальна записка	106	с.
			Інші документи		
			Апбробація результатів		
А4	2		досліджень	9	с.
А4	3		Лістинг програми 1	5	с.
А4	4		Лістинг програми 2	3	с.
А4	5		Демонстраційний матеріал	16	*у форматі *.ppt*

Змін	А	№ докум.	Підпи	Дата	ГЮИК. 466.459.022 ВД		
Розроб.		Пащенко О.С.			Автоматизація процесу розпізнавання деталей на поточному виробництві Відомість кваліфікаційної роботи	Літер Н	Аркуш 1 / Аркушів 1
Переві		Косенко Н.В.					
						Кафедра КІТАР ХНУРЕ	
Н.конт.		Демська Н.П.					
Затв.		Невлюдов І.Ш.					