

SCIENTIFIC PAPERS
OF THE UNIVERSITY OF PARDUBICE

Series B

The Jan Perner Transport Faculty

4 (1998)

**ANALÝZA MECHANICKÝCH SÚSTAV S VYUŽITÍM
COMPONENT WORKS™**

Tomáš LACK, Juraj GERLICI

Katedra koľajových vozidiel, motorov a zdvíhadiel, Strojnícka fakulta,
Žilinská univerzita Žilina, Slovenská republika

1. Úvod

Analýza mechanických systémov je pojem, ktorý môže nadobúdať rôzne formy realizácie. Takými formami sú napríklad experimentálne pozorovanie a popis existujúcej mechanickej sústavy (spaľovací motor, dopravné zariadenie apod.), kde je potrebné zostaviť riadiaci program pre meranie, priebežné zobrazovanie meraných hodnôt, ich zápis a ukladanie do pamäti počítača, ako aj vyhodnotenie a spracovanie výsledkov metódami štatistickej a matematickej analýzy. Inou formou realizácie je tvorba algoritmu, ktorý matematicky popíše (aj fyzicky neexistujúcu mechanicú sústavu), nasimuluje jej správanie sa pri danom zaťažení a výsledky (napr. určenie vlastných čísel, vlastných vektorov a pod.) upraví do formy prijateľnej pre vedeckú prezentáciu problému. Zdanlivo dve rozdielne oblasti približuje súbor knižníc s názvom Component Works™ (CW) čo je sada 32-bitových ActiveX komponent pre zber dát, numerickú analýzu, výpočty a zobrazenie údajov. Je možné ich zaradiť pomocou programovacích prostriedkov podporujúcich technológiu ActiveX. Použitie podobných prostriedkov je veľmi aktuálne, preto uvádzame aj vyskúšané príklady.

2. Component Works™

Component Works™ (CW) je sada 32-bitových ActiveX komponent pre zber dát, numerickú analýzu, výpočty a zobrazenie údajov. Je možné ich zaradiť pomocou programovacích prostriedkov podporujúcich technológiu ActiveX komponent ako sú MS Visual Basic, MS Visual C++, Borland Delphi, Borland C++ Builder a iných do programov určených pre daný konkrétny účel. Na obrázku 1 je zobrazená schéma takéhoto programu. Program má

kontakt s analyzovaným objektom prostredníctvom snímačov rôzneho typu (zrýchlení, rýchlostí, polohy, napätí, teplôt, ...) a akčných členov z ktorých sú dáta prostredníctvom komponent určených na ovládanie obvodov meracieho systému (CWAIPoint, CWAI, CWAOPoint, CWAO, CWDIO, CWDI, CWDO, CWCounter, CWPulse a CWDAQTools) predávané akčným členom alebo zaznamenávané programom. Vstupom do programu sú okrajové a začiatkové podmienky ako aj dáta charakterizujúce parametre modelu pre porovnanie a možnú korekciu výpočtového modelu, výpočtovej metódy, parametrov výpočtu alebo postupu výpočtu. Pre naprogramovanie výpočtovej časti programu sú k dispozícii výpočtové komponenty implementujúce operácie s komplexnými číslami, multidimenzionálnymi komplexnými poliami, jedno, dvoj a multidimenzionálnymi poliami a prvkami polí, a operácie s maticami (CWComplex, CWArray a CWMatrix). Pokiaľ dáta nie sú vo vhodnej forme je možné použiť pre ich spracovanie komponenty zamerané na spracovanie a generovanie signálov (CWStat a CWDSP). Komponenty CW komunikujú s hlavným programom pomocou vlastností, metód a udalostí. Vlastnosti komponent je možné nastavovať priamo z programu alebo pomocou stránok vlastností charakteristických pre ActiveX komponenty. Neoddeliteľnou časťou CW sú vizuálne komponenty (CWButton, CWSlide, CWKnob, CWGraph a CWNumEdit) rozširujúce základné komponenty vývojového prostredia o možnosť interaktívneho ovládania programu a vhodnej vizualizácie nameraných a vypočítaných dát.

3. Príklad použitia komponent pre ovládanie obvodov meracieho systému

3.1 Meranie frekvencie – použitie komponenty CWCounter

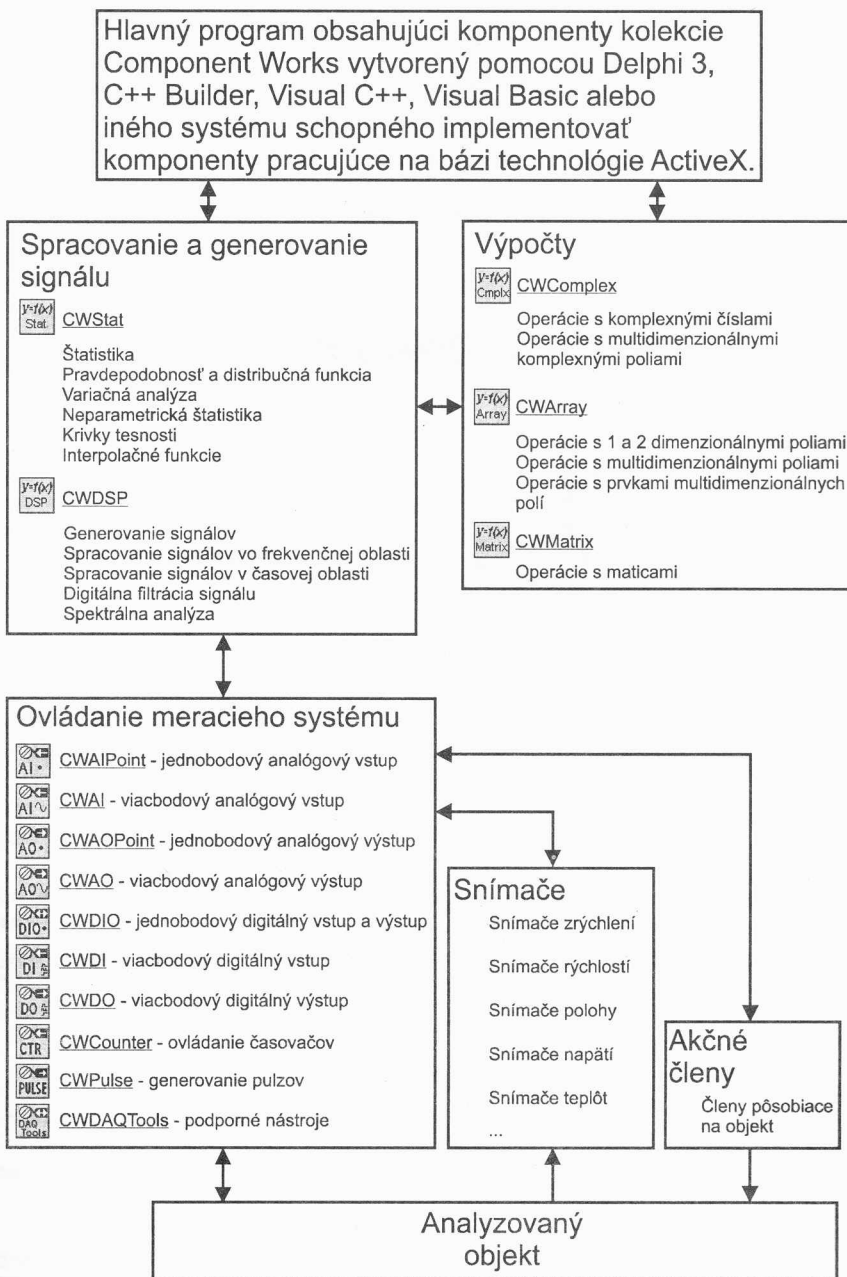
type

```
TForm1 = class(TForm) { Deklarácia metód }
  CWCounter1: TCWCounter;
  procedure StartButton1Click(Sender: TObject);
  procedure StopButton1Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure CWCounter1AcquiredData(Sender: TObject;
    var Measurement: OleVariant; Overflow: WordBool);
end;
```

implementation

```
procedure TForm1.StartButton1Click(Sender: TObject);
begin
  { Vlastnosti komponenty CWCounter1 sú nastavené v „Object inspector“ }
  CWCounter1.Configure;
  Timer1.Enabled := True;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  CWCounter1.Start; { Spustenie merania v CWCounter1 }
end;
```



Obr. 1 Schématický pohľad na systém Component Works Components a jeho použitie.

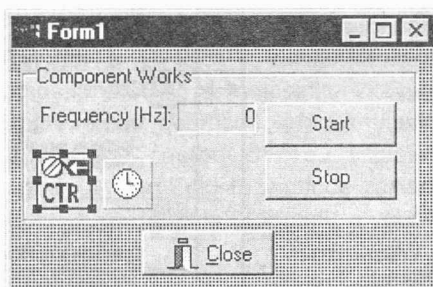
```

procedure TForm1.CWCounter1AcquiredData(Sender: TObject; var Measurement:
OleVariant; Overflow: WordBool);
begin
    if StopButton1.Enabled then { Vypísanie výsledku merania }
        StaticText1.Caption := FloatToStrF(Measurement, fFixed, 10, 0);
end;

procedure TForm1.StopButton1Click(Sender: TObject);
begin
    Timer1.Enabled := False; { Zastavenie merania }
end;

end.

```



Obr. 2 Pohľad na formulár s použitím CWCounter.

3.2 Snímanie analógového signálu a jeho dolnofrekvenčné filtrovanie

– použitie komponent CWAI a CWDSP.

```

type
    TForm1 = class(TForm) { Deklarácia metód }
        CWAI1: TCWAI;
        procedure StartButton1Click(Sender: TObject);
        procedure StopButton1Click(Sender: TObject);
        procedure ReadButton1Click(Sender: TObject);
        procedure CWAI1AcquiredData(Sender: TObject; var Voltages,
BinaryCodes: OleVariant);
    private
        vData : OleVariant; { Private declarations }
        ReadSpec: OleVariant;
    end;

implementation

procedure TForm1.StartButton1Click(Sender: TObject);
begin
    { Vlastnosti komponenty CWAI1 sú nastavené v „Object inspector“ }
    CWAI1.Configure;
    CWAI1.Start;
end;

procedure TForm1.StopButton1Click(Sender: TObject);
begin
    CWAI1.Stop;
end;

```

```

procedure TForm1.ReadButton1Click(Sender: TObject);
var
  sData: OleVariant;
begin
  ReadSpec := CreateOleObject('CWAIReadSpec');
  ReadSpec.ReadMode := cwaiFromEndOfData;
  ReadSpec.ReadOffset := -100;
  CWAI1.Read(100, vData, ReadSpec);           // Snímanie dát
  sData := CWDSP1.BwLPF(vData, 1000, 10, 5); // digital lowpass
  Butterworth filter
  CWGraph1.PlotY(vData, 0, 1, True);         // Vykreslenie
  nasnímaných dát
  CWGraph2.PlotY(sData, 0, 1, True);        // Vykreslenie
  filtrovaných dát
end;

procedure TForm1.CWAI1AcquiredData(Sender: TObject; var Voltages,
  BinaryCodes: OleVariant);
var
  sData: OleVariant;
begin
  CWGraph3.PlotY(Voltages, 0, 1, True); //Vykreslenie priebehu
  napätia
end;
end.

```

4. Príklad použitia komponent pre výpočty

4.1 Výpočet neznámych systému lineárnych algebraických rovníc pomocou LU rozkladu

- použitie komponenty CWMatrix.

```

procedure TForm1.SolveButton1Click(Sender: TObject);
var
  a, b, c, x, y, p, sign: OleVariant;
begin
  a := VarArrayCreate([0, 99, 0, 99], varDouble);
  y := VarArrayCreate([0, 99], varDouble);
  //... Vloženie hodnôt do matice a a vektora pravej strany y ...
  CWMatrix1.LU(a, b, p, sign);           // LU decomposition of a
  CWMatrix1.ForwSub(b, y, p, c);        // forward substitution
  x := CWMatrix1.BackSub(b, c);         // backward substitution
end;

```

4.2 Výpočet vlastných čísel a vlastných vektorov

- použitie komponenty CWMatrix.

```

procedure TForm1.SolveButton1Click(Sender: TObject);
var
  a, evalreal, evalimag, evecreal, evecimag: TOleVariant;
begin
  a := VarArrayCreate([0, 99, 0, 99], varDouble);
  // ... Vloženie hodnôt do matice a ...

```

```

CWMATRIX1.GenEigenValueVector(a, 1,
    evalreal, //Real part of the Eigenvalues of Input Matrix
    evalimag, //Imaginary part of the Eigenvalues of Input Matrix
    evecreal, //Real part of the Eigenvectors of Input Matrix
    evecimag); //Imaginary part of the Eigenvectors of the Input
Matrix
end;

```

5. Záver

Z uvedeného vyplýva, že variabilita použitia CW je vysoká, neviaže sa tematicky len na jedinú oblasť práce, jej použitie môže aj pri štandardnej úrovni programátorských skúseností a námahe viesť k zostaveniu vysoko kvalitných programových aplikácií.

Lektoroval: Ing. Jaromír Zelenka, CSc.

Předloženo v lednu 1999.

Literatúra

- [1] Lack T., Gerlici J.: Tvorba systémov pre programovo riadené meranie, výpočet a vyhodnotenie údajov; In: Zborník: XII. Medzinárodná konferencia "Súčasný problémy v koľajových vozidlách - PRORAIL'95", Žilina, VŠDS Žilina, 1995.
- [2] Sága M., Vavro J.: Spectral tuning of mechanical structures, 2nd Scientific Conference of Faculty of Mechanical Engineering, Žilina 1995.
- [3] Gerlici J., Lack T.: Analýza dynamiky kmitania sústav so sústredenými hmotnosťami; XIII. Mezin. konference "Současné problémy v kolejových vozidlech", Č. Třebová 1997.
- [4] Sága M.: Designing of Dynamic Structures, 3rd Scientific Conference of Faculty of Mechanical Engineering, Žilina 1997.
- [5] Sága M., Vavro J.: Navrhovanie spektrálnych vlastností prúťových, nosníkových a škrupinových mechanických sústav, Strojnícky časopis 5/45, Bratislava, 1994.
- [6] Strážovec, I., Kriššák, P., Grinč, L.: Visual Programming in Lab View. In. Zborník XVII. Medzinárodnej vedeckej konferencie katedier mechaniky tekutín a termomechaniky, Herľany 1998.

Resumé

ANALÝZA MECHANICKÝCH SÚSTAV S VYUŽITÍM

COMPONENT WORKS™

Tomáš LACK, Juraj GERLICI

Príspevok pojednáva o možnosti analýzy mechanických sústav pomocou programového balíka Component Works™ z hľadiska dynamiky a merania. Programový balík predstavuje ucelený celok využiteľný vo vývojových systémoch schopných implementovať komponenty typu ActiveX™.

Summary

MECHANICAL SYSTEMS ANALYSIS WITH COMPONENT WORKS™

Tomáš LACK, Juraj GERLICI

The paper deals with the possibility of mechanical systems analysis with program pack Component Works™ from the point of view of dynamics and measurement. Program pack is integrated whole, that can be used in development systems that are able to implement the components of ActiveX™.type.

Zusammenfassung

DIE ANALYSE DER MECHANISCHEN SYSTEME MIT COMPONENT WORKS™

Tomáš LACK, Juraj GERLICI

Der Beitrag behandelt über die Möglichkeit der Analyse der mechanischen Systeme mittels des Software Component Works™ in Hinsicht der Dynamik und der Messung. Das Software stellt ein Ganzheit vor, die die Komponenten des ActiveX Typs in dem Entwicklungssystemen einschließen zu fähig ist.

