

AIC-Automatisation avancée, intelligence artificielle et cognitive

8. Commande à algorithme génétique

Jean-Daniel Dessimoz



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch



HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

1

AIC-Automatisation avancée, intelligence artificielle et cognitive

Contenu

- **Introduction**
- **Notion de modèle ; métrique pour le traitement d'information et pour la cognitive**
- **Choix d'une structure de commande**
- **Intelligence artificielle et « machine learning »**
- **Commande à logique floue**
- **Commande neuronale, yc. « deep learning »**
- **Commande multimodale**
- **Commande à algorithme génétique**
- **Robots mobiles autonomes**
- **Robot humanoïde NAO**
- **Conclusion**

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

2

Contenu des *Exposés et exercices*

Notion de modèle ; métrique pour le traitement d'information et pour la cognitive	4p
Choix d'une structure de commande	2p
Intelligence artificielle et inférences bayésiennes	2p
Commande à logique floue	2p
Commande neuronale	2p
Commande multimodale	2p
Commande à algorithme génétique	2p
Robots mobiles autonomes et humanoïdes	4p
Réserve et contrôle continu (TE, corr.)	6p

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

3

Travaux de laboratoire associés

Estimation de grandeurs cognitives (essais en simulation avec programmes d'évitement d'obstacles)	L-AIC-1
Test d'intelligence artificielle selon Turing et utilisation d'Eliza	L-AIC-2
Commande neuronale	L-AIC-3
Commande à logique floue	L-AIC-4
Commande à algorithme génétique	L-AIC-5
Commande multimodale	L-AIC-6
Robot mobile autonome	L-AIC-7
Robot humanoïde NAO	L-AIC-8
Inférences bayésiennes	L-AIC-9
Sur demande, l'étudiant peut échanger l'une des manipulations ci-dessus par un autre sujet (cf. manipulations LaRA)	

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

4

AIC-Automatisation avancée, intelligence artificielle et cognitive

Contenu

- *Introduction*
- *Notion de modèle ; métrique pour le traitement d'information et pour la cognitive*
- *Choix d'une structure de commande*
- *Intelligence artificielle et « machine learning »*
- *Commande à logique floue*
- *Commande neuronale, yc. « deep learning »*
- *Commande multimodale*
- **Commande à algorithme génétique**
- *Robots mobiles autonomes*
- *Robot humanoïde NAO*
- *Conclusion*

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

5

Commande à algorithme génétique



HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

6

Commande à algorithme génétique

Méthodes non-conventionnelles pour la commande de processus



- ...
- **Les algorithmes génétiques**

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

7

Commande à algorithme génétique

Contenu

- **Principe de l'évolution naturelle**
- **Description du logiciel RaGe**
- **Les composants**
- **Les messages d'erreurs**
- **Dimensionnement du régulateur par AG**
- **Simulation**

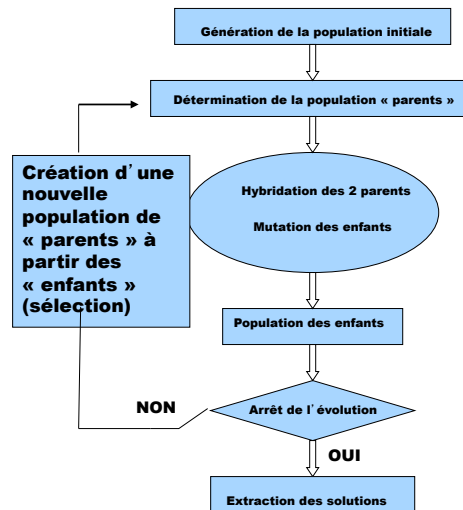
HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

8

Commande à algorithme génétique

Les algorithmes génétiques

- C'est une famille d'algorithmes basés sur l'évolution de l'espèce
- L'évolution d'une espèce est simplement une suite successive d'améliorations afin qu'elle soit le mieux adaptée au milieu dans lequel elle évolue

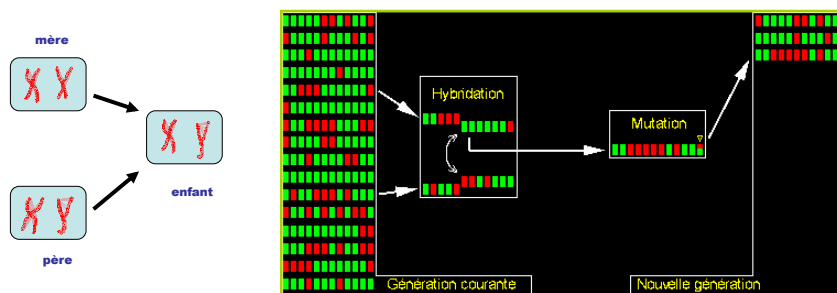


HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

9

Commande à algorithme génétique

- Ils travaillent sur des chaînes de caractères particulières (chromosomes) généralement définies de manière aléatoire.



- 1 Les individus vont s'évaluer, se sélectionner, se reproduire et subir des mutations

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

10

Commande à algorithme génétique

Description du logiciel RaGe

- **Ce logiciel permet de simuler graphiquement un pendule inversé en mode libre ou régulé**
- **Le type de régulation employé est une régulation proportionnelle-dérivée**
- **Le dimensionnement du régulateur peut se faire soit manuellement, soit, et c'est la partie la plus originale, par l'intermédiaire d'un algorithme génétique.**

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

11

Commande à algorithme génétique

Dimensionnement du régulateur par AG

- **Un algorithme génétique comprend classiquement 4 phases principales telles qu'indiquées dans le tableau ci-dessous:**

Phases	Actions	Paramètres de départ
Initialisation	Génération d'une population de régulateurs de manière aléatoire.	Population maximum
Hybridation	Composition par paires des chromosomes de la population pour générer des enfants.	Nombre de générations
Mutation	Modification aléatoire d'un des éléments du chromosome (phénomène « rare »).	Probabilité de mutation
Sélection	Les 75% les plus performants sont conservés.	Critère de sélection

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

12

Commande à algorithme génétique

Simulation graphique

Simulation du pendule avec régulateur P+D dimensionné par algorithme génétique

Choix des paramètres:

Coefficient de frottement: 0.95

Angle de départ: 45

Paramètre Kp: 4.9

Paramètre Td: 1

Actions:

Lâcher Stop

Reset Quitter

Régul. P+D Pousser

Dim. du régulateur par AG

RaGe

Valeurs:

Theta: 0.00

dTheta: 0.00

ddTheta: 0.00

Courant: 0.00

Réponse en amplitude du courant lors de la régulation

Effacer le graphe

Intervalle de mesure: 1 ms

Axe vertical: amplitude du courant [A]

Axe horizontal: temps [s]

i 5 A / DIV

--> 20 ms / DIV

Temp. de stabilisation (ms): 19

Régulation
 Mode libre
 Arrêt du balancier

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

13

Commande à algorithme génétique

Recherche de Kp par AG

Valeurs à déterminer:

N max: 100

Critère de sélection: 4.9

Probabilité de mutation (en %): 0.1

Nombre de génération(s): 1

Génération de parents de manière aléatoire:

4.8442	0.5684	2.2171	3.4073	2.5229
4.8442	2.8093	5.9529	0.0381	0.3003

Valeurs des enfants résultants:

4.375	0.5625	2.0625	3.8125	2
-------	--------	--------	--------	---

Progression du calcul:

Hybridation: OK

Mutation selon taux voulu: OK

Nombre de génération(s): OK

Génération d'une population de N régulateurs:

3.5942	3.2383	1.0365	3.3368
4.7619	0.5684	3.8961	5.7686
0.3844	3.2721	2.6267	0.0381
2.7401	5.5816	0.598	2.3737
3.8534	3.2083	4.3261	4.6674
0.1874	1.8598	1.2161	1.0914
1.6606	5.8188	4.8846	2.8714
1.3744	3.1877	4.6715	3.981
5.8382	3.7196	5.0821	5.6366
3.9226	3.4169	1.2932	3.2435
2.1132	3.5861	2.5229	1.3749
0.1238	0.8257	3.9657	1.2787
0.0367	0.3796	1.256	4.3487
1.2601	1.3809	1.6236	5.534
0.9655	0.3003	2.2171	4.5328
3.4073	4.7268	0.3274	3.8779
0.5502	2.8446	4.9274	2.8093

Recherche de Kp par AG

Retour à la simulation

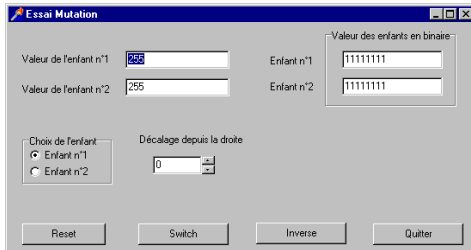
Meilleur résultat

Valeur de Kp trouvée par AG: 4.875

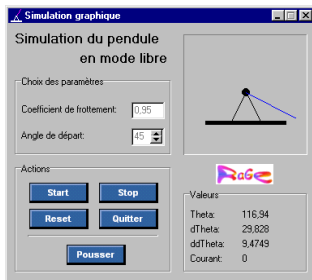
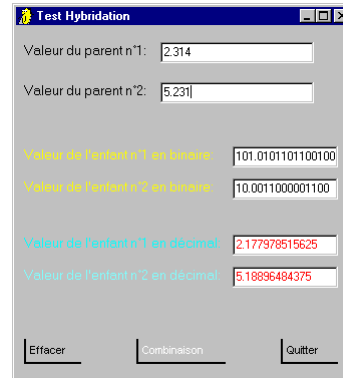
HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

14

Commande à algorithme génétique



Tests détaillés



Programme de mutation

Simulation en mode libre

Programme d'hybridation

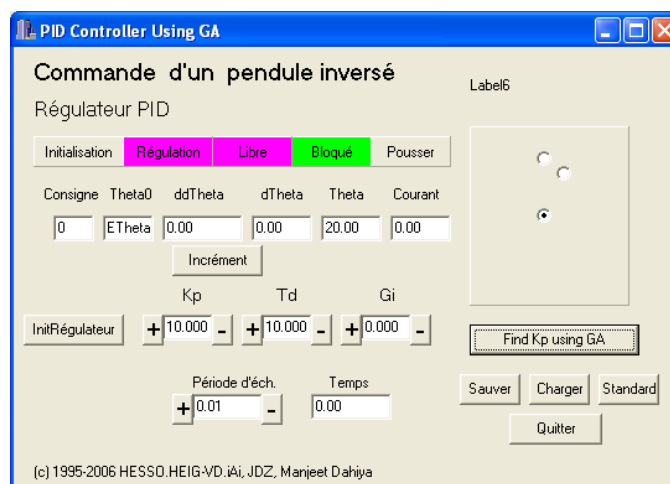
HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

15

Commande à algorithme génétique

Implémentation en C++

Fiche principale



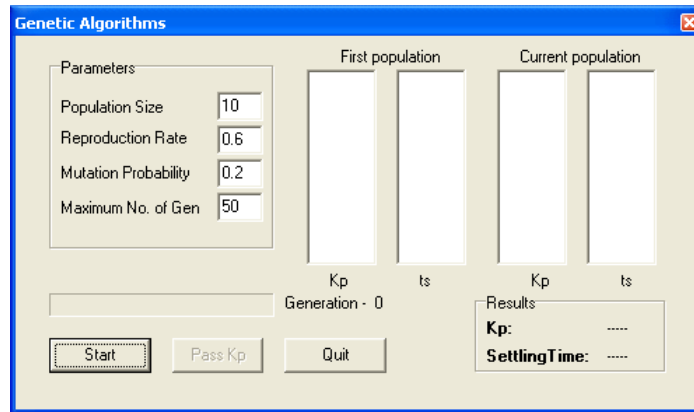
HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

16

Commande à algorithme génétique

Implémentation en C++

Fiche secondaire



Kp: Gain of PID controller

Ts: Settling time (calculated by simulation of pendulum)

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

17

Commande à algorithme génétique

Implémentation en C++

Crossover

```
void crossover1(TChromosome &c1,
TChromosome &c2, int d)
{
    AnsiString P1, P2, O1, O2;
    int length, i;

    P1 = c1.get_valueBin();
    P2 = c2.get_valueBin();

    length = P1.Length();
    i = length/3;

    O1 = P1.SubString(1,i)
+P2.SubString(1+i,length-2*i)+P1.SubString(length-
i+1,i);
    O2 = P2.SubString(1,i)
+P1.SubString(1+i,length-2*i)+P2.SubString(length-
i+1,i);

    if (d==1)
    {
        c1.set_values(BinToDec(O1));
        c1.evaluate_fitness_chromosome();
    }
    else if (d==2)
    {
        c1.set_values(BinToDec(O1));
        c1.evaluate_fitness_chromosome();
        c2.set_values(BinToDec(O2));
        c2.evaluate_fitness_chromosome();
    }
}
```

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

18

Commande à algorithme génétique

Implémentation en C++

Mutation

```
void mutation1(TChromosome & c)
{// i randomly chosen genes are mutated

  AnsiString bin = c.get_valueBin();
  int length = bin.Length();
  int i = round((Mut_Prob*100)/(length));
  if(i>length) i = length;
  if(i<0) i = 0;

  AnsiString a = random_1(length);

  for(int j=0;j<i;j++)
  {int k = StrToInt(a.SubString(j+1,1))
  +1;
  int i = StrToInt(bin.SubString(k,1));
  if(i==0) i = 1;
  else i = 0;
  bin.Delete(k,1);
  bin.Insert(IntToStr(i),k);
  }
  int test = BinToDec(bin);
  c.set_values(BinToDec(bin));
  c.evaluate_fitness_chromosome();
  //delete [] a;
}
```

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

19

Commande à algorithme génétique

Implémentation en C++

Evaluation du temps
de stabilisation
(pour sélection)

```
float SettlingTime(float value)
{
  Kp = value;
  Form1->TAffichage->Enabled = false;
  Form1->TBalancier->Enabled = false;
  Theta=177.97;
  Courant=0;
  dTheta=-854.56;
  ddTheta=-400.91;
  dt = .01;
  tstop = 0;

  for(t=0; ;t=t+dt)
  { UpdateBalancier();
    CommandePID();

    delta=ConsigneTheta-Theta;

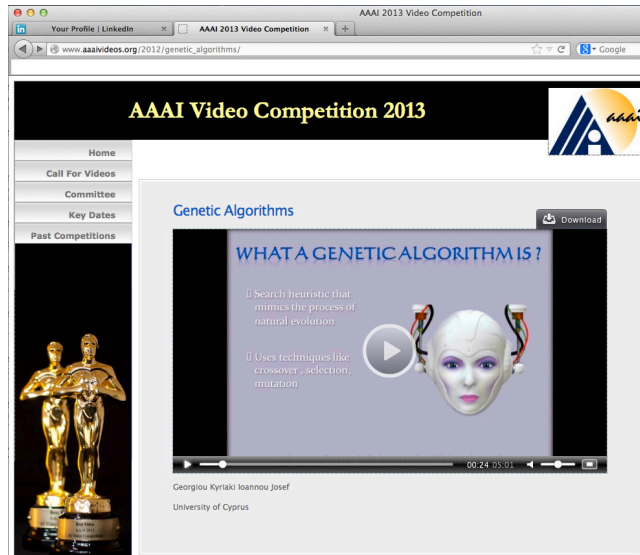
    if(((delta>=0)&&(delta<4.0))||((delta<=0)&&((-delta)<4.0)))
      Converged=Yes;
    else
      Converged=No;

    if(Converged==Yes)
    { if((t-tstop)>2000*dt)
      {Converged=No;
      Integrale=0;
      OldEcart=0;
      return tstop;
      }
    }
    else
    { tstop = t;
      if(t>50)
      { //break;
        return tstop;
      }
    }
  }
}
```

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

20

Commande à algorithme génétique



Description récente, sans changement, de l'approche à AG décrite plus haut.

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

21

Commande à algorithme génétique

**Exemple: Fly Algorithm (1 de 6)
(Eurobot, Heidelberg, 2008)**

Titre

The Fly Algorithm for Robot Navigation

**Rodrigo Montúfar-Chaveznava¹, Mónica Pérez-Meza²
and Ivette Caldelas³**

1

**Engineering Division, ITESM Santa Fe, Av. Carlos Lazo 100, Del. A. Obregón,
01389 México D.F., Mexico, rmontufar@itesm.mx**

2

Universidad de la Sierra Sur, Miahuatlán de Porfirio Díaz, Oaxaca, Mexico

3

**Instituto de Investigaciones Biomédicas, Universidad Nacional Autónoma de
México, México, D.F., Mexico**

HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

22

Commande à algorithme génétique

Exemple: Fly Algorithm (2 de 6) Abstract

The fly algorithm is a strategy employed for 3D reconstruction of scenes, which employs the *genetic* algorithms and stereovision principles to determine clusters of points corresponding to different objects present in scene.

The obtained reconstruction is partial, but enough to recognize obstacles in the robot space work. This 3D reconstruction strategy also allows knowing the dimensions of the detected objects. Many parameters are involved in the fly algorithm, and then it is difficult to assign the optimal values for the best performance. The algorithm considering the fly algorithm can be employed in *robot navigation*.

Keywords: The fly algorithm, *Evolutionary Strategies*, Vision stereo, Robotics.

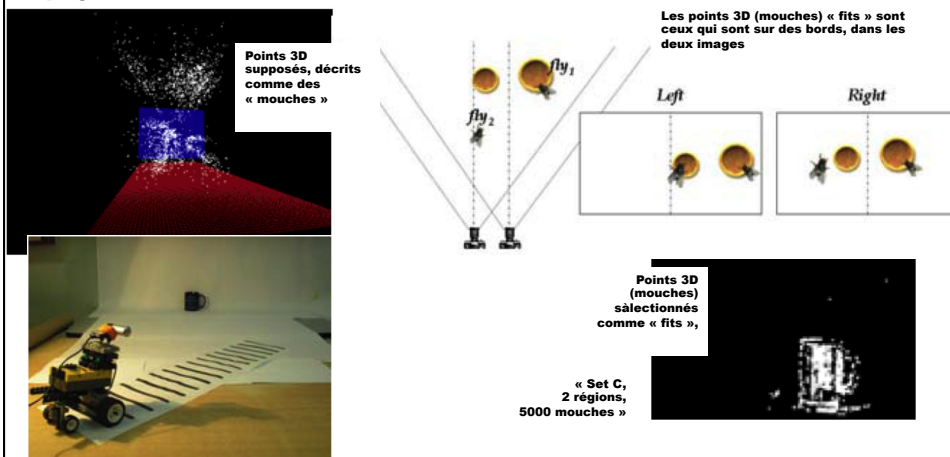
HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

23

Commande à algorithme génétique

Exemple: Fly Algorithm (3 de 6) Overview

Fig. 1. (a) The space of the fly algorithm. The scene model is represented as cloud of 3D points. (b) The projection of flies: The pixel neighborhoods for fly1 are same in stereo images; the pixel neighborhoods for fly2 are different. This observation is employed in the fitness function.

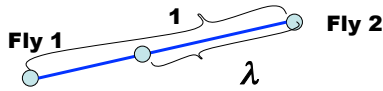


HESSO.HEIG-VD, J.-D. Dessimoz, 05.03.2017

24

Commande à algorithme génétique

Exemple: Fly Algorithm (4 de 6)



Genetic operators

Genetic algorithm uses the following operators to generate new flies:

- **Barycentric cross-over.** From two parents F_1 and F_2 , the heir F is created such as :

$$\overrightarrow{OF} = \lambda \overrightarrow{OF_1} + (1 - \lambda) \overrightarrow{OF_2}$$

where λ is a random value between $[0, 1]$.

- **Gaussian mutation.** The mutation operation adds Gaussian noise to each component of the 3D coordinate in a discriminated fly to produce a new one..

- **Immigration.** This operation extends the exploration area in search space, *creating* new individuals *randomly*; to ensure a constant exploration of the whole space of search.

Commande à algorithme génétique

Exemple: Fly Algorithm (5 de 6)

Selection

Selection classifies the flies according the fitness values, preserving the best individuals. It uses sharing rules that splits the population, forcing a part of it to explore other areas on the search space by replacement. The flies with low fitness values are replaced and new flies are generated using genetic operators.

Commande à algorithme génétique

Exemple: Fly Algorithm (6 de 6)

Fitness function

The fitness function, F , compares the *fly* projections on the left and right images. If the fly is posed on an object, the projections will have *similar pixel neighborhoods* on both images, producing a high fitness value. If the fly is “flying”, the projections will have *different pixel neighborhoods*, and the fitness value will be low. The idea is illustrated in Fig. 1b. There, fly1, which is settled on an object, has a better fitness value than fly2.

$$F = \frac{|\nabla(M_L)| \cdot |\nabla(M_R)|}{\left(\sum_{\text{colors}} \sum_{(i,j) \in N} (L(x_L + i, y_L + j) - R(x_R + i, y_R + j))^2 \right)}$$

$|\nabla(M_L)|$ and $|\nabla(M_R)|$ are the norms of the gradients of Sobel on the projections of the fly. That is intended to penalize flies when they are posed on uniform regions.

Commande à algorithme génétique

Conclusion

- La méthode de synthèse par sélection naturelle permet d'obtenir des résultats pratiques, comme démontré dans le cas du balancier motorisé commandé en position quelconque.
- Il s'agit essentiellement de générer des individus par hybridation (crossover) et mutation; puis de sélectionner les plus performants.
- De façon analogue au cas des réseaux neuronaux, ici l'aspect bio-inspiré semble attractif à certains.
- Mais l'ingrédient déterminant, c'est surtout la génération aléatoire; complétée d'une possibilité de simulation, pour sélection automatique des paramètres « optimaux ».

**AIC-Automatisation avancée, intelligence
artificielle et cognitive**

FIN