

## بناء نموذج تصميم مشترك للنظم المضمنة المستمرة

الدكتور محمد مازن المحاييري\*

الدكتور حسن البستاني\*\*

محمد ملحم\*\*\*

(تاريخ الإيداع 9 / 2 / 2014. قُبِلَ للنشر في 15 / 6 / 2014)

### □ ملخص □

تُعَدُّ النظمُ المضمنة المستمرة من البنى الأساسية في تكوين النظم المضمنة التخالفية (Heterogeneous Embedded System)، و يزداد تعقيد تصميم هذه النظم بازدياد تعقيد بنيتها. لذلك وبسبب السلوك المختلف لكل من HW و SW من حيث التنفيذ و الاتصال يتم السعي إلى تصميم هذه النظم عن طريق التوصيف على مستوى النظام (System Level) للعتاد الصلب و العتاد البرمجي من خلال وضع نموذج تصميم مشترك ( Hardware Software co-design Model). واستخدام النماذج المتوفرة يتطلب إعادة صياغتها لتصبح نموذج تصميم مشترك، بالإضافة إلى أن الحاجة إلى أكثر من نموذج في التصميم يتطلب آلية لربط هذه النماذج وذلك للحصول على نموذج متكامل. في هذا البحث نقوم بوضع نموذج تصميم مشترك للنظم المضمنة المستمرة باستخدام (PtolemyII) و هو إطار برمجي مبني على أساس لغة Java، من إنتاج (University of California, Berkely) وهو جزء من مشروع Ptolemy. ثم نطبق هذا النموذج على محركات السيارات بغرض ضبط سرعتها ضمن حلقة مفتوحة و حلقة مغلقة.

الكلمات المفتاحية: Embedded Systems, Continuous dynamic systems, Hw/SW Co-design Model, MoD, MoC, PtolemyII.

\*أستاذ مساعد - قسم هندسة الحواسيب والأتمتة- كلية الهندسة الميكانيكية والكهربائية- جامعة دمشق- سورية.

\*\*مدرس - كلية تكنولوجيا المعلومات والاتصالات- طرطوس- جامعة تشرين - سورية.

\*\*\*طالب دراسات عليا(دكتوراه) - قسم هندسة الحواسيب و الأتمتة- كلية الهندسة الميكانيكية و الكهربائية- جامعة دمشق- سورية.

## Building a Co-design Model for Continuous Embedded Systems

Dr. M. Mazin Mahairey\*  
Dr. Hasan Albustani\*\*  
Mohammed Melhem\*\*\*

(Received 9 / 2 / 2014. Accepted 15 / 6 / 2014)

### □ ABSTRACT □

Continuous-domain embedded systems are one of the essential structures of heterogeneous embedded system, and the designing complexities increase proportional to increasing the complexities of its structure. The difference behavior of hardware and software in execution and communication, motivates the usage of high-level system design approaches, such (Hardware Software co-design Model). Using available models in design, is needing to redefine this models to be (Hw/Sw co-design Model), and use multiple models in design require interconnection tools to obtain compatible model. In this research we design (co-design model) for Continuous-domain embedded systems, and applied it on car engine to control its speed as open-loop and close-loop. To achieve the model, we use Ptolemy II (is a Java-based software framework developed as part of the Ptolemy Project by University of California, Berkely).

**Key Words:** Embedded Systems, Continuous dynamic systems, Hw/SW Co-design Model MoD, MoC, Ptolemy II.

---

\*Associate Professor, Computers and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.

\*\*Assistant Professor, Faculty of Informatics technology and Communications Engineering, Tartous, Tishreen University.

\*\*\*Postgraduate Student, Computers and Automation Department, Faculty of Mechanical and Electrical Engineering, Damascus University.

## مقدمة:

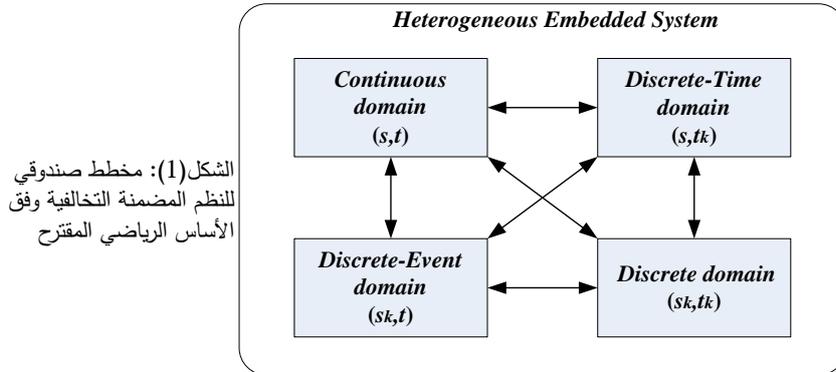
خلال العقود السابقة طُرحت تقنيات عديدة في النمذجة مثل المعادلات التفاضلية (ordinary differential equations)، حيث تقدم طريقة رياضية تحليلية لدراسة أي نظام أو كيان (entity). مثال على ذلك يستخدم قانون فوريير (Fourier's law) في دراسة توزع أو انتشار الحرارة في كيان ما. عملياً يتم النظر إلى الكيان كوسط مستمر من حيث الحرارة و زمن انتشارها في كل الاتجاهات، المتحولين المعنيين بالدراسة هما الحرارة  $Q$  و الزمن  $t$ ، يمكن أن نعبر عنه بالثنائية  $(Q, t)$ . لكن عند النظر إلى الكيان بشكل مختلف أي نهتم بدرجة الحرارة بنقاط محددة في كل اتجاه، فيظهر عامل جديد وهو سلوك الحرارة الذي يظهر بشكل متقطع، و عامل التقطع هو الزمن الذي تستغرقه الحرارة للانتشار من نقطة إلى أخرى يمكن أن يكون ثابت أو متغير و بكل الأحوال نرسم له  $t_k$  وهي الخطوة الزمنية التي تفصل النقاط بعضها عن بعض، فتصبح الثنائية السابقة  $(Q, t_k)$ . يصبح توصيف السلوك الجديد بمعادلة تفاضلية غير ممكن، بل نحتاج إلى استخدام معادلات الفرق (difference equations) ليأخذ قانون فوريير شكلاً جديداً. عند النظر لأنظمة وكيانات أكثر تعقيداً مثل نظم الاتصالات و التحكم و الأتمتة تصبح المعادلات التفاضلية و معادلات الفرق غير قادرة على توصيف سلوك هذه النظم، لذلك لابد من نظريات رياضية جديدة تعتمد على مبادئ فيزيائية ومنطقية (physical and logical principles) مختلفة. نحتاج في هذه النظم المعقدة إلى تقسيم النظام إلى كيانات منفصلة كل منها يشكل حالة منفصلة (discrete state) بحيث يتم الانتقال من حالة إلى أخرى حسب منطق عمل النظام إما بإدخال الزمن كعنصر متغير و نطلق عليه (timed version) و إما بإهمال عامل التغير الزمني (untimed version). في الشبكات الحاسوبية متحول الحالة الذي نهتم به هو حزم البيانات، فإما أن نهتم بحزم البيانات بمعزل عن زمن وصولها ومن ثم تظهر البيانات كمتحول متقطع في زمن مستمر فيكون لدينا الثنائية  $(d_k, t)$ ، وإما نهتم بحزم البيانات في أزمنة محددة فتظهر البيانات كمتحول متقطع بأزمنة متقطعة فيكون لدينا الثنائية  $(d_k, t_k)$ . بالعودة إلى تقنيات و أنواع النمذجة [1][2]، يمكن إعادة تصنيفها على أساس كيفية إتمام تمثيل الثنائية (state variables, time). فإذا اعتمدنا الثنائية (state variables, time) كأساس في التفكير لبناء النموذج، سيكون لدينا أربع حالات فقط، هي  $(s, t), (s, t_k), (s_k, t), (s_k, t_k)$ ، هذه الثنائيات تتيح تبويب النظام في أربعة مستويات:

1. نظم مستمرة (Continuous Systems):  $(s, t)$
2. نظم مقادة زمنياً (Discrete-Time Systems):  $(s, t_k)$
3. نظم مقادة بالحدث (Discrete-Event Systems):  $(s_k, t)$
4. نظم متقطعة (Discrete Systems):  $(s_k, t_k)$

باعتقاد الثنائيات السابقة كأساس رياضي في التصميم، تظهر النظم المضمنة المستمرة كبنية أساسية في النظم المضمنة التخالفية كما هو مبين في الشكل (1).

هناك نوعان من النمذجة، النوع الأول: النمذجة الحاسوبية (model of computation (MoC)، و الثاني: نمذجة تصميم (model of design (MoD) أو model-based design (MBD). النمذجة الحاسوبية (MoC) هي مجموعة من التعليمات البرمجية تستخدم في ضبط تفاعل مكونات حاسوبية على أساس قانوني معين، و الغاية هي المحاكاة لتحليل نتائج عمل نظام ما حاسوبياً، ومن ثم مقارنة النتائج التي يقدمها النموذج مع الواقع. والنمذجة التصميمية (MoD) تؤكد النمذجة الرياضية (mathematical modeling) في تصميم و تحليل النظم، حيث يقرن

العملية الفيزيائية بحسابات مضمنة (*embedded computations*) لإنتاج خرج مناسب للدخل ضمن الزمن الحقيقي، لذلك يبرز الزمن كمكون أساسي في العلاقات الرياضية التي تشرح نموذج التصميم [3][4][5].



تمت دراسة النظم المستمرة باستخدام تقنيات مختلفة وفيما يأتي نعرض بعض الأعمال التي استخدمت فيها النمذجة بنوعها في دراسة النظم المستمرة. حيث تظهر الدراسة [6] فاعلية النمذجة الرياضي للنظم الفيزيائية الديناميكية كنظم مضمنة مستمرة، حيث تتطلب التطبيقات المعقدة مثل النظم (*Cyber-physical systems CPS*) عدة أدوات نمذجة و لغات برمجة و بنى صلبة خاصة، حيث لا يوجد أي نموذج لتغطيتها تغطية كاملة، و تظهر الدراسة أهمية استخدام النمذجة التصميمية في شرح سلوك النظم المستمرة، وتقدم ذلك من خلال دراسة حالتين وهما (*Cal*) (*Climber*) و (*Tunniling Ball device*)، حيث تستخدم (*LabVIEW*) كبيئة برمجة و (*PtolemyII*) كبيئة نمذجة. البحث [7] يركز على الصعوبات التي تواجه نمذجة (*CPS*) كنظم ديناميكية مثل التباين الجوهري في البنية، و التوافق، و الحساسية للزمن، من خلال دراسة (*aircraft vehicle management systems (VMS)*) حيث يوجد العديد من خزانات الوقود و الصمامات و المضخات و الحساسات و المجسات و المفاتيح، ويعمل البحث على دراسة توزيع الوقود من الخزانات إلى المحرك مع المحافظة على توازن الطائرة، عن طريق نمذجة الحاسوبية (*MOG*) كنظام مستمر باستخدام بيئة النمذجة (*PtolemyII*). البحث [8] يطرح التحديات التي تواجه التصميم و النمذجة المشتركة التي تظهر في النظم المضمنة المعقدة التي تتفاعل مع الوسط الفيزيائي المحيط (*CPS*) مثل السيارات الحديثة، من هذه التحديات الحاجة إلى خوارزميات تحكم معقدة للتحكم بالأجزاء الدقيقة، بالإضافة إلى الشروط الواجب توفرها في لغة النمذجة للحصول على نموذج يحاكي النظام الحقيقي وخاصة في السلوك الزمني (*timing behavior*) عند جمع الأجزاء المختلفة للنظام الديناميكي، حيث يقدم لغة نمذجة جديدة يطلق عليها *MODEL and analyze (Modelyze)* قادرة على بناء نماذج مختلفة مضمنة على شكل *domain-specific languages (DSLs)*. البحث [9] يدرس الربط بين التصميم البرمجي المضمن (*software design*) والتصميم التحكمي (*control design*) حيث يركز على المفاهيم الوظيفية و المفاهيم الزمنية مثل (*periods, sampling-to-actuation (StA) delay,*) (*and Jitter*) في ردم الهوة بين المجالين المذكورين، و يستخدم تقنية النمذجة المشتركة لربط المفاهيم التالية (*PID, optimal, adaptive,...; ODEs, continuous time control theory, sampled data...*) من جهة التصميم التحكمي، و (*response time, memory footprint, WCET, inter-process communication,*) (*scheduling theory, model checking...*) من جهة التصميم البرمجي، و تعتمد المقاربة التي يقدمها البحث على تعريف الحالات الآتية (*separation of concerns, optimization and synthesis of timing*) (*parameters, run-time adaptation*). البحث [10] يقدم إطار عمل جديد للنموذج (*One-1D CFD*)

*Dimensional Computational Fluid Dynamics* الذي يستخدم في نمذجة و محاكاة سلوك تدفق الوقود في محركات الديزل، المقاربة التي يقدمها البحث هي حل النموذج (1D CFD) في الزمن الحقيقي باستخدام FPGA. في هذا البحث نعتد على الثنائية  $(s, t)$  كأساس رياضي في بناء نموذج تصميم جديد للنظم الديناميكية المستمرة. نستخدم فيه المعادلات التفاضلية (ODEs) كتوابع غرضية (*object Functions*) لأحتواء السلوك المستمر للنظام في كل لحظة زمنية. بالإضافة إلى احتواء كافة مكونات النظام من خلال تفاعل الدخل مع التوابع الغرضية للحصول على الخرج المناسب، ومن ثم تشكيل خرج النموذج على شكل مصفوفة نطلق عليها مصفوفة النظام المستمر.

### أهمية البحث وأهدافه:

تصميم نموذج تصميم مشترك (Hw/Sw co-design Model) اعتماداً على متحولي الحالة و الزمن للنظم المضمنة المستمرة، و يستخدم في توصيف النظم الديناميكية المستمرة كحلقة مفتوحة وحلقة مغلقة.

### منهجية البحث:

- يتم تصميم النموذج بناءً على ثلاثة ثوابت:
- أن يكون قادراً على توصيف النظم المضمنة المستمرة.
- أن يكون نموذج تصميم مشترك (Hw/Sw co-design Model).
- في النظم الحاسوبية بأنواعها كافة يتم ترأسل البيانات بين مكوناته على شكل إشارات (signals)، و الإشارة هي تسلسل من الأحداث  $S = \langle e_1, e_2, e_3, e_4, \dots \rangle$  بأزمنة متتالية تحدد بساعة محلية ضمن كل مكون.

### 1-التصميم الصوري للنموذج

التعريف الصوري للنموذج (*Basic Form*):

*Continuous Domain*

$M(X, I, O, F)$

$X$  : Internal state set

$I$  : Input set

$O$  : output set

$F_{CD} = \{f_{cdState}, f_{cdOut}, f_{cdLclk}, f_{cdT}, f_{cdR}\}$

توابع النموذج الأساسية (*Object Functions*):

States Functions:  $f_{cdState} : X \times I \rightarrow X$

Output Functions:  $f_{cdOut} : X \times I \rightarrow O$

Local Clock:  $f_{cdLclk}$

Transmitter Function:  $f_{cdT} : O \rightarrow Other$

Receive Function:  $f_{cdR} : Other \rightarrow I$

خرج النموذج سيكون مجموعة من التوابع الغرضية التي توصف سلوك وبنية النظام على الشكل الآتي:

$f_{cdState} = [f_{cdState1}, f_{cdState2}, f_{cdState3}, \dots]$

$$\begin{aligned}
f_{cdOut} &= [f_{cdOut1}, f_{cdOut2}, f_{cdOut3}, \dots] \\
f_{cdLcik} &= [f_{cdLcik1}, f_{cdLcik2}, f_{cdLcik3}, \dots] \\
f_{cdT} &= [f_{cdT1}, f_{cdT2}, f_{cdT3}, \dots] \\
f_{cdR} &= [f_{cdR1}, f_{cdR2}, f_{cdR3}, \dots]
\end{aligned}$$

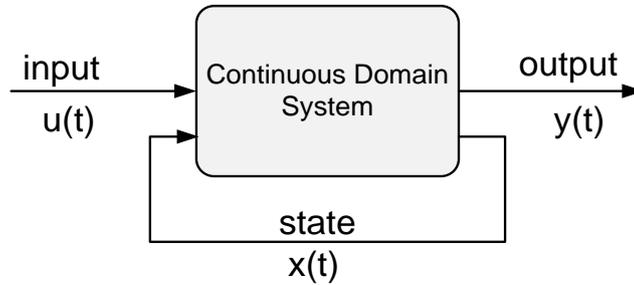
نشكلها على شكل مصفوفة، نطلق عليها مصفوفة النظام على الشكل الآتي:

$$S_{cd} = \begin{pmatrix} f_{cd11} & f_{cd12} & \dots & f_{cd1m} \\ f_{cd21} & f_{cd22} & \dots & f_{cd2m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{cdn1} & f_{cdn2} & \dots & f_{cdnm} \end{pmatrix}$$

$(n, m)$  هي أبعاد المصفوفة. حيث  $n$  عدد أنواع التوابع و  $m$  عدد التوابع من كل نوع.

## 2- تعريف التوابع الغرضية للنموذج

يبين الشكل (2) رسماً تخطيطياً للنظام ديناميكياً مستمراً.



الشكل(2): شكل توضيحي للنظام

نفترض أن للنظام  $r$  مجموعة دخل نعبر عنها بالشعاع  $U(t) = [u_1(t), u_2(t), \dots, u_r(t)]$ ، وبالوقت نفسه تمثل مجموع الدخل  $I$  في النموذج. كذلك بالنسبة للخروج حيث نفترض وجود  $m$  خرج تمثل مجموعة الخرج  $O$  و يعبر عنها بالشعاع  $Y(t) = [y_1(t), y_2(t), \dots, y_m(t)]$ . و يفرض أن للنظام  $n$  حالة ممثلة بالشعاع  $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ . بعد أن تم تحديد مجموعات الدخل، ومجموعة الخرج، و مجموعة الحالات للنظام؛ نقوم بتعريف التوابع الغرضية في النموذج [11].

### • توابع الحالة $f_{cdState}$ :

$$\text{States Functions: } f_{cdState} : X \times I \rightarrow X$$

في هذا النوع من النظم، تحدد توابع الحالة بمعادلات تفاضلية من الدرجة الأولى.

$$\dot{x}(t) = f_{cdState}(x(t), u(t), t)$$

عندما يُعبر عن نظام بمعادلة تفاضلية من الدرجة  $n$ ، يتم تحويل هذه المعادلة إلى  $n$  معادلة تفاضلية من الدرجة الأولى باستخدام متحولات الحالة  $state\ variables$ . وعدد المعادلات يعبر عن عدد الحالات في النظام و عن درجة النظام. بالتالي فضاء الحالة يتكون من  $n$  متحول حالة. تحدد توابع الحالة على الشكل الآتي:

$$\dot{x}_1(t) = f_{CdState1}(x(t), u(t), t)$$

$$\dot{x}_2(t) = f_{CdState2}(x(t), u(t), t)$$

⋮

$$\dot{x}_n(t) = f_{CdStaten}(x(t), u(t), t)$$

تُمثل توابع الحالة بالشعاع  $f_{CdState} = [f_{CdState1}, f_{CdState2}, \dots, f_{CdStaten}]$

في أغلب الأحيان لا يمكن حل هذه المعادلات غير الخطية بطريقة تحليلية عادية، لذلك يتم استخدام طرق التحليل العددي. وهذه الطرق تعتمد على سلسلة تايلور بدرجات تقريب مختلفة. العلاقة (1) تمثل سلسلة تايلور [12][13].

$$x(t+h) = \sum_{i=0}^{\infty} \frac{x^{(i)}(t)}{i!} h^i \quad (1)$$

حيث  $h$  تمثل خطوة التكامل.

نستخدم طريقة رونجا كوتا ( $Runge-Kutta2/3$ ) التي تعتمد على التقريب من الدرجة الرابعة لمنشور تايلور

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{1}{2}h^2\ddot{x}(t) + \frac{1}{6}h^3x^{(3)}(t) + \frac{1}{24}h^4x^{(4)}(t)$$

وطريقة رونجا كوتا من الطرق الشائعة التي تستخدم في حل المعادلات الفاصلية عددياً، وتقدم الدقة الفضلى في الحل من بين الطرق العددية المستخدمة على مستوى النظم الديناميكية المستمرة [11][14][15]. وتتطلب هذه الطريقة معرفة اشتقاق التابع أربع مرات، لذلك تستخدم خوارزمية التقطيع ( $discretized algorithm$ ) المكافئة لتايلور من الدرجة الرابعة فتكون النتيجة الآتية:

$$k_1 = f(t_k, x(k)),$$

$$k_2 = f\left(t_k + \frac{1}{2}h, x(k) + \frac{1}{2}hk_1\right),$$

$$k_3 = f\left(t_k + \frac{1}{2}h, x(k) + \frac{1}{2}hk_2\right),$$

$$k_4 = f(t_k + h, x(k) + hk_3),$$

$$t_{k+1} = t_k + h,$$

$$x(k+1) = x(k) + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4).$$

نطبق هذه الطريقة على توابع الحالة  $\dot{x}_i(t) = f_{CdStatei}(x(t), u(t), t)$  نحصل على النتيجة الآتية:

من أجل  $1 \leq i \leq n$

$$k_{1x_i} = f_i([x_1(t_k), x_2(t_k), \dots, x_n(t_k)], u(t_k), t_k)$$

$$k_{2x_i} = f_i\left(\left[x_1(t_k) + \frac{h}{2}k_{1x_1}, x_2(t_k) + \frac{h}{2}k_{1x_2}, \dots, x_n(t_k) + \frac{h}{2}k_{1x_n}\right], u(t_k + h/2), t_k + h/2\right)$$

$$k_{3x_i} = f_i\left(\left[x_1(t_k) + \frac{h}{2}k_{2x_1}, x_2(t_k) + \frac{h}{2}k_{2x_2}, \dots, x_n(t_k) + \frac{h}{2}k_{2x_n}\right], u(t_k + h/2), t_k + h/2\right)$$

$$k_{4x_i} = f_i\left(\left[x_1(t_k) + \frac{h}{2}k_{3x_1}, x_2(t_k) + \frac{h}{2}k_{3x_2}, \dots, x_n(t_k) + \frac{h}{2}k_{3x_n}\right], u(t_k + h/2), t_k + h/2\right)$$

$$f_{CdStatei} \equiv x_i(t_{k+1}) = x_i(t_k) + \frac{h}{6}\left(k_{1x_i} + \frac{1}{2}k_{2x_i} + \frac{1}{2}k_{3x_i} + k_{4x_i}\right) \quad (2)$$

• توابع الخرج  $f_{cdout}$  :

Output Functions:  $f_{cdout} : X \times I \rightarrow O$

في هذا النوع من النظم وخاصة في النظم ذات الدرجات المنخفضة تنتقل قيم متحولات الحالة إلى الخرج مباشرة، وتتحدد مجموعة الخرج بحسب عدد متحولات الحالة التي سيتم نقلها إلى الخرج.

$$f_{cdout} = y_j = x_i, \quad 1 \leq j \leq m, \quad 1 \leq i \leq n$$

و يمكن أن يكون أحد عناصر الدخل دونما إجراء أي تعديل.

$$f_{cdout} = y_j = u_i, \quad 1 \leq j \leq m, \quad 1 \leq i \leq r$$

أو أن يتشكل يدوياً عندما يكون تابعاً للدخل و متحولات الحالة معاً.

$$f_{cdout} = f(u, x)$$

تابع الخرج بشكله العام:

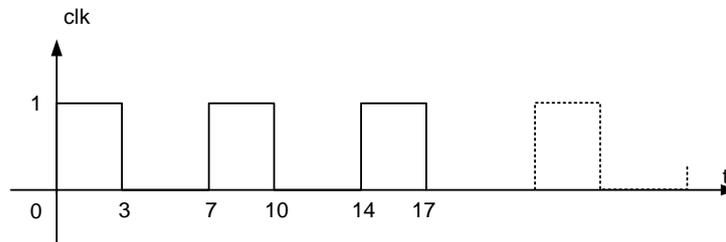
$$(3) f_{cdout} = \begin{cases} y_j = x_i, & 1 \leq j \leq m, \quad 1 \leq i \leq n \\ y_j = u_i, & 1 \leq j \leq m, \quad 1 \leq i \leq r \\ y_j = f(u, x), & 1 \leq j \leq m \end{cases}$$

• تابع التوافق المحلي ( $f_{cdclk}$  Local Clock)

يتم النظر إلى النظام زمنياً، على شكل مجموعة من الساعات المحلية التي تقاد من قبل ساعة رئيسية. تعمل فيه الساعة المحلية كجزء من المنظومة على ضبط ترتيب خروج الأحداث من الكيان عن طريق طبع الحدث بزمن خروجه. وتقوم الساعة الرئيسية ( $global\ clock$ ) أو ( $master\ clock$ ) على قيادة الساعات المحلية وتنظيم عملها. ولتمثيل إشارة الساعة رياضياً يتم استخدام التوابع المركبة مثل تابع ( $signum\ function$ ) و تابع الجزء الصحيح ( $INT$ ) [12]. بشكل عام إذا بدأت الإشارة عند الزمن  $t = t_0$ ، و بدور  $T$ ، وذات قيمة مرتفعة بنسبة  $p$  من كل دورة، حيث  $p$  تدعى دورة التشغيل ( $duty\ cycle$ ). بفرض أن الإشارة تتناوب بين القيمتين  $A_{min}$  و القيمة  $A_{max}$ ، فتكون الصيغة على الشكل الآتي :

$$x(t) = (A_{max} - A_{min}) \left( \left[ t > \left( t_0 + T * INT \left( \frac{t}{T} \right) \right) \right] \text{ and } \left[ t < \left( p * T + T * INT \left( \frac{t}{T} \right) \right) \right] \right) + A_{min}$$

الشكل (3) يمثل نبضات ساعة عند القيم  $(A_{max} = 1, A_{min} = 0, p \approx 43\%, T = 7, t_0 = 0)$ .



الشكل (3): نموذج لإشارة ساعة بتردد  $f = 1/7$

بعد الأخذ بالحسبان أن مطال نبضة الساعة هو "1" أي  $(A_{max} = 1, A_{min} = 0)$  فتصبح صيغة تابع الساعة المحلية كالآتي:

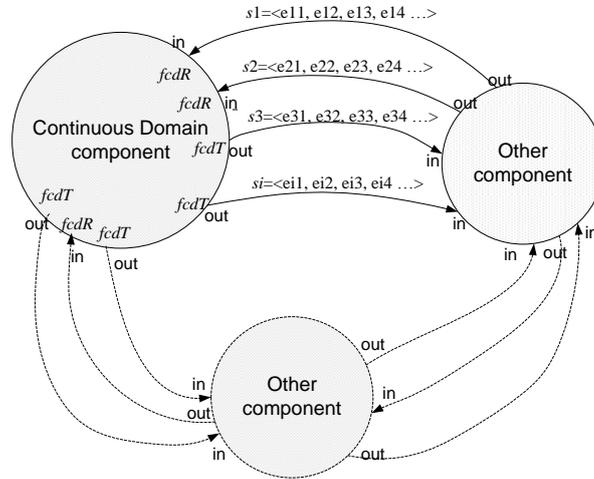
$$f_{cdtik} = \left( \left[ t > \left( t_0 + T * INT \left( \frac{t}{T} \right) \right) \right] \text{ and } \left[ t < \left( p * T + T * INT \left( \frac{t}{T} \right) \right) \right] \right) \quad (4)$$

• توابع الإرسال و الإستقبال (Transmit and Receive Functions):

Transmitter Function:  $f_{cdT}: O \rightarrow Other$

Receive Function:  $f_{cdR}: Other \rightarrow I$

إنّ توابع الإرسال و الاستقبال في النظم المستمرة هي المسؤولة عن التخاطب مع المكونات الأخرى في الوسط المحيط. وفي هذه النظم تعمل عملاً مركباً، فتوابع الاستقبال تحول إشارات الدخل من تسلسل أحداث إلى إشارات مستمرة و العكس بالعكس تقوم توابع الإرسال بتحويل إشارات الخرج المستمرة إلى سلسلة من الأحداث و من ثم ترسلها إلى الوسط المحيط.



الشكل (4): رسم يوضح طريقة التخاطب ضمن النظام المضمن التخالفي

الشكل (4) يبين تبادل الإشارات بين مكون من نوع  $ContinousDomain(CD)$  مع مكونات أخرى مختلفة باستخدام التوابع  $(f_{cdT}, f_{cdR})$ . كما ذكرنا سابقاً تتواصل مكونات النظام بعضها ببعض بواسطة الإشارات ( $signals$ )، و هذه الإشارات هي سلسلة من الأحداث. و تصنف الأحداث عند التعامل معها كنموذج حاسوبي، إلى ثلاثة أنواع: النوع الأول ( $Untimed Events$ ) و غالباً يستخدم في تدفق المعطيات ( $Data Flow$ )، النوع الثاني ( $Synchronous Events$ ) وفيه يظهر ما يسمى الحدث الغائب ( $absent event$ ) ويرمز له  $\{ \square \}$  ويستخدم هذا المفهوم للأحداث في نظم الإتصالات ( $communication systems$ )، أما النوع الثالث فهو ( $Timed Events$ ) ويستخدم في نظم التحكم ( $control systems$ ) [5] [16].

واستخدام الساعة المحلية كأحد التوابع الأساسية لنموذج تصميم تجعل من الممكن صياغة الحدث بطريقة تشمل الحالات السابقة، والصيغة التالية تؤمن التعامل مع الحدث في المستويات جميعها.

Event form

$$e(v, \tau)$$

$v$ : the value of event,  $v \in \mathbb{R}$

$\tau$ : time tags,  $\tau \in \mathbb{R}_+$

قيمة الحدث ( $v$ ) هي قيمة الإشارة عند الزمن ( $\tau$ )، حيث تقوم الساعة المحلية (*local clock*) بطبع كل حدث بزمن خروجه ( $\tau$ ) من الكيان المستمر.

تتابع الإرسال: (*Transmitter Function:  $f_{cd\tau}: O \rightarrow Other$* )

تعمل هذه التتابع على تحويل الإشارة المستمرة إلى أحداث عن طريق تقطيع هذه الإشارة بفواصل زمنية محددة، ومن الناحية النظرية لتقطيع إشارة يتم ضرب هذه الإشارة بقطار من النبضات الواحدية و تمثل بالعلاقة الآتية:

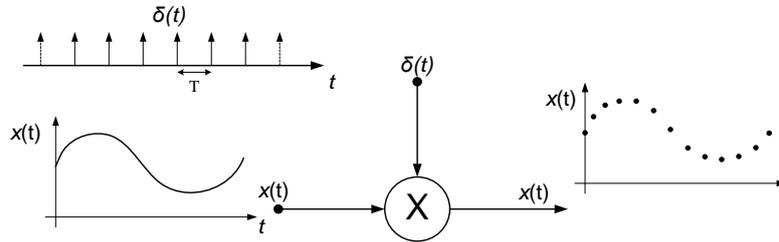
$$s(t) = \sum_{k=-\infty}^{\infty} \delta(t + kT_s)$$

حيث  $f_s = 1/T_s$  هو تردد التقطيع، و  $t_k = t + kT_s$

فتكون الإشارة المقطعة

$$x(t_k) = x(t)s(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t + kT_s)$$

الشكل (5) يوضح عملية التقطيع.



الشكل(5): شكل يوضح عملية التقطيع

أما من الناحية التطبيقية فيتم تحويل الإشارة إلى سلسلة من الأحداث عن طريق جداء الإشارة بقطار النبضات الناتج عن الساعة المحلية.

$$x(t_k) = x(t)f_{cd\tau k}$$

فحصل على أحداث متسلسلة بفواصل زمنية بتردد الساعة المحلية  $f = 1/T$ . فتصبح الإشارة على الشكل

الآتي:

$$\begin{aligned} x(t_k) &= ((x(t_1), x(t_2), x(t_3), \dots)) = ((e_1, e_2, e_3, e_4, \dots)) \\ &= ((v_1, \tau_1), (v_2, \tau_2), (v_3, \tau_3), (v_4, \tau_4), \dots) \end{aligned}$$

بالتالي يحدد تابع الإرسال بالعلاقة الآتية:

$$f_{cd\tau}: O \rightarrow Other$$

$$f_{cd\tau}(x(t)) = x(t)f_{cd\tau k} = ((v_1, \tau_1), (v_2, \tau_2), (v_3, \tau_3), (v_4, \tau_4), \dots) \quad (5)$$

$$\tau_k = t_0 + kT, \quad v_k = x(\tau_k)$$

توابع الإستقبال (Receive Function:  $f_{CAR}: Other \rightarrow I$ ):

تستخدم عملية التقدير الاستقرائي (*extrapolation*) في استعادة إشارة من أحداث منقطعة. ومثال على ذلك الماسك من الدرجة صفر (*zero-order hold (ZOH)*) من خلال نقطة واحدة يتم استقراء الباقي. و للحصول على ماسك من درجات أعلى بغرض الدقة الفضلى نستخدم كثيرات حدود لاغرانج (*Lagrangian extrapolation*) (*polynomial*) [11][12]، حيث يتم الإحتفاظ بعدد عينات سابقة  $l + 1$ . بفرض أن قيم الإشارة المرسل على شكل أحداث، وهي مبينة في الجدول (1).

جدول(1): قيم إشارة مرسل على شكل أحداث

$k$	$t_k$	$x_k$
$\vdots$	$\vdots$	$\vdots$
$k - l$	$t_{k-l}$	$x_{k-l}$
$\vdots$	$\vdots$	$\vdots$
$k - 2$	$t_{k-2}$	$x_{k-2}$
$k - 1$	$t_{k-1}$	$x_{k-1}$
$k$	$t_k$	$x_k$

فيكون كثير الحدود التقريبي  $x(t)$ :

$$x(t) = \sum_{i=0}^l \left( x_i \prod_{j=0, j \neq i}^l \frac{t - t_j}{t_i - t_j} \right)$$

إن الصيغة السابقة هي صيغة لاغرانج لعملية التقدير الاستقرائي وهي عامة جداً، و بما أن أخذ العينات يتم على مسافات زمنية ثابتة ومنتظمة،  $T_s = t_k - t_{k-1} = 1/f_s$ . تصبح صيغة لاغرانج:

$$x(t) = f_s^l \sum_{i=0}^l \frac{(-1)^i}{i! (l-i)!} \left[ x_{k-i} \prod_{j=0, j \neq i}^l (t - t_{k-j}) \right]$$

الحالة الخاصة من صيغة لاغرانج هي الماسك من الدرجة صفر (*zero-order hold (ZOH)*) عندما يكون

$l = 0$  من ثم تصبح الصيغة:

$$x(t) = f_s^0 \sum_{i=0}^0 \frac{(-1)^i}{i! (0-i)!} \left[ x_{k-i} \prod_{j=0, j \neq i}^0 (t - t_{k-j}) \right] = x_k$$

ونحصل على الماسك من الدرجة الأولى (*First-order hold (FOH)*) عندما يكون  $l = 1$ .

$$x(t) = f_s^1 \sum_{i=0}^1 \frac{(-1)^i}{i! (1-i)!} \left[ x_{k-i} \prod_{j=0, j \neq i}^1 (t - t_{k-j}) \right] = f_s [x_k (t - t_{k-1}) - x_{k-1} (t - t_k)]$$

الدقة المطلوبة في استعادة الإشارة هي التي تحدد درجة الاستعادة.

والإشارة المستقبلية في النظم المستمرة تكون على الشكل الآتي:

$$x(t_k) = ((x(t_1), x(t_2), x(t_3), \dots)) = ((e_1, e_2, e_3, e_4, \dots)) \\ = ((v_1, \tau_1), (v_2, \tau_2), (v_3, \tau_3), (v_4, \tau_4), \dots)$$

$f_{cdR}$ : Other  $\rightarrow I$

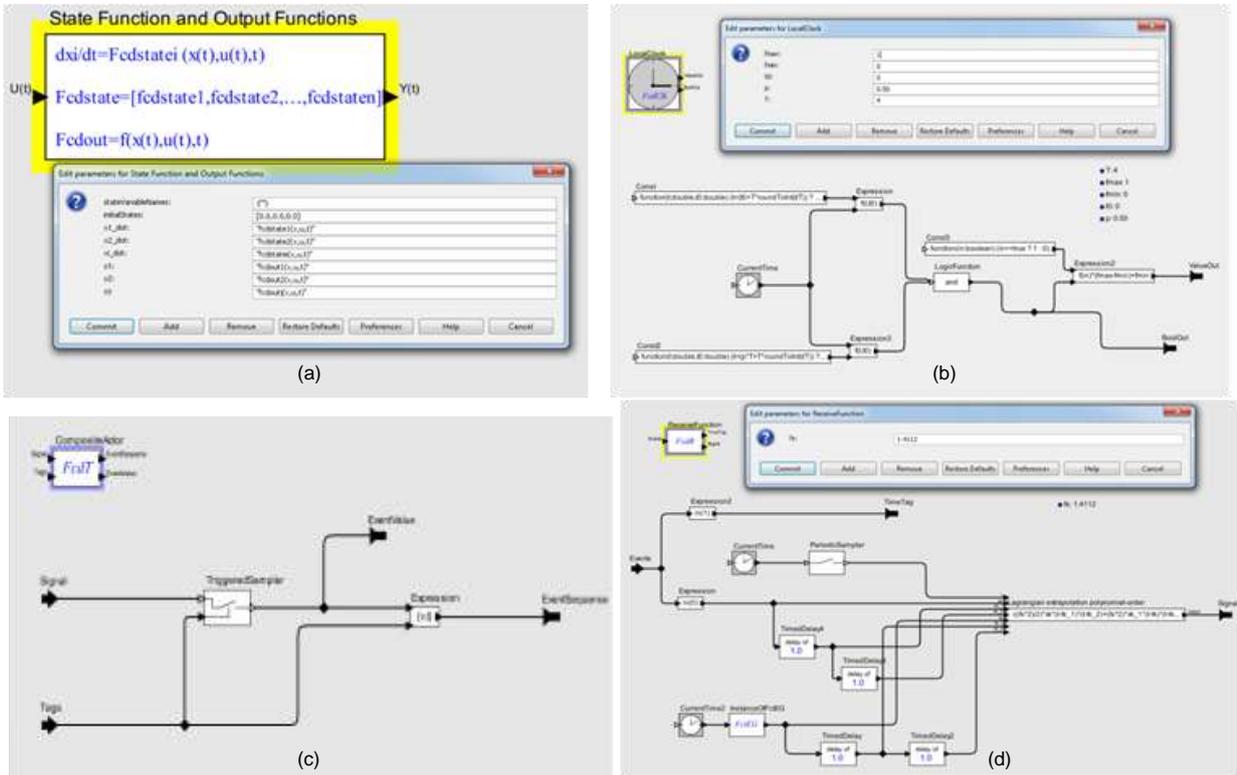
$$f_{cdR}(x(t_k)) = f_{cdR}(((v_1, \tau_1), (v_2, \tau_2), (v_3, \tau_3), (v_4, \tau_4), \dots))$$

$$= f_s^l \sum_{i=0}^l \frac{(-1)^i}{i!(l-i)!} \left[ x_{k-i} \prod_{j=0, j \neq i}^l (t - t_{k-j}) \right] \quad (6)$$

$$f_s = \frac{1}{n} f_d, \quad f_d = \frac{1}{T} \text{ frequency of local clock } f_{cacik}$$

### 3-تحقيق النموذج

لتحقيق النموذج نستخدم (PtolemyII) و هو إطار برمجي مبني على أساس لغة Java، من إنتاج (University of California, Berkely) وهو جزء من مشروع Ptolemy [17]. حيث يدرس النمذجة و المحاكاة و التصميم المتوافق، و يدرس الزمن الحقيقي و النظم المضمنة. و يستخدم Ptolemy لتصميم نماذج حاسوبية (models of computation) من خلال ضبط التفاعل بين المكونات.



الشكل(6):تحقيق توابع النموذج باستخدام PtolemyII

الشكل (6-a) يمثل كتلة توابع الحالة و توابع الخرج وفق المعادلات (2)(3). الشكل(6-b) يمثل كتلة الساعة المحلية وفق العلاقة (4) ويتم التحكم بمواصفات الساعة وفق البارامترات المرافقة. الشكل(6,c) كتلة تمثل تابع الإرسال حيث يحول الإشارة المستمرة إلى سلسلة من الأحداث على شكل تسلسل رتيب من الثنائيات  $e(v, \tau)$  العلاقة (5).

والشكل (6,d) يمثل كتلة تابع الاستقبال حيث يستعيد الإشارة من تسلسل الأحداث، تم استخدام توابع لاغرانج من المرتبة الثانية وفق العلاقة (6)، علماً أنه يمكن استخدام هذه التوابع من الدرجة صفر أو من الدرجة الأولى بحسب مقتضيات الدقة المطلوبة.

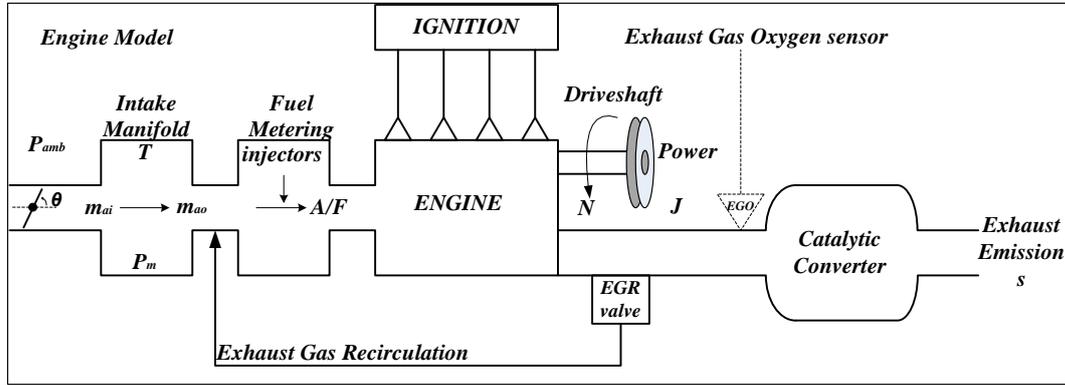
### تطبيق النموذج

من أجل تطبيق النموذج نقوم بالخطوات الآتية:

- نحدد كل من مجموعة الدخل و الخرج و ذلك بحسب عدد إشارات الدخل و الخرج.
- نوصف النظام المستمر باستخدام معادلات تفاضلية من الدرجة الأولى، حيث كل معادلة تفاضلية تشكل تابع حالة واحد.
- نربط إشارات الخرج بمتحولات الحالة و إشارات الدخل بواسطة توابع الخرج بحسب متطلبات التطبيق.
- يتفاعل الكيان المستمر مع باقي الكيانات وفق الرؤية التي نقدمها على شكل سلسلة من الأحداث. توابع الإرسال تقطع وترسل إشارة الخرج على شكل تسلسل أحداث مطبوعة بزمناً خروجها من الكيان المستمر، وتوابع الاستقبال تستعيد إشارات الدخل المقطعة الواردة من كيانات أخرى و تحولها إلى إشارة مستمرة.
- لاستخدام ساعة التوقيت المحلي نقوم بتحديد تردد الساعة أولاً بالشكل الذي يتلائم مع التطبيق و مع عمل توابع الإرسال و الاستقبال.

### 1-تطبيق النموذج على المحرك كحلقة مفتوحة

تُعدُّ وسائط النقل أحد أهم تطبيقات النظم المضمنة التخالفية، و باعتبار أن محرك السيارة من النظم الديناميكية المستمرة التي تخضع لمجموعة من المعادلات التفاضلية التجريبية فإننا سنقوم بتطبيق النموذج على محرك السيارة بغرض ضبط سرعته. الشكل (7) يمثل نموذج لمحرك سيارة حيث يتم ضبط السرعة من خلال فتحة خانق الهواء وعزم التحميل [18][19][20].



الشكل (7) نموذج لمحرك سيارة

- صمام دخل الهواء (*Throttle*): يتحكم الصمام بتدفق الهواء إلى مجمع الهواء عن طريق ضبط زاوية الفتح، و تحدد كتلة الهواء المتدفق مع إهمال كتلة الغازات المعاد تدويرها بالمعادلة التفاضلية الآتية [20]:

$$\dot{m}_{ai} = f(\theta)g(P_m)$$

$$f(\theta) = 2.821 - 0.0523\theta + 0.10299\theta^2 - 0.00063\theta^3$$

$$g(P_m) = \begin{cases} 1 & , P_m \leq \frac{P_{amb}}{2} \\ \frac{2}{P_{amb}} \sqrt{P_m P_{amb} - P_m^2} & , \frac{P_{amb}}{2} \leq P_m \leq P_{amb} \\ -\frac{2}{P_m} \sqrt{P_m P_{amb} - P_{amb}^2} & , P_{amb} \leq P_m \leq 2P_{amb} \\ -1 & , P_m \geq 2P_{amb} \end{cases}$$

• ضغط مسرب الهواء (*Intake manifold*): يحدد بالمعادلة الآتية باستخدام قانون الغازات العامة (*Boyle's law*).

$$\dot{P}_m = \frac{RT}{V_m} (\dot{m}_{ai} - \dot{m}_{ao})$$

• كتلة هواء الخرج (*Intake mass flow rate*): وهي كتلة الهواء التي تدخل إلى الأسطوانات و اللازمة للاحتراق و تحدد بالمعادلة التفاضلية

$$\dot{m}_{ao} = -0.366 + 0.08979NP_m - 0.0337NP_m^2 + 0.0001N^2P_m$$

• توليد عزم التدوير و التسارع (*Torque Generation and Acceleration*): من خلال العلاقة التي تحدد

عزم التدوير نكتب المعادلة التفاضلية التي تحدد السرعة

$$Torque_{eng} = -181.3 + 379.36m_a + 21.91(A/F) - 0.85(A/F)^2 + 0.26\sigma - 0.0028\sigma^2 \\ + 0.027N - 0.000107N^2 + 0.00048N\sigma + 2.55\sigma m_a - 0.05\sigma^2 m_a$$

$$J\dot{N} = Torque_{eng} - Torque_{load}$$

$\theta$  (deg): زاوية الخانق،  $P_m$  (bar): ضغط المجمع،  $P_{amb}$  (bar): الضغط الخارجي،  $\dot{m}_{ai}$  (g/s): معدل تدفق كتلة الهواء إلى المجمع،  $R$ : ثابت الغازات،  $T$  (K): درجة الحرارة،  $V_m$  (m<sup>3</sup>): حجم المجمع،  $\dot{m}_{ao}$  (g/s): معدل تدفق كتلة الهواء من المجمع،  $\dot{P}_m$  (bar/s): معدل تغير الضغط في المجمع،  $N$  (rad/s): سرعة المحرك،  $m_a$  (g): كتلة الهواء في الأسطوانة،  $A/F$ : نسبة الهواء إلى الوقود،  $\sigma$ : زاوية القدح (degrees before top-dead-center)،  $TDC$ ،  $Torque_{eng}$  (Nm): العزم الناتج عن المحرك،  $J$  (kg/m<sup>2</sup>): عزم العطالة الدوراني للمحرك.

لدينا  $(\frac{RT}{V_m} = 0.41328)$ ،  $(\sigma = 8.0 \text{ deg}, \frac{A}{F} = 14/7)$  [18] [20].

مجموعة الدخل:

$$I = \{\theta, P_{amb}, Torque_{load}\}$$

مجموعة الخرج:

$$O = \{m_{ao}, P_m, N\}$$

تتابع الحالة للمحرك:

$$f_{CdState} = [ \dot{P}_m = f_{Cdstate1}, \dot{m}_{ao} = f_{Cdstate2}, \dot{N} = f_{Cdstate3} ]$$

تتابع الخرج للمحرك:

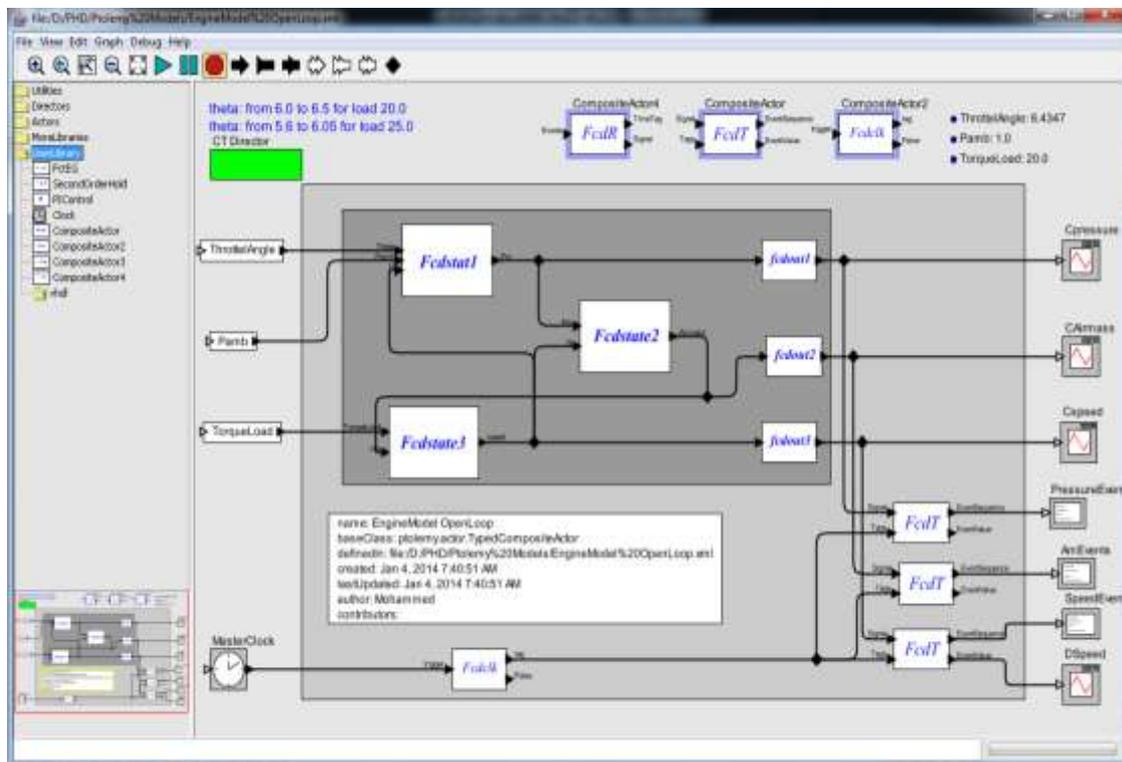
$$f_{CdOut} = [ f_{Cdout1} = P_m, f_{Cdout2} = m_{ao}, f_{Cdout3} = \frac{N*30}{\pi} ]$$

تتابع الإرسال:

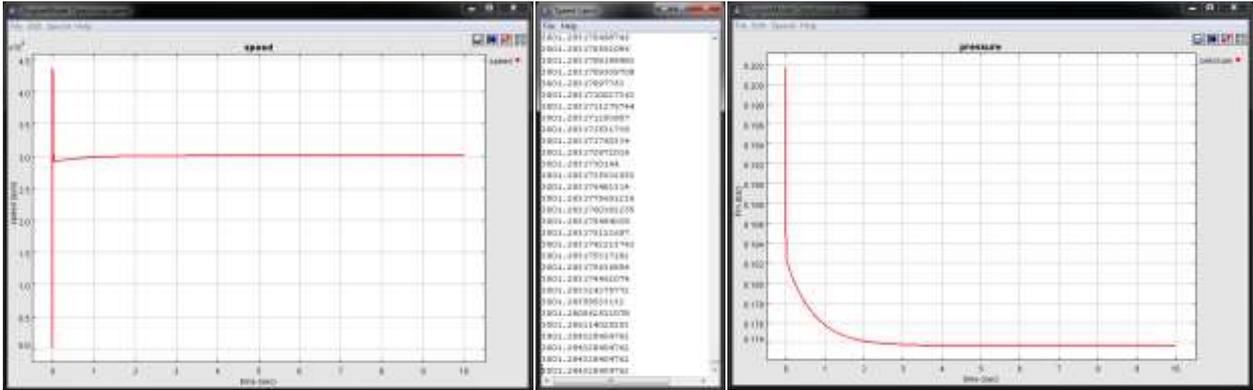
$$f_{cdT} = [f_{dcT1} = P_m f_{cdclik}, f_{dcT2} = m_{a0} f_{cdclik}, f_{dcT3} = \frac{N+30}{\pi} f_{cdclik}]$$

تطبيق النموذج بشكل كامل على محرك السيارة كحلقة مفتوحة مبين في الشكل (8).

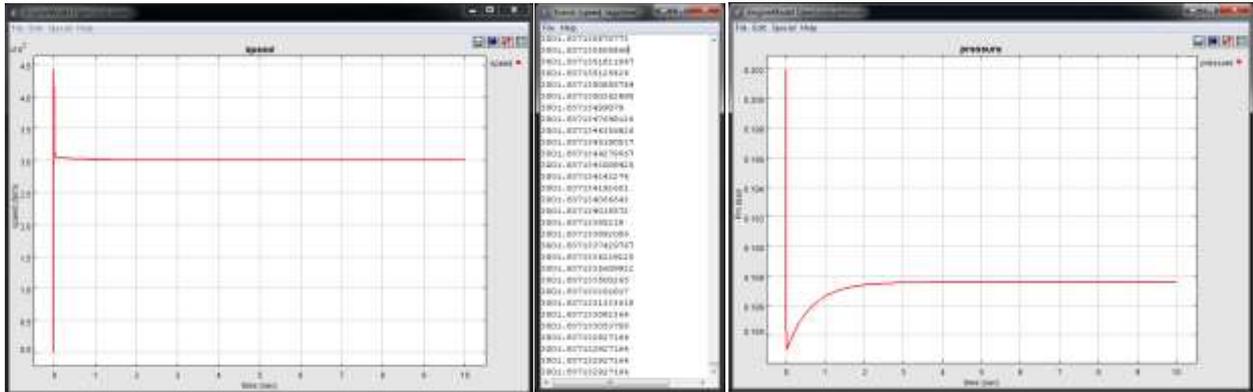
طبعاً تم الأخذ بالحسبان أن إشارات الدخل هي إشارات مستمرة لذلك لم يتم استخدام توابع الاستقبال ( $FcdR$ ), أما توابع الإرسال ( $FcdT$ ) فتقوم بتحويل الإشارات المراد إرسالها إلى سلسلة من الأحداث على شكل ثنائيات. أما الساعة الرئيسية ( $master\ clock$ ) فنقوم الساعة المحلية ( $local\ clock$ ) بتدريج يتم اختياره بحسب التطبيق. ونتائج تطبيق النموذج على محرك السيارة لضبط سرعته كحلقة مفتوحة ( $open\ loop$ ) مبينة في الأشكال (9-a-b) (10). في النظم المستمرة التي تعتمد على المعادلات التفاضلية تظهر مشكلة القيم البدائية ( $initial-value\ problems$ ) ومشكلة التقسيم على صفر أثناء المحاكاة، لذلك نتجاوز التقسيم على صفر يتم اختيار قيم بدائية مناسبة وهذا يؤدي إلى ظهور تغير في الحالة العابرة للنظام بحسب القيم البدائية.



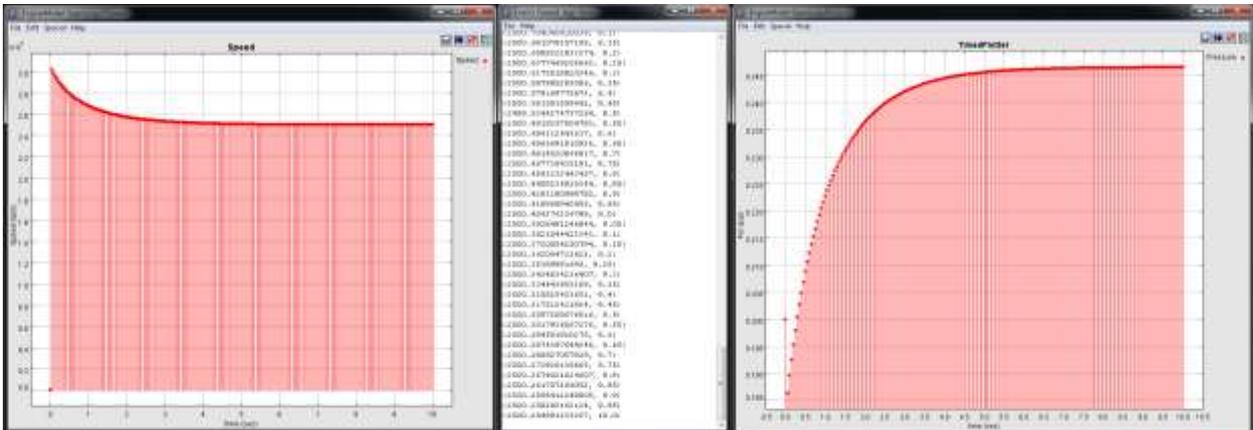
الشكل (8): تطبيق النموذج على محرك السيارة كحلقة مفتوحة



الشكل (9-أ): اختبار السرعة عند القيم ( $\theta = 5.865 \text{ deg}, \text{load} = 25\text{Nm}$ )



الشكل (9-ب): اختبار السرعة عند القيم ( $\theta = 6.289 \text{ deg}, \text{load} = 20\text{Nm}$ )



الشكل (10): اختبار توابع الإرسال بالنسبة للسرعة عند القيم ( $\theta = 6.4347 \text{ deg}, \text{load} = 20\text{Nm}, f = 10\text{Hz}$ )

## 2-تطبيق النموذج على المحرك كحلقة مغلقة

طبقتنا النموذج بشكله الأساسي على اعتبار أن المثال السابق حلقة مفتوحة (*open loop*). والسؤال : ماذا لو أردنا تحديد السرعة المطلوبة بشكل مسبق؟. يتم تحويل المثال السابق ليصبح حلقة مغلقة (*close loop*)، و النموذج بشكله الأساسي سيكون غير كافٍ لتحقيق النمذجة؛ لذلك نحن بحاجة إلى إضافة تابع آخر لتحقيق النمذجة و هو تابع التحكم (*fcdPI*) أو المتحكم *PI*، حيث يقوم تابع التحكم بمقارنة السرعة المطلوبة مع السرعة الناتجة [21]، و من الفرق بينهما يحدد زاوية فتحة صمام الهواء اللازمة لتحقيق السرعة المطلوبة. بإضافة تابع التحكم يصبح النموذج كالاتي:

*Continuous Domain*

$M(X, I, O, F)$

$X$  : Internal state set

$I$  : Input set

$O$  : output set

$F_{CD} = \{f_{cdState}, f_{cdOut}, f_{cdClick}, f_{cdT}, f_{cdR}, f_{cdPI}\}$

تحتسب زاوية فتحة الصمام بواسطة تابع التحكم بالعلاقة:

$$\theta = f_{cdPI}(N_{set}, N)$$

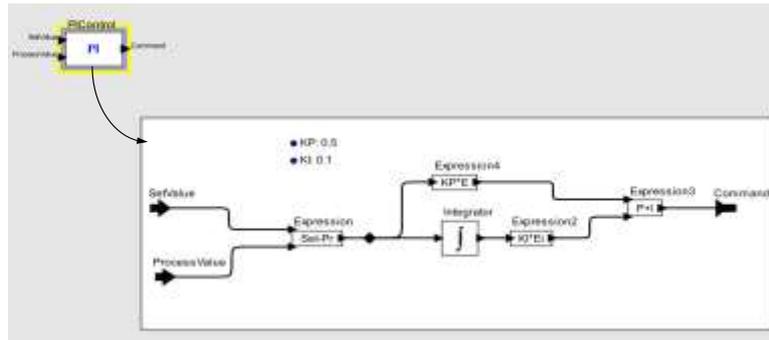
$$\theta = K_p(N_{set} - N) + K_I \int (N_{set} - N) dt \quad (7)$$

$N_{set}$  = speed set point

$K_p$  = proportional gain

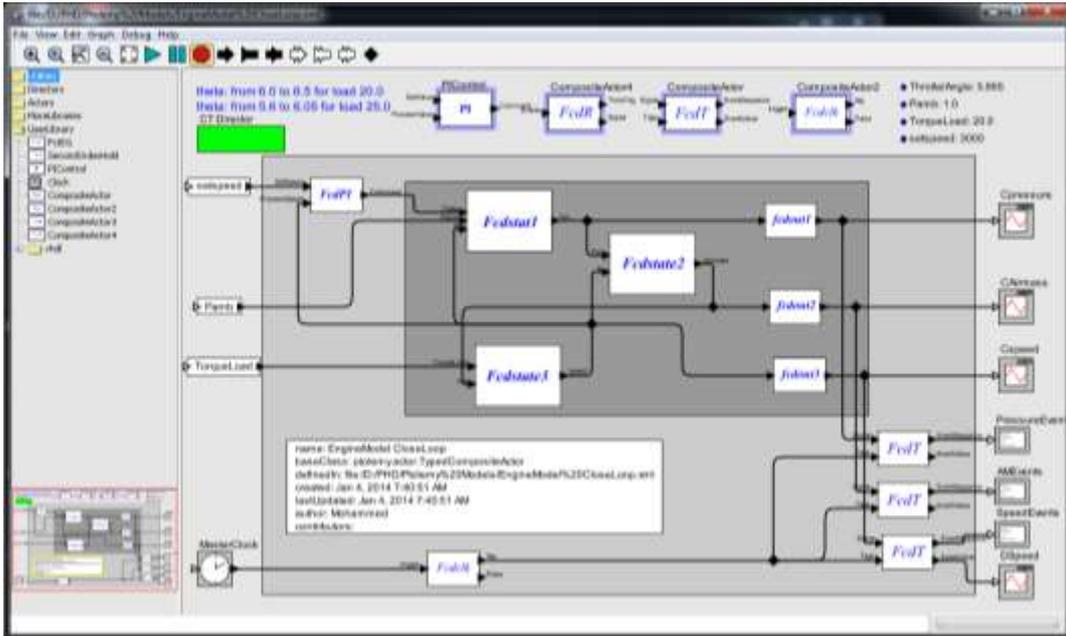
$K_I$  = integral gain

الشكل (11) يمثل تحقيق تابع التحكم بواسطة *PtolemyII* وفق العلاقة (7).



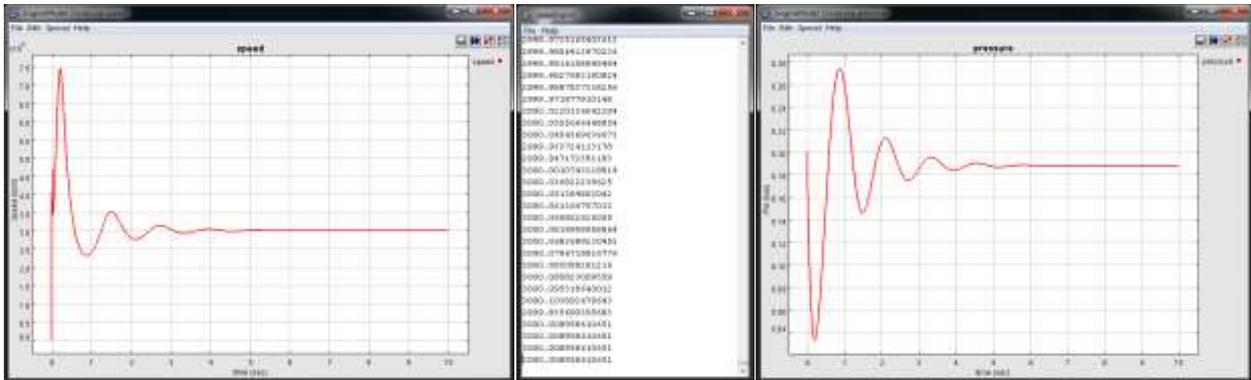
الشكل(11): تابع الحكم ( $f_{cdPI}$ )

باستخدام تابع التحكم يصبح النموذج كما هو مبين في الشكل (12).

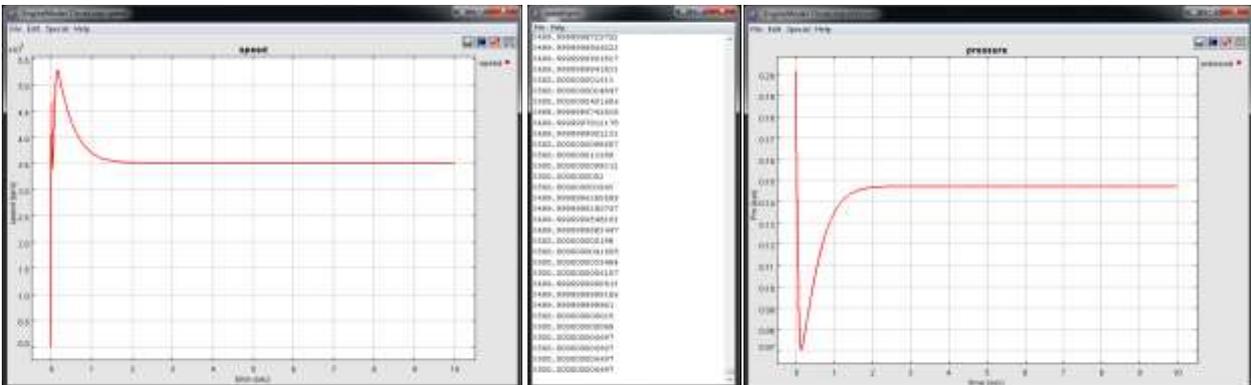


الشكل (12): تطبيق النموذج على محرك السيارة كحلقة مغلقة

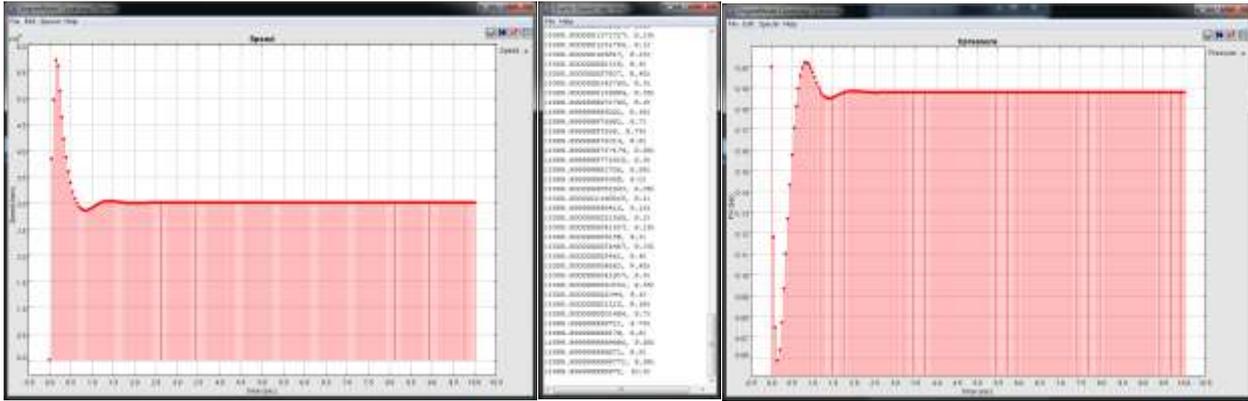
نتائج إختبار النموذج كحلقة مغلقة مبينة بالأشكال (13-a-b) (14).



الشكل (13-a): النتيجة بحسب القيم (Set speed=3000 rpm, Kp=0.001, Ki=0.052)



الشكل (13-b): النتيجة بحسب القيم (Set speed=3500 rpm, Kp=0.02, Ki=0.06)

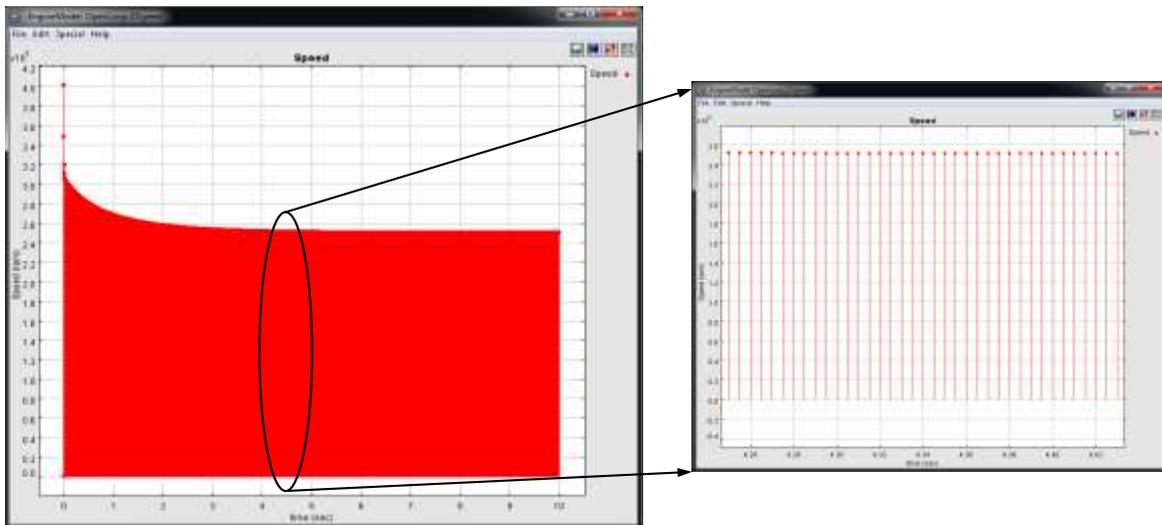


الشكل(14): نتيجة اختبار توابع الإرسال عند القيم ( $Set\ speed=3000\ rpm, Kp=0.008, Ki=0.072, f=10Hz$ )

## النتائج و المناقشة:

### أ. نتائج الاختبار في الحلقة المفتوحة:

تحدد السرعة عن طريق عزم التدوير اللازم للتغلب على عزم الحمل الذي يعبر عن جميع القوى و العزوم المعاكسة لعزم التدوير. يزداد عزم التدوير بازياد كمية مزيج الأحتراق (هواء + وقود) في المحرك، و تتحدد كمية المزيج عن طريق فتحة الخانق، فكلما كانت زاوية الخانق أصغر كانت كمية الهواء الداخلة إلى المحرك أكبر، و كمية الهواء الداخلة إلى المحرك تحدد بشكل آلي كمية الوقود اللازمة للاحتراق وفق المعدل ( $A/F = 14/7$ ). للحصول على السرعة ( $3000\ rpm$ ) عند عزم حمل ( $load = 25Nm$ ) نحتاج لفتح الخانق بزاوية ( $\theta = 5.865\ deg$ ). و عند عزم حمل أصغر ( $load = 20Nm$ ) نحتاج إلى زيادة زاوية فتحة الخانق إلى ( $\theta = 6.289\ deg$ ) لتخفيض كمية الهواء ومن ثمّ المزيج المحترق و منه إنقاص عزم التدوير كما هو مبين في الأشكال (9-a-b). أما في الشكل (10) فهو خرج تابع الإرسال عند استخدام تردد الساعة المحلية بقيمة ( $10Hz$ ) حيث تم ضبط السرعة على القيمة ( $2500\ rpm$ ) عند عزم حمولة ( $load = 20Nm$ ) و فتحة الخانق بزاوية ( $\theta = 6.4347\ deg$ )، فإذا أعدنا الاختبار للسرعة ( $2500\ rpm$ ) لكن عند قيمة تردد ( $100Hz$ ) فنحصل على قيمة السرعة من خرج تابع الإرسال كما هو مبين في الشكل (15).



الشكل(15): خرج توابع الإرسال بالنسبة للسرعة عند القيم ( $\theta = 6.4347\ deg, load = 20Nm, f = 100Hz$ )

ب. نتائج الإختبار في الحلقة المغلقة:

في الحلقة المغلقة لا داعي لتحديد زاوية فتحة خانق الهواء مهما كان عزم حمل، إذ يكفي تحديد السرعة المطلوبة فيقوم تابع التحكم ( $f_{CdPI}$ ) بمقارنة السرعة الناتجة مع السرعة المطلوبة و بناءً على الفرق بينهما يحدد زاوية فتحة الخانق المناسبة لتحقيق السرعة المطلوبة بشكل آلي، و في الحلقة المغلقة فإن لثوابت تابع التحكم (ثابت التفاضل + ثابت التكامل) دورها في زيادة أو إنقاص التعرج في المرحلة الانتقالية التي تسبق الاستقرار. ففي الشكل (13-a) نلاحظ أن عامل التعرج كبير في المرحلة الانتقالية عند استخدام القيم ( $Kp = 0.001, Ki = 0.052$ ) أما في الشكل (13-b) فنلاحظ أن التعرج انخفض باستخدام قيم جديدة لتابع التحكم ( $Kp = 0.02, Ki = 0.06$ ) إن زيادة ثابت التكامل مع المحافظة على ثابت التناسب يؤدي إلى زيادة التعرج؛ لذلك لتخفيض التعرج فإنه يتم برفع ثابت التناسب بالتوازي مع ثابت التكامل. وقيم تابع التحكم التي تم اختيارها هي قيم تجريبية، لكن لتوليفه (*tuning*) يتم اختيار قيمة ثابت التفاضل  $Kp$  أولاً ومن ثم يتم ضبط قيمة ثابت التكامل  $Ki$  وهكذا بشكل تكراري حتى يتم الحصول على أفضل تعرج للمرحلة الانتقالية. عمل تابع الإرسال في الحلقة المغلقة نفسه في الحلقة المفتوحة ففي الشكل (14) يبين خرج تابع الإرسال للسرعة عند تردد ساعة (10 Hz).

مصفوفة النظام

إذا افترضنا أن إشارة دخل غير مستمرة فإننا بحاجة إلى ثلاثة توابع استقبال، فتصبح توابع النموذج على الشكل

الآتي:

$$F_{CdState} = [f_{CdstatePm}, f_{CdstateM}, f_{CdstateN}]$$

$$F_{CdOut} = [f_{CdoutPm}, f_{CdoutM}, f_{CdoutN}]$$

$$F_{CdT} = [f_{CdTPm}, f_{CdTM}, f_{CdTN}]$$

$$F_{CdR} = [f_{CdR\theta}, f_{CdRPamb}, f_{CdRLoad}]$$

$$f_{Cdclk}, f_{CdPI}$$

فتصبح مصفوفة النظام على الشكل الآتي:

$$S_{Engine} = \begin{pmatrix} f_{CdstatePm} & f_{CdstateM} & f_{CdstateN} \\ f_{CdoutPm} & f_{CdoutM} & f_{CdoutN} \\ f_{CdTPm} & f_{CdTM} & f_{CdTN} \\ f_{CdR\theta} & f_{CdRPamb} & f_{CdRLoad} \\ f_{Cdclk} & f_{CdPI} & \times \end{pmatrix}$$

الاستنتاجات و التوصيات:

في هذا البحث تم تعريف أساس رياضي شامل لبناء نموذج تصميم مشترك للبنى الصلبة و البنى البرمجية (*Hw/Sw co-design model*) للنظم المضمنة، وانطلاقاً من التعريف الرياضي تم تصميم نموذج جديد للنظم المضمنة المستمرة ككيان أساسي في النظم المضمنة التخالفية، ومن ثم طبقناه على محركات السيارات كنظام ديناميكي مستمر.

تأتي أهمية النموذج الجديد من خلال:

1. مرونته و قابليته على التوسع، لأن أساسه الرياضي يتيح لنا تعريف أي تابع جديد نحتاجه في أي تطبيق، وغير موجود ضمن توابعه الأساسية. وهذا ممكن كما في التطبيق الذي قدمناه في البحث، بالإضافة إلى أن النظم المضمنة المستمرة هي نظم شاملة ويندرج تحت هذا الاسم العديد من النظم الميكانيكية و الفيزيائية ...إلخ.
  2. من خلال الخرج الذي يقدمه، حيث يقدم النظام المضمن المستمر على شكل مصفوفة من التوابع الغرضية، أطلقنا عليها مصفوفة النظام، لأنها تعدُّ أساساً للمراحل التالية في التصميم المشترك للنظم المضمنة المستمرة. حيث يتم بسهولة الحصول على المصفوفة العتادية (HW, SW) من مصفوفة النظام بحسب متطلبات التطبيق.
  3. اعتماد النمذجة الرياضية في الحسابات المضمنة للحصول على الخرج المناسب من تفاعل الدخل مع الكيان المستمر، و تتيح اعتماده كنموذج تصميم مشترك (Co-design model) جديد للنظم المستمرة.
- وبناءً على التعريف الرياضي الذي تم إيجاده من خلال هذا البحث، يمكن تعميم النموذج على أنواع نظم أخرى ليشمل النظم المضمنة التخالفية ككيان متكامل. و من ثم وضع آلية تقسيم للحصول على المصفوفة العتادية من خلال مصفوفة النظام.

#### المراجع:

1. R.V.Hanxleden; E.A.Lee. C.Motika; H.Fuhrmann. *Multi-View Modeling and Pragmatics in 2020 Position Paper on Designing Complex Cyber-Physical Systems*.University of California at Berkeley; Funkwer Information Technologies GmbH 2012 (P1-5).
2. Adnan Shaout, Ali H; El-Mousa; and Khalid Mattar. *Specification and Modeling of HW/SW CO-Design for Heterogeneous Embedded Systems* . [Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.](P1-6).
3. F.Balarin; M.Chiodo; P.Giusto; H.Hsieh. *Hardware-Software Co-design of Embedded Systems/the POLIS Approach/*. Kluwer Academic Publishers, Boston 1997(P1-75).
4. Daniel D.Gajski; F.Vahid; S.Narayan; J.Gong. *Specification and Design of Embedded Systems*. University of California at Irvine 1994(P15-61).
5. Axel Jantsch. *Modeling Embedded Systems and SoCs/Concurrency and Time in MoC*. Royal Institute of Technology. Stockholm, Sweden 2004(P47-134) (P181-188) (P223-235).
6. Jeff C.Jensen. *Element of Model-Based Design*. Department of Electrical Engineering & Computer Science/ University of California, Berkely/ 2010(P1-15).
7. P.Derler; E.D.Lee; A.S.Vincentelli. *Addressing Modeling Challenges in Cyber-Physical Systems*. University of California at Berkeley 2011(P1-19).
8. David Broman. *High-Confidence Cyber-Physical Co-design*. University of California at Berkeley , USA and Linkoping University, Sweden, 2012(P1).
9. P.Derler; E.A.Lee; M. Torngren. *Cyber-Physical System Design Contracts*. University of California at Berkeley , KTH Royal Institute of Technology, 2013(P1-10).
10. I.Liu; E.A.Lee; M.Viele; G.Wang; H.Andrade. *A Heterogeneous Architecture for Evaluating Real-Time one-Dimensional Computational Fluid Dynamics on FPGAs*. IEEE 2012(P1-8).
11. Frank L. Severance. *System Modeling and Simulation/an Introduction*. Western Michigan University 2001(P34-67) (P141-177).

12. B. P. Zeigler; H. Praehofer; T. G. Kim. *Theory of Modeling and simulation/integration discrete event and continuous complex dynamic systems/; 2<sup>nd</sup> ed. Academic Press 2000(P49-64) (P99-129).*
13. A. M. Law; W. David Kelton. *Simulation Modeling and Analysis, 3<sup>rd</sup> ed. McGraw-Hill Higher Education 2000(P1-91).*
14. P. Blanchard; R. L. Devaney; G. R. Hall. *Differential Equations, 3<sup>rd</sup> ed. Boston University 2006(P646-688).*
15. N. Finizio; G.Ladas. *An Introduction to Differential Equations/Fourier Series and Partial Differential Equations/. University of Rhode Island 1982(P286-314).*
16. G. A.Wainer. *Discrete-Event Modeling and Simulation/ A Practitioner's Approach/. Taylor & Francis Group, 2009(P35-53).*
17. Edward A. Lee. *Ptolemy Project . <http://ptolemy.eecs.berkeley.edu>.*
18. William B. Ribbens. *Understanding Automotive Electronics, 6<sup>th</sup> ed. Elsevier Science(USA) 2003(P157-194).*
19. Allan Bonnick. *Automotive Science and Mathematics. Elsevier Ltd 2008(P158-170).*
20. *Using Simulink and Stateflow in Automotive Applications; <ftp://ftp.mathworks.com/pub/product-info/examples>; 1998.*
21. Antonio Visioli. *Practical PID Control. Springer-Verlag London Limited 2006(P1-18).*