

# كيف يعمل السميوليك؟؟

## مقدمة

يقوم السميوليك بمحاكاة الأنظمة الديناميكية كما تعرفنا سابقا وتتم هذه العملية بمرحلتين هما :  
**المرحلة الأولى** يقوم بها المستخدم بعمل النموذج والذي يحتوى على مجموعة البلوكات المطلوبة .  
**والمرحلة الثانية** يقوم البرنامج بتنفيذ عملية المحاكاة فى الفترة الزمنية المطلوبة .

## نمذجة المنظومات الديناميكية

من المعروف ان المنظومات الديناميكية تتكون من مجموعة من المعادلات الرياضية ويتم تمثيل هذه المعادلات فى السميوليك على هيئة بلوكات وهذه الفكرة مأخوذة من مبادئ التحكم الالى والمعروف بـ

Block Diagram

وتنقسم البلوكات فى السميوليك الى نوعان نوع افتراضى و نوع غير افتراضى (nonvirtual block and virtual blocks) .

الانواع الغير افتراضية هى التى تمثل عناصر النظام الديناميكي اما الانواع الافتراضية وهى التى تستخدم فى تحويل الاشارات وغيرها دون ان تدخل فى تكوين النظام او معادلاته الرياضية .

## ما معنى "time-based block diagram"؟؟؟

1- اى ان هناك علاقة زمنية بين الاشارات وبين المتغيرات (state variables) ويكون حل النموذج او block diagram هو حل لهذه العلاقات خلال الزمن المحدد time step والذي يمثل بزمان البداية الى زمن النهاية .

2- الاشارات تعبر عن كميات تتغير مع الزمن وتكون معرفة خلال الفترة الزمنية المحددة .

3- العلاقة بين الاشارات والمتغيرات تكون عبارة عن مجموعة من المعادلات اى ان كل بلوك يحتوى على مجموعة من المعادلات وهذه المعادلات توضح العلاقة بينه و بين الداخل له وبين الخارج منه .

ويوجد نوعان من انواع البلوكات تبعا لنوعية بناءها

1- البلوكات الموجودة فى البرنامج وتسمى **built-in blocks**

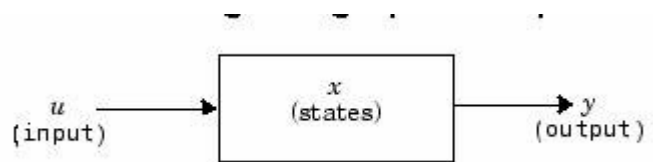
2- البلوكات التى يقوم المستخدم بعملها وتسمى

**custom blocks User-defined**

## States

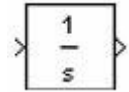
وهى التى تمثل قيم النظام الموجود وهى عبارة عن مجموعة من المتغيرات والتى تستخدم فى حساب الخرج الخاص بالبلوك عند الخطوة الزمنية المحدده وهناك نوعان من انواع ال states وهما :  
**Discrete و continuous** متقطعة ومستمرة .

والمستمرة هى التى تتغير باستمرار اما المتقطعة هى التى تتغير عند فترات زمنية محددة intervals .  
وتعتبر البلوكات States كما فى الشكل التالى



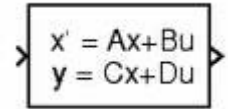
والبلوكات التى تعبر continuous states يجب ان تحتوى على احد البلوكات الاتية :

### Integrator



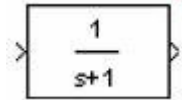
ووظيفة هذا البلوك هو تكامل الاشارة الداخلة

### State-Space



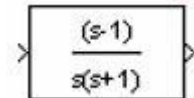
ووظيفة هذا البلوك هو عمل نظام خطى من النوع State-Space

### Transfer Fcn



ووظيفة هذا البلوك هو عمل نظام خطى من النوع transfer function

### Zero-Pole



وهذا البلوك يعتبر حالة خاصة من السابق حيث يعبر عن نظام خطى من النوع transfer function ولكن يحتوى على zero-pole-gain اى يوجد قيم ل S تساوى صفر سواء فى البسط او المقام .

## الحالات المستمرة Continuous States

للتعامل مع الحالات المستمرة يجب علينا معرفة معدل تغيرها او مشتقتها و قيمة الحالة المستمرة يساوى تكامل مشتقتها فى الفترة الزمنية المحدده وتعتمد دقة هذه الحسابات على مقدار الخطوة الزمنية وكما معروف كلما صغرت الخطوة الزمنية فان الدقة سوف تزيد ولكن سيسبب هذا فى حمل اضافى على معالج الكمبيوتر وقد يسبب بعض البطء .  
ومن مميزات السميولنك انه يوجد به مجموعة من طرق الحل و تكون بها الخطوة الزمنية متغيرة تبعا لمعدل التغير .

### الحالات المتقطعة Discrete States

للتعامل مع الحالات المتقطعة يجب علينا معرفة العلاقة بين قيمتها فى الفترة الزمنية الحالية وبين قيمتها فى الفترة الزمنية السابقة وهذه العلاقة تسمى فى البرنامج update functions وتعتمد أيضا على قيمة الدخل للنموذج .

## نمذجة الانظمة التي تحتوى على حالات متقطعة ومستمرة معا Modeling Hybrid Systems

وفى هذا النوع يقوم السميولنك بوضع مقدار للفترة الزمنية يحقق الدقة المطلوبة لتكامل الحالات المستمرة وفى نفس الوقت يكون مناسب للحالات المتقطعة .

### قيم معاملات البلوك Block Parameters

وهى التى تعبر عن خصائص اى بلوك فمثلا البلوك الخاص باضافة ثابت يعتبر Parameter ولكل بلوك له Parameters يكون له خصائص ويمكن استعمال الماتلاب فى تعديل هذه الخصائص .  
ويقوم السميولنك بحساب هذه القيم قبل بداية المحاكاة كما يمكنك تغييرها اثناء عمل المحاكاة .  
وتسمى بى A tunable parameter وهى التى تغييرها دون الحاجة الى اعادة ترجمة النموذج الى لغة الالة recompiling ويجب العلم ان هذا التغيير لن يكون سريع ولكن سوف يقوم بالانتظار الى بداية الفترة الزمنية الجديدة ويمكن جعل جميع parameters الموجوده غير قابلة للتعديل مما يودى الى زيادة فى سرعة التنفيذ .

### الفترة الزمنية Block Sample Times

يوجد لكل بلوك فترة زمنية خاصة به ويمكننا تغييرها وللبلوكات ذات الحالات المستمرة تكون الفترة الزمنية مالا نهاية وتسمى continuous sample time .  
وبالنسبة للبلوكات الغير متاح بها الفترة الزمنية تسمى implicit sample time  
او

fundamental sample time of the inputs

اى يعتمد على نوع الدخل فمثلا لو كان الدخل مستمر فان الفترة الزمنية تكون مثل الحالات المستمرة .  
اما اذا كان الدخل متقطع فهنا الفترة الزمنية تسمى An implicit discrete sample time وتساوى اصغر فترة زمنية فى الدخل .

### Systems and Subsystems

يمكن للنموذج فى السميولنك ان يتكون من عدة طبقات كل طبقة تسمى نظام فرعى subsystem ويوجد نوعان منها ايضا وهما افتراضى وغير افتراضى virtual and nonvirtual ومن مميزات البرنامج انه يمكنك عمل Subsystems يكون تنفيذها متوقف على شروط معينة مثل استعداد دالة معينة او فعل ويكون اغلب الانظمة الفرعية الغير مرتبطة باى شروط أنظمة افتراضية .

### طرق البلوكات Block Methods

البلوكات هى عبارة عن مجموعة من المعادلات كما عرفنا وهذه المعادلات تعبر عن طريقة البلوك ويتم تنفيذ هذه الطريقة خلال تنفيذ النموذج ومن انواع هذه الطرق :

#### Outputs

والتي فيها يتم حساب خرج البلوك من الدخل عند الفترة الزمنية الحالية والفترة الزمنية السابقة

#### Update

وهى الطريقة الخاصة بالبلوكات ذات الحالات المتقطعة فى الفترة الزمنية الحالية

#### Derivatives

وهى الطريقة الخاصة بالبلوكات ذات الحالات المستمرة فى الفترة الزمنية الحالية ويكون التعبير فى السميولنك عن نوع الطريقة المستخدمة كما يلى ::

BlockType.MethodType

# محاكاة الانظمة الديناميكية

## ماذا يحدث عند الضغط على Start Simulation ???

### اولا Model Compilation

اى يقوم السيميولنك بتحويل البرنامج او النموذج الى الصورة التى يمكنه حلها executable form ويكون تسلسل خطوات هذه العملية كما يلى :

1- حساب قيم التعبيرية لخصائص البلوكات وايجاد قيمتها  
block parameter expressions

2- تحديد خصائص الاشارات مثل نوعها و نوع البيانات و ابعادها

3- ويقوم بتحديد الخصائص الغير موجودة ويستخدم طريقة  
attribute propagation

4- يقوم بعملية تخفيض للبلوكات  
optimizations

5- يقوم وضع النموذج فى تسلسل هرمى  
hierarchy

6- يقوم بتحديد رتبة كل بلوك وسوف نتعرض له لاحقا  
block sorted order

7- يقوم بتحديد الفترات الزمنية لكل بلوك  
Sample Time

### ثانيا : Link Phase

وفى هذه المرحلة يقوم البرنامج بتحديد الاماكن اللازمة فى الذاكرة لتنفيذ النموذج

### ثالثا : Simulation Loop Phase

وفى هذه المرحلة يقوم البرنامج بحساب قيم الحالات والخرج خلال الفترات الزمنية حتى نهاية زمن المحاكاه

وتحتوى هذه المرحلة على مرحلتين فرعيتين :

### 1- Loop Initialization phase

وهذا يحدث مرة واحدة فقط فى البداية

### 2- Loop Iteration phase

اما هذه المرحلة فيعاد تكرارها عند بداية كل فترة زمنية جديدة ويحدث بها

1- حساب خرج البرنامج او النموذج

2- حساب حالة البرنامج او النموذج

3- البحث عن حالات غير مستمرة فى البلوكات المستمرة باستخدام  
zero-crossing detection  
(هذه الخطوة اختيارية ) وسوف نتعرض لها لاحقا

4- حساب زمن الفترة الزمنية التالية  
ويتم تكرار هذه الخطوات طوال زمن المحاكاة.

## طرق الحل فى السميولنك

حل النموذج المقصود به هو عملية حساب الحالات المتعاقبة  
successive states

وطرق الحل هى عبارة عن مجموعة من البرامج الموجودة فى البرنامج وتسمى  
solvers  
ومن أهم الانواع

### 1- Fixed-step solvers

وهى التى تقوم بحل النموذج فى فترات زمنية منتظمة من البداية حتى النهاية وحجم هذه الفترات يعرف بى step size وكما نعلم مع تقليل ال step size فان الدقة سوف تزيد .

### 2- Variable-step solvers

وهى التى تقوم بحل النموذج فى فترات زمنية متغيرة فتقوم بتصغير حجم الفترة الزمنية لزيادة الدقة عندما تتغير حالة النموذج بسرعة وتقوم بتكبير حجم الفترة الزمنية عندما يكون التغير فى الحالة بطيء.

وهناك ايضا من انواع Solvers تبعاً للحالة مثل

### 1- Continuous solvers

وتكون عبارة عن عملية تكامل عددى فى الفترة الزمنية الحالية لحساب حالة النموذج من الحالة عند الفترة الزمنية السابقة ومن مشتقتها كما ذكر سابقا .

### 2- Discrete solvers

وتقوم بحساب حجم الفترة الزمنية التالية فقط

## Zero-Crossing Detection

وهى الطريقة المستخدمة فى البحث عن الحالات الغير المستمرة فى كل فترة زمنية وعندما يجد البرنامج منطقة بها عدم استمرارية يقوم بتحديد زمنها بدقة ويقوم بأخذ فترات زمنية إضافية قبلها وبعدها .

لماذا نستخدم هذه الطريقة ؟؟

وذلك لان عند مناطق عدم الاستمرارية يحدث تغير هام جدا فى الخصائص الديناميكية للنظام وغالبا تتزامن مناطق عدم الاستمرارية مع الاحداث المهمة فى النظام .

ومن الممكن الاستغناء عن هذه الطريقة بتقليل حجم الفترة الزمنية الى قيم صغيرة جدا مما قد يؤدي الى زيادة زمن المحاكاة.

## Algebraic Loops

بعض البلوكات فى السميولنك لديها مداخل تسمى

direct feedthrough

وهذا معناه ان خارج هذه البلوكات لا يمكن حسابه بدون معرفة قيم الاشارات الداخلة ومن أهم هذه البلوكات :

• The Math Function block

• The Gain block

• The Integrator block's initial condition ports

• The Product block

• The State-Space block when there is a nonzero D matrix

• The Sum block

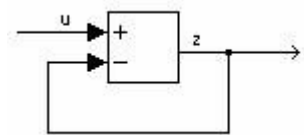
• The Transfer Fcn block when the numerator and denominator are of the same order

• The Zero-Pole block when there are as many zeros as poles

وال algebraic loop يحدث عندما يكون الداخل

direct feedthrough

وأیضا معرض للخارج كما فى الشكل التالى :



ومن الشكل السابق نرى ان  $z = u - z$  ويكون الحل  $z = u/2$

ويمكننا أيضا استخدام بلوك

Algebraic Constraint

فى عمل algebraic loop

# Modeling and Simulating Discrete Systems

تكمّن مقدرة السميولنك على محاكاة الانظمة ذات الزمن المتقطع والتي تسمى (sampled data) وايضا قدرة على محاكاة الانظمة التي يكون معدل تغيرها غير ثابت (multirate systems) - والتي يكون فيها بلوكات ذات فترة زمنية معينة وبلوكات أخرى ذات فترة زمنية مختلفة - ومحاكاة الانظمة التي تجمع بيانات متصلة و متقطعة معا (hybrid systems) فى الخاصيتين التاليتين ::

## 1- SampleTime block parameter

يوجد نوعان من نوع حجم الفترة الزمنية Sample Time block parameter وهما explicit و implicit والبلوكات ذات الزمن المتصل تكون من النوع implicit .

## 2- Sample-time inheritance

يمكن لأغلب بلوكات السميولنك ان تأخذ حجم الفترة الزمنية الخاص بها من البلوك المتصل بمدخلها أما بالنسبة للبلوكات التى ليس لها مدخل يمكنها ان تتوارث الفترة الزمنية من البلوكات المتصلة بمخرجها .

## Determining Step Size for Discrete Systems

يقوم السميولنك باختيار حجم للفترة الزمنية step size

متزامن مع الزمن الخاص بمعدل تقطيع الاشارة sample time hits ويكون اختياره بناء على fundamental sample time

ويكون fundamental sample time هو أكبر عدد صحيح مقسوما عليه ال sample time مثلا :

لدينا 0.25 sample times و 0.5 فيكون fundamental sample time 0.25

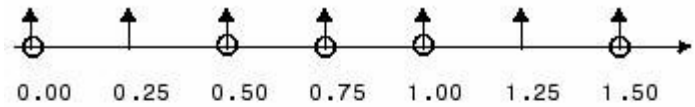
ويمكننا فى محاكاة الانظمة ذات الزمن المتقطع استخدام كلا النوعين من ال solver وهما

variable-step discrete

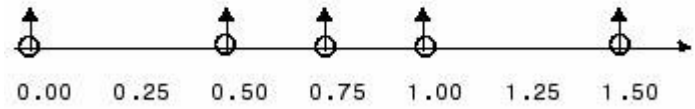
او fixed-step

وفى حالة fixed-step يكون simulation step size يساوى fundamental sample time

وفى حالة variable-step solver يكون step size مساوى للمسافة بين نقط التقطيع sample time hits. والفرق بينهم موضح فى الشكل التالى :



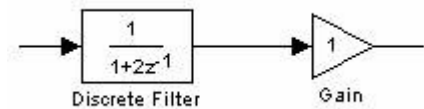
Fixed-Step Solver



Variable-Step Solver

## Sample Time Propagation

توليد زمن التقطيع أو حجم الفترة الزمنية يقوم السميولينك هذه العملية في بداية المحاكاة لتحديد زمن التقطيع للبلوكات التي تتوارث زمن تقطيعها وهى البلوكات التي ليس لها مداخل ومثلا في النموذج الاتي ::



نرى بلوك gain ووظيفته هى ضرب الدخل في ثابت والناتج يكون هو الخرج ولذلك الخارج يكون له نفس زمن تقطيع البلوك السابق له ويقوم السميولينك بالاتي ::  
 1- اذا كان الداخل له نفس زمن التقطيع فان السميولينك يقوم بتخصيصه  
 2- اذا كان الداخل له زمن تقطيع مختلف ولكن عدد صحيح و اسرع من زمن البلوك نفسه فان السميولينك يقوم بتخصيص الزمن الاسرع

## Constant Sample Time

زمن التقطيع الثابت ::

والمقصود به هو زمن التقطيع الخاص بالبلوكات التي لا يتغير زمن تقطيعها اثناء عملية المحاكاة وشروط هذه البلوكات ::

1- ان يكون جميع معاملات البلوك parameters غير قابلة للتعديل اثناء المحاكاة nontunable  
 2- ومن الممكن وضع زمن التقطيع لهذه البلوكات مالا نهية (inf) او تكون قابلة لتوارث زمن التقطيع من بلوكات أخرى بشرط ان تكون تلك البلوكات ذات زمن تقطيع ثابت اثناء عملية المحاكاة .  
 ويقوم السميولينك عمل بحث عن هذه البلوكات قبل بداية المحاكاة حتى يسهل من عملية الحسابات

أما اذا وجد السميولينك بلوكات لديها زمن تقطيع مالا نهية ولكن لا تعتبر ذات زمن ثابت وذلك نتيجة وجود معاملات من الممكن تعديلها

Tunable Parameters

فانه يقوم بتنفيذ عملية Sample Time Propagation وقد سبق ذكرها

والان انتهى الدرس الثانى والى اللقاء فى الدرس القادم مع أساسيات السميولينك