백서

임베디드 제어 시스템을 위한 모델 기반 설계

MathWorks®

여러분의 팀에서 로봇 시스템의 일부를 구동하는 제어기를 개발하고 있다고 생각해 보십시오. 이 시스템은 속도가 제어되는 전기 모터로 구동됩니다. 여러분은 설계를 시작하기 전에 다음과 같은 몇 가지 중요한 질문에 대해 생각을 해보아야 합니다.

- 모터의 크기는 어떻게 정해야 하는가?
- 요구사항이 변경되면 어떻게 해야 하는가?
- 원하는 성능을 보장할 수 있도록 설계를 어떻게 최적화하는가?
- 어떻게 하면 위험을 최소화하면서 설계를 철저히 검사할 수 있는가?

산업용 로봇, 풍력 터빈, 생산 기계, 자율주행 차량, 굴착기 또는 전기 서보 드라이브에 대한 제어기를 개발하면서 여러분의 팀이 수작업으로 코드를 작성하고 문서 기반 요구사항 수집을 사용하고 있다면 이런 질문들에 대한 답을 얻으려면 직접 시행착오를 겪거나 실물 프로토타입에 테스트를 해보는 방법 외에는 없습니다. 그리고 만일 요구사항 하나가 변경되면 전체 시스템을 다시 코딩하고 구축해야 하기 때문에 프로젝트가 며칠 또는 몇 주까지도 지연될 수 있습니다.

MATLAB® 및 Simulink®를 통해 모델 기반 설계를 사용하면 수작업으로 작성한 코드 및 문서 대신에 시스템 모델을 만들 수 있습니다. 산업용 로봇을 예로 들면, 로봇 팔, 모터, 제어기 설계로 이루어진 모델을 만들게 됩니다. 언제든지 모델을 시뮬레이션하여 값비싼 하드웨어에 의존하지 않고 지연 및 위험 없이 즉시 시스템 동작을 확인하고 다양한 가정 시나리오를 테스트할 수 있습니다.

이 백서에서는 모델 기반 설계를 소개하고 시작에 도움이 되는 팁과 모범 사례를 다룹니다. 실제 예제를 사용하여 다양한 업계의 팀들이 모델 기반 설계를 채택하여 어떻게 개발 시간을 단축하고 컴포넌트 통합 문제를 최소화하며, 더욱 고품질의 제품을 제공하는지 보여드립니다.

모델 기반 설계란?

모델 기반 설계를 이해하는 가장 좋은 방법은 실제로 그것이 어떻게 이루어지는지 보는 것입니다.

한 자동차 엔지니어 팀이 승용차의 ECU(엔진 제어 유닛)를 구축하는 사례를 살펴보겠습니다. 그들은 모델 기반 설계를 이용하고 있기 때문에 먼저 시스템 요구사항으로부터 아키텍처 모델을 구축하게 됩니다. 이 사례에서의 시스템은 4기통 엔진입니다. 이어서 시뮬레이션/설계 모델을 도출합니다. 이 고수준, 저충실도 모델에는 ECU 및 플랜트(이 사례에서는 엔진과 구동 환경)에서 실행될 제어 소프트웨어의 일부분들이 포함되어 있습니다.

팀은 다양한 시나리오에서 이 고수준 모델을 시뮬레이션하여 초기 시스템 테스트 및 통합 테스트를 수행함으로써 이 시스템이 정확히 표현되었고 입력 신호에 적절히 응답하는지 검증합니다.

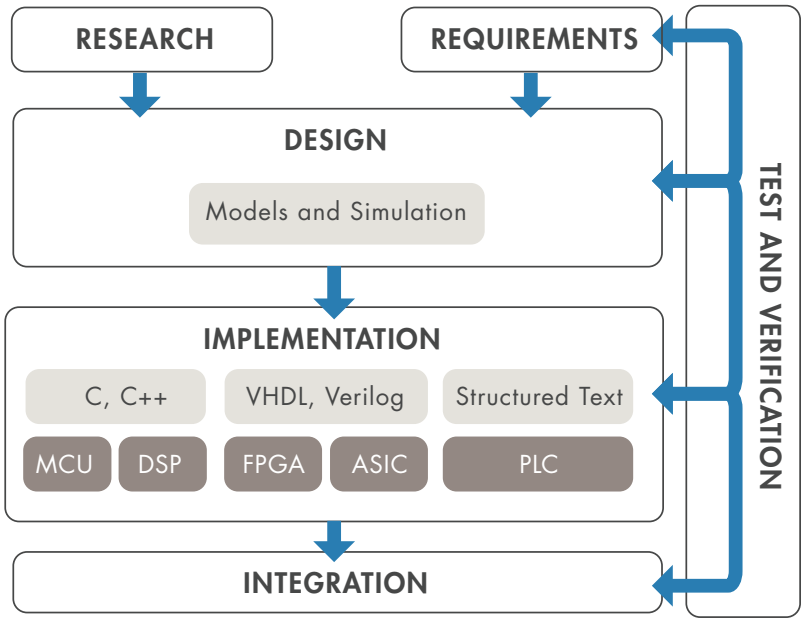
이어서 모델에 세부 사항을 추가하고 계속적으로 테스트하며, 사양과 비교하여 시스템 수준 거동을 검증하게 됩니다. 시스템의 규모가 크고 복잡하다면 엔지니어는 개별 컴포넌트를 따로 개발하고 테스트할 수 있지만,
그래도 전체 시스템 시뮬레이션에서 이를 빈번히 테스트합니다.

궁극적으로 팀은 시스템 및 시스템이 구동되는 환경에 대한 상세한 모델을 구축하게 됩니다. 이 모델은 시스템에 대한 누적된 지식(IP)을 수집합니다. 엔지니어는 제어 알고리즘

모델에서 자동으로 코드를 생성하여 소프트웨어 테스트 및 검증에 활용할 수 있습니다. HIL(Hardware-in-the-Loop) 테스트에 이어 엔지니어는 생성된 코드를 생산 하드웨어에 다운로드하여 실제 차량에서 테스트하게 됩니다.

이 시나리오에서 알 수 있듯이, 모델 기반 설계에서는 종래의 개발 워크플로와 같은 요소를 사용하지만 큰 차이가 두 가지 있습니다.

- 예를 들어, 코드 생성처럼 워크플로에서 많은 시간이 소요되거나 오류가 발생하기 쉬운 단계가 자동화되어 있습니다.
- 요구사항 수집에서 설계, 구현, 테스트에 이르는 개발 과정의 중심에 시스템 모델이 있습니다.



모델 기반 설계의 워크플로.

요구사항 수집 및 관리

요구사항을 문서를 통해 수집하는 종래의 워크플로에서는 전달 과정에서 오류와 지연이 발생할 수 있었습니다. 설계 문서 또는 요구사항을 만드는 엔지니어와 시스템을 설계하는 엔지니어가 다른 경우가 종종 있습니다. 요구사항이 그저 다음 팀으로 전달될 수가 있는데, 이는 두 팀 간에 분명하거나 꾸준한 소통이 없다는 것을 의미합니다.

모델 기반 설계에서는 Simulink 모델 내에서 요구사항을 작성, 분석 및 관리하게 됩니다. 사용자 지정 특성으로 서식 있는 텍스트 요구사항을 작성하고 이를 설계, 코드 및 테스트에 연결할 수 있습니다. 요구사항은 요구사항 관리 툴과 같은 외부 소스에서 가져오거나 동기화할 수도 있습니다. 설계에 연결된 요구사항이 변경되면 자동 알림을 받게 됩니다. 따라서 여러분은 변경에 직접적으로 영향을 받는 설계 또는 테스트 부분을 식별하고, 그걸 해결하기 위해 적절한 조치를 취할 수 있습니다. 시스템 및 소프트웨어 컴포넌트의 아키텍처와 컴포지션을 정의, 분석, 지정할 수 있습니다.

사례 연구: *Embraer*

Embraer Legacy 500.

"우리 그룹은 종래의 방법 대비 두 배 많은 요구사항을 만들었으나 요구사항당 문제 숫자는 오히려 50 배 감소했습니다. 모델 기반 설계를 사용하면서 가장 길었던 요구사항 관련 지연 기간은 단 하루였습니다. 반면 문서를 사용했을 때는 제일 짧은 지연 기간이 2주였습니다."

— Julio Graves, *Embraer*

Embraer Legacy 500은 지능형 비행 조종 및 완전 전자식 비행 제어 기술을 갖춘 최초의 중형 비즈니스 제트기입니다. FCS(비행 조종 시스템)의 기계식 조종을 대체하는 이 기술을 이용하면 더 많은 조종면을 동시에 작동시킬 수 있어서 더욱 매끄러운 비행이 가능하고 조종사 업무량이 줄며, 안전이 향상됩니다.

Embraer는 고객과 함께 Legacy 500을 위한 고수준 요구사항을 개발했습니다. 고수준 요구사항을 FCS 소프트웨어를 개발할 공급사를 위한 잘 작성된 저수준 요구사항으로 변환하는 것이 주된 난관이었습니다.

초기 설계는 모델링과 시뮬레이션을 집중적으로 사용하지 않은 채로 개발되었습니다. 그 결과, 종종 요구사항이 공급사에 전달된 후에 재작성되어야 했고 이는 시간의 낭비와 증가된 비용을 초래하게 되었습니다.

Embraer의 엔지니어는 모델 기반 설계를 이용하여 Legacy 500 FCS를 위한 저수준 요구사항을 정의하기로 했습니다. 그들은 항공기 역학 및 조종사 입력 모델뿐만 아니라 상세한 FCS 모델을 만들었습니다. 전체 모델은 100만 개 이상의 블록과 수십 개의 컴포넌트, 700개 이상의 입력, 500개의 출력 등으로 구성되었습니다.

그들은 기능 테스트 케이스를 만들어서 모델에 대해 실행하여 고수준 요구사항이 충족되었는지 검증하고 저수준 요구사항을 확인하였습니다. 이어서 Embraer는 작성된 요구사항을 공급사에 전달했고 공급사는 시스템을 구현하기 전에 DO-178 Level A 및 기타 항공 표준에 맞춰 자체 확인을 수행했습니다.

설계

종래 방식에서는 모든 설계 아이디어를 코딩하고 실물 프로토타입에 테스트해야 했습니다. 따라서 각각의 테스트 반복은 프로젝트 개발 시간 및 비용이 늘어남을 의미하기 때문에 매우 제한된 설계 아이디어와 시나리오만을 살펴볼 수 있었습니다.

모델 기반 설계에서는 살펴볼 수 있는 아이디어의 개수가 사실상 무제한입니다. 요구사항, 시스템 컴포넌트, IP, 테스트 시나리오를 모두 모델에 수집할 수 있으며, 모델을 시뮬레이션할 수 있기 때문에 값비싼 하드웨어를 구축하기 전에 미리 설계 문제와 질문을 자세히 살펴볼 수 있습니다. 여러 설계 아이디어를 신속히 검토하고 장단점을 살펴보며, 설계 변경이 시스템에 어떤 영향을 미치는지 알 수 있게 됩니다.

사례 연구: *Ather Energy*

Ather 450 지능형 전기 스쿠터.

> "우리는 유망한 아이디어가 많았지만 소규모 스타트업으로서 이 모두를 테스트할 프로토타입을 제작할 시간, 자금, 인력이 없었습니다. 모델 기반 설계를 통해 시뮬레이션으로 최적의 아이디어를 식별하고 검증하여 더욱 완전한 기능을 갖춘 스쿠터를 더 짧은 시간에 제작할 수 있었습니다."
>
> — Shivaram N.V., *Ather Energy*

방갈로어의 차량 중 70% 가량을 차지하는 500만 대의 이륜 스쿠터 중 대부분은 석유를 사용하며 이로 인해 소음 공해와 CO_2 배출 문제가 심각합니다. 더 깨끗한 대체 에너지에 대한 수요를 충족하기 위해 스타트업 기업인 Ather Energy는 인도 최초의 지능형 전기 스쿠터를 제작했습니다. 4초 내에 0km/h에서 40km/h로 가속할 수 있는 Ather 450은 최고 속력이 80km/h이며 한 번 충전으로 최대 75km까지 주행할 수 있습니다.

450은 기존 시장에 없던 종류의 제품이었기 때문에 팀은 여러 미지의 문제에 직면했습니다. 스쿠터 및 주요 컴포넌트로 이루어진 플랜트 모델을 구축한 후에 그들은 시뮬레이션을 실행하여 여러 가지 주행 및 사용 시나리오를 검토했습니다. 예를 들면 언덕에서, 여러 탑승자를 태운 상황에서, 극한 기온에서, 또는 배터리가 거의 방전된 상황에서 450을 주행하는 시나리오 등을 평가했습니다.

그들은 시뮬레이션 결과로 얻어진 정보를 이용하여 설계를 다듬었습니다. 예를 들면 배터리 용량을 늘리면 주행 거리가 늘지만 비용 및 크기가 커지고 스쿠터의 무게 중심이 변경된다는 사실을 시뮬레이션으로 확인했습니다. 그들은 비용, 크기, 온도 제약 조건을 충족시키면서도 목표 가속도와 주행 거리 요구사항을 충족시키는 모터와 배터리 구성을 찾을 때까지 설계를 정교화했습니다.

스쿠터 자체의 설계 이외에도 배터리 충전, 온도 관리, 기타 주요 기능에 관한 임베디드 제어 알고리즘도 개발했습니다. 상세한 컴포넌트 데이터가 없는 상태에서 이 팀은 경험적 접근법을 이용하여 배터리 셀을 모델링했습니다. 그들은 다양한 온도와 충전 상태에서 배터리를 테스트했고 측정한 입력-출력 데이터를 이용하여 셀의 전기적, 열적 특징을 나타내는 블랙박스 모델을 만들었습니다.

다음으로 그들은 배터리 충전, 출력 제어 및 온도 제어 알고리즘을 개발했습니다. 그들은 플랜트 모델을 이용한 폐루프 시뮬레이션을 실행하여 제어 설계를 검증했습니다. 그들은 제어기 모델로부터 코드를 생성하여 스쿠터의 ARM® Cortex® 프로세서 또는 충전소의 TI C2000™ 마이크로컨트롤러에 코드를 배포했습니다.

Ather 450은 지금 양산 중이며 첸나이의 31개 충전소와 7개 스테이션과 함께 벵갈룰루에서 최초로 출시되었습니다.

코드 생성

종래의 워크플로에서는 임베디드 코드를 시스템 모델 또는 백지 상태로부터 수작업으로 작성해야 했습니다. 소프트웨어 엔지니어는 제어 시스템 엔지니어가 작성한 사양에 맞춰 제어 알고리즘을 작성합니다. 이 공정의 각각의 단계, 즉 사양의 작성, 수작업 알고리즘 코딩 및 수작업 코드 디버그는 시간이 많이 들고 오류가 발생하기 쉽습니다.

모델 기반 설계를 이용하면 손으로 수천 줄의 코드를 작성하는 대신에 모델로부터 직접 코드를 생성하고, 모델은 소프트웨어 엔지니어와 제어 시스템 엔지니어를 이어 주는 다리 역할을 합니다. 생성된 코드는 신속 프로토타이핑이나 생산에 사용할 수 있습니다.

신속 프로토타이핑은 실시간으로 하드웨어에서 알고리즘을 테스트하고 몇 주가 아니라 몇 분 안에 설계 반복을 수행할 수 있는 빠르고 경제적인 방법입니다. 프로토타입 하드웨어나 생산 ECU를 사용할 수 있습니다. 신속 프로토타이핑 하드웨어와 설계 모델을 통해 여러분은 HIL 테스트나 기타 테스트 및 검증 활동을 수행하여 생산 전에 하드웨어와 소프트웨어 설계를 검증할 수 있습니다.

생산 코드 생성을 통해 여러분의 모델을 양산용 임베디드 시스템에 구현될 실제 코드로 변환하게 됩니다. 생성된 코드는 특정 프로세서 아키텍처에 맞춰 최적화하고 손으로 작성한 레거시 코드와도 통합할 수 있습니다.

사례 연구: *Murata Manufacturing*

리튬 이온 배터리가 장착된 Murata의 플렉시블 3상 에너지 관리 시스템.

> "우리는 프로그래밍 경험이 일천했기 때문에 제어기를 수작업으로 코딩했더라면 훨씬 더 많은 버그가 있었을 것입니다. 우리는 코드를 100% 생성하여 신뢰성을 보장했습니다. 우리가 출력을 확인했을 때 Embedded Coder로 생성한 코드에는 버그가 하나도 없었습니다."
>
> — Dr. Yue Ma, *Murata Manufacturing Co., Ltd.*

Murata Manufacturing은 태양광 패널, 배터리 제어기, 계통연계형 인버터, 지능형 제어 시스템을 종합한 EMS(에너지 관리 시스템)를 개발하고 있습니다. 태양광 패널이 사용자 수요보다 많은 전력을 생성하면 제어 시스템이 초과 에너지를 이용하여 배터리를 충전하거나 다시 전력망으로 공급합니다. 그 반대로, 사용자가 태양광 패널이 생산하는 것보다 많은 전력을 요구하면 제어 시스템이 배터리를 방전시키거나 전력망에서 전력을 얻습니다.

제어 시스템의 개발 시간을 단축시키기 위해 엔지니어링 팀은 모델에서 바로 제어 코드를 생성하기를 원했습니다. 팀에는 단 세 명의 엔지니어가 있었고 프로그래밍 경험도 적었기 때문에 그들은 수작업으로 제어 코드를 작성하고 디버그한다면 시간도 오래 걸리고 품질도 열악할 것이라고 생각했습니다.

그들은 태양광 컨버터, 배터리 DC-DC 컨버터, 3상 계통연계형 인버터 등 주요 시스템 컴포넌트로 된 플랜트 모델을 만들었습니다. 이 팀은 시스템의 제어기를 모델링하고, 제어기와 플랜트로 폐루프 시뮬레이션을 실행했습니다. 추가로 폐루프 시뮬레이션을 수행하여, 정전 및 전력망 위상 불균형 등 비정상 상황에 대한 설계의 응답을 검토했습니다.

그들은 제어기 모델로부터 C 코드를 생성하고 그것을 32비트 마이크로컨트롤러에 배포했습니다. 이 팀은 개루프 테스트를 실행하여 기본 검사를 수행하고 시스템 폐루프 제어기 및 상태 전환을 검증함으로써 마이크로컨트롤러와 EMS 회로를 함께 테스트하여 코드와 양산용 하드웨어를 검증했습니다.

테스트 및 검증

종래의 개발 워크플로에서 테스트 및 검증은 보통 공정의 후반부에 이루어졌기 때문에 설계와 코딩 단계에서 도입된 오류를 식별하고 수정하기가 어려웠습니다.

모델 기반 설계에서는 요구사항 및 사양 모델링부터 설계, 코드 생성, 통합에 이르기까지 개발 주기 내내 테스트와 검증이 진행됩니다. 모델에서 요구사항을 작성하고 이를 설계, 테스트, 코드로 추적할 수 있습니다. 정형 기법을 이용하면 설계가 요구사항에 부합하는지 증명할 수 있습니다. 리포트와 아티팩트를 생성하고 소프트웨어를 기능 안전 표준에 맞춰 인증할 수 있습니다.

사례 연구: *Bombardier Transportation*

독일의 Bombardier 열차.

"종래의 방식과 비교했을 때 우리는 모델 기반 설계를 통해 설계, 현, 테스트, 문서화 반복 작업을 훨씬 줄여서 비용을 45% 절감하고 리드 타임을 35% 단축했습니다. 매우 까다로운 기능이 처음부터 아주 매끄럽게 작동하는 것을 본 고객은 아주 만족했습니다."

— Claes Lindskog,
Bombardier Transportation

경전철, 지하철, 통근 열차, 광역 열차, 기관차, 고속열차의 신규 설계는 차량이 운행되는 도시나 지역의 특수한 수요를 충족해야 합니다. 또한 그러한 차량의 소프트웨어 시스템은 EN 50128 및 EN 50657 등의 산업 표준뿐만 아니라 현지 및 국가 규정에도 부합해야 합니다.

철도 운송 산업에서 시스템 테스트는 완전한 차량을 제작하고 궤도에 올릴 때까지는 수행할 수 없는 경우가 많습니다. Bombardier Transportation의 엔지니어는 MATLAB과 Simulink를 이용한 모델 기반 설계를 채택하여 MITRAC 추진 및 제어 시스템의 리드 타임을 단축하고 개발 비용을 절감했습니다.

모델 기반 설계를 채택하기 전에 Bombardier는 제어 소프트웨어 개발에 종래의 폭포수 워크플로를 따랐습니다. 한 팀이 요구사항과 설계를 담당했고, 그것을 두 번째 팀에 전달하여 전통적인 수작업 코딩을 통해 구현하곤 했습니다. 대부분의 테스트는 HIL 환경에서 수행했고, 이어서 하드웨어/소프트웨어 조합 시스템 테스트를 했으며, 일부 테스트는 열차 자체에 수행해야 했습니다. 이런 공정에서는 후반에 오류가 발견되면 수 주 내지는 수개월까지 이르는 재작업 및 지연으로 이어져 상당한 비용이 소비될 수도 있었습니다.

이제 모델 기반 설계를 채택하였기 때문에 Bombardier의 엔지니어는 제어 모델을 구성하고 변경하여 IBM® Rational® DOORS® 또는 Microsoft® Word 문서에 수집된 고객 정의 요구사항을 충족시킬 수 있습니다. 이후에는 추진 시스템의 전기 하드웨어(플랜트) 모델을 만들고 폐루프 시뮬레이션을 실행하여 요구사항을 검증하고 기능을 점검하며 제어기 성능을 검토합니다. 이 중 한 사례에서는 MATLAB 및 Simulink에서 전기 및 제어 시스템 모델을 만들었습니다. 모델을 시뮬레이션함으로써 그들은 보통 시스템 조합 테스트를 할 때까지는 발견할 수 없었던 전기적 결함을 식별했습니다.

시작하기

여러분의 팀은 모델 기반 설계로의 전환에 대한 이점을 인지하더라도, 동시에 전환에 수반되는 조직적, 물류적, 기술적 위험과 난점에 대해 걱정할 수 있을 것입니다. 이 섹션에서는 모델 기반 설계 채택을 고려하고 있는 엔지니어링 팀이 자주 묻는 질문에 답을 하고 많은 팀들이 전환할 수 있도록 도움을 준 팁 및 모범 사례를 소개합니다.

Q. 모델 기반 설계를 도입하면 엔지니어링 역할에 어떤 영향이 있습니까?
A. 모델 기반 설계는 제어 설계와 소프트웨어 아키텍처 분야의 엔지니어링 전문 지식을 대체하지 않습니다. 모델 기반 설계를 이용하면 제어 엔지니어의 역할은 문서 요구사항을 제공하는 역할에서 모델과 코드 형태로 된 실행 가능한 요구사항을 제공하는 역할로 확대됩니다. 소프트웨어 엔지니어는 응용 프로그램 소프트웨어를 코딩하는 시간을 줄이고 아키텍처를 모델링하고, OS, 기기 드라이버 및 기타 플랫폼 소프트웨어를 코딩하고 시스템 통합을 수행하는데 더 많은 시간을 할애할 수 있게 됩니다. 제어 엔지니어와 소프트웨어 엔지니어는 개발 공정 초기 단계부터 시스템 수준 설계에 영향을 미치게 됩니다.

Q. 기존 코드는 어떻게 됩니까?
A. 기존 코드는 설계의 일부로 편입될 수 있고, 여러분의 시스템 모델은 내부적으로 모델링된 컴포넌트와 레거시 컴포넌트를 모두 포함할 수 있습니다. 이는 시스템 시뮬레이션, 검증, 코드 생성을 계속 수행하면서 레거시 컴포넌트를 단계적으로 도입할 수 있다는 의미입니다.

Q. 권장되는 모델 기반 설계 채택 방법이 있습니까?
A. 새로운 접근 방식과 설계 툴을 사용한다는 것은 항상 위험 요소를 수반합니다. 채택에 성공한 팀들은 점진적으로 모델 기반 설계를 도입하고, 프로젝트가 지연되지 않도록 진행을 돕는 집중 단계를 통해 그러한 위험을 줄였습니다. 조직의 규모에 관계 없이, 모든 조직은 작은 그룹 수준에서 모델 기반 설계를 최초로 채택하기 시작합니다. 빠르게 성공을 거두고 그러한 초기 성공을 기초로 활용할 수 있는 단일 프로젝트부터 시작하는 것이 보통입니다. 경험을 갖춘 후에는 부서 수준에서 모델 기반 설계를 전개함으로써 모든 그룹의 임베디드 시스템 개발에 모델이 중심이 되도록 합니다.

다음 네 가지 모범 사례는 많은 팀에게 도움을 주었습니다.

- 프로젝트의 작은 부분부터 실험해 봅니다. 임베디드 시스템의 새로운 영역을 선택하고 소프트웨어 거동 모델을 구축한 후 모델로부터 코드를 생성하는 것이 좋은 시작 방법입니다. 팀 구성원은 새로운 툴과 기법을 학습하는 최소한의 투자로 이러한 작은 변화를 만들 수 있습니다. 그 결과를 이용하여 다음과 같은 모델 기반 설계의 주요 이점을 보여줄 수 있습니다.
  - 고품질의 코드를 자동으로 생성할 수 있습니다.
  - 코드가 모델의 거동과 일치하게 됩니다.
  - 모델을 시뮬레이션하면 여러분은 알고리즘에 있는 버그를 훨씬 간단하게 고칠 수 있고, 데스크톱에서 C 코드를 테스트하는 것에 비해 더 큰 통찰력을 얻게 됩니다.
- 시스템 수준 시뮬레이션을 추가하여 최초 모델링의 성공을 활용합니다. 이 백서의 앞부분에서 밝혔듯이, 여러분은 시스템 시뮬레이션을 이용하여 요구사항을 확인하고 설계 질문을 살펴보고 조기 테스트 및 검증을 수행할 수 있습니다. 시스템 모델은 충실도가 높을 필요가 없으며 인터페이싱 신호가 올바른 단위를 갖고 올바른 채널에 연결되며, 시스템의 동적 거동을 수집하는 데 충분한 정도로만 자세하면 됩니다. 시뮬레이션 결과를 통해 플랜트와 제어기의 거동 방식을 조기에 볼 수 있습니다.
- 모델을 이용하여 구체적인 설계 문제를 해결합니다. 여러분의 팀은 플랜트, 환경, 알고리즘의 완전한 모델을 개발하지 않고도 목표한 이점을 얻을 수 있습니다. 예를 들면, 액추에이션에 사용되는 솔레노이드의 파라미터를 선택해야 한다고 가정해 봅시다. 여러분의 팀은 솔레노이드를 중심으로 구동기와 작용 대상을 포함하여, 개념적인 '제어 체적'을 그리는 간단한 모델을 개발할 수 있습니다. 여러분의 팀은 다양한 극한 작동 조건을 테스트하고 방정식을 도출하지 않고도 기초 파라미터들을 도출할 수 있습니다. 그리고 이 모델을 저장하여 다른 설계 문제나 미래의 프로젝트에 활용할 수 있습니다.
- 모델 기반 설계의 핵심 요소부터 시작합니다. 모델 기반 설계를 하면 얻는 즉각적인 이점에는 컴포넌트와 시스템 모델을 생성하고 시뮬레이션으로 설계를 테스트 및 확인하며, 프로토타이핑과 테스팅에 사용할 C 코드를 자동으로 생성할 수 있다는 점 등이 있습니다. 추후에는 고급 툴 및 사례를 고려하고 모델링 지침, 자동화된 준수 여부 검사, 요구사항 추적성, 소프트웨어 빌드 자동화 등을 도입할 수 있습니다.

사례 연구: *Danfoss*

The Danfoss VLT® AutomationDrive FC302.

> "우리는 새 엔지니어들을 교육하고 새로운 설계 공정을 채택했음에도 모델 기반 설계를 통해 일정 내에 첫 번째 태양광 인버터 프로젝트를 완료했습니다. 두 번째 프로젝트에서는 개발 시간을 10~15% 단축했습니다."
>
> — Jens Godbersen, Danfoss

제품에 대한 증대되는 수요를 충족하기 위해 Danfoss 전력 전자 그룹은 새로 엔지니어를 채용하고 그때까지 수작업 코딩에 의존해 오던 임베디드 소프트웨어 개발 공정을 재검토했습니다. 종래의 개발 공정과 수작업 코딩으로는 하드웨어 프로토타입과 인증 테스트 때까지도 오류가 검출되지 않고 남아 있었습니다.Danfoss는 새로운 공정이 필요하다는 걸 알고 있었지만 모델 기반 설계를 채택하면 마감 시한을 지키기 어려울 수도 있다는 걱정이 있었습니다. 팀이 새로운 공정에 익숙해지려면 시간이 걸릴 것 같았습니다. 또한 신제품인 태양광 인버터 개발이 이미 시작된 상태였습니다. 모델 기반 설계는 개발 중에 도입되어야 했고 프로젝트 마감일에 영향을 미치지 않아야 했습니다.

MathWorks 컨설턴트와의 협력을 통해 Danfoss는 우선 성공적인 모델 기반 설계 채택을 보장할 수 있는 계획을 수립했습니다. Danfoss의 엔지니어는 MathWorks 엔지니어가 주관하는 Simulink, Stateflow®, Embedded Coder® 에 관한 현장 교육과정에 참석했습니다.

이 팀은 수작업으로 코딩했던 기존의 소프트웨어 컴포넌트를 재구축하는 시범 프로젝트를 완료했습니다. 시범 프로젝트에서는 그들은 모델 기반 설계의 세 가지 핵심 능력인 모델링, 시뮬레이션, 코드 생성에 집중하기로 했습니다. 시범 프로젝트를 마친 후, 이 팀은 새로운 태양광 인버터의 개발 공정을 모델 기반 설계로 완전히 전환했습니다.

MathWorks 컨설턴트는 주간 전화 통화를 통해 가장 좋은 시작 방법에 관한 조언을 하고 모델의 초기 버전에 대한 의견을 제공했으며 팀이 업계 모범 사례를 적용하여 모델 재이용을 극대화하고 생성된 코드의 성능을 개선할 수 있도록 도왔습니다.

이 팀은 일정 내에 개발을 완료했고, 준비 과정에서 광범위한 시뮬레이션을 했기 때문에 테스트와 인증 활동도 매끄럽게 진행되었습니다.

이제 그들은 새로운 워크플로로 성공할 수 있다는 것을 보여 주었고 더 많은 엔지니어들이 조직 내에서 모델 기반 설계에 참여하고 있으며 향후 프로젝트에서 재사용할 수 있는 모델 라이브러리와 지식 기반을 구축했습니다.

요약

요구사항 수집에서 설계, 구현, 테스트에 이르는 개발 과정의 중심에는 시스템 모델이 있습니다. 그것이 바로 모델 기반 설계의 핵심입니다. 시스템 모델을 이용하면 다음과 같은 작업이 가능합니다.

- 설계를 직접 요구사항에 연결
- 공유된 설계 환경에서 협업
- 여러 가정 시나리오의 시뮬레이션
- 시스템 수준 성능의 최적화
- 임베디드 소프트웨어의 코드, 리포트, 문서 자동 생성
- 조기 테스트를 통한 조기 오류 탐지

> "3년 전에 SAIC Motor는 임베디드 제어 소프트웨어 개발에 관한 풍부한 경험이 없었습니다. 검증된 효율적인 개발 방식이기 때문에 우리는 모델 기반 설계를 선택했습니다. 이 방법을 통해 우리 엔지니어 팀은 매우 복잡한 HCU 제어 로직을 개발하고 일정보다 빨리 프로젝트를 완료할 수 있었습니다."
>
> — Jun Zhu, SAIC Motor Corporation

모델 기반 설계를 위한 툴

기본 제품

MATLAB®

데이터 분석, 알고리즘 개발, 수학 모델 생성

Simulink®

임베디드 시스템 모델링 및 시뮬레이션

요구사항 수집 및 관리

Simulink Requirements™

모델, 생성된 코드 및 테스트 케이스에 대한 요구사항을 작성, 관리 및 추적

System Composer™

시스템 및 소프트웨어 아키텍처의 설계 및 분석

설계

Simulink Control Design™

모델 선형화 및 제어 시스템 설계

Stateflow®

상태 머신과 순서도를 사용한 의사 결정 로직의 모델링 및 시뮬레이션

Simscape™

멀티도메인 물리적 시스템의 모델링 및 시뮬레이션

코드 생성

Simulink Coder™

Simulink 및 Stateflow 모델로부터 C와 C++ 코드 생성

Embedded Coder®

임베디드 시스템에 최적화된 C 및 C++ 코드 생성

HDL Coder™

FPGA 및 ASIC 설계용 VHDL 및 Verilog 코드 생성

테스트 및 검증

Simulink Test™

시뮬레이션 기반 테스트 개발, 관리 및 실행

Simulink Check™

스타일 지침과 모델링 표준을 사용한 준수 여부 검증

Simulink Coverage™

모델 및 생성된 코드의 테스트 커버리지 평가

Simulink Real-Time™

실시간 응용 프로그램의 작성, 실행 및 테스트

Polyspace® **제품군**

심각한 런타임 오류의 부재 증명

자세히 알아보기

*mathworks.com*에는 모델 기반 설계를 신속히 시작하는 데 도움이 되는 다양한 자료가 준비되어 있습니다.다음과 같은 자료부터 시작해 보십시오.

대화형 방식 튜토리얼

MATLAB Onramp

Simulink Onramp

Stateflow Onramp

웨비나

신규 사용자를 위한 *Simulink* (54:07)

제어 시스템의 모델 기반 설계 (54:59)

제어 시스템 설계의 속도 및 범위 가속화 (51:03)

태양광 인버터 모델링, 시뮬레이션 및 코드 생성 (45:00)

온사이트 또는 자기 주도형 교육과정

MATLAB Fundamentals

Simulink for System and Algorithm Modeling

Control System Design with MATLAB and Simulink

추가 자료

컨설팅 서비스