

Intro til Model-Based Design i Simulink SEES Workshop

11. november 2020

Det følgende dokument udgør vejledningen til SEES' workshop i Model-Based Design, november 2020. Da workshoppen er rettet mod et bredt spænd af forhåndskendskab til MATLAB/Simulink, indeholder hver delopgave en hoveddel, der er tiltænkt alle, samt en valgfri opgave til de, der anløber workshoppen med et væsentligt forhåndskendskab til Simulink.

En eksemplarisk løsning kan clones fra <https://github.com/CMJensen3141/SimulinkWorkshop>.

1 Byg et Simulink Library

I denne workshops første opgave skal I bygge et Simulink Library, der indeholder en PID-controller givet på nedenstående-form:

$$u = K_p \cdot e + K_i \int e \, dt + K_d \frac{de}{dt}, \quad u_{min} \leq u \leq u_{max} \quad (1)$$

Her er u kontrolsignalet fra controlleren (jeres output), der skal befinde sig mellem u_{min} og u_{max} , mens e er forskellen på den angivne reference og det egentlige feedback. Grænserne bør kunne indstilles fra bibliotets interface¹.

Valgfri opgave: Udbyg jeres PID-controller med *bumpless reference*, dvs. en begrænsning på hastigheden, hvormed referencen kan ændre sig. Denne funktionalitet skal gerne kunne slås til og fra på bibliotets interface.

2 Lav en Test Harness

I skal nu lave en Test Harness til jeres library. Husk, at en Test Harness kun kan knyttes til en Simulink model-instans, ikke direkte til et library. I anbefales at samle både input og output i en Test Sequence og gemme Harnessen eksternt, men dette er ikke obligatorisk. Når I har lavet jeres harness, skal I implementere en test af u_{min} og u_{max} .

Valgfri opgave: Implementer en test af jeres bumpless reference. I må *ikke* teste den hele tiden², da det kun giver mening at tjekke om referencen er bumpless, når funktionaliteten er slået til.

¹Hint: Her er "Saturation Dynamic"en nyttig blok.

²Hint: Find ud af hvordan man laver en "when decomposition".

3 Integrer jeres Test Harness i Test Manager

I skal nu lave en fil i Simulink Test Manager, der kører jeres tests fra den foregående opgave. Lav først en ny Test File, derefter en Test Suite, og inkluder endeligt jeres test fra opgave 2 som en Test Case deri.

4 Lav et parameter-sweep

Endeligt skal I lave et parameter-sweep, hvor I skruer på K_i og holder resten af controllerens parametre faste. I skal derfor lave en ny Test Harness - denne gang anbefales det at I bruger en Signal Editor-blok - og finde ud af at lave flere testscenarier i denne. Find derefter ud af, hvordan man får Simulink Test til at sweepe over alle testscenarier i Signal Editoren.

Valgfri opgave: Find ud af hvordan man optager en baseline-måling, og prøv at få Simulink test til at holde jeres parameter-sweep op mod baselinen. Er der nogle udfordringer med denne type test, som man bør have i baghovedet?

5 Fin

Er I blevet færdige med alt det ovenstående før tid, kan I prøve at autogenerere testdokumentation via Simulink Test. Leg gerne med de forskellige mulige indstillinger og diskutér. Når I også dette, opfordrer vi til fri leg!