

I4.0 PÅ DATAMATIKER- UDDANNELSEN

Inddragelse af Industri 4.0 i programmering på datamatikeruddannelsen

Over den seneste årrække er Internet of Things (IoT) og Industri 4.0 (i4.0) kommet i voksende fokus. Derfor er det yderst relevant, at datamatikeruddannelsens dimittender forberedes til at kunne understøtte aftager-virksomhederne inden for i4.0. Derfor eksperimenter vi med at indføre brug af i4.0-teknologier i programmeringsundervisningen. Eksperimentet antager, at motivation og læring øges, når den studerende arbejder med noget konkret (fysisk). Spørgsmålet er, om den forøgede kompleksitet vil påvirke læringen negativt. De studerende inddeles i en interventionsgruppe, som deltager i forsøget, og en referencegruppe, som deltager i et uændret forløb. Begge grupper testes i forhold til læringsmål før og efter forløbet. De foreløbige resultater viser, at det ikke kan påvises, at interventionsgruppen lærer mindre inden for programmeringsfagets centrale læringsmål end referencegruppen. Endvidere er der indikationer på, at motivation og læring forøges i interventionsgruppen. Derfor overvejes det og diskuteres, hvordan brugen af disse eksempler og teknologier i undervisningen kan inddrages bredere i undervisningen.

FORFATTERE

Mogens Holm Iversen, lektor,
IT-uddannelserne, UCN

Istvan Knoll, lektor,
IT-uddannelserne, UCN

Finn E. Nordbjerg, lektor,
IT-uddannelserne, UCN

INDLEDNING

Internet of Things (IoT) og Industri 4.0 (i4.0) (Kagermann, Wahlster and Helbig, 2013) er begge områder, der er kommet mere og mere i fokus hen over de seneste år (se fx Seidelin, 2020). IoT dækker over, at flere og flere apparater bliver forbundet til internettet. Det være sig forbrugerelektronik, biler mv., men i høj grad også maskiner i produktionsapparatet. Dette er et centralt element i,

hvad der er døbt "den fjerde industrielle revolution" eller blot "Industri 4.0". Dette dækker over integration mellem den digitale og den fysiske verden i fremstillingssektoren. Elementer i produktionsapparatet forbindes, der opsamles data, produktionen automatiseres i højere og højere grad, og data udnyttes til optimering af produktionen (*Hvad er Industri 4.0 - Teknologisk Institut, 2020*). Dette er gældende både i Danmark,

EU og resten af verden. Samtidigt er også flere og flere virksomheder i Nordjylland begyndt at udnytte disse muligheder både i produktionen og i produkterne. Endvidere har en stor del af disse virksomheder store udfordringer med at komme i gang med at udnytte potentialet (Hofman-Bang, 2020),(Hansen and Bøgh, 2019)). Groft generaliseret er deres digitale modenhedsniveau ofte for lavt. For uddannelser, der leverer medarbejdere til disse virksomheder, er det derfor nødvendigt, at dimittenderne fra uddannelserne har kendskab til dette område, således at de kan gå ind og hjælpe virksomhederne i en situation, hvor man enten ønsker at begynde en transformation, er i gang med en – eller allerede navigerer i IoT- og i4.0-universet. Dette gælder i særlig grad for datamatikeruddannelsen, som uddanner til at varetage udvikling af software til erhvervsmæssig anvendelse. Traditionelt har datamatikeruddannelsen fokus på administrativ software, hvilket afspejles i de opgaver, eksempler, cases og projekter, som tages op på uddannelsen. Som beskrevet ovenfor er i4.0 et nyt og vigtigt anvendelsesområde for software. Derfor igangsatte vi i 2019 et eksperiment, hvor vi vil prøve at række i4.0 ind som anvendelsesområde i programmeringsundervisningen. Målet er gennem valg af opgaver, eksempler, cases og projekter at give de studerende indsigt i og erfaring med i4.0-universet. Samtidigt er det vigtigt, at de studerende stadig når de samme læringsmål fastsat i den nationale studieordning.

BAGGRUND

På denne baggrund har vi startet et eksperiment omkring at lade noget af undervisningen i programmeringsfaget foregå på en platform, som er meget anvendt inden for i4.0. Platformen er også udbredt til mange hjemme-IoT-projekter og vil være kendt for mange af de studerende. Den del af i4.0, som belyses,

kan kort beskrives som en decentral computer koblet op til en ekstern enhed (et perifert board), hvorpå der er monteret noget elektronik. Parametre for valg af denne decentrale computer vil typisk være processor-kraft, energiforbrug og pris per enhed. Dette kunne f.eks. pege på anvendelse af en eller anden form for Arduino board (Arduino, 2021) eller andet baseret på denne teknologi. Da formålet med projektet er læring, er den vigtigste udvælgelsesparameter i stedet et board, der kan programmeres med et sprog, som de studerende allerede kender, nemlig Java. De studerende benytter Java gennem et BlueJ-udviklingsmiljø (IDE) (BlueJ, 2021), og en Raspberry Pi (Raspberry Pi, 2021) har computerkraft nok til at køre BlueJ. BlueJ er præinstalleret på Raspberry Pi. Derfor vælges at køre med Raspberry Pi og seneste version (4B), så fokus kan holdes på motivationen, og det at anvende kendt teknologi (Java-programmering) i en ny kontekst. Den typiske kontekst for de studerende er at anvende Java til at løse opgaver inden for salg og udlejning. Disse løses 100 % på den studerendes computer. Ved at benytte Raspberry platformen med tilhørende elektronik tydeliggøres det over for den studerende, at der er andre områder, hvor Java-teknologien kan benyttes. Her er kongs-tanken, at dette belyses ved, at den studerende fysisk skal flytte filer fra sin computer til Raspberry Pi. Der er ikke tale om at benytte en eller anden form for weboverførsel, da denne vurderes til at blive opfattet som magi af den studerende og derfor ikke vil leve op til formålet. Så den studerende skal lave opgaver på sin computer og teste det af i et kontrolleret miljø, for derefter at konfigurere det til at køre på Raspberry Pi, inden koden fysisk flyttes (via en USB-stick) til den indlejrede platform. Typisk, når man arbejder i en indlejret i4.0-verden, så er det forbundet med tid og andre ressourcer at overføre programmerne til

den indlejrede platform. I nogle situationer vil hardwaren heller ikke være designet og tilgængelig på det tidspunkt, hvor softwareudviklingen finder sted. Løsningen på dette har traditionelt været at anvende en eller anden form for simulator og i nyere tid en digital tvilling (Barricelli, Casiraghi and Fogli, 2019). Til dette forsøg er der udviklet en simpel digital tvilling, hvor der er en bit, der styrer om koden skal køre på tvillingen eller på hardwaren.

En række opgaver til programmeringsundervisningen er blevet udviklet. Opgaverne støtter de samme faglige mål (objektinteraktion og loops og collections) som tidligere opgaver, men konteksten er ændret fra det administrative univers til i4.0 universet. Formålet er endvidere at vise de studerende, at programmering ikke kun foregår på deres medbragte lap-top, men også kan køre på andre platforme, fx en indlejret enhed som Raspberry Pi. Programmering er en abstrakt aktivitet, hvor en computer instrueres vha. abstrakte koder. Resultaterne manifesterer sig ofte i form af ændringer i dataværdier (tal og andre symboler) – også meget abstrakt. Mange erfaringer fra programmeringsundervisning viser, at arbejdet med konkrete fysiske enheder er motiverende for mange studerende, fx LEGO Mindstorms forløb (Lykke et al., 2014). Det er derfor vores forventning, at en række studerende vil opleve arbejdet med de fysiske elementer mere motiverende end den traditionelle programmeringsundervisning.

Studieordningen for datamatikeruddannelsen fastlægger en række læringsudbytter, som er udmøntet i konkrete læringsmål i fag- og semesterplaner. På dette sted i semestret arbejdes der med konkrete programmeringsfærdigheder inden for emnerne objektinteraktion samt loops og collections. Introduktion af yderligere materiale (her Raspberry Pi og diverse fysiske komponenter) kan stjæle fokus fra

disse centrale læringsmål. Derfor besluttede vi sideløbende med eksperimentet at undersøge, om de studerende fortsat når de krævede læringsmål. Herudover forventer vi, at de oplever øget motivation og opnår indsigt i i4.0-universet.

Konkret ønsker vi kvantitativt at undersøge, om de studerende, som deltager i forsøget, når mindst de samme læringsmål som de studerende, der deltager i vores normale undervisning inden for de berørte emner. Herudover vil vi kvalitativt undersøge om, de studerende, som deltager i forløbet, oplever øget motivation for programmeringsundervisningen og indsigt i i4.0.

METODE

På datamatikeruddannelsen er de studerende opdelt i hold på 20-30 studerende. Hvert efterår starter vi to danske og to internationale hold, og hvert forår starter vi et dansk hold. De studerende er delt i to grupper (randomiseret): interventionsgruppen: de studerende, som deltager i det nye forløb, og referencegruppen: de studerende, som deltager i det traditionelle forløb. Et hold er altid i samme gruppe.

Resultatet af forsøget undersøges på to måder: kvalitativt og kvantitativt. De studerendes opnåelse af de konkrete faglige læringsmål måles kvantitativt:

For at vurdere de deltagende studerendes udbytte i forhold til de konkrete læringsmål inden for programmeringsfærdigheder, så gennemfører vi to faglige test for begge grupper. I hver test stilles en række små programmeringsopgaver, som de studerende skal løse så mange som muligt af inden for et fastsat tidsrum. Opgavesættets omfang er fastsat ved at lade et antal undervisere løse sættet. Programmering er tæt beslægtet med matematik, og erfaringer herfra viser, at en underviser bør kunne løse et sæt (næsten) fejlfrit på ca. halvdelen af den tid, som de studerende får til sættet.

Det er vigtigt at bemærke, at denne type test kun afprøver konkrete færdigheder, men opnåelse af konkrete programmeringsfærdigheder er også helt centrale læringsudbytter på dette stadie i uddannelsen.

Som nævnt, så tester vi to gange med to forskellige test. Vi vil gennemføre en test inden forsøget og en anden test efter forsøget og derpå sammenligne, hvor meget de to grupper har flyttet sig.

Den første test (baseline) har et dobbelt formål. Vi ønsker at måle, i hvilken grad de enkelte studerende opfylder fagets læringsmål på tidspunktet for forsøgets start. Erfaringsmæssigt kommer en ikke ubetydelig gruppe studerende med en række programmeringsmæssige færdigheder erhvervet på anden uddannelse og/eller gennem hobbyaktiviteter. For at måle effekten af forsøget vil vi gerne kunne regulere for disse evt. forskelle i forkundskaber. Tilsvarende vil baselinetesten gøre det muligt at tage højde for evt. tilfældige udsving mellem grupperne pga. forskellige læringsforudsætninger, forskelligt talent og forskellige undervisere mv. Dette styres gennem en rettenøgle: Der kan i alt opnås 100 point, hvor 70 point er det, som vi forventer, at en studerende uden særlige forudsætninger, som opfylder læringsmålene, vil opnå, mens de sidste 30 point kun gives til løsningselementer, der går ud over det forventelige. Der er ikke en bonus for at slutte før tid, da vi ikke ønsker at give den studerende ekstra stress i testsituationen.

Den anden test (effekttest) tages, efter at de studerende har gennemført Raspberry Pi-undervisningen i klassen. Formålet med testen er at måle, om de studerende har opnået de forventede læringsmål på dette tidspunkt af semestret.

Vi ønsker at undersøge, om de studerende i interventionsgruppen har lært mindst det samme som de studerende i referencegruppen. For

at kunne sammenligne resultaterne af de to test beregner vi forskellen (gain) mellem scoren i baseline- og effekttesten. Da 100 % målopfyldelse i baselinetesten svarer til en score på 70 point, justeres baseline-score med følgende formel: $(\text{baselinescore}/70)*100$, inden gain beregnes. Hermed vil scoren i de to test kunne sammenlignes direkte.

Herefter vil vi sammenligne de to gruppers gennemsnitlige gain. Endvidere vil vi også se de enkelte studerendes progression.

Selve forsøget er tilrettelagt som et forløb, hvor de studerende i referencegruppen gennemfører den sædvanlige undervisning i emnerne objektinteraktion, loops og collections, mens de studerende i interventionsgruppen undervises i disse emner i et nyt forløb, hvor Raspberry Pi og i4.0-universet inddrages.

De to grupper følger samme studieordning og skal opnå samme læringsmål. Referencegruppen følger den for semestret sædvanlige lektionsplan. Undervisningsforløbet i programmeringsfaget i starten af semestret er opbygget med henblik på at etablere et stabilt fundament hos alle studerende, hvor især nybegyndere inden for programmering har brug for at forstå grundelementerne i faget og få øvelse i grundlæggende programmeringsstrukturer. Dette opnås igennem emner, bestående af to til tre dage med en gradvis udforskning af det givne emne – f.eks. "objektinteraktion" eller "collections og loops", efterfulgt af en dag med fordybelse, hvor de studerende først og fremmest løser opgaver. Vi har placeret Raspberry Pi-dagene på disse fordybelsesgange, hvor referencegruppen får en emnedag med Raspberry Pi. Som det fremgår af sammenligningen mellem referencegruppens og interventionsgruppens lektionsplaner, indgår de samme elementer i begge med små variationer i rækkefølge affødt af praktiske forhold på det enkelte semester. Dagene med Raspberry Pi

Tabel 1: Lektionsplaner for reference- og interventionsgruppe. (Tabellen er udarbejdet af forfatterne.)

Session	Referencegruppe	Interventionsgruppe
1	Fundamental building blocks of object-oriented programming	Fundamental building blocks of object-oriented programming
2	Introduction to the class definition, methods, fields	Introduction to the class definition, methods, fields
3	Review on classes and objects	Review on classes and objects
4	Object interaction	Object interaction
5	Object interaction	Object interaction
6	Collections of objects, sweep and search pattern	Raspberry Pi – day (Object interaction, conditional statement)
7	Collections of objects, sweep and search pattern	Object interaction review
8	Collections of objects, sweep and search pattern	Collections of objects, sweep and search pattern
9	Collections of objects, sweep and search pattern	Collections of objects, sweep and search pattern
10	Object interaction, debugger	Collections of objects, sweep and search pattern
11	Documentation, library classes, packages, other types of collections, class interfaces	Raspberry Pi – day, debugger (Collections of objects, sweep and search pattern)
12	Documentation, library classes, packages, other types of collections, class interfaces	Fixed sized collections - arrays
13	Collections of objects, sweep and search pattern	Documentation, library classes, packages, other types of collections, class interfaces
14	Fixed sized collections - arrays	Documentation, library classes, packages, other types of collections, class interfaces

for interventionsgruppen ligger på dage, hvor referencegruppen afslutter et emne med fordybelse, se tabel 1.

I forløbet med Raspberry Pi benyttes samme programmeringssprog (Java) og udviklingsmiljø (BlueJ (BlueJ, 2021)), som de studerende er vant til, og som benyttes i det traditionelle forløb. Dette er valgt for at nedbringe eventuelle negative effekter af at bevæge sig ud af det kendte miljø (den studerendes laptop) og over på det indlejrede miljø (Raspberry Pi). I de undervisningsmoduler, som inddrager Raspberry Pi, udlånes færdige sæt af hardware til de studerende. Vi har under udviklingen af hardwaren brugt læringsmålene i studieordningen, så opgaverne, der knytter sig til disse undervisningsgange, holder sig inden for de læringsmål, der gælder for det pågældende semester. En konsekvens er f.eks., at de studerende

ikke selv skal bygge eller samle udstyret, da det ikke indgår i læringsmålene. Hardwaren består af en Raspberry Pi-enhed, et til formålet fremstillet prototypekort – et hulprint – med et simpelt kredsløb, forbundet af et GPIO-kabel, et skærmerkabel samt parrede periferier som mus og tastatur. De studerende låner en skærm, som de kan tilkoble hardwaren. Prototypekortet omfatter en LED, som er en lysende diode, samt en trykknop. Raspberry Pi-aktiviteterne er placeret på dage, som normalt har repetition og fordybelse som primært formål. Som det fremgår af tabel 1, er læringsmålene for første Raspberry Pi-dag relateret til grundlæggende programstruktur i et objektorienteret program. Der arbejdes med objekter, metodekald på tværs af og internt i objekter, og programmets afvikling styres af if-sætninger. Med Raspberry Pi bliver de studerende instrueret i at

reagere på knappetryk på hardwaren og lade hardwaren reagere på forskellige simple måder. Der inddrages et element af leg, da de studerende opfordres til at udvikle et lille reaktionsspil. På den anden Raspberry Pi-dag er loops og collections i fokus, hvor de studerende med Raspberry Pi skal udvikle en oversætter til Morse-alfabetet med den indbyggede LED som interface. Referencegruppen skal løse de sædvanlige øvelser med søgning i lister med eksempler fra primært forretningsverdenen, mens interventionsgruppen skal bruge samme teknikker til at slå bogstaver op i Morse-alfabetet for at få lampen til at blinke den ønskede Morse-kode.

På datamatikeruddannelsen er programmeringsundervisningen tilrettelagt i dage af seks lektioner. Der er afsat to forskellige dage, hvor de studerende arbejder med et opgavesæt, som kræver, at de

programmatisk interagerer med hardwaren. Når udstyret er samlet og tændt, fungerer knappen og LED'en ikke. Det bliver således en central del af opgaven for den studerende at implementere funktionaliteten. Igennem anvendelsen af det softwareudviklingsmiljø, de studerende allerede er fortrolige med, implementerer de studerende først simpel tænd-sluk-funktionalitet, siden mere komplekse opgaver, herunder et simpelt spil, der måler reaktionsniveauet hos spilleren. Hardwaren kan ses på figur 1 og 2.

Hardwaren suppleres, som tidligere omtalt, af et simulationsprogram, en simpel form for digital tvilling. Denne programmeres gennem det samme IDE, som de studerende allerede benytter og er mere eller mindre fortrolige med (BlueJ). Digitale tvillinger benyttes ofte i i4.0-løsninger, idet de åbner mulighed for at afprøve forskellige løsningsinteraktion med hardwaren (ofte et produktionsanlæg) uden at tilgå hardwaren og derved forstyrre driften af anlægget. Da installation på hardwaren spares, er det også muligt langt hurtigere at afprøve forskellige løsninger. Endvidere giver digitale tvillinger også mulighed for at simulere forskellige ændringer i det fysiske anlæg (simuleringer). I undervisningssammenhæng giver den digitale tvilling mulighed for hurtigt at afprøve programmer, der udvikles som led i opgaveløsningen, inden programmet overføres til hardwaren. Da antallet af hardware-sæt er begrænset, er en gruppe på fem til seks studerende nødt til at deles om et sæt. Med den digitale tvilling har de studerende mulighed for at arbejde selvstændigt og afprøve på egen computer og siden overføre og teste på hardwaren. Herved opleves, hvordan det udviklede program interagerer med hardwaren, så det ses, at LED'en lyser, slukker eller blinker efter det ønskede mønster, når der trykkes på kontakten. Introduktionen af en

Figur 1: Prototypeboardet med monteret diode og knap.

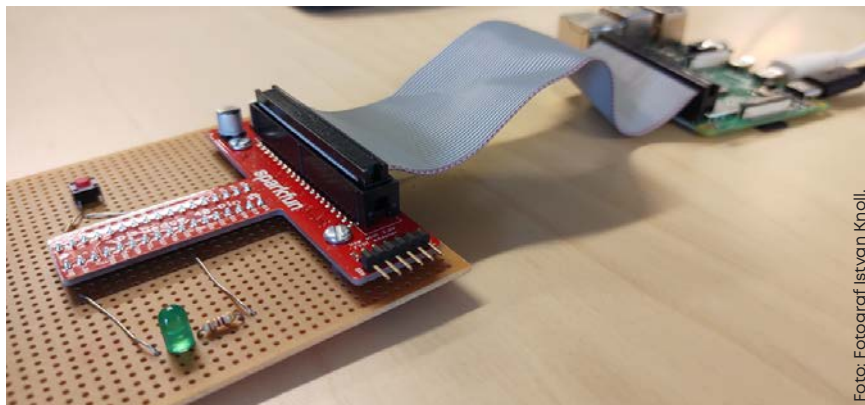


Foto: Fotograf Istvan Knoll.

Figur 2: Den samlede hardwareopstilling med perifere enheder.



Foto: Fotograf Istvan Knoll.

digital tvilling demonstrerer også en central i4.0-teknologi og giver en række muligheder for videre forløb (se "Perspektivering").

Den digitale tvilling er programmeret i det samme udviklingsmiljø, som de studerende anvender til deres undervisning. Dette forventes at give en følelse af genkendelse hos den studerende og derigennem nedsætte antallet af ubekendte, som den studerende skal forholde sig til. Den

digitale tvilling er udviklet i Java under anvendelse af Pi4J biblioteket (Pi4J, 2021). Dette er et standard Java-bibliotek udviklet til at kunne kommunikere med GPIO porten på en Raspberry Pi og supporterer i skrivende stund op til og med Raspberry Pi 4B. Det har den yderligere fordel, at den studerende relativt nemt vil kunne købe egen Raspberry og tilbehør og så udvide programmet, så det nu understøtter den

opsætning som den studerende vil forsøge sig med. I denne sammenhæng betragtes den digitale tvilling som en "blackbox", hvis indre opbygning og funktion er irrelevant. Design, arkitektur og funktionalitet af den digitale tvillings software er dog udviklet under anvendelse af principper og mønstre, som den studerende vil stifte bekendtskab med senere på uddannelsen (se "Perspektivering").

Oplevelse af motivation og kompetencer inden for i4.0 søger vi at vurdere kvalitativt gennem semistrukturerede interviews med hver enkelt studerende, som deltager i forsøget. Interventionsgruppen er blevet interviewet individuelt med henblik på at vurdere, hvordan de har oplevet forløbet (motivation, interesse mv.). Dette er gjort i forbindelse med de midtvejsamtaler, som afholdes ca. midtvejs i semestret. Midtvejsamtaler falder efter ca. 15 undervisningsgange, og det nye forløb afsluttedes efter ca. 10 undervisningsgange (tabel 1), hvilket er passende tid efter, at forløbet er afsluttet. Der har været tid til at reflektere, og de studerende har fået indsigt i, hvordan den traditionelle

undervisning forløber. Der bliver udarbejdet en spørgeramme til midtvejsamtalen, som bliver brugt til at styre indholdet og retningen i de studerendes refleksioner. I dette år blev interventionsgruppen bedt om at besvare følgende spørgsmål:

- 1.) *Hvordan oplever du undervisningen – hvordan synes du, at uddannelsen passer til dig?*
- 2.) *Hvordan oplever du at være i klassen og i den gruppe, du er kommet i?*
- 3.) *Hvordan går det med det enkelte fag – er der noget, du gerne vil have, vi gør mere af eller mindre af?*

I forlængelse af spørgsmål 3 blev interventionsgruppen også bedt om at fortælle om deres oplevelse af at arbejde med Raspberry Pi i forhold til den almindelige undervisning, som de kendte fra de andre undervisningsgange. Også her blev de bedt om at forholde sig til, hvorvidt de gerne ville have mere eller mindre af denne form for opgaver.

RESULTATER

Der er kørt forsøg med Raspberry Pi undervisning i fem klasser af datamatikerstuderende fordelt på tre

studiestarter (efterår 2019, forår 2020 og efterår 2020). Disse udgør interventionsgruppen. Referencegruppen udgøres af fire klasser af datamatikerstuderende (efterår 2019 og efterår 2020). En klasse består af mellem 20 og 30 studerende, og begge grupper indeholder både internationale og danske studerende. Der har i alt været 170 studerende fordelt på 77 i referencegruppen og 93 i interventionsgruppen. Det har været frivilligt for de studerende at deltage i undersøgelsen (men ikke i undervisningen), ligesom de studerende har haft mulighed for at springe fra testene efter første undervisningsgang. Yderligere kunne den studerende være syg, have fortrudt studievalg eller have anden grund til ikke at fuldføre forsøget. Derfor er der for begge grupper kun medtaget testsvar, hvor den studerende har deltaget i både baseline- og effekttesten. Resultaterne er anonymiseret og kan ses af tabel 2 og 3 nedenfor.

For at få et bedre overblik over resultaterne plottes disse ind i et diagram, der er justeret efter, hvor mange procent af svarene som ligger inden for det enkelte interval.

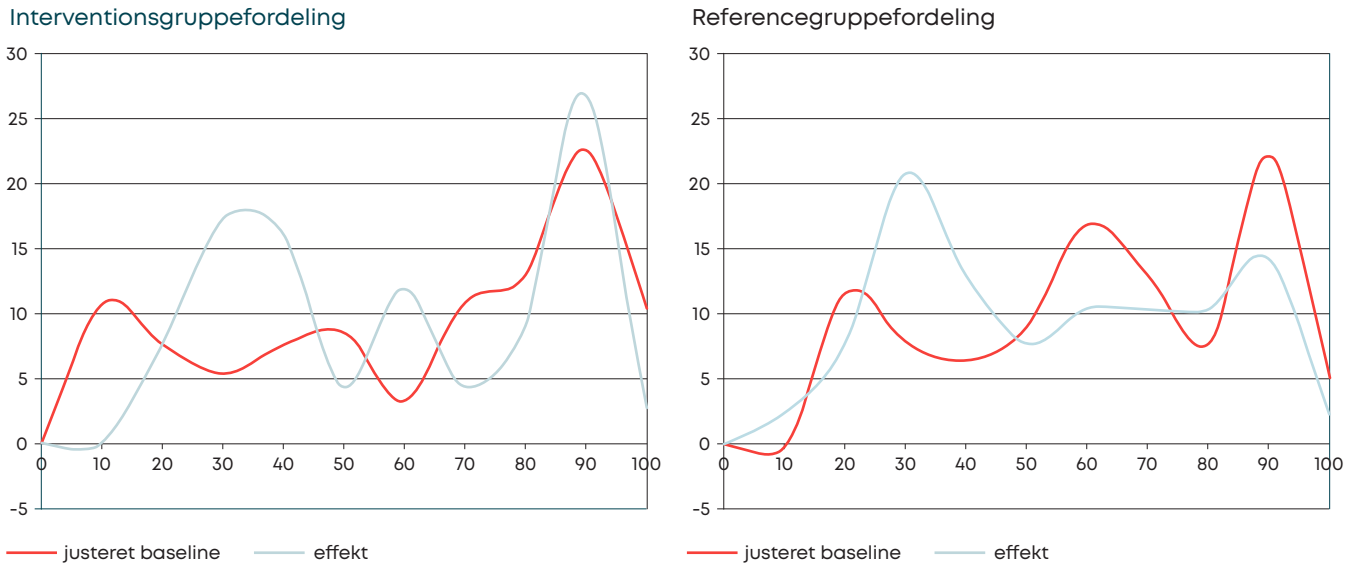
Tabel 2: Lektionsplaner for reference- og interventionsgruppe. (Tabellen er udarbejdet af forfatterne.)

	baseline	adjusted baseline	effekt	gain
antal	77	77	77	77
minimum	14	20	15	-60
maksimum	80	100	100	31
median	47	67	54	-5
gennemsnit	47	66	59	-7
spredning	17,84	24,71	25,06	18,88

Tabel 3: Resultater for interventionsgruppe. (Tabellen er udarbejdet af forfatterne.)

antal	93	93	93	93
minimum	8	11	22	-56,57
maksimum	87	100	100	52
median	52	74,29	65	-3
gennemsnit	46,82	66,36	64,13	-2,23
spredning	21,12	29,43	25,69	18,35

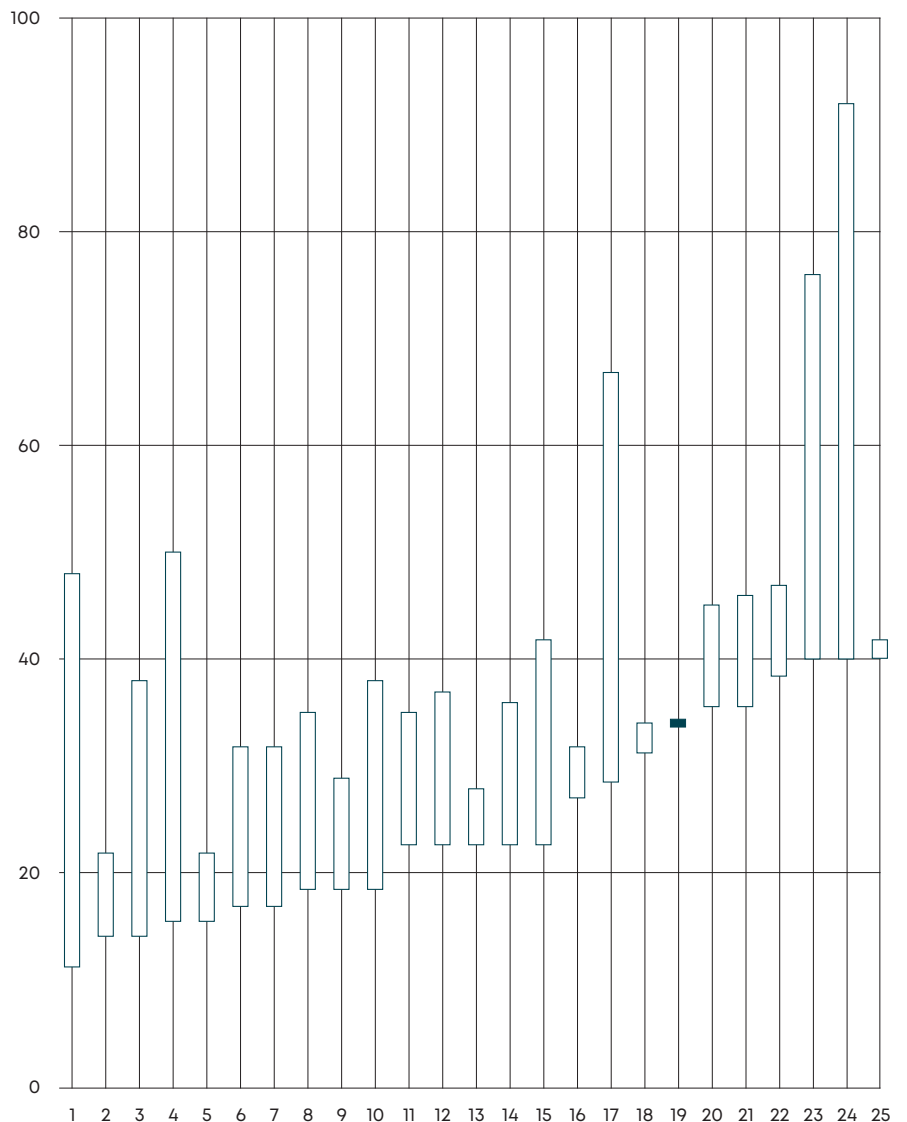
Figur 3: Fordelingen af svarene i grupperne. (Figuren er udarbejdet af forfatterne.)



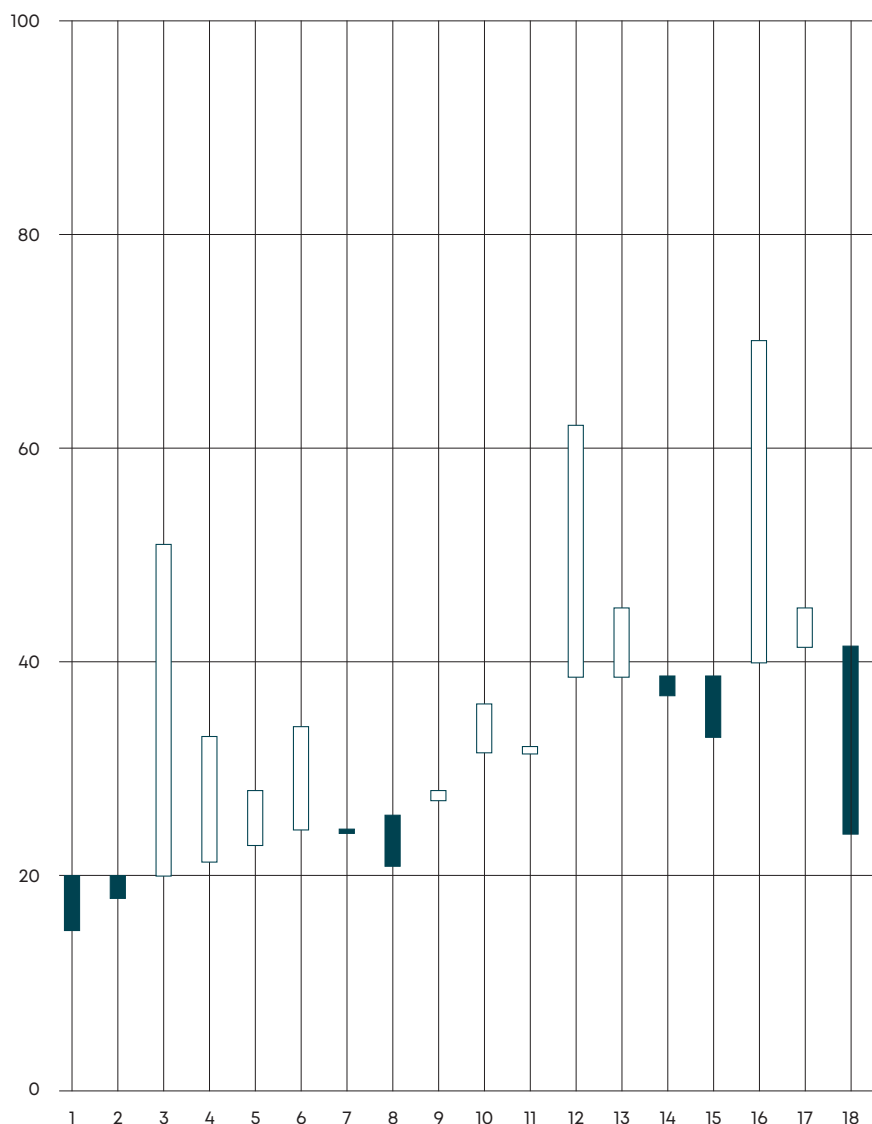
For nemheds skyld er antallet i intervallet placeret ved afslutningen af intervallet. Figuren kan derfor kun bruges til at skabe et groft overblik over fordelingen. Er detaljer nødvendige, skal tabel 2 og 3 bruges. Som tidligere omtalt, er det nødvendigt at justere baselinescoren, for at den er sammenlignelig med effekttestscoren. Dette fordi baselinescoren også skal måle, om der er studerende, der i forvejen har programmeringskundskaber, noget, der er typisk blandt en del af 1. semesters datamatikerstuderende. Fordelingen efter justering i både referencegruppe og interventionsgruppe fremgår af figur 3.

Fordelingen rejser en række spørgsmål omkring læringen hos de studerende fordelt i tre grupper: den svage gruppe (de, der scorer op til 41 point i baselinetesten), middelgruppen (placeret mellem 42 og 80 point) og de øvede/gode (placeret over de 80 point). Bemærk, at disse pointintervaller er for den justerede score i baseline. En simpel måde at visualisere dette på er ved at anvende et diagram kendt fra aktiehandel, nemlig et open/close-diagram. Data sorteres fra lav baselinescore til høj, og diagrammet konfigureres med baseline som åbningspris, 100 som maksimumspris, 0 som minimumspris og effekttest som lukkeprisen.

Figur 4: Den individuelle *gain* sorteret efter baselinescoren for interventionsgruppen. (Figuren er udarbejdet af forfatterne.)



Figur 5: Den individuelle *gain* sorteret efter baselinescoren for referencegruppen. (Figuren er udarbejdet af forfatterne.)



Dermed skabes et diagram, der viser den enkelte studerendes *gain* fra baseline til effekt. På figur 4 og 5 er et udsnit af disse diagrammer vist for de lavestscorende studerende.

Af disse diagrammer (figur 4 og 5) ses, at de studerende, som har den lavest justerede score i baselinetesten, har det højeste *gain* i interventionsgruppen, sammenlignet med de tilsvarende studerende i referencegruppen.

Midtvejssamtalernes svar er overvejende positive, ca. 90 % fremhæver, at det, at de fik et apparat til at fungere, har været motiverende. Af de studerendes

feedback fremgår det endvidere, at ca. 70 % gerne så, at der var flere dage med Raspberry Pi.

Sammenfattende tyder resultaterne af den kvalitative analyse på, at de studerende oplever øget motivation for at arbejde med programmering og har fået et forøget kendskab til i4.0 som anvendelsesområde for softwareudvikling. Samtidigt viser de kvantitative resultater, at de studerende, som har deltaget i forsøget, har nået de konkrete læringsmål i mindst samme grad, som de studerende, der har deltaget i den traditionelle undervisning i de samme emner.

DISKUSSION AF RESULTATERNE

Den overvejende positive modtagelse af Raspberry Pi i undervisningen hos de studerende peger i retning af, at vores antagelse om, at arbejdet med de fysiske elementer højner motivationen for at lære programmering holder. Den noget mere begrænsede interesse (70 % mod 90 %) for endnu flere opgaver af samme slags kan skyldes, at dag nr. 2 anvender samme hardwareløsning og derfor ikke har den samme nyhedsværdi, som den havde på første dag. De studerendes kommentarer peger i retning af, at flere Raspberry Pi-opgaver bør udvikles, så nyhedsværdien kan opretholdes for fortsat at virke motiverende. Disse resultater er dog noget usikre, hvorfor det vil være hensigtsmæssigt at undersøge den evt. motivationsfaktor nærmere gennem flere kvalitative analyser.

Som nævnt havde vi frygtet, at introduktion af en ny platform kunne føre til, at de studerende ikke nåede de obligatoriske læringsmål i samme omfang som gennem det traditionelle forløb, men dette ser ikke ud til at være tilfældet. Referencegruppens *gain* har en middelværdi på -7 og interventionsgruppens på -2. Dvs., at interventionsgruppen tilsyneladende har lært mest (målt ved de beskrevne test). For at udelukke, at forskellen skyldes statistiske tilfældigheder, har vi kigget nærmere på *gain* for både referencegruppen og for interventionsgruppen. I begge tilfælde er *gain* normalfordelt med nogenlunde samme varians. Dvs. der kan udføres en variansanalyse (ANOVA). Denne viser, at de to middelværdier er forskellige med en p-værdi på 0,076. Dvs., sandsynligheden for at begå en fejl ved at konkludere, at der er forskel på de to middelværdier er 7,6 %. Man kan med god statistisk sikkerhed fastslå, at tallene ikke bekræfter vores frygt om, at interventionsgruppen ville opnå en lavere grad af målopfyldelse i forhold til de traditionelle

læringsmål for programmering end referencegruppen. Tværtimod ser der ud til at være svag evidens for, at interventionsgruppen har lært lidt mere.

De individuelle ændringer (figur 4 og 5) tyder endvidere på, at de lavestscorende studerende har et højere udbytte af det nye forløb end referencegruppen. Dette skyldes måske, at for disse studerende er den konkrete oplevelse af at have fysiske apparater i hånden endnu mere fremmende for motivation og læring.

PERSPEKTIVERING

Disse resultater underbygger, at det vil være hensigtsmæssigt at fortsætte udvikling af flere i4.0-forløb baseret på Raspberry Pi-plattformen.

Noget af det, vi forestiller os, er et forløb inden for databaseprogrammering (2. semester), hvor Raspberry Pi'en bruges til dataopsamling og evt. til noget af arbejdet med at rense og aggregere data (Iftikhar, Liu and Nordbjerg, 2015). Dette kan udvides ved at benytte Raspberry Pi som en edge-computer i et distribueret netværk (3. semester). Knudepunkterne i dette netværk vil så være mikroprocessorer, f.eks. baseret på Arduino-plattformen og tilkoblede sensorer. Data opsamles på Pi'en og skal delvis behandles der, før de sendes til en cloud-baseret server.

En anden mulighed vil være i slutningen af 1. semester at se nærmere på den digitale tvilling (se Metode). Gennem det tidligere forløb har den digitale tvilling været en sort kasse, hvis interne opbygning og funktion var skjult for de studerende, men ved at åbne den og undersøge koden vil de studerende lære en del om designmønstre og avancerede objektorienterede programmeringsteknikker, hvilket er en del af de studieordningsfastsatte læringsudbytter for faget, mens de studerende arbejder i i4.0-universet og ser eksempler på

teknikker vedr. simulering samt styring og regulering.

Selve montagen af elektronikken på et element vil være oplagt at lade foregå i et samarbejde med produktionsuddannelserne, der finder ud af, hvorledes den faktiske elektronik skal monteres på det emne, der skal arbejdes med, og IT-teknologuddannelsen, der får lavet programmeringen til sensorer og den generelle styring på Arduino. Dette arbejde startes i regi af det EU-finansierede projekt DigiDemo (DigiDemo, 2020)).

Som det fremgår af interviewresultaterne, vil det virke motiverende, hvis man kunne inddrage nye hardwarekomponenter på en simpel måde. Derfor vil vi undersøge muligheder for en hardwarekonfiguration, hvor de studerende lettere kan koble nye periferier på og selv være med til at bestemme den fysiske funktionalitet af deres devices.

Når vi har opbygget og afprøvet en tilstrækkeligt omfattende portefølje af opgaver, eksempler mv., så skal Raspberry Pi på boglisten og være obligatorisk, så den kan anvendes gennem hele uddannelsen.

Det langsigtede mål for datamatikeruddannelsen er altså, at de studerende fra start har en Raspberry Pi og et board, hvor forskellige sensorer og aktuatorer kan monteres. I løbet af uddannelsen vil de på hvert semester møde et eller flere forløb, hvor der arbejdes med i4.0-relaterede problemstillinger med Raspberry Pi som platform. Dette kan danne basis for, at de studerende i deres specialisering på 4. semester kan arbejde sammen med produktionsvirksomheder – en gruppe af virksomheder, som datamatikeruddannelsen ønsker at udvikle og styrke samarbejdet med. Dette samarbejde kan endvidere danne basis for praktikophold og afsluttende projekter og ultimativt job, hvor de erhvervede i4.0-kompetencer udnyttes til fordel for

erhvervslivet. Derudover ser vi, at IT- og softwareudviklingsvirksomheder interesserer sig mere og mere for produktionsvirksomheder, hvorfor i4.0-kompetencer blandt vore dimittender også vil styrke disse virksomheder.

KONKLUSION

De statistiske resultater viser, at introduktion af i4.0-universet og den nye platform ikke har medført, at de studerende i interventionsgruppen har lært mindre i forhold til de givne læringsmål end studerende i referencegruppen. Dette er et meget vigtigt resultat for os, da introduktion af nye elementer ofte fører til, at andre elementer bliver nedprioriteret. Endvidere tyder midtvejssamtalerne på, at de fleste studerende oplever i4.0-forløbet som ekstra motiverende. Endelig tyder de individuelle gain på, at specielt de studerende i interventionsgruppen, som har scoret lavt i baselinetesten, har haft forøget læring gennem forløbet sammenlignet med tilsvarende studerende i referencegruppen.

Alt i alt er der altså ikke noget i resultaterne, som indikerer, at forløbet har haft negative sideeffekter på de studerendes opnåelse af studieordningens læringsudbytter. Som tidligere nævnt mener vi, at i4.0 vil blive mere og mere udbredt blandt produktionsvirksomheder både lokalt, regionalt, nationalt og globalt. Hermed er og vil i4.0 blive et meget vigtigt anvendelsesområde for software. Derfor skal fremtidens softwareudviklere (bl.a. vores dimittender) have kompetencer inden for dette felt. Da det samtidigt ser ud til at øge de studerendes motivation og læring, så vil vi fortsætte og udvide inddragelsen af i4.0 (jf. forrige afsnit).

Litteraturliste

- Arduino (2021) *Arduino - Home*. Available at: <https://www.arduino.cc/> (Accessed: 6 May 2021).
 - Barricelli, B. R., Casiraghi, E. and Fogli, D. (2019) 'A survey on digital twin: Definitions, characteristics, applications, and design implications', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 7. doi: 10.1109/ACCESS.2019.2953499.
 - BlueJ (2021) *BlueJ*. Available at: <https://bluej.org/> (Accessed: 24 February 2021).
 - DigiDemo (2020) *Digitizing products - Creating demonstrators for Future Education - DigiDemo - Digitizing products*. Available at: <https://www.digidemo-project.eu/> (Accessed: 24 February 2021).
 - Hansen, E. B. and Bøgh, S. (2019) 'Artificial Intelligence and Internet of Things for Small & Medium-Sized Enterprises : A Survey', *Submitted to Journal of Manufacturing Systems*.
 - Hofman-Bang, T. (2020) 'Industriens Fond, Netværks-og Informationsmøde'. Available at: https://www.industriensfond.dk/sites/default/files/praesentation_af_fonden_-_temaendkaldelsesmoede_150120.pdf (Accessed: 2 March 2020).
 - *Hvad er Industri 4.0 - Teknologisk Institut (2020) Teknologisk Institut*. Available at: <https://www.teknologisk.dk/hvad-er-industri-4-0/36707> (Accessed: 16 April 2021).
 - Iftikhar, N., Liu, X. and Nordbjerg, F. E. (2015) 'Relational-Based Sensor Data Cleansing', in *New Trends in Databases and Information Systems*. Springer, pp. 108–118. doi: 10.1007/978-3-319-23201-0_13.
 - Kagermann, H., Wahlster, W. and Helbig, J. (2013) *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 -- Securing the Future of German Manufacturing Industry*. München. Available at: http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf.
 - Lykke, M. et al. (2014) 'Motivating programming students by problem based learning and LEGO robots', in *IEEE Global Engineering Education Conference, EDUCON*. IEEE Computer Society, pp. 544–555. doi: 10.1109/EDUCON.2014.6826146.
 - Pi4J (2021) *The Pi4J Project - Home*. Available at: <https://pi4j.com/1.3/> (Accessed: 24 February 2021).
 - Raspberry Pi (2021) *Teach, learn and make with raspberry pi*. Available at: <https://www.raspberrypi.org/> (Accessed: 24 February 2021).
 - Seidelin, C. A. (2020) *Ny teknologi på dagsordenen i flere SMV'er - DI*. Available at: <https://www.danskindustri.dk/arkiv/analyser/2020/1/ny-teknologi-pa-dagsordenen-i-flere-smver/> (Accessed: 15 April 2021).
- 