

Kod	EAR-1-602-S	Nazwa przedmiotu	Tworzenie aplikacji do symulacji i sterowania procesów dyskretnych	
Rodzaj zajęć			Ćwiczenia laboratoryjne	
Prowadzący przedmiot			Prof. dr hab. inż. Andrzej M. SKULIMOWSKI	
Kierunek			Automatyka i Robotyka	Stopień: I, Rok 3
Ilość pkt. ECTS			3	Email: decyzje@agh.edu.pl

INSTRUKCJA DO ĆWICZEŃ LABORATORYJNYCH



AGH

Katedra Automatyki i Robotyki WEAIiB AGH

Semestr Letni 2019

wersja 1.0, marzec 2019

Spis treści

I.	Harmonogram ćwiczeń laboratoryjnych	3
II.	Ogólne zasady zaliczania ćwiczeń laboratoryjnych	4
2.1.	Ogólne zasady redakcji sprawozdania (wszystkie ćwiczenia).....	4
2.2.	Zasady współpracy podczas zajęć.....	5
III.	INSTRUKCJA DO ĆW.1 i 2	5
3.1.	Ćw. 1 - Symulacja SZD.....	5
3.2.	Ćw. 2 – symulacja sieci antycypacyjnych.....	8
3.3.	Ćw. 1,2- FAQ	10
3.4.	OCENY z Ćwiczeń 1 i 2 :.....	11
IV.	INSTRUKCJA DO ĆW. 3.....	11
4.1.	Wybrane tematy symulacji systemów dyskretnych w ćw. 3.....	12
4.2.	Redakcja sprawozdania z ćwiczenia 3	14
4.2.1.	Poszukiwanie innych rozwiązań, analiza bibliograficzna do sprawozdania, wybór aplikacji do analizy porównawczej.	14
4.2.2.	Opisy aplikacji/rozwiązań	15
4.3.	OCENA z Ćwiczenia 3	15
V.	INSTRUKCJA DO Ćw. 4 i 5.....	15
5.1.	Instrukcja do prezentacji	15
VI.	Bibliografia.....	16

I. Harmonogram ćwiczeń laboratoryjnych

Lp.	Opis tematyki zajęć	Termin realizacji	Uwagi
1.	Omówienie tematyki i harmonogramu ćwiczeń. Metodyka realizacji ćwiczeń laboratoryjnych z zakresu symulacji. Instalacja software'u. Przedstawienie sposobu oceniania i warunków uzyskania zaliczenia z laboratorium. Wybór koordynatora grupy. Omówienie podziału zadań w ćw. 3.	26.02- .05.03.2019	Każda grupa wybiera koordynatora, który jest osobą do kontaktu oraz zbiera i przekazuje prowadzącemu sprawozdania z poszczególnych ćwiczeń
2.	Ćwiczenie 1: Realizacja symulacji prostego systemu zdarzeń dyskretnych ze sterowaniem nadzorczym i przypadkowymi zakłóceniami w Matlab/SimEvents	05-18.03.2019	Do końca 1 tygodnia grupa dostarcza notatkę zbiorczą z planem realizacji ćwiczenia i pytaniami/wątpliwościami. Sprawdzian w 2. tygodniu. Sprawozdanie powinno być dostarczone do końca 2. tygodnia
3.	Ćwiczenie 2: Realizacja symulacji procesu decyzyjnego w prostej sieci antycypacyjnej z elementami optymalizującymi, realizującymi strategię mieszane, z przypadkowymi zakłóceniami w Matlab/SimPy	19-29.03.2019	Do końca 1 tygodnia grupa dostarcza notatkę zbiorczą z planem realizacji ćwiczenia i pytaniami/wątpliwościami. Sprawdzian w 2. tygodniu. Sprawozdanie powinno być dostarczone do końca 2. tygodnia
4.	Ćwiczenie 3: Realizacja symulacji ewolucji wybranego systemu dyskretnego, wybór tematu i oprogramowania z uwzględnieniem preferencji i kompetencji programistycznych studentów (por. lista tematów, realizacja w Matlab/SimEvents/SimPy)	02.04.2019 – 24.05.2019	Do końca 1 tygodnia każda podgrupa dostarcza notatkę zbiorczą z planem realizacji ćwiczenia, opisem metody i pytaniami/wątpliwościami. Kolokwium w 8. tygodniu. Sprawozdanie powinno być dostarczone do końca 8. tygodnia
5.	Ćwiczenie 4: Wizualizacja symulacji i sterowania ewolucją wybranego systemu dyskretnego, wybór optymalnej strategii. Temat jak w ćw. 3. (por. lista tematów, realizacja w Matlab/SimEvents/SimPy, biblioteki graficzne)	28.05.2019 – 07.06.2019	Do końca 1 tygodnia grupa dostarcza notatkę zbiorczą z opisem zakresu wizualizacji, wyborem środowiska i bibliotek graficznych oraz z pytaniami/wątpliwościami. Sprawozdanie i prezentacja powinny być dostarczone do końca 2. tygodnia
6.	4. Seminarium nt. „Metody Komputerowego Wspomagania Decyzji i Modele Prognostyczne” studenckiego Koła Naukowego „Modelowania w Finansach”	do 27.06.2019 (termin do potwierdzenia)	Prezentacje najlepszych aplikacji i wyników symulacji. Nieobowiązkowe wsparcie dla osób wygłaszających referaty z zakresu przedmiotu
7.	Sesja letnia – zaliczenia laboratorium, ocena końcowa	15-30.06.2019	Sesja poprawkowa: 02-15.09.2019

II. Ogólne zasady zaliczania ćwiczeń laboratoryjnych

- ⊙ Obecność jest obowiązkowa i również obowiązkowe jest jej sprawdzanie (listy + ewidencja własna). W przypadku 3 nieobecności nieusprawiedliwionych prowadzący ma obowiązek zgłosić to do Dziekana.
- ⊙ Każda grupa powinna wybrać koordynatora, który będzie zbierać materiały do sprawozdania, zredaguje je i prześle do prowadzącego.
- ⊙ W ramach każdego ćwiczenia studenci realizują swoje indywidualne zadanie (np. podobne, choć różne przykłady SZD, sieci antycypacyjnych) i otrzymują indywidualne oceny. Koordynator zbiera wszystkie zadania i redaguje je w postaci zbiorczego sprawozdania grupy, w którym rozdziały mają przypisanych autorów. W zależności od trudności przykładu do przeliczenia i możliwości oceny indywidualnej pracy, zadania można przydzielać dwóm osobom, na pewno koordynator może otrzymać taką pomoc.
- ⊙ Każde sprawozdanie może być albo przyjęte (ocena co najmniej 4,0) lub odesłane do poprawy, która powinna nastąpić w ciągu 7 dni. Poprawione sprawozdanie jest oceniane ponownie od 0 do 5 i ta ocena jest ostateczna po akceptacji przez prowadzącego przedmiot.
- ⊙ Do czasu uruchomienia strony przedmiotu studenci będą wysyłali ostateczne (po poprawie) sprawozdania z każdego ćwiczenia na adres decyzje@agh.edu.pl.
- ⊙ Zmiana grupy jest możliwa, o ile:
 - liczebność grupy po zmianie nie przekracza 15 osób,
 - ma miejsce do końca drugiego tygodnia semestru, tj. do 8 marca,
 - po zmianie pozostanie nadal 7 grup.

2.1. Ogólne zasady redakcji sprawozdania (wszystkie ćwiczenia)

Sprawozdanie (.docx, .pdf, m-pliki) powinno zawierać rozdziały z zadaniami indywidualnymi z:

- ręcznie przeliczonymi przykładami,
- schematem blokowym lub pseudokodem aplikacji,
- schematem analizowanego systemu,
- kod w m-plikach jako załącznik i wklejony do sprawozdania,
- wyniki obliczeń, ich statystyka i wizualizacja (wykresy w Matlabie),
- krótkie wnioski – mogą być sformułowane wspólnie i zredagowane przez Koordynatora.

Do sprawozdania powinny być dołączone spakowane indywidualne kody. Nazwa każdego katalogu: Symulacja-Cw{X}-Grupa{Y}-Nazwisko_Inicjał{Z}. Jeśli jest tylko jeden plik (tylko Ćw. 1), nie tworzymy katalogów, a konwencja dotyczy nazw m-plików.

Termin oddania sprawozdania to 7 dni po zakończeniu ostatnich zajęć w ramach ćwiczenia (czyli najpóźniej dzień przed kolejnymi zajęciami). Na tych zajęciach lub najpóźniej na kolejnych zostanie wystawiona ocena z tego ćwiczenia, np. 50% ocena z kartkówki, 50% ocena za sprawozdanie (część sporządzona indywidualnie).

W przypadku, gdy sprawozdanie ocenione jest na niedostatecznie (2,5 lub mniej), lub zawiera krytyczne błędy, zostanie odesłane do poprawy. Części niezaakceptowane poprawiane są przez ich autorów. Czas na poprawę to 7 dni – po tym czasie ocena za poprawione sprawozdanie staje się ostateczna. Poprawa obniża ocenę indywidualną z ćwiczenia ze względu na niższą ocenę terminowości.

2.2. Zasady współpracy podczas zajęć

Podczas wykonywania ćwiczeń i sprawozdania studenci mogą (a nawet powinni) konsultować się ze sobą. Części indywidualne sprawozdanie nie muszą być wykonane wyłącznie samodzielnie, jednak należy zaznaczyć, kto pomagał w wykonaniu zadań indywidualnych - taka informacja nie będzie obniżać oceny, ale może ją podwyższyć dla pomagającego. Sprawozdanie powinno zawierać na schemat przydziału zadań (np. różne warianty symulacji, poszczególne elementy wizualizacji z odpowiedzialnymi za nie osobami) oraz notkę koordynatora z opisem wkładu poszczególnych osób. Ten wkład należy wcześniej określić w rozmowie podczas ćwiczeń oraz przedyskutować przy omawianiu sprawozdania na zajęciach.

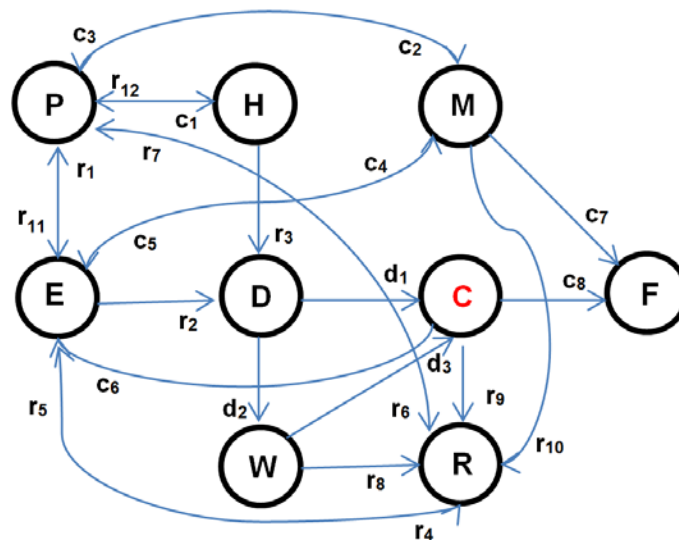
III. INSTRUKCJA DO ĆW.1 i 2

3.1. Ćw. 1 - Symulacja SZD

Na pierwszych zajęciach nt. SZD każdy ze studentów powinien przeliczyć przykład prostego systemu. Te przykłady zostaną przygotowane przez prowadzącego laboratorium przed zajęciami i przekazane studentom na zajęciach. SZD powinien posiadać co najmniej jedno sterowanie i jedno kryterium funkcjonowania. Każdy ze studentów otrzyma inne zadanie, ale będą one mieć podobny stopień trudności. Z reguły będą to różne warianty tego samego, prostego SZD. Przykłady muszą nadawać się do szybkiego przeliczenia na kartce w celu sprawdzenia wyniku obliczeń. Następnie studenci będą tworzyć proste aplikacje w Matlabie z lub bez SimEvents (najlepiej dwie wersje w podgrupach). W tej części zajęć aplikacje mogą być tworzone przez dwie osoby, zwłaszcza gdy są w dwóch wersjach (każdy tworzy jedną, lub dwie wspólnie).

Po przeliczeniu prostego przykładu, na tych samych i na kolejnych zajęciach analizowany będzie przykład o interpretacji praktycznej (działanie robota inspekcyjnego). Na Rys. 1 poniżej przedstawiony jest przykład SZD, który powinien zostać przeanalizowany w trzech różnych wariantach:

- 1- bez zmian,
- 2- dodatkowe losowe przejście z E do D,
- 3- dodatkowe przejście z E do H.



Rys. 1. Przykład SZD do analizy w Ćw.1. Źródło: [15]

W każdym przypadku należy uwzględnić dodatkowe losowe przejście z D do R, które może się uaktywnić tylko w przypadku, gdy poprzednim zdarzeniem było (E,D). Szczegółowe opisy oznaczeń są w publikacjach w LNCS:

https://link.springer.com/chapter/10.1007/978-3-319-51969-2_6,

https://link.springer.com/chapter/10.1007/978-3-319-72329-7_11.

Symulowany jest jeden robot ze stanem początkowym E, który porusza się po okręgu (może zmieniać kierunek ruchu).

Symulacji podlegają:

- Wszystkie przejścia przypadkowe (należy przyjąć założenia o oczekiwanym czasie pojawienia się przejścia, a dla niektórych zdarzeń - również o stopniu uszkodzenia w wyniku takiego zdarzenia),
- Operacje (sterowania) wybierane przez system lub operatora (c_1, c_2, c_3, \dots) – tu nie losujemy, lecz przeglądamy wszystkie możliwe kombinacje takich zdarzeń;
- Operacje będące efektem działania innych robotów (d_1, d_2, d_3, \dots) – symulujemy jednego robota, zatem i te operacje należy symulować poprzez przeglądnięcie kombinacji TAK/NIE dla wszystkich d_i . Do każdego c_i i d_i należy przypisać czas przejścia $t(c_i)$, który może być deterministyczny lub losowy. W pierwszym przypadku musi on być stały we wszystkich przebiegach symulacji, a w drugim stały musi być rozkład.

d) Poziom uszkodzenia (lub sprawność s) robota, wg zasady: $s(0)=1$, $s(n+1)=s(n)u(n)$, gdzie $u(n)$ jest poziomem uszkodzenia w wyniku zdarzenia w chwili poprzedniej. Można dodatkowo przyjąć, że w stanie „R” robota sprawność poprawia się w każdym okresie o 0,1, chyba że wynosi 0,1 lub mniej – wtedy robot jest nienaprawialny i kończymy symulację.

Przed rozpoczęciem symulacji losowane jest miejsce pojawienia się zagrożenia i jego poziom.

- gdy miejsce to odległe jest o mniej niż $\pi/20$ od aktualnego położenia robota, wówczas pierwszym przejściem jest E-D i losowany jest stopień uszkodzenia zależny monotonicznie od poziomu zagrożenia,
- gdy miejsce to odległe jest o mniej niż $\pi/4$ od aktualnego położenia robota, wówczas pierwszym przejściem jest E-H
- miejsce to odległe jest o więcej niż $\pi/4$ od aktualnego położenia robota, wówczas pierwszym przejściem jest E-P

[te zdarzenia można dodatkowo uzależnić od losowanego poziomu zagrożenia]

W przypadku stanu, z którego możliwe są albo przejścia losowe, albo sterowane (lub będące efektem oddziaływań zewnętrznych), konieczne jest przyjęcie dodatkowej reguły, że realizowane jest to zdarzenie, którego przyczyna (sterowanie c_i / operacja d_i lub r_i) wystąpi najwcześniej – wynika to z losowania czasów pojawienia się przyczyn losowych r_i , które porównywane są z deterministycznym lub też losowanym czasem c_i lub d_i ;

c_i ze sobą nie porównujemy, gdyż przeglądane są wszystkie ich kombinacje.

Sugeruję przeliczenie wariantów, w którym

i) porównujemy c_i z d_i tak jak z r_i

oraz

ii) przeglądane są wszystkie kombinacje c_i i d_i (czyli d_i traktujemy jak c_i).

Wszystkie założenia do symulacji proszę szczegółowo opisać w sprawozdaniu (będą inne dla poszczególnych podgrup).

Na wyjściu każdego przebiegu każdy z uczestników eksperymentu otrzymuje wartości dwóch kryteriów (jako funkcje czasu):

1) czas trwania całej operacji od wykrycia zagrożenia do zakończenia akcji (stan F) lub do fatalnego uszkodzenia robota (stan R i sprawność poniżej 0,1)

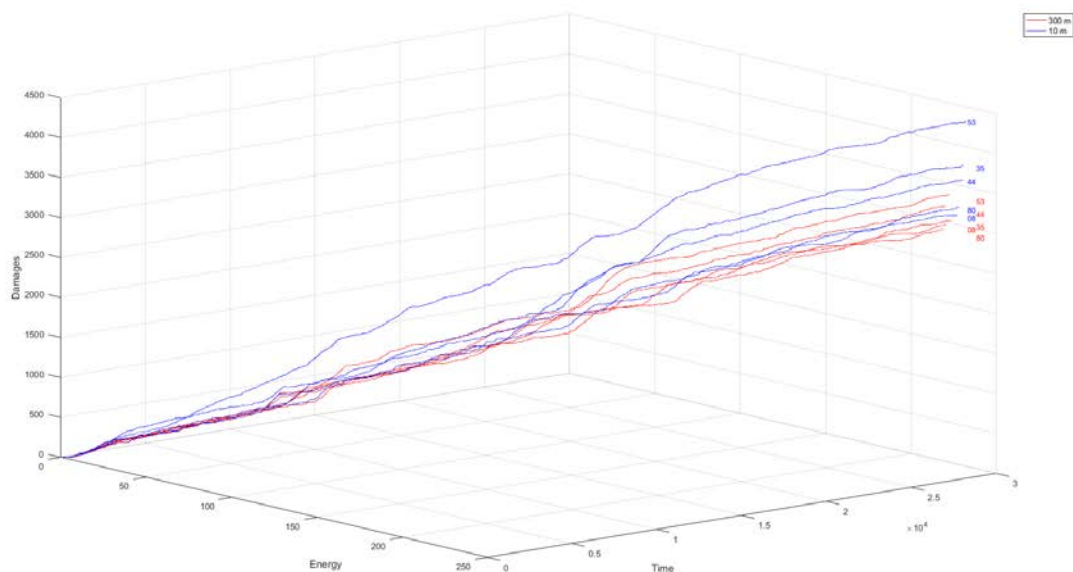
2) stopień uszkodzenia podczas akcji

oraz

3) przebytą odległość lub zużyta energię (informacyjnie).

Obliczana ma być wartość oczekiwana i wariancja tych kryteriów po 50ciu przebiegach z tym samym zagrożeniem.

Przykładowy wykres kryteriów w funkcji czasu ('Time_Energy...') zamieszczony jest niżej– podobne wykresy powinny znaleźć się w sprawozdaniu z Ćw. 1,2, 3 i 4 oraz w prezentacji (Ćw. 5).



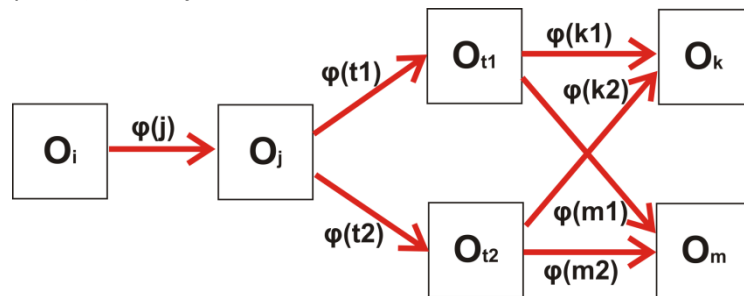
Rys. 2. Przykład wykresu do analizy optymalizacyjnej wyników symulacji w Ćw.1 i wybranych zadaniach w Ćw.3.
Źródło: [15]

Przykład można wykorzystać w dalszych konfiguracjach zdarzeń i założeń zmienionych zgodnie z własnymi pomysłami.

Ponadto przykład powinien być przeliczony dla innej odległości zagrożenia od robota, co spowoduje, że pierwsze zdarzenia będą różne (każdy taki eksperyment może wykonać jedna osoba).

3.2. Ćw. 2 – symulacja sieci antycypacyjnych

Ćwiczenie rozpoczyna się rozwiązaniem prostego zadania z sieci antycypacyjnych, opartego o diagram kauzalny przedstawiony na Rys. 3 niżej.



Rys. 3. Przykład diagramu kauzalnego AN do analizy w Ćw.2. Źródło: [17]

Podczas pierwszych zajęć rozwiązywane będzie zadanie, polegające na uzupełnieniu diagramu wyżej o:

- sprzężenia antycypacyjne (różne dla różnych osób, ale muszą mieć sens – zgodnie z teorią s.a.),
- o wpływ czynnika zewnętrznego na preferencje (przynajmniej na jeden problem decyzyjny, który wpływa na inny problem - w wyniku tego czynnika może zmieniać się np. poziom aspiracji lub sprzężenie antycypacyjne dochodzące do tego decydenta),
- o założenia dotyczące synchronizacji wpływu wcześniejszych problemów na problem zależny od co najmniej dwóch z nich (wszystkie kombinacje OR i AND, wg [17])

oraz o

- wartości kryteriów (przynajmniej dwa w każdym problemie), które otrzymywane są w wyniku podjęcia każdej decyzji.

Należy też przyjąć współczynniki dodatkowej funkcji h (przynajmniej dwukryterialnej), którą uwzględniamy przy rozwiązywaniu sieci zgodnie z Alg. 1 w [17] w sytuacji, gdy od pewnego miejsca brak rozwiązań (łańcuchów antycypacyjnych) spełniających wszystkie warunki sprzężeń.

Pierwszy wariant symulacji sieci antycypacyjnych realizowany na pierwszych zajęciach dotyczyć będą jedynie wszystkich kombinacji decyzji podejmowanych przez wszystkich decydentów w sieci w kolejnych przebiegach symulacji oraz wszystkich (lub wybranych) możliwości zdefiniowania sprzężeń antycypacyjnych w systemie. Następnie symulacja zostanie uzupełniona o czynniki losowe wpływające na parametry symulowanych problemów optymalizacji,

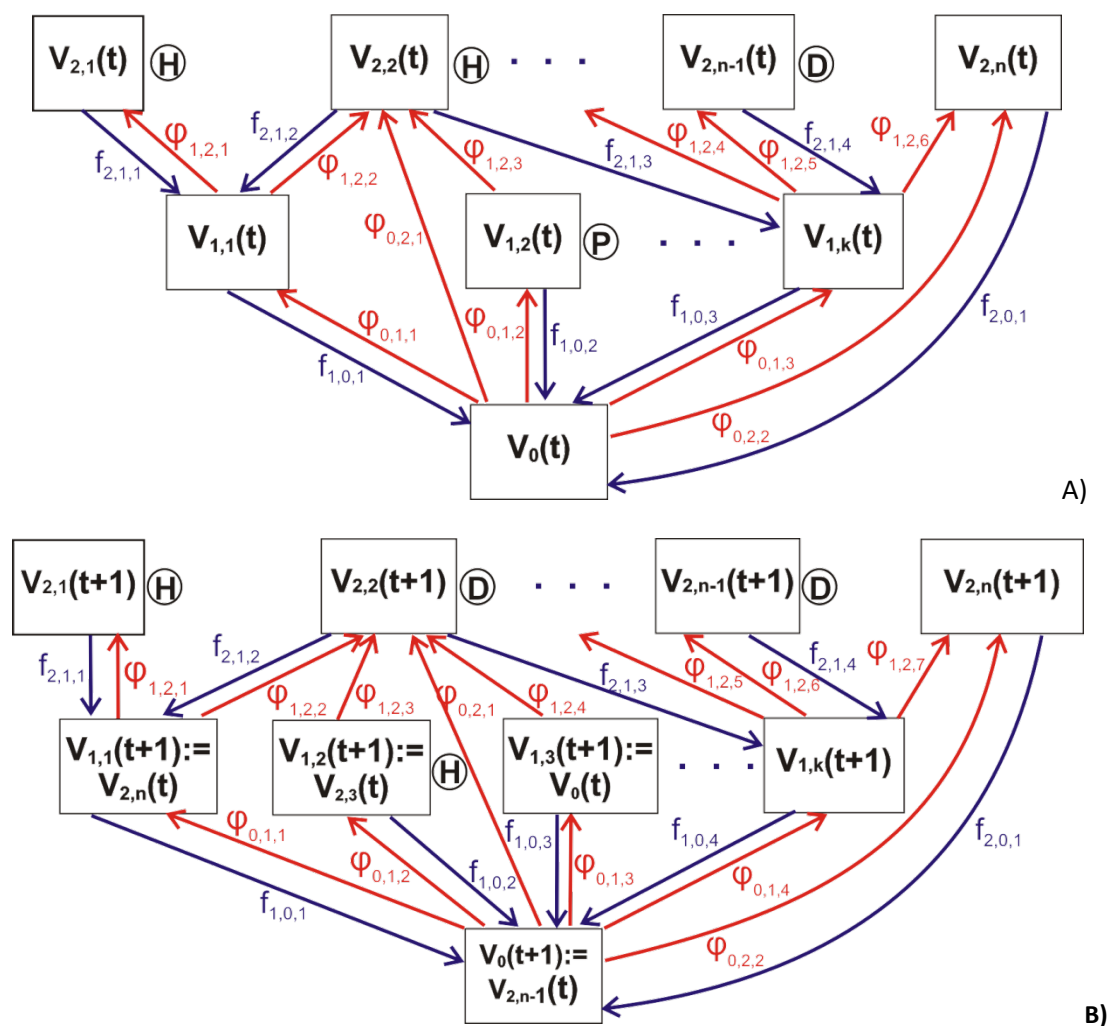
UWAGA: przyjęcie symulacji jako metody porównywania rozwiązań s.a. powoduje, że w pierwszym zadaniu algorytmy 1, 1a i 2 z IJSS [14] oraz [17] nie będą potrzebne.

Wynikiem symulacji są wszystkie łańcuchy

- dopuszczalne
- antycypacyjne

z wartościami kryteriów h , ilość problemów, dla których można wybrać rozwiązania racjonalne (niezdominowane) dla każdego przebiegu i jako średnia po wszystkich przebiegach.

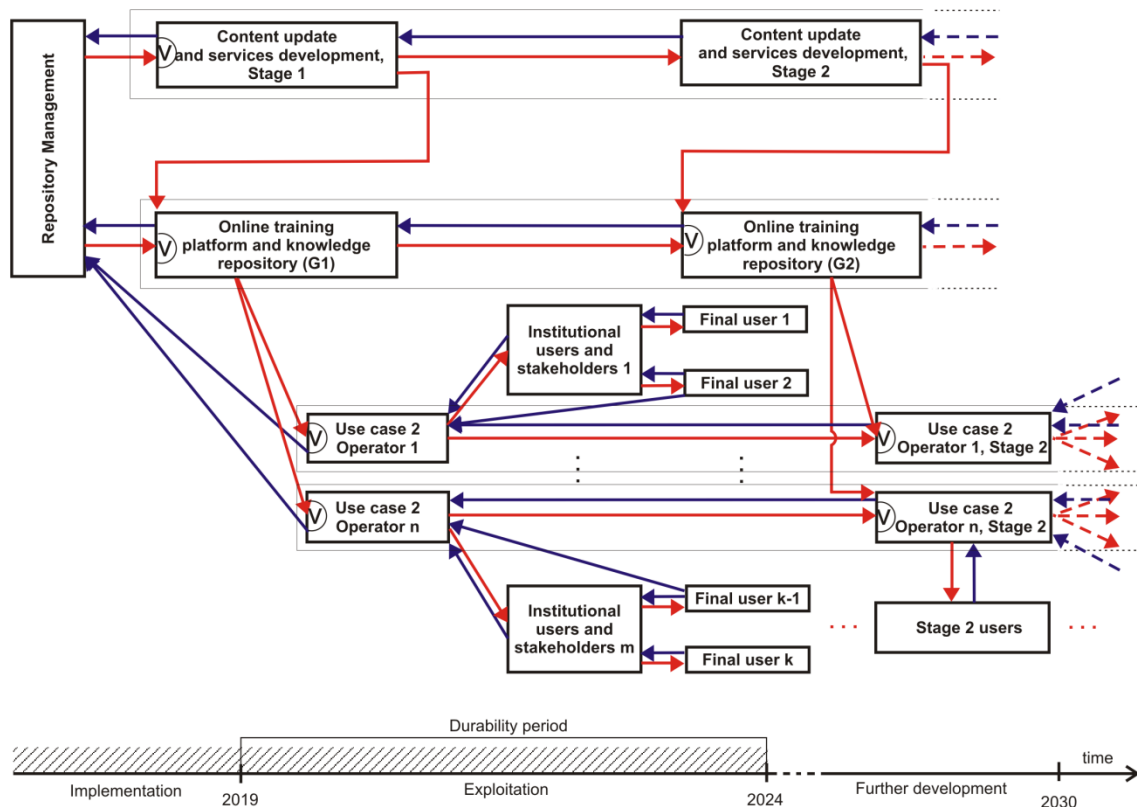
Na drugich zajęciach z s.a. studenci (wszyscy lub większość grup, w przypadku gdy prowadzący opracują własne przykłady) analizować będą przykłady sieci z publikacji nt. robotów antycypacyjnych [15] i [20], również w różnych konfiguracjach.



Rys. 4. Przykłady sieci antycypacyjnych robotów do analizy w Ćw.2. Źródło: [15] i [20].]

Litery w kółkach odnoszą się do stanów SZD z przykładu SZD podanego w rozdz. 3.1.

Drugą opcją przykładu odnoszącego się do rzeczywistego problemu jest analiza przetargu pomiędzy nakładami na rozwój technologiczny i marketing – dotyczy to grupy, która w Ćw. 3 nie będzie tworzyć symulacji roju robotów. Schemat s.a. do tego przykładu podany jest na Rys. 5 niżej.



Rys. 5. Przykład sieci antycypacyjnej związanej z decyzjami inwestycyjnymi do analizy w Ćw.2. Źródło:

3.3. Ćw. 1,2- FAQ

- W jakiej formie ma być przedstawiona symulacja SZD?

ODP.: Najważniejsza jest wizualizacja przejść na grafie stanów SZD:

Po aktywacji animacji dla pojedynczego przebiegu podświetla się stan aktualny i przejście, które do niego prowadzi; cieńszą i niepodświetloną linią zaznaczona jest dotychczasowa trajektoria. Przy krawędziach, przez które trajektoria przeszła wielokrotnie można w nawiasie umieścić krotność, np. (4). W oknie umieszczonym w tym samym polu wykresu pokazywane są aktualne wartości kryteriów i czasu.

Warto zaimplementować możliwość prostego zwalniania/przyspieszania animacji (np. dwa przyciski w polu wykresu, które aktywują funkcję etime z mniejszymi/większymi wartościami)

Oprócz tego należy wykonać wykresy 2D przebiegów [czas (lub tylko nry przejść) – stan], przy czym "oś stanów" powinna wyglądać tak, by stan początkowy („E”) był najniżej, a końcowy („F”) – najwyżej (bezpośrednio pod nim „R”); stany PHD mieszczące powinny być obok siebie. Ponadto wykresy kryteriów jw., mogą być też 2D zblokowane w tym samym oknie z wykresem stanów.

Dla każdego eksperymentu ciekawym wynikiem będzie także % przebiegów zakończonych sukcesem (w stanie „F”).

Proszę uzgodnić podział pracy przy wizualizacji, np. podział na 3-4 podgrupy po 3-4 osoby, tak jak podczas ćwiczeń 1 i 2. Warto przypomnieć sobie AppDesigner lub nauczyć się go od podstaw. Wizualizacja wykonana w ramach ćw. 1 i 2 przyda się także podczas dalszych ćwiczeń.

- czy istnieje jakiś praktyczny przykład podobnej symulacji (napisany w matlabie lub języku programowania), który mógłby Pan udostępnić, aby zweryfikować metodykę i sposób działania (taką symulację piszemy po

raz pierwszy i nie do końca wiemy, jakie podejście do rozwiązania problemu jest oczekiwane)

ODP.: Przykłady są albo obszerne i chronione prawem autorskim, albo są pracami studenckimi, nie zawsze polecanymi jako wzór. Postaram się przygotować wzór aplikacji do Ćw. 3, a na razie proszę skorzystać z instrukcji niżej.

Ogólnie: aplikacja w Matlabie nie różni się wiele od modelu np. rozwiązania równania różnicowego.

Proszę pamiętać o wektoryzacji obliczeń wszędzie tam, gdzie jest to możliwe i sensowne, w przeciwnym wypadku kod wykonywać się może mało efektywnie.

- jak mają być realizowane przejścia pomiędzy kolejnymi stanami (kiedy mają zachodzić)

ODP.: System jest asynchroniczny, zatem w zasadzie liczy się tylko kolejność przejść, którym odpowiadają nry iteracji pętli głównej w aplikacji symulacyjnej (1,2,3 itd.). Jednak zdarzenia przypadkowe, np. wykrycie zagrożenia, mają przypisany określony oczekiwany czas wystąpienia (tj. można im też przypisać czas w postaci nr przejścia, ale ten sposób będzie mniej czytelny), zatem należy przekształcić (zdyskretyzować) założony rozkład prawdopodobieństwa czasu wystąpienia zdarzenia przypadkowego do rozkładu binarnego na odcinku czasu $[n, n+1)$ dla każdego n z okresu symulacji

[Uwaga: generator liczb losowych wywołany w pętli bardzo spowalnia działanie Matlab. Lepiej obliczyć wartości prawdopodobieństw dla wszystkich (z pominięciem dyskretyzacji), a potem wszystkie iteracje działają deterministycznie]

- jakie liczby mają być w wektorach c, r, d (Ćw.1)

ODP.: Proszę przyjąć sensowne (wg własnego uznania i zgodnie z instrukcją podczas ostatniego laboratorium i opisem niżej) wartości parametrów, oczywiście jako parametry wejściowe funkcji wczytywane z prostego GUI (najlepiej byłoby połączyć takie GUI z animacją przebiegu jw., ale prawdopodobnie wymagałoby to zastosowania appdesignera)

Dodatkowym parametrem wczytywanym w ten sposób będzie ilość przebiegów do eksperymentu

Parametry dla 50-100 przebiegów wykonywanych do obliczeń statystycznych, będą ustalone dla wszystkich przebiegów, ale różna dla różnych podgrup.

3.4. OCENY z Ćwiczeń 1 i 2 :

Ocena sposobu, poprawności i kompletności budowy modelu i realizacji obliczeń symulacyjnych, na podstawie sprawozdania z ćwiczenia – waga 0,1. sprawdzian – waga 0,05. Łączna waga oceny z Ćw. 1 w ocenie z laboratorium: 0,15 (tak samo ćw. 2)

IV. INSTRUKCJA DO ĆW. 3

W Ćwiczeniu 3 obowiązuje podział na 2 podgrupy tematyczne po 7-8 osób każda. W wyjątkowych przypadkach (brak w roku 2019) temat może realizować cała grupa laboratoryjna.

1. Sposób realizacji.

W zależności od wyboru podgrupy, realizacja Ćw. 3 polegać będzie na:

- znalezieniu danych do budowy modelu symulacyjnego
- projekt i wykonanie aplikacji symulacyjnej
- wykonanie eksperymentów obliczeniowych zgodnie z instrukcją
- analiza rezultatów
- redakcja sprawozdania.

4.1. Wybrane tematy symulacji systemów dyskretnych w ćw. 3

Lp.	Temat	Narzędzia, materiały	KN, PI (T/N)	Prowadzący
1.	Symulacja rozwoju technologicznego oraz ewolucji rynków systemów wizyjnych i pozostałych sensorów przestrzennych (w tym: trendy i rynki podstawowych technologii systemów wizyjnych, wyszukiwarek multimedialnych, systemów sensorów dla robotów mobilnych - kamery, lidary, radary itp.)	<i>Ekstrakcja reguł rozwoju rynków i technologii, model oparty o symulację działania tych reguł</i>		Dr inż. P. Pukocz
2.	Symulacja rozwoju technologicznego oraz ewolucji rynków repozytoriów danych, serwerów wiedzy i platform e-learningowych (SWD w nauce, LMS, serwery kursów MOOC)	<i>Model w Matlabie, baza wiedzy w MySQL, bot patentowy, webometria</i>	T	Prof. A. Skulimowski
3.	Budowa modeli prognostycznych opartych o sieci antycypacyjne, przykłady: rozwój baz wiedzy, SWD, IoT, wg http://www.tandfonline.com/doi/abs/10.1080/00207721.2012.670308 (kontynuacja tematu wykonywanego w Kole Naukowym – istnieją materiały do wykorzystania)	C#, Matlab-/ SimPy	T	Dr inż. P. Pukocz
4.	Model symulacyjny systemu socjoekonomicznego (kontynuacja tematu wykonywanego w Kole Naukowym, cała grupa - 15 osób)	<i>Discovering complex system...</i>		Prof. A. Skulimowski
4.1.	a) interfejs graficzny systemu – Etap I: wprowadzanie danych oraz metody łączenia danych ilościowych (trendy, szeregi czasowe) z jakościowymi, Etap II: wizualizacja trajektorii systemu (1 osoba)	konieczna dobra znajomość Java/MySQL		
4.2.	b) badanie przyczynowości metodą Batesa-Grangera (1 osoba)	Matlab, PHP	T	
4.3.	c) konstrukcja macierzy stanu, kalibracja modelu, filtr i predykcja Kalmana (1-2 osoby)	<i>Discovering complex system...</i>	T	
4.4.	d) identyfikacja systemu zdarzeń dyskretnych, model zależności trendów i zdarzeń (2 osoby)	<i>Discovering complex system...</i>	T	
4.5.	e) Symulacja systemu zdarzeń dyskretnych	<i>Można wykorzystać wyniki Ćw. 1</i>		
4.6.	f) integracja danych w bazie wiedzy do modelu (koordynator - 1 osoba)	Testy w Matlabie/PHP	T	
5.	Budowa prognoz w oparciu o symulacje złożonych systemów dyskretnych¹	nieaktualne w 2019		Prof. A. Skulimowski
5.1.	Metody trend impact i cross-impact (1 osoba)	Gordon i in.	T	
5.2.	Klasteryzacja elementarnych przebiegów symulacji procesów ewolucji systemów ekonomiczno-społecznych (fuzzy k-means i in., 1 osoba)	np. Dziechciarz, Ekonometria, 2002		
5.3.	Analiza korelacji pomiędzy trendami i klastrami scenariuszy (1 osoba)	<i>Matlab Optimization Tbx, Statistics Toolbox</i>		

¹ Temat pozostaje w programie na rok przyszły

Lp.	Temat	Narzędzia, materiały	KN, PI (T/N)	Prowadzący
5.4.	Zastosowanie metod zdarzeń dyskretnych do konstrukcji scenariuszy technologicznych	wg instrukcji	T	
6.	Symulacja ewolucji software'u: generowanie kolejnych wersji, identyfikacja i symulacja reguł biznesowych, symulacja wpływu trendów rynkowych i technologicznych. Przykłady zastosowań: - CMS (kontynuacja wcześniejszego projektu), - systemy wspomagania decyzji w medycynie (medyczne DSS, zwłaszcza m-health)	Publikacje: Skulimowski, Badecka (KICSS'2015, CONFENIS 2016). Matlab/Simulink/SimEvents		Mgr I. Badecka
7.	Model symulacyjny grupy współpracujących robotów antycypacyjnych (dwie podgrupy po 7-8 osób): a) likwidacja zagrożeń b) unikanie kolizji na skrzyżowaniach	Matlab/Simulink/SimEvents		Mgr inż. M. Drapała, prof. A. Skulimowski
8.	Symulacja zakładu produkcyjnego (parametry techniczne, zdarzenia zewnętrzne, decyzje menedżerskie)	SimPy, Java		Dr inż. P. Pukocz
9.	Model symulacyjny autonomicznego robota inspekcyjnego działającego w różnych środowiskach:	Gazebo i in.		Dr inż. P. Pukocz
10.	Symulacja dyfuzji i oddziaływania webowej aplikacji społecznej	Matlab/Simulink/SimPy		Prof. A. Skulimowski
11.	Symulacyjna metoda agregacji prognoz o różnych horyzontach czasowych w oparciu o konwersję prognoz do postaci standardowej. Realizacja obejmuje bazę danych prognoz, interakcyjną metodę oceny prognoz z przeszłości. Szczegółowe analizy przykładów symulacyjnych prognoz rozwoju a) Internetu rzeczy (IoT) b) Zaawansowanej sztucznej inteligencji (możliwa własna propozycja innego przykładu)	Matlab/PHP, konieczna dobra znajomość statystyki	T/T	Prof. A. Skulimowski
12.	Projekt metodyczny: opracowanie przeglądu metod budowy sieci kauzalnych dla celów symulacji, implementacja 3-4 wybranych metod i tworzenia na ich podstawie modelu do prostych prognoz symulacyjnych (metody trendu, ARIMA), umieszczenie go na stronie (w tym layout strony) www.decyzje.agh.edu.pl , przetestowanie interfejsu i aplikacji	konieczna dobra znajomość PHP7, CMS (preferowany Drupal). Praca w oparciu o wcześniejszą wersję systemu (do 7 osób)	T	Dr inż. P. Pukocz
13.	Symulacja rozwoju technologicznego oraz ewolucji rynków i w zakresie BCI (symulacyjna ekstrapolacja trendów patentowych, technologicznych i bibliometrycznych), ekstrakcja reguł rozwoju rynków i technologii	Model oparty o symulację działania znalezionych reguł. Przydatne mogą być bazy: patentowy i bibliograficzny	T	Mgr I. Badecka
14.	Symulacja użytkowników inteligentnego repozytorium cyfrowego: rejestrowane są zdarzenia po stronie użytkownika i systemu, w tym zapytania do systemu i jego odpowiedzi. Na podstawie zbioru uczącego generowane jest zachowanie systemu i reakcje użytkowników, w tym stymulacja (po stronie systemu) i wzrost kreatywności użytkowników	Konieczna dobra znajomość JScript, JSON; pożądana: MongoDB, Elasticsearch		Prof. A.M. Skulimowski
<p>Symulacja ewolucji technologicznej i rozwoju rynków IT: – w oparciu o dane historyczne, wpływ globalnych trendów technologicznych na rozwój gospodarczy i inne technologie, symulacja wzajemnego wpływu trendów i ograniczeń ekonomicznych, symulacyjne wyznaczanie scenariuszy rozwoju technologicznego i ekonomicznego, prognozy w scenariuszach.²</p>				

² Temat aktualny od 2020

Szczegółowe instrukcje realizacji tych tematów studenci otrzymają od prowadzących wskazanych w tabeli wyżej.

4.2.Redakcja sprawozdania z ćwiczenia 3

Ze względu na większą ilość zajęć, które będą poświęcone na to ćwiczenie, sprawozdanie to zawierać będzie dodatkowe elementy.

4.2.1. Poszukiwanie innych rozwiązań, analiza bibliograficzna do sprawozdania, wybór aplikacji do analizy porównawczej.

A. Każdy ze studentów powinien dokonać przeglądu dostępnych rozwiązań problemu, ustalić ich ilość i przydatność.

Dokumentacja wyników poszukiwania (w sprawozdaniu z Ćw.3):

<zapytanie, wyszukiwarka, czas> (to w tabeli) -> ilość wszystkich odpowiedzi -> ilość odpowiedzi trafnych (wartości Precision, Recall, F-measure)-> ilość systemów wybranych do tabeli

Uwaga:

Precision: = ilość wyszukanych elementów spełniających warunki poszukiwania (*true positives*) / ilość wszystkich znalezionych elementów w danym etapie poszukiwania (*true positives + false positives*)

Recall: = ilość wyszukanych elementów spełniających warunki poszukiwania (*true positives*)/ilość wszystkich elementów (znalezionych lub nie), które są zgodne z celem poszukiwania (*true positives + false negatives*)

F-measure [średnia harmoniczna Precision i Recall::= $2/(1/Precision + 1/Recall)$ = $2*(true\ positives)/[2*(true\ positives) + (false\ negatives) + (false\ positives)]$

Dodatkowo premiowany będzie nadobowiązkowy opis całego przebiegu poszukiwania zgodnie z modelem tzw. procesu kreatywnego (wg

https://www.researchgate.net/publication/221274898_Freedom_of_Choice_and_Creativity_in_Multicriteria_Decision_Making)

uzupełniony o dodatkowe informacje wg schematu:

: -> wyniki <ilość, oszacowanie wartości precision/recall, <side effects>, ocena> -> <określenie cech dla 'false positives' (jaki dodatkowy filtr jest potrzebny?)> i dla 'false negatives' (dlaczego nie były szukane? lub dlaczego nie zostały wyszukane?)> ->(na tej podstawie)-><wskazanie sposobu modyfikacji zapytania> -> <nowe zapytanie, baza, czas> wg jw.

W części podgrup będzie można skorzystać z wyników przeglądu aplikacji do symulacji lub prognoz wykonanego wcześniej w Kole Naukowym Modelowania w Finansach, dla wcześniej wybranych i przeanalizowanych aplikacji konieczne będzie tylko uzupełnienie i aktualizacja informacji, m.in. o wersjach aplikacji.

B. Wybór aplikacji do analizy porównawczej.

Udział biorą wszyscy członkowie grupy. Każdy ze studentów wpisuje do tabeli opisy znalezionych aplikacji symulacyjnych lub wyników symulacji, podgrupa analizuje dokładnie co najmniej 5 rozwiązań problemu.

4.2.2. Opisy aplikacji/rozwiązań

Opisy te należy umieścić w tabeli o formacie wg wzoru 'aplikacje-symulacja2019.xls' (zał. do instrukcji). Nazw i kolejności pól nie wolno zmieniać, natomiast w razie potrzeby można dodawać własne.

[w ramach Ćw.3 jedna z grup stworzy makro w VBA do wielokryterialnego wyboru aplikacji symulacyjnych]

4.3. OCENA z Ćwiczenia 3

Ocena sposobu, poprawności i kompletności realizacji obliczeń symulacyjnych, na podstawie sprawozdania z ćwiczenia – waga 0,4. Kolokwium – waga 0,1. łączna waga oceny z Ćw. 3: 0,5 w ocenie z laboratorium

V. INSTRUKCJA DO Ćw. 4 i 5

5.1. Instrukcja do prezentacji

1. Zawartość prezentacji: wyniki Ćw. 3 i wizualizacja z Ćw. 4. Przede wszystkim tabele (+screeny), animacje dla przykładów obliczeniowych:

- sformułowanie i krótka charakterystyka rozwiązywanego problemu,
- omówienie zaimplementowanej aplikacji, uzasadnienie wyboru sposobu rozwiązania,- krótka charakterystyka problemów, które aplikacja rozwiązuje, historia wersji, ze wskazaniem najważniejszych zmian,
- sposoby analizy i zakres danych, stosowane metody - software, metody AI, optymalizacji, prognozowania itp.,
- porównanie z innymi aplikacjami analizowanymi w ramach grupy, wybranymi do eksperymentów obliczeniowych oraz z kilkoma niewybranymi do eksperymentów, ze wskazaniem ich zalet i wad,
- przykład obliczeniowy z wykorzystaniem własnej aplikacji,
- wnioski z porównania aplikacji i inne zagadnienia z analizy wyników symulacji,

2. **Format.** Prezentacje należy dostarczyć wyłącznie w formacie .pptx lub .ppt. Wszystkie wzory i tabele muszą być zredagowane we wbudowanym edytorze . Należy również dostarczyć pliki źródłowe wszystkich diagramów i rysunków.

3. Do prezentacji należy wykonać streszczenie na ok. 0,5 – 1 strony i dostarczyć je przed przedstawieniem prezentacji, najpóźniej do 1 czerwca 2019. Streszczenia będą udostępnione studentom z wszystkich grup.

4. Czas na prezentację indywidualnego tematu: 12-15 minut (15-18 ekranów na podgrupę w formacie .ppt lub .pptx), w tym dyskusja..

5. Prezentacje należy przysłać do dnia 07 czerwca 2018 na adres decyzje@agh.edu.pl. Prezentacje nie spełniające ww. warunków nie będą dopuszczane do przedstawienia podczas zajęć.

6. OCENA z Prezentacji: Ocena sposobu przedstawienia tematu, aspekty merytoryczne i redakcyjne, dyskusja. Oceniają zarówno prowadzący, jak i słuchacze.

Łączna waga oceny z prezentacji: 0,1 w ocenie z laboratorium i OK.

VI. Bibliografia

1. Abu-Taieh, Evon M.O., El Sheikh, Asim, Abdel Rahman (red., 2010). Handbook of Research on Discrete Event Simulation Environments: Technologies and Applications IGI Global, Hershey, s.585.
2. Birta, Louis G.; Arbez, Gilbert (2013). Modelling and Simulation: Exploring Dynamic System Behaviour, 2nd Edition, Springer-Verlag, London, seria: Simulation Foundations, Methods and Applications, s.437, <https://doi.org/10.1007/978-1-4471-2783-3> (dostęp przez konto BG AGH).
3. Brailsford, Sally; Churilov, Leonid; Dangerfield, Brian (red., 2014). Discrete-event simulation and system dynamics for management decision making, Wiley Series in Operations Research and Management Science, Wiley, s. 342.
4. Maciąg A., Pietroń R., Kukla S. (2012). Prognozowanie i symulacja w przedsiębiorstwie, Wyd. PWE, Warszawa.
5. Andrzej M. J. SKULIMOWSKI (1985). Solving Vector Optimization Problems via Multilevel Analysis of Foreseen Consequences. Found. Control Engrg., 10, No. 1, 25-38. [w researchgate.net]
6. Andrzej M.J. SKULIMOWSKI (1991). Optimal Control of a Class of Asynchronous Discrete-Event Systems. IFAC Papers Online, 27, Automatic Control in the Service of Mankind. Proceedings of the 11th IFAC World Congress, Tallinn (Estonia), August 1990, Vol.3, s. 489-495; Pergamon Press, London. [ScienceDirect, Researchgate.net]
7. Andrzej M.J. SKULIMOWSKI (1994). Optimizing the structure of a partitioned population. In: System modelling and optimization: proceedings of the 16th IFIP-TC7 conference : Compiègne, France, July 5–9, 1993, eds. J. Henry, J.-P. Yvon. London: Springer-Verlag, 1994. Lecture Notes in Control and Information Sciences; LNCIS 197. Springer-Verlag, Berlin Heidelberg New York, s. 771–782. [link.springer.com, Researchgate.net]
8. Andrzej M.J. SKULIMOWSKI (1996). Decision Support Systems Based on Reference Sets, Wydawnictwo AGH, seria: Monografie, nr 40, https://www.researchgate.net/publication/236153959_Decision_Support_Systems_Based_on_Reference_Sets
9. Andrzej M. J. SKULIMOWSKI (2011). Freedom of choice and creativity in multicriteria decision making. In: Knowledge, Information, and Creativity Support Systems : 5th international conference, KICSS 2010, Chiang Mai, Thailand, November 25–27, 2010 : revised selected papers, eds. Thanaruk Theeramunkong [et al.]. Berlin ; Heidelberg : Springer-Verlag. Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence, 6746, s. 190–203. [link.springer.com, Researchgate.net]
10. Andrzej M. J. SKULIMOWSKI (2012). Discovering complex system dynamics with intelligent data retrieval tools. In: Intelligent Science and Intelligent Data Engineering: second Sino-foreign-interchange workshop, IScIDE 2011 : Xi'an, China, October 23–25, 2011: revised selected papers, red. Yanning Zhang [et al.]. Berlin, Heidelberg: Springer-Verlag, Lecture Notes in Computer Science 7202, s. 614–626, http://link.springer.com/chapter/10.1007%2F978-3-642-31919-8_78
11. Andrzej M.J. Skulimowski (2013). Exploring the Future with Anticipatory Networks. In: Physics, Computation, and the Mind - Advances and Challenges at Interfaces: Proc. of the 12th Granada Seminar on Computational and Statistical Physics, 17–21.09.2012, La Herradura, Spain. Pedro L. Garrido, Joaquín Marro, Joaquín J. Torres, J.M. Cortés (Eds.), American Institute of Physics, AIP Conf. Proc.1510, s. 224-233. <http://scitation.aip.org/proceedings/volume.jsp>, https://www.researchgate.net/Andrzej_Skulimowski
12. Andrzej M.J. SKULIMOWSKI (2014a). An insight into the evolution of intelligent information processing technologies until 2025 In: IISA 2014: 5th International Conference on Information, Intelligence, Systems and Applications: 7–9 July 2014, Chania, Crete, Greece. IEEE, Piscataway, s. 343–348, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6878810>
13. Andrzej M.J. SKULIMOWSKI (2014b). Applying anticipatory networks to scenario planning and backcasting in technological foresight. Paper presented at the 5th FTA Conference, Brussels, November 27-28, 2014, s.10, <https://ec.europa.eu/jrc/en/event/site/fta2014/programme/panel-session/cutting-edge-fta-approaches#28-Nov>
14. Andrzej M.J. SKULIMOWSKI (2014c). Anticipatory network models of multicriteria decision-making processes, International Journal of Systems Science, Vol. 45 (1), 39-59, DOI:10.1080/00207721.2012.670308, [http://www.tandfonline.com/doi/full/10.1080/00207721.2012.670308, dostęp z konta BG AGH]
15. Andrzej M.J. SKULIMOWSKI (2016a). Anticipatory Control of Vehicle Swarms with Virtual Supervision. In: C.-H. Hsu et al. (Eds.): IOV 2016, Lecture Notes in Computer Science 10036, Springer International Publishing, s.65-81, https://link.springer.com/chapter/10.1007/978-3-319-51969-2_6

16. Andrzej M.J. SKULIMOWSKI (2016b). Impact of future intelligent information technologies on the methodology of scientific research. In: Proceedings 16th IEEE International Conference on Computer and Information Technology, Nadi, Fiji, IEEE CPS, s. 238–247, 7–10 December 2016. doi: 10.1109/CIT.2016.118
17. Andrzej M.J. SKULIMOWSKI (2018). Anticipatory networks. In: Poli R. (ed.), Handbook of anticipation: theoretical and applied aspects of the use of future in decision making. Springer International Publishing AG, Cham, s. 1–36. <https://goo.gl/rd3ktp>
18. Andrzej M.J. SKULIMOWSKI, I. Badecka (2016) Software Innovation Dynamics in CMSs and Its Impact on Enterprise Information Systems Development, In: A.M. Tjoa et al. (Eds.): CONFENIS 2016, LNBIP 268, Springer International Publishing AG, s. 309–324, DOI: 10.1007/978-3-319-49944-4_23
19. Andrzej M.J. SKULIMOWSKI, I. Badecka (2018). Competition boosts innovativeness: the case of CMS evolution. In: Thanaruk Theeramunkong, T. [et al.], eds., Recent advances and future prospects in Knowledge, Information and Creativity Support Systems : selected revised papers from the 10th international conference on Knowledge, Information and Creativity Support Systems (KICSS 2015), 12–14 November 2015, Phuket, Thailand—: Advances in Intelligent Systems and Computing, vol. 685; Springer, Cham, s. 188-204, https://doi.org/10.1007/978-3-319-72329-7_11
20. Andrzej M.J. SKULIMOWSKI, A. Ćwik (2017). Communication Quality in Anticipatory Vehicle Swarms: A Simulation-Based Model. In: Peng S.L., Lee G.L., Klette R., Hsu C.H. (eds) Internet of Vehicles. Technologies and Services for Smart Cities. IOV 2017. Lecture Notes in Computer Science, vol. 10689. Springer, Cham, https://doi.org/10.1007/978-3-319-72329-7_11
21. Andrzej M. J. SKULIMOWSKI, Przemysław Pukocz (2012a). Enhancing creativity of strategic decision processes by technological roadmapping and foresight. W: KICSS 2012: seventh international conference on Knowledge, Information and Creativity Support Systems, Melbourne, Australia, 8–10 November 2012, Vincent CS Lee, Kok-Leong Ong (red.); IEEE Computer Society. CPS Conference Publishing Services. s. 223–230 (w IEEE Xplore ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6405533&queryText%3DSkulimowski)
22. Andrzej M. J. SKULIMOWSKI, Przemysław Pukocz (2012b). On-line technological roadmapping as a tool to implement foresight results in IT enterprises. In: Internet – technical developments and applications 2, eds. Adrian Kapczyński, Ewaryst Tkacz, Maciej Rostanski. Berlin, Heidelberg: Springer-Verlag. Advances in Intelligent and Soft Computing; 118, s. 95–111.
23. Train, Kenneth E. (2009). Discrete Choice Methods with Simulation, 2nd Edition. Cambridge University Press, Cambridge, s.388.
24. Wainer, Gabriel A. : Discrete-event modeling and simulation: a practitioner's approach, CRC Press, 2009, s.483, ISBN 978-1-4200-5336-4.

Linki:

<https://simpy.readthedocs.io/en/latest/>,
<https://bitbucket.org/simpy/simpy>,
<https://github.com/SanDisk-Open-Source/desmod>,
www.goldsim.com,
<http://gamma.cs.unc.edu/RVO2>,
<http://www.simscrip.com>,
<http://www.software.rockwell.com>,
<http://www.arenasimulation.com>