

Wirtualne zespoły robocze – nawigacja obliczeniowa

zadania dodatkowe:

1. Optymalizacja doboru momentów czasowych wysyłania informacji ze zmienną częstością chwilową oraz częstością średnią nie przekraczającą 1,0 ramki/s, tak aby po stronie odbiorcy uzyskać jak najmniejszy błąd predykcji położenia i orientacji obiektu bez zwiększenia średniego obciążenia sieci.
Założenia:
 - ramki dochodzą bez opóźnień,
 - stosowany jest algorytm predykcji opracowany w ramach zadania podstawowego,
 - do porównania metody z optymalizacją i metody o stałej częstości chwilowej należy użyć testu predykcji,
 - błędy predykcji położenia i orientacji dla metody z optymalizacją powinny być mniejsze niż przy stałej chwilowej częstości wysyłania ramek.
2. Wygładzanie trajektorii ruchu obiektów - likwidacja nieciągłości wynikających z błędów predykcji (ekstrapolacji). Podczas prezentacji wykonanego zadania należy pokazać różnicę w płynności ruchu, gdy wygładzanie jest lub nie jest stosowane. Należy obliczyć różnice w jakości ekstrapolacji gdy wygładzanie jest i nie jest stosowane. W tym celu można stworzyć dodatkowy cudzy obiekt stanowiący sieciowy obraz własnego obiektu i odtwarzać go bez wygładzania. Należy ocenić możliwości śledzenia pewnej liczby obiektów gdy wygładzanie jest lub nie jest stosowane (za pomocą klawisza I można ukryć numer ID). Założenia:
 - ramki dochodzą bez opóźnień ale ze zmniejszoną częstością,
 - do porównania błędu predykcji z i bez wygładzania należy użyć testu predykcji,
 - błąd predykcji przy wygładzaniu nie powinien być większy niż o 50%.
3. Predykcja z uwzględnieniem opóźnień w sieci. Należy ustawić `czy_opoznienia = 1` w pliku `Interakcja.cpp`. Opóźnienia można wyznaczyć na podstawie czasu uniwersalnego (czas systemowy + poprawka synchronizacyjna) i czasu wysłania ramki. Ramki mniej aktualne od już otrzymanych powinny być filtrowane. Założenia:
 - ramki dochodzą z opóźnieniami (tak jak w sieci rozległej) i z ograniczoną częstością,
 - opóźnienia są losowe ale o stałych średnich i wariancjach,
 - synchronizację można wykonać poza systemem (np. porównując czasy systemowe na monitorach).
4. Predykcja z uwzględnieniem opóźnień w sieci wraz z automatyczną synchronizacją czasu. Należy ustawić `czy_opoznienia = 1` w pliku `Interakcja.cpp`. Opóźnienia można wyznaczyć na podstawie czasu uniwersalnego (czas systemowy + poprawka synchronizacyjna) i czasu wysłania ramki. Ramki mniej aktualne od już otrzymanych powinny być filtrowane. Założenia:
 - ramki dochodzą z opóźnieniami (tak jak w sieci rozległej) i z ograniczoną częstością,
 - opóźnienia są losowe ale o stałych średnich i wariancjach,
 - poprawki synchronizacyjne należy obliczyć wykorzystując specjalne ramki wysyłane do poszczególnych klientów z żądaniem natychmiastowego odesłania.
5. Predykcja parametrów stanu obiektu z wykorzystaniem wielu ramek. W kodzie obsługi klawisza <strzałka do góry> w pliku `interakcja.cpp` należy ustawić $F = 3000$. Dzięki tak dużej sile napędowej możliwe jest dostanie się na orbitę planety. Ekstrapolacja ruchu o niejednostajnym przyspieszeniu wymaga zastosowania algorytmu ekstrapolacji wielopunktowej. Można do tego celu użyć funkcji wielomianowej lub sztucznej sieci neuronowej. Założenia:

- ramki dochodzą bez opóźnień ale ze zmniejszoną częstością,
 - do porównania ekstrapolacji jedno i wielopunktowej należy użyć testu predykcji.
6. Predykcja położenia i prędkości środka pojazdu na podstawie historii zachowań uczestnika lub trajektorii ruchu pojazdu w określonych miejscach terenu (np. gdy użytkownik zawsze przejeżdża daną trasę w prawie ten sam sposób). Po pewnej liczbie przejazdów program powinien rozpoznać scenariusz, a następnie przyjąć odpowiedni sposób uzupełniania informacji. Założenia:
- ramki dochodzą bez opóźnień ale ze zmniejszoną częstością.
 - należy przyjąć, że każdy z uczestników realizuje jeden z 3 scenariuszy jazdy (można wykorzystać scenariusze z testu predykcji)
7. Obsługa kolizji przy ograniczonej częstości i opóźnieniach w sieci w architekturze multicast. Należy ustawić `czy_opoznienia = 1` w pliku `Interakcja.cpp`. Ustalanie czasu kolizji oraz synchronizacja zdarzenia dla poszczególnych uczestników (moment kolizji musi być taki sam dla wszystkich uczestników). Uwzględnienie dużych prędkości (nie wystarczy porównywanie sumy promieni pojazdów do odległości pomiędzy nimi!). Założenie:
- ramki dochodzą z opóźnieniami (jak w sieci rozległej) i z ograniczoną częstością.