

# Metody wirtualnej rzeczywistości

## Urządzenia haptyczne

Paweł Daniluk

Wydział Fizyki

Wiosna 2011



## Technologia haptyczna

Technologia wykorzystująca mechaniczne komunikowanie się z użytkownikami poprzez zmysł dotyku przy użyciu zmieniających się sił, wibracji i ruchów. (z greckiego haptikos – dotyk)

# Urządzenia haptyczne



- Możliwość wskazania położenia i orientacji
- Zwrotna projekcja sił (i momentów sił)

# Rodzaje sił

## W funkcji położenia lub prędkości

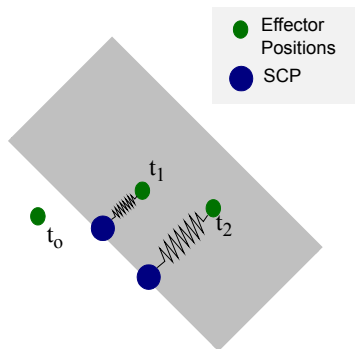
- Siła sprężystości –  $F = kx$  (prawo Hooke'a)  
 $\mathbf{F} = k(\mathbf{p}_0 - \mathbf{p}_1)$
- Tłumienie (tarcie lepkie)  
 $\mathbf{F} = -b\mathbf{v}$
- Tarcie Kulombowskie – stała siła przeciwdziałająca ruchowi  
 $\mathbf{F} = -c\frac{\mathbf{v}}{v}$
- Tarcie statyczne i dynamiczne – przełączanie między dwoma trybami
- Bezwładność – wyliczana na podstawie przyspieszenia  
 $\mathbf{F} = -a\mathbf{m}$

# Rodzaje sił c.d.

## W funkcji czasu

- Siły stałe
- Siły okresowe (wibracje)
- Impulsy

# Interakcja z obiektami



## Proxy

Obiekt podążający w wirtualnej przestrzeni za ruchem urządzenia. Urządzenie oddziałuje na proxy np. siłą sprężystości.

# Integracja z wizualizacją

## Częstotliwość odświeżania

- Grafika – 30 - 60 Hz
- Projekcja sił – 1000 Hz

# Integracja z wizualizacją

## Częstotliwość odświeżania

- Grafika – 30 - 60 Hz
- Projekcja sił – 1000 Hz

Nie da się w jednej pętli obsługiwać równocześnie grafiki i projekcji sił.



# Integracja z wizualizacją

## Częstotliwość odświeżania

- Grafika – 30 - 60 Hz
- Projekcja sił – 1000 Hz

Nie da się w jednej pętli obsługiwać równocześnie grafiki i projekcji sił.

Za projekcję sił odpowiada niezależny wątek. Nie można opóźniać jego działania. Aktualizacja sił następuje niezależnie od odświeżania grafiki.

# Integracja z dynamiką modelu

Użytkownik wpływa na przebieg symulacji wywierając siły na elementy modelu.

# Integracja z dynamiką modelu

Użytkownik wpływa na przebieg symulacji wywierając siły na elementy modelu.

Jak zmierzyć siłę z jaką użytkownik oddziałuje na model?

# Integracja z dynamiką modelu

Użytkownik wpływa na przebieg symulacji wywierając siły na elementy modelu.

Jak zmierzyć siłę z jaką użytkownik oddziałuje na model?

Pomiar siły jest możliwy wyłącznie przez dobranie siły równoważącej tak, aby ramię pozostało nieruchome.

# Integracja z dynamiką modelu

Użytkownik wpływa na przebieg symulacji wywierając siły na elementy modelu.

Jak zmierzyć siłę z jaką użytkownik oddziałuje na model?

Pomiar siły jest możliwy wyłącznie przez dobranie siły równoważącej tak, aby ramię pozostało nieruchome.

## Virtual coupling

Ramię urządzenia z elementem modelu łączy sprężyna (z tłumieniem). *Virtual coupling* umożliwia łączenie wątku haptycznego o wysokiej częstotliwości odświeżania z symulacją.

# Elementy haptycznego interfejsu użytkownika

- Studnia grawitacyjna – siła zadana prawem Hooke'a, ograniczona z góry
- Studnia grawitacyjna z rzutowaniem
- Operacje niezależne od głębokości
- Przekształcenia względne
- Zgodność bodźców wzrokowych z haptycznymi
- Stabilizacja przy pomocy tarcia

# Haptic Device API

Niskopoziomowa biblioteka do komunikacji z urządzeniami haptycznymi. Zapewnia funkcje służące do:

- inicjalizacji
- kalibracji
- zarządzania stanem urządzenia
- pobieranie i aktualizowanie informacji o położeniu i siłach

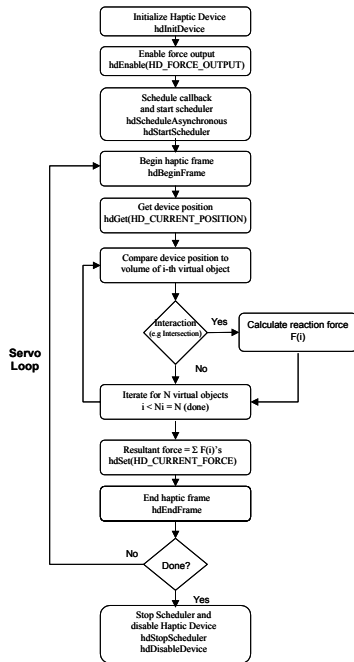
## Pętla zarządzająca (scheduler)

Osobny wątek z pętlą iterowaną z częstotliwością 1000Hz. Zapewnia cykliczne wykonywanie zadanych przez programistę wywołań (*callbacks*).

## Schemat typowego programu

- ❶ Inicjalizacja urządzenia.
- ❷ Utworzenie wywołania zwrotnego (callback), które na oblicza siły na podstawie stanu urządzenia.
- ❸ Włączenie projekcji sił.
- ❹ Uruchomienie wątku zarządzającego.
- ❺ Likwidacja połączenia z urządzeniem i wątku zarządzającego po zakończeniu programu.





# Inicjalizacja urządzenia

## Inicjalizacja

```
HHD hHD = hdInitDevice(HD_DEFAULT_DEVICE);
```

## Włączanie projekcji sił

```
hdEnable(HD_FORCE_OUTPUT);
```

## Uruchamianie pętli zarządzającej

```
hdStartScheduler();
```

## Zmiana bieżącego urządzenia

```
hdMakeCurrentDevice(hHD);
```

# Ramki haptyczne

Ramki haptyczne określają zakresy, w których stan urządzenia jest niezmienny.

## Otwarcie

```
hdBeginInitFrame()
```

## Zamknięcie

```
hdEndInitFrame()
```

# Dostęp do stanu urządzenia

## Pobieranie – rodzina funkcji hdGet

Funkcje: `hdGetDoublev()`, `hdGetIntegerv()`, `hdGetFloatv()`, `hdGetString()`

## Zapisywanie – rodzina funkcji hdSet

Funkcje: `hdSetDoublev()`, `hdSetIntegerv()`, `hdSetFloatv()`, `hdSetString()`

## Stałe określające wartość

`HD_CURRENT_FORCE`, `HD_LAST_FORCE`, `HD_CURRENT_TORQUE`,  
`HD_CURRENT_POSITION`, `HD_CURRENT_VELOCITY`, ...

## Uwaga

Zmiany są wysyłane do urządzenia po zamknięciu ramki.

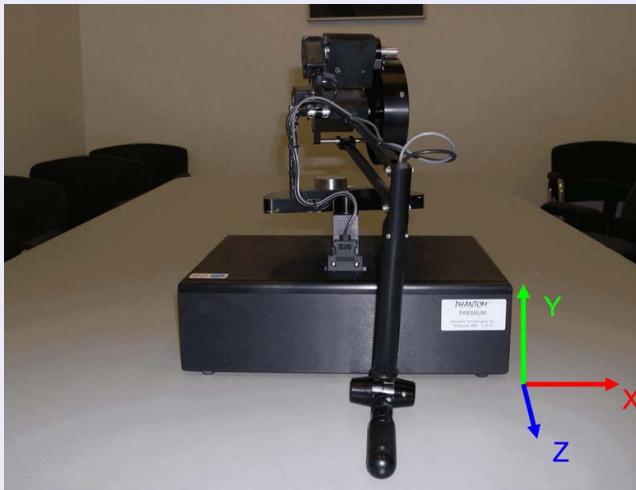
## Dostęp do stanu urządzenia c.d.

### Przykłady

```
HDint buttonState;  
HDstring vendor;  
hduVector3Dd position;  
HDfloat velocity[3];  
HDdouble transform[16];  
  
hdGetInterv(HD_CURRENT_BUTTONS,&buttonState);  
hdGetString(HD_DEVICE_VENDOR,vendor);  
hdGetDoublev(HD_CURRENT_POSITION,position);  
hdGetFloatv(HD_CURRENT_VELOCITY,velocity);  
hdGetDoublev(HD_LAST_ENDPOINT_TRANSFORM,transform);  
  
HDdouble force[3] = 0.5, 0.0, 1.0;  
hdSetDoublev(HD_CURRENT_FORCE,force);  
HDfloat rampRate = .5;  
hdSetFloatv(HD_FORCE_RAMPING_RATE,&rampRate);
```

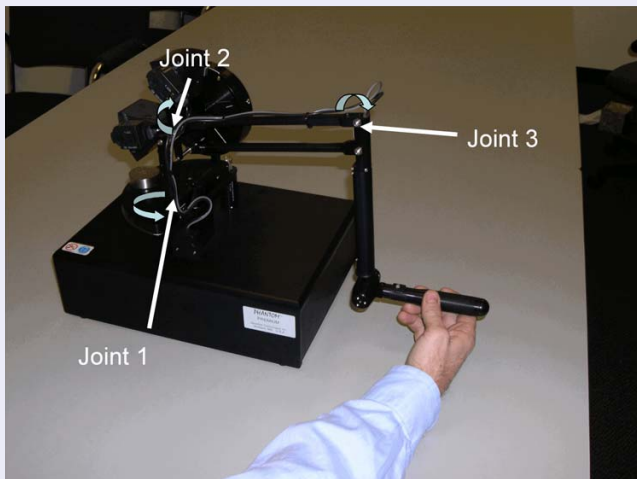
# Przestrzeń położeń

## Kartezjański układ współrzędnych

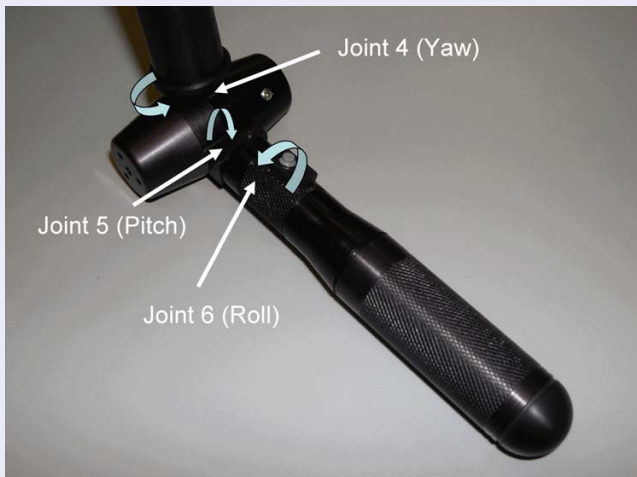


# Przestrzeń położeń c.d.

## Przeguby



## Obroty





## Przestrzeń położeń c.d.

Rodzaj urządzenia	Parametr	Dotyczy	Układ odniesienia	Jednostki
3DOF/6DOF	HD_CURRENT_FORCE	Siły	Kartezjański	N ( $F_x, F_y, F_z$ )
6DOF	HD_CURRENT_TORQUE	Momenty sił	Kartezjański	mNm ( $T_x, T_y, T_z$ )
3DOF/6DOF	HD_CURRENT_MOTOR_DAC_VALUES	Siły i momenty sił	Przestrzeń przegubów	( $M_1, M_2, M_3, \dots$ )
3DOF/6DOF	HD_CURRENT_JOINT_TORQUE	Przeguby ramienia	Przestrzeń przegubów	mNm ( $T_{J1}, T_{J2}, T_{J3}$ )
6DOF	HD_CURRENT_GIMBAL_TORQUE	Przeguby uchwytu	Przestrzeń przegubów	mNm ( $T_{J4}, T_{J5}, T_{J6}$ )

# Pobieranie informacji z urządzenia

## Callback

```
struct DeviceDisplayState {
    HDdouble position[3];
    HDdouble force[3];
}

HDCallbackCode HDCALLBACK DeviceStateCallback (void *pUserData) {
    DeviceDisplayState *pDisplayState=(DeviceDisplayState *)pUserData;
    hdGetDoublev(HD_CURRENT_POSITION, pDisplayState->position);
    hdGetDoublev(HD_CURRENT_FORCE, pDisplayState->force);

    return HD_CALLBACK_DONE;
}
```

## Rejestracja (synchroniczna)

```
DeviceDisplayState state;
hdScheduleSynchronous(DeviceStateCallback, &state,
    HD_MIN_SCHEDULER_PRIORITY);
```

# Generowanie siły

```
HDCallbackCode HDCALLBACK CoulombCallback(void *data) {
    HHD hHD = hdGetCurrentDevice();
    hdBeginFrame(hHD);
    HDdouble pos[3];

    hdGetDoublev(HD_CURRENT_POSITION, pos); //Pobieranie pozycji ramienia

    HDdouble force[3];
    forceField(pos, force); // Obliczanie siły

    hdSetDoublev(HD_CURRENT_FORCE, force); // Ustawianie siły w urządzeniu

    hdEndFrame(hHD); // Wysyłanie siły.

    // Wywołanie ma być wykonane w następnym kroku pętli
    return HD_CALLBACK_CONTINUE;
}

hdScheduleAsynchronous(AForceSettingCallback, (void*)0,
    HD_DEFAULT_SCHEDULER_PRIORITY);
```

# Kończenie pracy

```
hdStopScheduler();  
  
hdUnschedule(scheduleCallbackHandle);  
  
hdDisableDevice(hdGetCurrentDevice());
```

[http://bioexploratorium.pl/wiki/  
Metody\\_Wirtualnej\\_Rzeczywistości\\_-\\_2011](http://bioexploratorium.pl/wiki/Metody_Wirtualnej_Rzeczywistości_-_2011)