# Programmazione e Amministrazione di Sistema e Ambiente Dot Net - 2014

*Milano 30 Aprile 2010 v.1.0 - novembre 2014 v 2.1.x*

*Dott. S.Pinardi Università Milano Bicocca, Dipartimento di Informatica Disco*

# Tema di esame.

*Descrizione del progetto, formato dati, definizioni delle funzioni, specifiche, raccomandazioni, descrizioni dei file delle acquisizioni.*

## Progetto

L’esame richiede la simulazione di una operazione di rappresentazione grafica, analisi e salvataggio di dati provenienti da più accelerometri indossati da una persona.

I dati saranno forniti su file, sono dati ottenuti da casi reali registrati usando un set di 5 sensori inerziali XSens e sono riferiti a campionamenti utilizzati per motivi di ricerca. La maggior parte dei campionamenti sono semplici si riferiscono a azioni ripetute, come alzare e abbassare più volte un braccio, o flettere più volte la testa a destra o a sinistra. Altri ancora si riferiscono ad azioni elementari isolate, come sedersi su una sedia, saltare, salire le scale, correre, cadere. Altri prevedono attività semi-naturali di vita quotidiana (Activity of Daily Living: ADL) eseguiti in sequenza come nel caso seguente: il soggetto è inizialmente fermo su una sedia, si alza, cammina per alcuni passi in linea retta, si volta su se stesso di 180 gradi, torna indietro in linea retta e si risiede. Notiamo che nel sedersi effettua tipicamente una rotazione di 180 gradi, senza soluzione di continuità (senza fermarsi). Altri campionamenti si riferiscono ad attività di tipo sportivo semplice, come eseguire un calcio karate frontale, o un calcio di karate laterale; altre acquisizioni invece riguardano attività sportive complesse, composte da sequenze di azioni semplici relative all’andare a cavallo: in alcuni casi il sistema persona-attrezzo, in questo caso persona-cavallo, sono considerati un insieme interagente non separato, di conseguenza un sensore si trova posizionato sull’attrezzo sportivo (o sul cavallo).

Durante le acquisizioni effettuate sul campo le tre componenti principali dell’architettura sono: un set di sensori Mtx per la cattura delle informazioni (ciascuno dotati di accelerometri, giroscopi e magnetometri triassiali) connesse ai movimenti della persona, un palmare in grado di elaborare preliminarmente questi dati e di trasportarli via Wifi, infine, un server che memorizzi questi dati per avere uno storico, che mostri a video e che effettua analisi a runtime e offline dei dati acquisite e

Il set di sensori indossabili possiedono un dispositivo bluetooth integrato, che permette di inviare i dati campionati al palmare. Le informazioni che genera il palmare verranno successivamente inviate ad un concentratore e al server in due possibili modalità: attraverso una onde Wireless Sensor Network via zigbee o attraverso una connessione socket (TCP/IP), il dispositivo è un membro a tutti gli effetti di un grid di comunicazione (zigbee) e di una WSN (Wireless Sensor Network).



Analizyer

(server da sviluppare)

Socket (Ip e porta da settare)

ByteSender

(Motus emulator)

Files

Figura 1: fusso dei dati

**Per il progetto d’esame** la parte svolta dai sensori inerziali e dal palmare, viene simulata da un applicativo che preleva i dati da un file di log; l’emulatore li invia ogni secondo su una socket TCP/IP riproducendo esattamente quanto capita nel caso reale. L’emulatore e la documentazione allegata con i dettagli tecnici per il formato dei dati, la dimensione della finestra dei dati, il formato di comunicazione socket e quant’altro **sono stati già sviluppati** e sono forniti a parte (sul sito del corso). Lo studente non deve sviluppare questa parte.

Quelle che viene richiesto allo studente ai fini dell’esame è sviluppare **il lato server** che si occupi della rappresentazione grafica dei dati in arrivo, dell’analisi concorrente, e del salvataggio dei dati.

L’analisi del segnale ha lo scopo di identificare le principali attività di una persona in movimento: orientamento nello spazio del corpo (in piedi/seduto), cambiamento di direzione (sta girando), stato (fermo o si muove) o aspetti pericolosi (come ad esempio una caduta) dell’individuo.

Il sistema sensoriale ha un sistema di riferimento S che è in solido col sensore, il sensore vengono montati diversamente a seconda del tipo di acquisizioni, i dettagli verranno forniti nei singoli casi illustrati sotto. Per dare una idea iniziale, per le attività di vita quotidiana ADL il sensore numero 1è di solito montato sul fianco destro del bacino all’altezza della cintura e a causa della particolare forma del sensori gli assi del sistema S sono posizionati nel modo seguente l’asse y in direzione del moto del corpo, x parallela alla spina dorsale e quindi quando il corpo è in piedi è parallelo all’accelerazione gravitazionale e ha verso rivolta in alto, e z direzionato fuori dal corpo (parallelo rispetto all’asse del bacino). Il posizionamento del sensore sul fianco permette di rilevare, in definitiva, il movimento del baricentro del corpo e dell’anca attorno al baricentro del corpo, unitamente ad alcune specifiche accelerazioni verticali, laterali ed orizzontali collegate al moto.

Walking direction

omino

x

y

I sensori sulle braccia (i polsi) e sugli arti inferiori (caviglie) meglio si prestano ad identificare azioni come aprire o chiudere le porte, o eseguire un movimento sportivo come il calcio frontale o laterale del karate. Naturalmente il sistema di riferimento di questi sensori è orientato in modo naturale rispetto agli arti e quindi, in generale non è fisso, ed è differente da come è orientato il sensore del bacino.

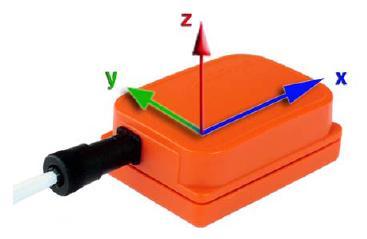


Fig.2 Il sensore e il suo sistema di riferimento solidale

Daremo una descrizione nella sezione relativa alle acquisizioni, del posizionamento specifico dei sensori e di come procedere per l’analisi: più attività verranno identificate meglio sarà valutato il progetto. Almeno 3 tipologie di attività devono essere identificate (**Attitudine** del corpo nello spazio: in piedi, seduto, sdraiato. **Heading**: va dritto, gira a sinistra, gira a destra. **Stato di moto**: è fermo, si muove. Vedete anche le note all’esame, **in fondo**). Altri elementi di valutazione saranno la bontà architetturale del progetto, l’usabilità, e la qualità e la chiarezza dei grafici e dei dati.

### Descrizione del progetto

Entriamo nel dettaglio e vediamo come funzionano le varie parti che entrano in causa.

Il sensore (nel nostro caso il modello è MTx costruito da XSens www.xsens.com) contiene un accelerometro (triassiale), un giroscopio (triassiale), un magnetometro (triassiale), e una quadrupla di valori detti quaternioni (che non verranno usati).

La frequenza di campionamento dei dati è di 50Hz (50 campionamenti in un secondo) ossia uno ogni 0.02 secondi.

I dati vengono emessi in formato byte: vengono usati 4 byte per formare un numero reale in complemento a duesingola precisione secondo lo standard IEEE 754. Quindi come prima cosa questi dati byte devono essere tradotti in numeri decimali (floating). Questo processo deve essere fatto per ogni terna di valori: accelerometri, giroscopi e magnetometri. I quaternioni sono esclusi, dato che non servono.

Nota:

Un singolo campionamento di un singolo sensore è pari a 58 byte di dati. Selezionando quindi una frequenza di 50, 100 o 200 Hz sull’emulatore vengono inviati ad ogni secondo 2900, 5800, 11600 byte.

I 58 byte di un campionamento sono così suddivisi:

3 byte necessari alla comunicazione (da scartare), 2 byte occupati da un contatore + 52 byte di dati (13 campi da 4 byte) + 1 byte per il CRC.

Nota: Al momento della connessione l’emulatore invierà come preambolo 14 byte così divisi:

10 byte che rappresentano ID di 10 caratteri che identificano il dispositivo trasmittente;

4 byte che rappresentano la frequenza emulata (la frequenza emulata non corrisponde alla frequenza di campionamento dei file, se un file è stato campionato a 50 Hz e viene letto a 200 Hz i dati saranno solo spediti ad una velocità maggiore).

### Descrizione pacchetti di dati

Il formato di ogni sequenza di byte inviati dipende dal numero di sensori Mtx collegati; un pacchetto inizia sempre con un preambolo di un byte (0xFA), ed è seguito da un BID (Bus Identifier, identifica mittente e destinatario dei messaggi) e da un MID (Message Identifier, identifica ogni specifico messaggio). Il MID nel caso dei dati sensoriali è 0x32.

Segue un quarto campo Length, che rappresenta la quantità di byte da leggere successivamente; quando questo campo è uguale a 255 significa che è attiva la modalità Extended-Length (modalità a 5 sensori), per conteggiare i byte da leggere viene usato un ulteriore dato a 16 bit rappresentato nel quarto byte (LENext) e quinto byte (EXT LEN) : il preambolo del pacchetto contiene quindi un byte in più. Ad esempio se riceviamo i dati da 5 sensori, avremo 5 \* 52 = 260 byte solo per i dati dei sensori (DATA), a cui andranno aggiunti i già citati byte di preambolo e il CHECKSUM finale. In Fig. 3 è mostrata la differenza tra un pacchetto normale e un pacchetto Extended-Length.

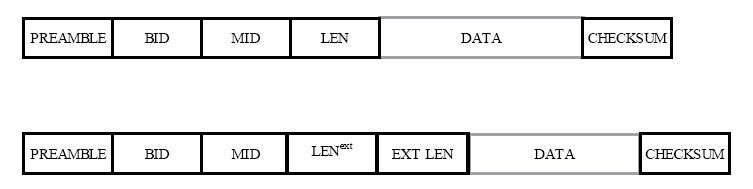


Fig. 3 Formato dati con un uno o con più sensori

I primi due byte della zona DATA fungono da contatore (in Fig. 3 sono compresi nel campo “DATA”, nella Fig. 4 sono stati esplicitati); il valore contenuto nel contatore viene incrementato ad ogni pacchetto inviato, e azzerato ogni volta che Xbus master (il Bus Hardware che comunica coi sensori) viene fermato e fatto ripartire. Alla fine della sezione DATA è presente un byte di Checksum, per l’individuazione di eventuali errori.

Nell’immagine di Fig. 4 è rappresentato un esempio di pacchetto per due sensori in formato calibrated, i.e. con quaternioni:

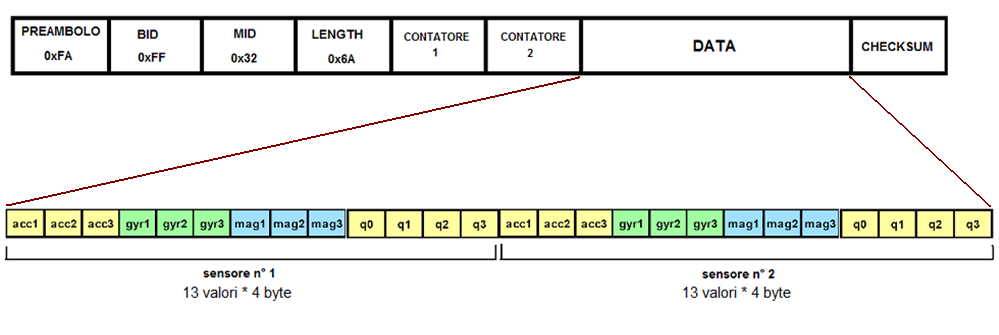


Fig. 4. Il campo DATA con due sensori (esempio).

### Pipeline di analisi dei dati.

Si deve procedere in questo modo: acquisire una finestra di dati fissata (ad esempio 10 secondi, a 50 Hz corrisponde a 500 campioni), questa finestra rappresenta per noi una finestra di analisi.

Di questa finestra occorre rappresentare in dati in modo grafico (tempo sulle ascisse, valore sulle ordinate), mandare la finestra agli algoritmi di analisi, salvare i dati floating point su un file in formato csv, salvare quanto emerge dalle analisi su un file (vedi nella sezione apposita per i dettagli). Alcune di queste operazioni posso svolgersi concorrentemente. Terminata questa fase occorre spostarsi di 5 secondi in avanti nella analisi (e non di 10 secondi, le finestre di analisi si sovrapporranno a metà per non perdere i fenomeni che si pongono a cavallo di due finestre) dopodiché si ripete tutto da capo: rappresentazione grafica, analisi, salvataggio, e così via.

Vediamo quindi ciascuna operazione più in dettaglio.

*La finestra in pratica è una coda.*

### Funzionalità di base.

Per prima cosa dopo aver trasformato i dati in formato decimale, occorre rappresentarli graficamente in modo che siano visivamente comprensibili.

Dopo la conversione noi avremo a disposizione i valori in un vettore di dimensione 9 x N x S. Dove 9 sono i valori dei 3 accelerometri (x,yz), 3 giroscopi (x,y,z), 3 magnetometri (x,y,z), N la grandezza del campione, S il numero dei Sensori (nel nostro caso 5). S[i]N[t] rappresenterà quindi i dati di accelerometro (xyz) giroscopio (xyz) e magnetometro (xyz) del sensore i-esimo al tempo t Chiamiamo questo vettore di dati ***sampwin[9][N][S]*** ***.***(NB l’uso di un vettore tridimensionale è un suggerimento concettuale non un vincolo progettuale).

In pratica fissato il sensore S, che è quello che di preferenza faremo, gli indici t della colonna N di questo vettore rappresentano il tempo t e graficamente parlando saranno rappresentati come valori in ascissa i- valori contenuti costituiranno i dati dei sensori, e verranno rappresentati in ordinata (vedasi fig. sotto ad esempio).

Per fare un esempio numerico semplice:il vettore I [] { 1, 10, 9, 8 ,0} contiene 5 valori che in ascissa avranno posizione 0,1,2,3,4 e in ordinata valore 1, 10,9,8,0. Come in un grafico Excel, per capirci.

Per prima cosa si deve procedere a rappresentare il modulo della accelerazione e il modulo dei giroscopi su due grafici distinti: su un terzo grafico provvederemo a rappresentare il magnetometro nel modo che specificheremo più sotto (vedi **girata**).

Il modulo è una quantità **scalare**, ovvero una quantità indipendente dall’orientamento nello spazio del sensore e per questo appare preferibile per alcune operazioni di analisi, dato che il sensore può variare il proprio orientamento nello spazio, per quanto accuratamente lo si posizioni, essendo il corpo umano mobile e articolato.

Quindi la **prima** funzione che dobbiamo realizzare èquella che restituisce il calcolo del valore assoluto (i.e. del modulo |XYZ|) data una tirpla di floating point di valori in ingresso (x,y,z). Rimandiamo alla letteratura di base di matematica per ulteriori dettagli, ma ricordiamo che si tratta banalmente del teorema di Pitagora nelle tre dimensioni.

Modulo(x,y,z) = sqrt( x2 + y2 + z2)

Dove sqrt() qui indica la funzione radice quadrata.

Quindi fissato il sensore S, presi i dati in ingresso dal vettore sampwin[][][S] (riga 0,1,2 per gli accelerometri; riga 3,4,5 per i giroscopi) di N campioni, restituiamo un vettore monodimensionale modacc[] di N campioni che contiene i valori del modulo (dell’accelerazione o del giroscopio).

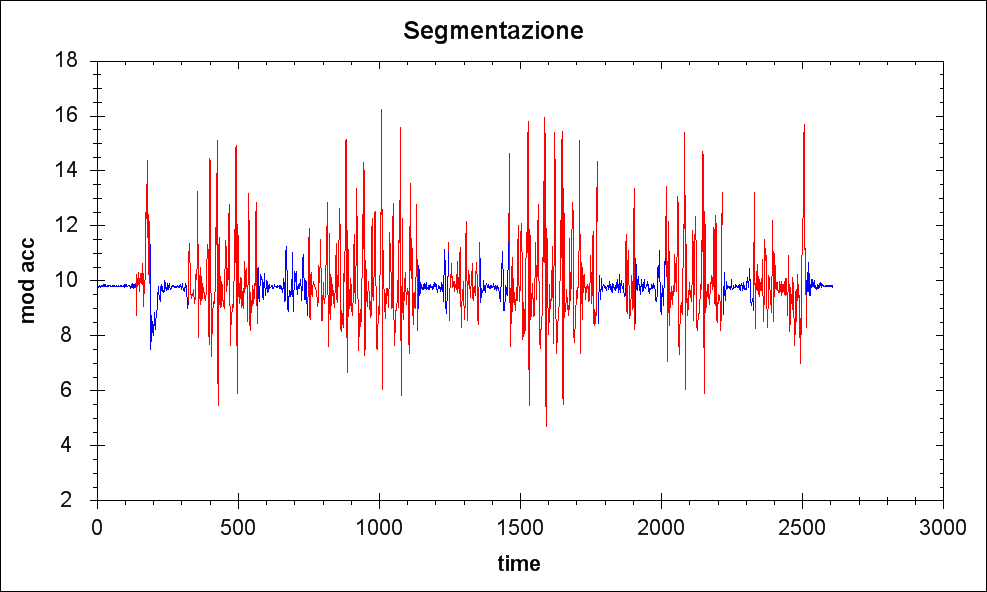


Figura 5: Esempio di segnale (segmentato in due colori diversi da un algoritmo sviluppato appositamente) del sensore dell’accelerometro del bacino, rappresentato come modulo di accelerazione. In questo sempio l soggetto si alza da una sedia, procedere avanti per alcuni metri, gira a destra proseguire per alcuni metri, effettua una svolta di 180°, cammina per alcuni metri, svoltara sinistra, camminar per alcuni metri, giradi 180° e si siede. Tra ogni evento saliente è presente una pausa.

La **seconda** funzione che dobbiamo implementare è una operazione di smoothing, che è utile in molte circostanze, ad esempio per eliminare il rumore (non è la funzione più adatta, ma in casi semplici può essere applicata senza problemi). Si procede così: dati N campioni (N = 500 nel nostro esempio dato che la finestra dura 10 secondi e si campiona a 50 Hz), ci si posiziona sul campione i-esimo. Una volta posizionati sul campione i-esimo si prende la finestra di K valori prima e dopo i (ad esempi i 10 campioni prima e i 10 campioni dopo), si calcola il valore medio di questa finestra di 2K+1 valori, e lo si scrive in una struttura idonea (un vettore smooth[] ad esempio) nel posto i.

Si incrementa l’indice i e si procede ripetutamente, fino a quando i = N.

Alla fine avremo un vettore di N valori “mediati”, quindi meno sensibili alle micro-variazioni.

Per inciso questa non è una tecnica utile per eliminare il rumore, ma per mediare i valori, ovvero per appiattirli. Per renderli “blurred”. Ma la utilizzeremo come strumento per eliminare alcune micro variazioni sui datu causati dal rumore di segnale.

*Lo smoothing pulisce il segnale ma abbassa i picchi, sarebbe più carino usare filtri passa basso e alto.*

Public static double[] smootami(double[] input, int datasize, int windows\_analysis\_size){

double[] smoothed = new double[datasize];

int range = windows\_analysis\_size;

List<double> \_windows = new List<double>();

for(int k = 0 +range ; k < (datasize- range) ; k++){

\_window = creawindow(input, k, range);

smoothed[k] = media(\_window);

}

return smoothed;

}

La **terza** operazione di base che dobbiamo implementare è il calcolo della derivata. Il calcolo della derivata è una operazione matematica “concettuale” dato che implica il concetto di limite, e può anche essere vista come operazione “algebrica” . Qui procederemo numericamente, costruiremo una funzione Rapporto Incrementale (RIfunc) che prende un vettore di valori in ingresso, ad esempio una colonna di sampwin[][], o il vettore modacc[], e restituisce il vettore monodimensionale RI[].

Per esemplificare diciamo che prendiamo “in ingresso” modacc[] che contiene N campioni, ad esempio con N=500.

Si parte da i = 0 si prende la coppia di valori *i* e *i+1*, e si calcola il rapporto incrementale di questi due valori. Il rapporto, una volta calcolato, viene salvato in un vettore di output RI[] nel posto i. Quindi, si incrementa i di uno e si procede di nuovo al calcolo del rapporto incrementale, e così via finche non si arriva ad N. Alla fine RI[] conterrà i valori di rapporto incrementale (ovvero la derivata numerica) della funzione in ingresso.

Vedete wikipedia per la formulina del R.I. <http://it.wikipedia.org/wiki/Rapporto_incrementale>

La **quarta** funzione è il calcolo della deviazione standard

La deviazione standard, come è noto è un indicatore statistico. Avremo però bisogno di una funzione accessoria (**quinta operazione**) per il calcolo della media necessaria per effettuare il calcolo della deviazione standard (della media ne abbiamo parlato nell’operazione di smoothing, non ripeteremo qui quanto già detto).

Quindi per esemplificare, presa la finestra di campioni modacc[] di dimensione N, si calcola la media del valore per tutta la finestra, che poi verrà usata per il calcolo della deviazione standard.

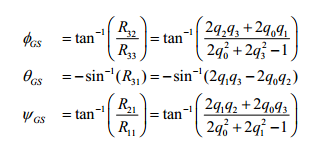
Si consiglia di usare media e deviazione standard mobili, ovvero non su tutta la finestra di dimensioni N, ma usando una sottofinestra mobile con un centro e un intorno di dimensioni T, procedendo da i = 0 a i = N in modo analogo a quanto fatto nella funzione di smoothing

Rimandiamo a wikipedia per la definizione matematica della deviazione standard (è semplice)

<http://it.wikipedia.org/wiki/Deviazione_standard>

La **quinta** operazione - obbligatoria da implementare - è la determinazione degli angoli di Eulero. **Si chiamo angoli di Eulero** gli angoli che il sistema di riferimento S, solidale col sensore, forma rispetto al sistema geo-referenziato G che è inerziale (in altre parole, è fermo). Questi angoli ci consentono, in concreto, di stabilire di quanto si sia inclinato il sensore nello spazio nelle tre dimensioni. . Dato che lo spazio è tridimensionale i gradi di libertà sono tre, Roll (i.e. φ), Pitch (i.e. θ), Yaw (i.e. ψ). Il valore restituito è in radianti, dimensione per dimensione. Possiamo determinare il valore di φ, θ, ψ utilizzando i quaternioni, q0, q1, q2, q3, che occupano gli ultimi quattro campi di ogni campionamento (vedi fig.4).

Le formule per determinare gli angoli di eulero a partire dai quaternioni del nostro sensore Mtx sono le seguenti:



Quindi ad ogni istante, i.e. ad ogni campionamento è possibile estrarre da ogni sensore, la quadrupla q0 q1 q2 q3 di quel campione, che consente di determinare gli angoli di eulero φ, θ, ψ relativi a quel preciso istante.

**(Operazioni ulteriori, facoltative).**

Segnaliamo che le seguenti due operazioni sono **facoltative** per gli appelli da gennaio fino a aprile, **dopo aprile verranno mano a mano introdotte e rese obbligatorie**. Per gli appelli di gennaio-aprile sono utili per ottenere la lode (è sufficiente una delle due, non entrambe). La lode viene impartita se il progetto è convincente, completo, funzionale *e* contiene una delle due seguenti funzioni. Implementare queste funzioni su un progetto non chiaro, non funzionale, o incompleto nelle funzionalità di base o avanzate non dà luogo a lode. Si ricorda che la lode “non fa media” nel curriculum universitario e costituisce una valutazione “qualitativa” sul progetto a *totale* *discrezione* del docente. Può non essere impartita se la commissione di esame non considera il lavoro del gruppo in generale meritevole di lode.

Descriviamo le funzioni opzionali.

Per un migliore funzionamento degli algoritmi è necessario introdurre un algoritmo di segmentazione del segnale (**sesta** **operazione**). Dato che l’introduzione di un algoritmo di segmentazione induce una maggiore complessità architetturale, solo chi lo desidera, e allo scopo di raggiungere una lode nella valutazione, venga a colloquio direttamente dal docente: otterrà ulteriori informazioni dettagliate.

Sempre per completezza, si segnala che in alcuni di questi algoritmi nei casi reali si provvede all’eliminazione del rumore di segnale(**settima** **operazione**).: la pulizia dal segnale richiede l’utilizzo delle FFT (Fast Fourier Transform) per l’elisione delle frequenze alte, di norma responsabili dei comportamenti di dettaglio del segnale.

Questa operazione diviene necessaria per sviluppare un algoritmo in grado identificare o contare i passi (evento); anche in questo chi lo desidera, allo scopo di raggiungere una lode nella valutazione, venga a colloquio direttamente dal docente: otterrà ulteriori informazioni dettagliate.

## Visualizzazione

Ogni volta che acquisiamo N campioni rappresenteremo quindi visualmente i seguenti valori su un grafico bidimensionale. Ogni 5 secondi (se si fissa la dimensione della finestra ad esempio a N = 500, cioè 10 secondi con overlap a metà) osserveremo un refresh dei dati rappresentati graficamente e i nuovi dati in ingresso verranno rappresentati e salvati.

Questi i grafici che vanno obbligatoriamente rappresentati: Modulo accelerazione, Modulo giroscopio, Orientamento (vedi dopo: Girata), posizione del corpo nello spazio (vedi dopo: sit/stand). È possibile rappresentare altri grafici, se ha senso (a scelta dello studente). Ad esempio il rapporto incrementale del magnetometro o delle accelerazioni.

### Attività (eventi)

Lo scopo dell’analisi è *riconoscere* alcune attività tra un insieme predefinito di attività identificate a priori, e *indicare* quando queste siano avvenute.

Diamo l’elenco delle attività e una spiegazione dettagliata delle stesse.

### Moto - Stazionamento

Tra gli eventi da prendere in considerazione e da cercare di riconoscere nel segnale, il moto - stazionamento è sicuramente il più semplice. E’ sufficiente controllare nel modulo della accelerazione il valore della deviazione standard, rispetto al valore medio (di norma ). Rammentiamo che una accelerazione (apparente) è sempre presente -quella gravitazionale - e il sensore la misura sempre: il valore medio ideale è quindi 9.81. Per segnali ideali ci si aspetta che il segnale sia “fisso” sul valore 9.81 e quindi che abbia una deviazione standard pari a 0 (i.e. molto bassa). Ma anche da fermi i valori reali si attesteranno in un intorno di 9.81 e quindi la deviazione standard non sarà mai quella ideale. Bisogna quindi scegliere una soglia sotto la quale il segnale è da considerarsi stazionario. Normalmente le attività di non-stazionamento hanno una deviazione standard superiore a 1, quindi è sufficiente scegliere una soglia che come limite abbia il valore 1. Consigliamo di calibrare questi valori dai dati, tramite un euristica.

Rimandiamo a wikipedia per la definzione matematica della deviazione standard

<http://it.wikipedia.org/wiki/Deviazione_standard>

### Girata

Per comprendere se una persona (o un cavallo) gira nel camminare (nel muoversi) dobbiamo ricorre ai dati del magnetometro (campi 6,7,8 della SampWin[][][S]). Ci poniamo l’obiettivo di comprendere se una persona giri *significativamente*, dato che il corpo nel camminare induce comunque movimenti oscillatori del bacino (o dei fianchi, delle scapole, o del garrese, nel cavallo) a cui il sensore è fissato: *non* siamo interessati ad identificare questi micro movimenti come girate.

Il magnetometro di fatto restituisce la direzione del vettore del polo nord magnetico proiettato sui tre assi del sistema di riferimento S. Il nord magnetico non si muove, ma il corpo nel deambulare si muove rispetto al nord magnetico, e dato che il sistema di riferimento S è solidale col corpo, se il soggetto gira su se stesso osserviamo il variare dell’angolo del vettore che punta al nord rispetto agli assi del sistema di riferimento stesso.

A noi interessa rappresentare l’angolo che il sensore forma rispetto sul piano x-z. Come se fosse una bussola.

Un piccolo ripasso di geometria per capire come.

**Riferendoci al grafico sottostante**

Dato che *z = R cos θ* , *y = R sin θ* 🡪 *tan(θ) = y/z* 🡪 *θ = tan-1(y/z)*

z

y

North pole vector

θ

R

Sz

Sx

A noi interessa sostanzialmente rappresentare il valore di Theta (l’angolo tra R e z) nel tempo (ricordiamo che ogni 0.02 sec avviene un campionamento e noi abbiamo un vettore di campioni temporalmente ordinati).

θ

Dato che per ogni campionamento, quindi in ogni istante di tempo, noi conserviamo nel vettore sampwin[][] nelle colonne 6,7,8 i valori di x, y ,z del magnetometro così come sono stati campionati, possiamo calcolare in ogni istante (cioè per ogni campionamento) il valore di Theta e salvarlo in un vettore idoneo e rappresentarlo graficamente (istante per istante). Il calcolo dell’angolo Theta nel tempo lo chiamiamo ***FunzioneOrientamento***.

In questa funzione, è obbligatorio che l’angolo Theta non sia “a salto” (con “punti di discontinuità” dovuti alla rappresentazione dell’arcotangente), non è possibile infatti girare in modo discontinuo nel mondo reale.

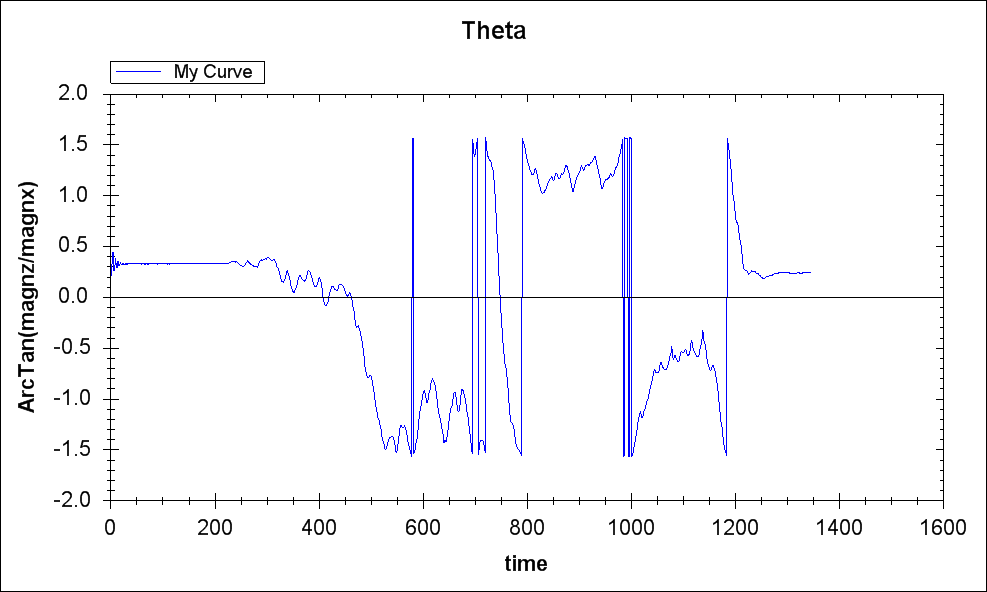


Figura 6: Segnale arcotangente(magnz/magnx) con le discontinuità (*quando si supera il limite in questo caso pigreco/2 si “rientra” o dal basso o dall’alto, bisogna controllare quando c’è un salto grosso allora vanno riappiciccati i pezzi in alto e basso)*

*Per le acquisizioni naturalistiche abbiamo che l’ascissa del nostro grafico sarà* Math.Atan(val[0][7]/val[0][8]).

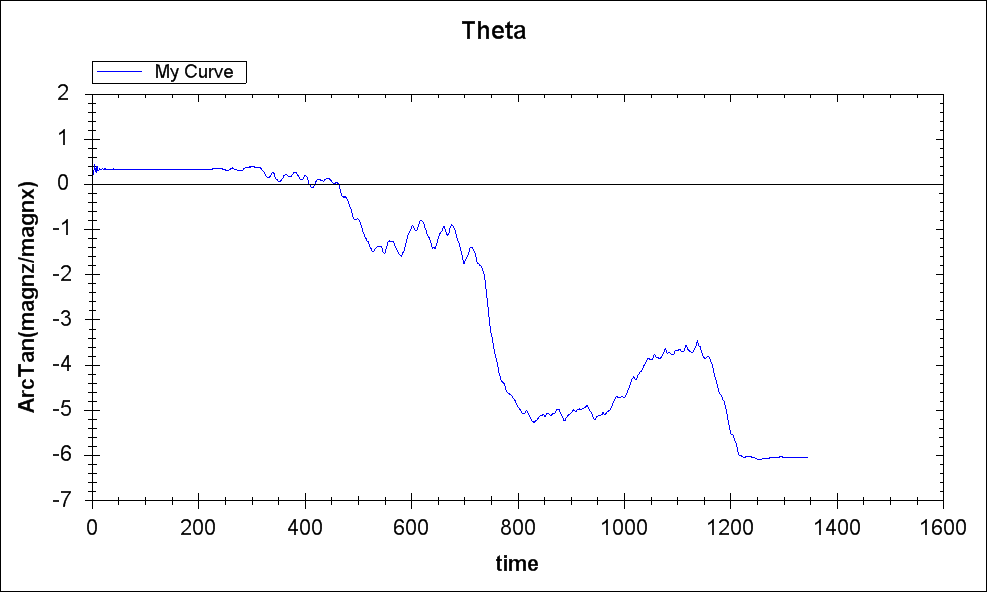


Figura 7: Segnale dell’arcotangente ricostruito algoritmicamente senza discontinuità *(in questo caso c’è stata una modifica affinchè il grafico non presenti discontinuità cioè senza salti da una parte all’altra del grafico, ci accorgiamo che la persona sta girando un casino quando la funziona cambia pendenza in modo pesante, tipo tra 400 e qualcosa e 700 e qualcosa)*

Per capire se un soggetto gira, dobbiamo verificare (l’algoritmo è “a cura dello studente”) che ci sia una differenza di incremento significativa. Consideriamo significative solo le girate superiori ai 5-10 gradi.

Consigliamo (ma non è obbligatorio) prima di tutto di usare una funzione di smoothing per eliminare le microscillazioni per poi procedere a capire se gli incrementi sono significativi.

Una volta identificate le girate occorre salvare questi dati su un file indicando in che direzione il soggetto gira (destra o sinistra), e approssimativamente di quanti gradi gira (tolleriamo errori anche grandi). In particolare indicheremo: istante di inizio e fine girata, direzione girata (dx/sin), e gradi approssimativi della girata. (Vedi in fondo, salvataggio dati).

(Fine parte obsoletata).

### Lay / Stand /Sit

Per identificare se il soggetto è lay (sdraiato/seduto) o stand (in piedi, indifferentemente se cammina o fermo ) andiamo ad osservare come si proietta l’accelerazione gravitazionale (che è sempre presente) sull’asse y. Sostanzialmente, se le componenti del vettore g (accelerazioni), proiettate sulla y restano entro valori “prossimi allo zero” (entro valori di tolleranza relativamente piccola) si considera che la persona è sdraiata altrimenti la persona è in piedi.

Diamo lo schema dei valori di y con le realtive interpretazioni

X < 2.7 Lay

2.7 <X <3.7 LaySit

3.7 < X < 7 Sit

X > 7 Stand

L’algoritmo è lasciato allo studente. Consigliamo di calibrare questi valori usando un’euristica.

*Si prende la dimensione del sensore parallelo alla spina dorsale del corpo, il valore da fermo sarà circa 9.81, quando sdraiato il valore diminuisce, si avvicina allo zero.*

### Dead reckoning (eliminato dal progetto)

La posizione stimata, o *dead reckoning* per usare un termine più diffuso, è una metodica che consente di determinare la posizione della persona nello spazio utilizzando esclusivamente la velocità (la funzione dello spostamento nel tempo), la direzione ed il verso. La posizione stimata risulta sempre affetta da piccoli errori di misura. L’errore della posizione stimata è in media di circa 1/20 del cammino percorso, e quindi aumenta con il tempo. La correzione di questo errore **non** è una questione da risolvere per questo progetto.

Utilizzando gli angoli di Eulero (vedi pag. 7) e la funzione che stima il moto/stazionamento (vedi pag. 9), lo studente deve determinare lo spostamento dell’utente nello spazio (più precisamente del sensore indossato dall’utente) e disegnarlo graficamente e “a runtime” su una windows form, usando ad esempio le zedgraph, assumendo che l’utente sia visto dall’alto, ovvero rappresentando il suo moto su una cartina bidimensionale. Lo studente deve usare la funzione che stima il moto/stazionamento (vedi pag. 9) come una funzione che definisce, per convenzione e indirettamente, la *velocità* dell’utente: maggiore è il valore di deviazione standard, maggiore sarà la velocità. Saranno necessari “aggiustamenti opportuni” dei valori di questa funzione per evitare che la velocità dell’utente sia eccessivamente “altalenante” su piccoli intervalli di tempo (<1 sec). Si consiglia di usare una funzione di smoothing allo scopo. Una funzione interpolante è la funzione ideale. Avendo la velocità, anche se determinata indirettamente, lo spazio percorso è calcolato usando l’integrale della velocità: in termini numerici e quindi programmativi questo corrisponde alla “cumulazione” dei valori di velocità nel tempo (la sommatoria cumulata). Per “tradurre” questa stima numerica in unità di misura reali, si tenga conto che la velocità media di una persona che cammina è di circa 1 m/sec.

Alla fine verrà quindi rappresentato da parte dello studente, istante per istante, la posizione dell’utente nello spazio usando un marcatore (un piccolo cerchio, di diverso colore, ad esempio) che indichi la posizione attuale, e una linea che indichi il percorso già effettuato. E’ possibile introdurre un ritardo (latenza), tra l’arrivo dei dati e la rappresentazione degli stessi, di massimo 6 secondi. Una latenza è necessaria, ed accettata, solo se si effettuano operazioni di “pulizia” dei dati (ad es. lo smoothing) prima di rappresentarli o per altre ragioni che devono essere verbalmente spiegate e gisutificate. La latenza comunque deve essere proporzionale al peso algoritmico di queste operazioni, non deve e non può essere **ingiustificatamente** alta.

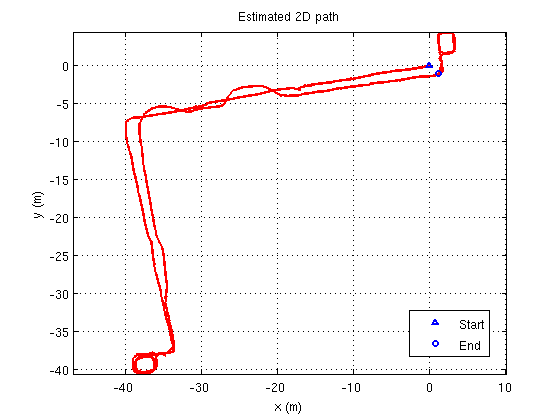
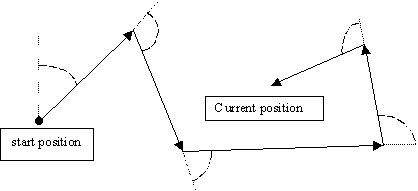


Fig 8. A sinistra, una schematizzazione di come si può effettuare il dead reckoning utilizzando spostamento e angolo in 5 campionamenti successivi (è una esemplificazione). A destra un esempio più rappresentativo di cosa ci si aspetta di vedere nel progetto a video.

**Nota:** per testare il corretto funzionamento delle operazioni di dead reckoning potranno essere usati, durante l’esame, qualunque file di simulazione messo a disposizione, comprese quindi le acquisizioni *naturalistiche continue*, e *quelle a cavallo (Pavia)*.

**Nota:** il sistema dovrà prevedere di poter leggere i dati di più emulatori in funzione in parallelo, per simulare la presenza di più persone in moto nello stesso ambiente, contemporaneamente. Per semplicità programmativa, i dati saranno intesi come riferiti a persone “intangibili”: è accettabile che due persone si trovino *temporaneamente* nello stesso posto allo stesso istante.

### Salvataggio dati

Una volta riconosciuti i fenomeni, occorre salvare le informazioni relative allo stato di moto, girata e lay/stand/sit su un file in un qualsiasi formato ragionevole: ad esempio con stringhe separate da virgole o spazi; o in formato xml. Va indicato il fenomeno riconosciuto, preceduto dal relativo tempo di inizio e fine del fenomeno nel formato “ore, minuti, secondi”. Quindi, quando comincia l’acquisizione dei dati dovrete farvi dare dal sistema operativo il tempo corrente con precisione al secondo: farete corrispondere il primo campione in cui i dati vi arrivano dall’emulatore col tempo corrente. Notiamo che il momento preciso in cui viene avviene un fenomeno può essere dedotto, sapendo che le finestre sono fatte di N campioni e che i campionamenti avvengono ogni 0.02 secondi (50 Hz). Ad esempio se rileviamo un fenomeno (ad una seduta/sit) iniziare al campione 210 della seconda finestra e la prima finestra è stata acquisita alle 10h:25min:33sec, dato che la prima finestra dura 10 secondi, la seconda inizia in sovrapposizione a metà della prima cioè al 5 secondo della prima finestra, il 210° campione corrisponde all’istante 210\*0.02 = 4.2 (sec) della seconda finestra. Quindi il fenomeno avviene al secondo 9.2 dall’inizio della acquisizione, e quindi all’istante 10h:25min:33sec + 9.2sec = 10:25:42.2

Approssimando: 10:25:42.

Di seguito alcuni esempi di scrittura.

“10:30:00 10:30:02 fermo”

“10:30:10 10:30:11 girata dx”

“10:30:00 10:31:00 in-piedi”

Indichiamo, ogni volta che non è fermo, il fenomeno opposto “non-fermo”

“10:30:02 10:30:30 non-fermo”

Occorre prestare attenzione al fatto che usiamo sliding windows che si sovrappongono, quindi lo *stesso* fenomeno - se dura qualche secondo - può essere riconosciuto due volte: questo problema va gestito in modo ragionevole cercando di evitare duplicazioni di scrittura dello *stesso* fenomeno.

### Note. Per la rappresentazione grafica e le funzioni statistiche.

Una difficoltà tecnica consiste nel rappresentare i grafici, C# non fornisce una libreria “a-là Excel” per i grafici, occorre quindi cercare una libreria in rete usabile in C#. Questo è considerato parte dell’obiettivo di esame. Possono essere usate le Zedgraph. Gli Herz di campionamento sono un parametro del programma, deve essere possibile impostare da menù sul server la frequenza. Il programma deve funzionare correttamente anche se la frequenza viene raddoppiata o dimezzata, ad esempio. Gli Herz obbligatoriamente richiesti sono i seguenti: 50 100, 200. Di default il programma parte impostato a 50 Hz.

### Note per l’esame

Il giorno dell’esame bisogna presentarsi con il progetto funzionante, una presentazione powerpoint (o modalità analoga) di circa 10 slide, della durata di circa 10 minuti, con descrizione del progetto e funzionalità, e una documentazione di almeno 10 pagine, con scopi e la descrizione delle principali funzioni offerte dal progetto.

Se non vengono implementate tutte le funzionalità richieste, se il resto del progetto funziona senza errori, in modo completo, razionale e funzionale, il progetto avrà una valutazione proporzionalmente più bassa, comunque **inferiore** a 26/30. Il voto è **personale**, non di gruppo: la commissione si riserva il diritto di effettuare durante l’esame - specialmente in caso di dubbio sull’effettivo svolgimento del lavoro da parte di *tutti* i componenti del gruppo - domande di approfondimento ai singoli componenti del gruppo, sia sul progetto in generale che sulla specifica parte sviluppata a cura del singolo, per verifica.

Le funzioni di analisi dei dati indicate vanno **implementate** (in particolare, la prima, la seconda e la terza), ovvero non vanno richiamate da librerie esistenti, o usate tramite components, web services, DLL, etc. In caso di dubbio su come procedere in questo senso contattate il docente. Per ogni evenienza tecnica o dubbio progettuale contattate il docente o i tutor che vi sono stati assegnati allo scopo.

***ATTENZIONE. Tutte le acquisizioni fatte coi sensori, i dati personali, i video e il materiale connesso servono esclusivamente per uso INTERNO. Non distribuire, copiare o pubblicare nessun dato, video, registrazione o acquisizione, senza le autorizzazioni scritte che vanno richieste al dipartimento di informatica Università Milano Bicocca e ai responsabili del Laboratorio di ricerca Nomadis, di questo dipartimento. Le autorizzazioni sono necessarie per rispetto delle norme sulla privacy, per rispetto della privacy dei minori, e per protezione del lavoro di ricerca e rispetto di brevetti. Le violazioni verranno perseguite legalmente.***

## Acquisizioni e File

Allegati all’emulatore (Xbus Simulator) sul sito ci sono 3 tipologie di acquisizioni effettuate in tre diverse condizioni: **Acquisizione Singole Separate** (12 tipi diversi di azioni singole e brevi, svolte da due soggetti e ripetute due volte), **Acquisizioni a Cavallo** (4 sequenze di azioni, per due soggetti. Due video allegati), **Acquisizione Naturalistiche Continue** (2 sequenze di azioni di un solo soggetto). Tutte le acquisizioni contengono i file in formato Raw (bytecode) che devono essere usati con l’emulatore di Motus.

Note: nelle Acquisizioni Singole Separate sono stati allegati anche 4 file in formato csv contenenti gli stessi dati dei file raw rappresentati in formato decimale, quindi già convertiti in floating point IEE754 (servono solo per una vostra verifica, non vanno usati).

### Acquisizioni Singole Separate

*Acquisizioni effettuate in laboratorio Nomadis.*

*Lorenzo Airoldi, Francesco Renzi, Stefano Pinardi.*

*Milano, Apr. 2009 - Nov. 2009*

Si tratta di **15 azioni** separate e brevi (11 azioni sono una per file, le 4 girate sono radunate in unico file), svolte da due soggetti (Lorenzo, Francesco) ripetute due volte (in due giornate diverse). Queste acquisizioni sono utili per capire se gli algoritmi sono corretti per la brevità e la specificità, verranno usati anche per verifica in sede di esame. Sono intesi come file preliminari di studio.

**POSIZIONAMENTO SENSORI**

La disposizione dei sensori è a “X” (polsi, caviglie, e fianco sinistro). Il primo sensore (sensore 1) è sul bacino, a sinistra, con X verso l’alto, Z uscente dal fianco, Y in direzione del moto; il secondo e il terzo sensore sono posti rispettivamente sui polsi destro e sinistro, con X rivolta verso il prolungamento del braccio, e la Z uscente dalla parte superiore del braccio; il quarto e il quinto sensore sono posti rispettivamente sulle caviglie destra e sinistra, con la X rivolta verso il basso e la Z che esce lateralmente (si prenda la Fig. 1 come riferimento).

**Azioni**: 1. Alzarsi dal letto. 2. Sedersi sul letto. 3. Alzarsi da sedia 4. Sedersi su Sedia 5. Sedersi pesantemente su Sedia. 6. Girata SX-90 gradi 7. Girata SX-180 gradi 8. Girata DX-90 gradi 9. Girata DX-180 10.Salire le scale 11. Scendere le scale 12. Camminare 13.Cadere 14. Correre 15. Saltare.

**File**: Cartella Lorenzo 1 (12 file), Cartella Lorenzo 2 (12 file), Cartella Francesco 1 (12 Azioni), Cartella Francesco 2 (12 Azioni).

**Nota:** le 4 girate sono in un unico file

**Nota:** sono stati allegati anche 4 file in formato csv contenenti gli stessi dati dei file raw ma già convertiti in floating point IEE754 (servono solo per verifica, non vanno usati).

### Acquisizioni a Cavallo

*Acquisizioni maneggio Pavia Istituto Mondino – Università Bicocca Nomadis*

*Marino Alessandro, Renzi Francesco, Landi Davide, Pinardi Stefano*

*Pavia, 26 aprile 2010*

**POSIZIONAMENTO SENSORI**

1. Scapola sinistra; (x verso l’alto parallelo alla spina dorsale, y uscente a sinistra, z perpendicolare alla schiena)
2. Scapola destra; (x verso l’alto parallelo alla spina dorsale, y uscente a sinistra, z perpendicolare alla schiena)
3. Ultime vertebre lombari/Inizio osso sacro; (x verso l’alto parallelo alla spina dorsale, y uscente a sinistra, z perpendicolare alla schiena)
4. Femore sinistro (metà); (x verso il basso direzione parallela all’osso del femore, y verso il retro della gamba, sinistra, z perpendicolare all’asse femorale uscente dal corpo)
5. Garrese del cavallo.(x e y parallela al piano del pavimento, x in direzione coda del cavallo, la y in direzione lato destro del cavallo, z verso l’alto parallela alla linea ideale delle zampe del cavallo)

**ESERCIZI EFFETTUATI**

**Soggetto 3**

Nome: Edoardo;

Sesso: M;

Età: <12;

Stato: patologia non diagnosticata;

*Attività 6*: Mezzo giro antiorario, senza mani con bastone, con appoggio, cavallo al passo + Mezzo giro antiorario stop e partenza, senza mani con bastone, con appoggio, cavallo al passo;

Orario inizio: 17.07;

File: edoardo6;

**Soggetto 4**

Nome: Giulia;

Sesso: F;

Età: <12;

Stato: emiplegia superiore sinistra;

*Attività 3*: 3 giri in senso antiorario, senza mani (appoggiate sulle gambe), con appoggio, cavallo al passo + 3 giri in senso orario, senza mani (appoggiate sulle gambe), con appoggio, cavallo al passo;

Orario inizio: 17.28;

Video: Giulia3;

File: giulia3;

*Attività 5*: Giro antiorario stop e partenza, senza mani (appoggiate sulle gambe), con appoggio, cavallo al passo;

Orario inizio: 17.32;

Video: Giulia5;

File: giulia5;

*Attività 8*: Giro antiorario, con mani, con appoggio, cavallo al trotto;

Orario inizio: 17.43;

File: giulia8;

### Acquisizioni Naturalistiche Continue

*Acquisizioni effettuate in laboratorio Nomadis.*

*Davide Landi, Francesco Renzi, Mirko Galbusera , Stefano Pinardi .*

*Milano, 30 Apr. 2010*

I file sono relativi alle attività di un soggetto che si muove in un piccolo ambiente e durano da alcuni secondi, al massimo due minuti.

**POSIZIONAMENTO SENSORI**

La disposizione dei sensori è a “X” (polsi, caviglie, e fianco sinistro). Il primo sensore (sensore 1) è sul bacino, a sinistra, con X verso l’alto, Z uscente dal fianco, Y in direzione del moto; il secondo e il terzo sensore sono posti rispettivamente sui polsi destro e sinistro, con X rivolta verso il prolungamento del braccio, e la Z uscente dalla parte superiore del braccio; il quarto e il quinto sensore sono posti rispettivamente sulle caviglie destra e sinistra, con la X rivolta verso il basso e la Z che esce lateralmente (si prenda la Fig. 2 come riferimento).

**DESCRIZIONE**

**Primo azione** (camminata semplice)

Il soggetto **A** parte da seduto, si alza, cammina in linea retta per alcuni passi, volta su se stesso di // 180 gradi, torna indietro e si siede. **Non** ci sono “pause” di delimitazione tra un fenomeno e

l’altro.

**File**: camminata semplice1, camminata semplice2

**Secondo file** (camminata con svolta)

Il soggetto **A** parte da seduto, si alza cammina in linea retta per alcuni passi, volta a destra,

cammina per pochi passi, volta su se stesso di 180 gradi, torna indietro di pochi passi, volta

sinistra di 90 gradi, procede dritto per alcuni passi e di risiede. **Non** ci sono “pause” di

delimitazione tra un fenomeno e l’altro.

**File**: camminata svolta1, camminata svolta2

**Terzo file** (tre camminate diverse)

Il soggetto **A** parte da seduto, si alza cammina in linea retta per alcuni passi, volta a sinistra, cammina per pochi passi, volta su se stesso di 180 gradi, torna indietro di pochi passi, volta a destra di circa 90 gradi, procede dritto per alcuni passi si siede e si ferma. Volta di 180 gradi.

Poi cammina in linea retta per alcuni passi, volta a destra, cammina per pochi passi, volta su se stesso di 180 gradi, torna indietro di pochi passi, volta a sinistra di 90 gradi, procede dritto per alcuni passi e si ferma. Volta di 180 gradi.

Poi cammina in linea retta per alcuni passi, volta su se stesso di 180 gradi, torna indietro e si siede. Ci sono “pause” di delimitazione tra una camminata e l’altra.

**File**: tre camminate1, tre camminate2.

**Quarto file** (sdraiato, si alza cammina e poi si risdraia)

Il soggetto **B** è sdraiato su un letto, poi si alza e cammina, cammina dritto, volta a destra, cammina per pochi passi, volta su se stesso di 180 gradi, torna indietro di pochi passi, volta a sinistra di 90 gradi, procede dritto per alcuni passi e di risdraia.

**Non** ci sono “pause” di delimitazione tra un fenomeno e l’altro.

**File:** partenza sdraiato1, partenza sdraiato2.