
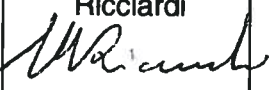



**SPECIFICA DEI REQUISITI DI SISTEMA SCMT**

**VOLUME 3 Algoritmo di odometria per SCMT**

A termine di legge RFI S.p.A. si riserva la proprietà di questo documento che non potrà essere copiato, riprodotto o comunicato a terzi senza specifica autorizzazione

Rev.	Data	Descrizione	Redazione	Verifica Tecnica	Autorizzazione
B	15 Maggio 2012	Modificato criterio di assegnazione della variabile v1d_gt_v2d	Ricciardi 	Ricciardi 	Bonafè 

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
2 di 46

## ELENCO DELLE REVISIONI

Rev	Data	Motivo della revisione
A	9 Febbraio 2009	Prima emissione
B	15 Maggio 2012	Modificato criterio di assegnazione della variabile <b>v1d_gt_v2d</b> per la minimizzazione dei rumori

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
3 di 46

## INDICE

<b>1</b>	<b>Scopo del documento .....</b>	<b>5</b>
<b>2</b>	<b>Riferimenti .....</b>	<b>5</b>
<b>3</b>	<b>Algoritmo di Odometria .....</b>	<b>5</b>
3.1	<i>Costanti Odometria.....</i>	<i>5</i>
3.2	<i>Calcola Odometria.....</i>	<i>13</i>

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

4 di 46

**INDICE DELLE FIGURE**

**INDICE DELLE TABELLE**

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
5 di 46

## 1 Scopo del documento

Lo scopo del documento è la definizione dell'Algoritmo di Odometria da utilizzare per il Sistema Controllo Marcia Treno. Esso è espresso mediante il seguente sorgente Matlab.

L'algoritmo è derivato dai documenti [R1] e [R2].

## 2 Riferimenti

Titolo	Codice	Rev	Data	Ente emittente
[R1] Odometria SCMT - Algoritmo per il calcolo della velocità stimata in caso di pattinamento o slittamento degli assi di misura.	Specifica N° 373450	0	16 Dicembre 2002	Trenitalia
[R2] Odometria SCMT - Principi generali dell'algoritmo per il calcolo della velocità stimata in caso di pattinamento o slittamento degli assi di misura.	Specifica N° 372574	3	10 Dicembre 2002	Trenitalia-UTMR

## 3 Algoritmo di Odometria

### 3.1 Costanti Odometria

ASSEGNAZIONE COSTANTI PER ALGORITMO DI ODOMETRIA

%% variabili logiche

FALSO = 0;

VERO = 1;

%% costanti utili per la visualizzazione del tipo di elaborazione:

%% 1: il treno sta accelerando o frenando?

ACCELERA = VERO;

FRENA = FALSO;

%% 2: tipo di elaborazione effettuato, dipende dalla fase (accelerazione/frenatura),

%% dall'eventuale slittamento/pattinamento dei due assi e, in caso di aderenza,

%% da quanto tempo lo stato di aderenza si è verificato

PATTINAMENTO = 3;

SLITTAMENTO = -3;

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

6 di 46

ADERENZA\_NON\_CERTA\_SLITTAMENTO = -2;  
ADERENZA\_NON\_CERTA\_PATTINAMENTO = 2;  
NON\_PATTINAMENTO = 1;  
NON\_SLITTAMENTO = -1;  
TRATTO\_A\_V\_COSTANTE = 4;

%% 3: numero di assi disponibili per l'elaborazione

ASSE1 = 1;  
ASSE2 = 2;  
ENTRAMBI = 3;  
NESSUNO = 0;

%% 4: stabilità della velocità dell'asse, dipende da:

%% - differenza di velocità dei due assi;  
%% - accelerazione dell'asse;  
%% - variazione dell'accelerazione dell'asse.

STABILE = 1;  
QUASI\_STABILE = 0.5;  
NON\_STABILE = 0;

%% 5: bloccaggio di un asse

%% - l'asse è BLOCCATO se la sua velocità è inferiore  
%% a una soglia minima;  
%% - l'asse è QUASI LIBERO se la sua velocità è superiore  
%% alla soglia minima, ma non è ancora passato un tempo  
%% 'sufficiente' (dipendente linearmente dalla  
%% velocità) dal bloccaggio;  
%% - l'asse è LIBERO se la sua velocità è superiore alla  
%% soglia minima ed è passato un tempo 'sufficiente'  
%% dall'ultimo bloccaggio.

BLOCCATO = 1;  
QUASI\_LIBERO = 0.5;  
LIBERO = 0;

%% 6: pattinamento degli assi

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

7 di 46

%% - gli assi PATTINANO se non sono rispettate le condizioni  
%% sulla differenza di velocità e sull'accelerazione;  
%% - gli assi FORSE NON PATTINANO se sono rispettate le  
%% condizioni sulla differenza di velocità e sull'acce-  
%% razione, ma per un tempo non 'sufficiente' (costante);  
%% - gli assi NON PATTINANO se sono rispettate le  
%% condizioni sulla differenza di velocità e sull'acce-  
%% razione, per un tempo 'sufficiente' (costante).

PATTINA = 0;

NON\_PATTINA = 1;

FORSE\_NON\_PATTINA = 0.5;

%% 7: slittamento degli assi

%% - gli assi SLITTANO se non sono rispettate le condizioni  
%% sulla differenza di velocità e sull'accelerazione;  
%% - gli assi FORSE NON SLITTANO se sono rispettate le  
%% condizioni sulla differenza di velocità e sull'acce-  
%% razione, ma per un tempo non 'sufficiente' (costante);  
%% - gli assi NON SLITTANO se sono rispettate le  
%% condizioni sulla differenza di velocità e sull'acce-  
%% razione, per un tempo 'sufficiente' (costante).

SLITTA = 0;

NON\_SLITTA = 1;

FORSE\_NON\_SLITTA = 0.5;

%% 8: inizio pattinamento

%% - siamo in una fase di NON\_PATTINAMENTO se sono rispettate  
%% le condizioni descritte nel punto 6;  
%% - siamo in una fase di INIZIO PATTINAMENTO se l'inizio del  
%% pattinamento si è verificato da 'poco' tempo (la lunghezza  
%% di questo intervallo di tempo dipende linearmente dalla  
%% velocità. In questa fase la velocità stimata è costante;  
%% - siamo in una fase di PATTINAMENTO se l'inizio del pattinamento

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

8 di 46

```
%%      si è verificato da un tempo 'sufficiente'.
IS_NOT_SLIDING = 0;
IS_BEGINNING_TO_SLIDE = 0.5;
IS_SLIDING_FOR_A_WHILE = 1;
%% 9: variabili di commutazione
%% - POSSO COMMUTARE dalla fase di accelerazione a quella di
%%      frenatura o viceversa se l'accelerazione media filtrata
%%      dei due assi cambia segno per un tempo 'sufficiente'
%%      fissato;
%% - NON POSSO COMMUTARE se l'accelerazione media filtrata
%%      dei due assi non cambia segno;
%% - POTREI COMMUTARE se l'accelerazione media filtrata
%%      dei due assi cambia segno per un tempo 'sufficiente'.
CAN_COMMUTATE = 1;
CANNOT_COMMUTATE = 0;
COULD_COMMUTATE = 0.5;
```

## ASSEGNAZIONE PARAMETRI NECESSARI PER L'ELABORAZIONE

```
%% soglia di velocità: se la differenza tra le velocità dei due assi supera
%% questo valore gli assi stanno pattinando/slittando
soglia_var_velo = 2/3.6; % m/s
%% il seguente valore viene aggiunto alla dr (decelerazione derivata da peso
%% frenato) per ottenere il valore di accelerazione con cui vengono confrontate
%% le accelerazioni dei singoli assi. In altri termini, si suppone che un asse
%% stia pattinando se la sua decelerazione supera di una certa quantità (fissata)
%% la dr.
incremento_decelerazione_ammesso = 0.8; % m/s^2
%% in fase di slittamento (accelerazione) la soglia di accelerazione al di
%% sopra della quale si suppone che un asse stia slittando è fissata
soglia_acce_slittamento = 2.6; % m/s^2
```



**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

9 di 46

```
%% si fissa un valore di accelerazione che viene sommato al termine dovuto
%% alla pendenza per discriminare tra le fasi di trazione/frenatura
    soglia_per_slittamento = -0.1;  % m/s^2
%% in fase di pattinamento/slittamento o aderenza non certa, se l'accelerazione
%% media cambia segno e la variazione di segno permane per un tempo 'sufficiente'
%% si effettua la commutazione trazione/frenatura o viceversa. La lunghezza
%% dell'intervallo di tempo di osservazione è stata fissata pari a 4 s.
%% Si valuta il numero di campioni corrispondenti a una finestra temporale di
%% ampiezza pari a 4 s.
    N_commutazione = round(4/T);
%% numero di campioni corrispondenti al tratto a v costante che deve essere
%% effettuato all'inizio di un pattinamento (il tratto di riferimento ha
%% ampiezza 3 s)
    N_inizio_pattinamento_iniziale = round(3/T);
%% nel calcolo della decelerazione media clippata si introduce un termine
%% di maggiorazione...
    maggiorazione_decelerazione = 0.15;  % m/s^2
%% limite inferiore per il clippaggio
    drm = 0.15;  % m/s^2
%% valore moltiplicato per la dr per ottenere il limite superiore di clippaggio
    kprime = 1.2;  % m/s^2
%% soglia di accelerazione utilizzata per valutare se un asse è stabile
    SOGLIA_JERK = 0.5;  % m/s^2
%% SOGLIA_JERK = 0.6;  % m/s^2
%% la stabilità della velocità di un asse viene controllata osservando
%% per un tempo fissato (2 s) la massima variazione di accelerazione
%% dell'asse
    N_asse_stabile = round(2/T);
%% quando viene riconosciuta la fine di un pattinamento o un asse stabile
%% la velocità stimata si 'adegua' a quella dell'asse utilizzando un valore
%% di decelerazione fissato
```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
10 di 46

```

d_adeguamento = 5; % m/s^2
%% se, durante un adeguamento, si perdono le condizioni di aderenza, viene
%% effettuato il tratto a v costante previsto all'inizio di un pattinamento.
%% Il tratto costante non viene effettuato se la differenza tra la velocità
%% stimata e quella degli assi supera un valore fissato
delta_v_adeguamento = 1.5; % m/s
%% galleggiamento...in fase di aderenza non certa o pattinamento, la differenza
%% tra la velocità stimata e quella dell'asse + veloce deve essere maggiore
%% di una soglia che dipende linearmente dalla velocità
maggiorazione_v_iniziale = 1.5; % m/s
%% ampiezza dell'intervallo di tempo necessario per verificare che una ripresa
%% di aderenza sia stabile in fase di pattinamento
N_non_pattinamento = round(2/T);
%% soglia minima utilizzata per decidere se la velocità di un asse può essere
%% considerata nulla
soglia_vel_nulla = 1 ; % m/s
%% valore di "tolleranza" utilizzato per verificare se le velocità di due assi
%% si incrociano - serve per compensare eventuali errori commessi nel riallineamento
tolleranza = 1/3.6;
%% valore di velocità utilizzato per definire il livello di "saturazione" del
%% termine di galleggiamento
v_discriminante_galleggiamento = 20; % m/s
%% ampiezza dell'intervallo di tempo necessario per verificare che una ripresa
%% di aderenza sia stabile in fase di slittamento
N_non_slittamento = round(2/T);
%% soglia di accelerazione utilizzata in fase di slittamento
a_massima_rif = 0.5; % m/s^2
%% accelerazione utilizzata in fase di slittamento se quella effettiva dell'asse
%% supera il valore massimo fissato
a_adeguamento_rif = 0.5; % m/s^2
%% tempo di salita del filtro del 1^ ordine usato per le singole

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
11 di 46

```
%% accelerazioni
tau_filtro_singole_accelerazioni = 0.5; % s

%% tempo di salita del filtro del 1^ ordine usato per
%% l'accelerazione media
tau_filtro_accelerazione_media = 4; % s

%% tempo di salita del filtro del 1^ ordine usato per
%% la stima di accelerazione
tau_filtro_stima = 8; % s

%% quando uno dei due assi si blocca momentaneamente, durante il bloccaggio
%% considero solo le informazioni relative all'altro asse per elaborare.
%% Quando l'asse riprende l'aderenza le informazioni relative alla sua velocità
%% non vengono subito considerate, ma si aspetta un intervallo di tempo 'sufficiente'
%% per assicurarsi che la velocità si sia riportata al livello dell'altro.
%% La lunghezza di tale intervallo di tempo dipende dalla velocità, ma non può
%% in ogni caso superare i 3 s.
%% Aspetto 1 s ogni 10 m/s di velocità
%% fino a un massimo di 4 s
N_asse_bloccato_max = round(4/T);

%% velocità
v_asse_bloccato = 145/3.6;

%% parametri per il calcolo del tempo di inizio pattinamento
v_max_inizio_pattinamento = 200/3.6;
v_rif_inizio_pattinamento = 160/3.6;
N_inizio_pattinamento_max = round(N_inizio_pattinamento_iniziale
*v_max_inizio_pattinamento/v_rif_inizio_pattinamento);
soglia_pattinamento_effettivo = 0.5;

%% Tempo di campionamento limite inferiore
T_CampF = 0; % s

%% Tempo di campionamento limite superiore
T_CampF = 1; % s

%% Costante di Tempo TAU_F limite inferiore
TAU_F = 0; % s
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
12 di 46

%% Costante di Tempo TAU\_F limite superiore  
TAU\_F = 1; % s

### CALCOLO PARAMETRI NECESSARI PER L'ELABORAZIONE

```
%% calcolo parametri per i filtri del primo ordine
%% 1 - filtro sulle singole accelerazioni
%% pulsazione di taglio
w_filtro_singole_accelerazioni = 1/tau_filtro_singole_accelerazioni;
%% numeratore z-trasformata filtro
num_filtro_iniziale = w_filtro_singole_accelerazioni*T;
%% secondo termine denominatore z-trasformata filtro
den_filtro_iniziale = -(1 - w_filtro_singole_accelerazioni*T);
%% 2 - filtro sull'accelerazione media
%% pulsazione di taglio
w_filtro_accelerazione_media = 1/tau_filtro_accelerazione_media;
%% numeratore z-trasformata filtro
num_filtro_accelerazione_media = w_filtro_accelerazione_media*T;
%% secondo termine denominatore z-trasformata filtro
den_filtro_accelerazione_media = -(1 - w_filtro_accelerazione_media*T);
%% 3 - filtro per la stima di accelerazione
%% pulsazione di taglio
w_filtro_stima = 1/tau_filtro_stima;
%% numeratore z-trasformata filtro
num_filtro_stima = w_filtro_stima*T;
%% secondo termine denominatore z-trasformata filtro
den_filtro_stima = (1 - w_filtro_stima*T);
%% calcolo il valore di accelerazione da usare per verificare il
%% pattinamento degli assi: questo valore è dato dalla somma di un
%% termine costante e della dr
soglia_acce_pattinamento = dr + incremento_decelerazione_ammesso;
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

13 di 46

%% dimensione massima vettori

MAX\_N

max([N\_asse\_bloccato\_max, N\_non\_pattinamento, N\_non\_slittamento, N\_commutazione, N\_inizio\_pattinamento\_max, N\_asse\_stabile]);

## 3.2 Calcola Odometria

### INIZIALIZZAZIONE VARIABILI PER IL PRIMO STEP

%% velocità dei due assi

v1d = v1(1)/3.6;

v2d = v2(1)/3.6;

v1d\_old = v1d;

v2d\_old = v2d;

%% accelerazione dei due assi

a1d = 0;

a2d = 0;

a1d\_old = 0;

a2d\_old = 0;

%% accelerazione filtrata dei due assi

a1df = 0;

a2df = 0;

a1df\_old = 0;

a2df\_old = 0;

%% filtro con BW più stretta

a1\_filtrata = 0;

a2\_filtrata = 0;

a1\_filtrata\_old = 0;

a2\_filtrata\_old = 0;

%% filtro con BW molto più stretta

a1\_molto\_filtrata = 0;

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

14 di 46

```

a2_molto_filtrata = 0;
a1_molto_filtrata_old = 0;
a2_molto_filtrata_old = 0;
%%      accelerazione media filtrata
a_media_filtrata = 0;
a_media_filtrata_old = 0;
%%      accelerazione media filtrata
a_media_molto_filtrata = 0;
a_media_molto_filtrata_old = 0;
%%      stima dell'accelerazione
a_stimata = 0;
a_stimata_store = 0;
a_stimata_old = 0;
count_inizio_pattinamento = 0;
%%      flag inizio pattinamento
FIP = FALSO;
%%      flag inizio slittamento
FISL = FALSO;
%%      variabili per la definizione delle condizioni di aderenza
%%      dei due assi
pattina1 = FALSO;
pattina2 = FALSO;
slitta1 = FALSO;
slitta2 = FALSO;
%%      variabile in cui viene memorizzato lo spazio percorso
spazio = 0;
spazio_old = 0;
assi_disponibili = ENTRAMBI;
%%      variabile per distinguere le fasi di accelerazione/frenatura
accelera_o_frena = ACCELERAZIONE;
%%      velocità stimata.. come primo valore si prende la media aritmetica

```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

15 di 46

```
%%      delle velocità dei due assi
vs = (v1d + v2d)*0.5;
vs_old = vs;
v_ref_slipping = min([v1d,v2d]);
%%      soglia per la commutazione
soglia_commutazione = soglia_per_slittamento;
%%      sto_adequando_asse = 1 quando la velocità stimata si sta adeguando
%%      a quella di uno dei due assi
sto_adequando_asse1 = FALSO;
sto_adequando_asse2 = FALSO;
a1df_finestra = zeros(N_asse_stabile,1);
a2df_finestra = zeros(N_asse_stabile,1);
asse1_stabile = zeros(MAX_N+2,1);
asse2_stabile = zeros(MAX_N+2,1);
asse1_bloccato = LIBERO *ones(MAX_N+2,1);
asse2_bloccato = LIBERO *ones(MAX_N+2,1);
asse1_libero = ones(MAX_N+2,1);
asse2_libero = ones(MAX_N+2,1);
assi_non_slittano = ones(MAX_N+2,1);
assi_non_pattinano = ones(MAX_N+2,1);
begin_of_a_sliding = 0;
begin_of_a_sliding_old = 0;
is_sliding_old = 0;
is_sliding = 0;
cosa = NON_PATTINAMENTO;
not_pattina1_and_not_pattina2 = VERO;
not_slitta1_and_not_slitta2 = VERO;
a_mean_gt_commutation_treshold = FALSO;
funziona = VERO;
commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);
td = zeros(length(tempo),1);
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
16 di 46

%% Tempo di campionamento valore di default

T\_CampF = 0.1 ;

%% Costante di Tempo TAU\_F valore di default

TAU\_F = 0.2 ;

store\_initialize\_20032003

for i = 2:length(tempo),

## 1 inizializzazione variabili

sto\_adequando\_asse1 = FALSO;

sto\_adequando\_asse2 = FALSO;

pattina1 = FALSO;

pattina2 = FALSO;

slitta1 = FALSO;

slitta2 = FALSO;

assi\_disponibili = NESSUNO;

could\_commutate = CANNOT\_COMMUTATE;

%% aggiornamento timer

td(i) = tempo(i);

## 2 calcolo della velocità dei due assi

% legge dai contatori gli impulsi

% count\_asse1 = count\_1(i);

% count\_asse2 = count\_2(i);

%

% impulsi\_metro1 = impulsi\_giro1 \* ris\_1(i) / (pi\*diametro1);

% impulsi\_metro2 = impulsi\_giro2 \* ris\_2(i) / (pi\*diametro2);

% % costanti per il calcolo della velocità

% if DeltaT(i) == 0

% kvel1 = 10000;



**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

17 di 46

```
%      kvel2 = 10000;
%  else
%      kvel1 = 1/impulsi_metro1/DeltaT(i);
%      kvel2 = 1/impulsi_metro2/DeltaT(i);
%  end
%  % prima ruota
%  if ris_1(i) == 0
%      kvel1 = 0;
%  else
%      kvel1 = kvel1;
%  end
%  % seconda ruota
%  if ris_2(i) == 0
%      kvel2 = 0;
%  else
%      kvel2 = kvel2;
%  end

% calcolo della velocità dei due assi
%asse 1
%  if kvel1 == 10000
%      v1d = v1d_old;
%  else
%      v1d = count_2_speed(count_asse1,count_asse1_old,kvel1,MAX_DIM);
%  end
%  % asse 2
%  if kvel2 == 10000
%      v2d = v2d_old;
%  else
%      v2d = count_2_speed(count_asse2,count_asse2_old,kvel2,MAX_DIM);
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

18 di 46

```
% end
v1d = v1(i)/3.6;;
v2d = v2(i)/3.6;
%%      leggo il valore della pendenza
di = di_vettore_odo(i);
%%      modificato per tenere conto della VARIAZIONE DEL TEMPO DI
CAMPIONAMENTO
%%      calcolo parametri per i filtri del primo ordine
%%      1 - filtro sulle singole accelerazioni
%%      pulsazione di taglio
w_filtro_singole_accelerazioni = 1/tau_filtro_singole_accelerazioni;
%%      numeratore z-trasformata filtro
num_filtro_iniziale = w_filtro_singole_accelerazioni*DeltaT(i);
%%      secondo termine denominatore z-trasformata filtro
den_filtro_iniziale = (1 - w_filtro_singole_accelerazioni*DeltaT(i));
%%      2 - filtro sull'accelerazione media
%%      pulsazione di taglio
w_filtro_accelerazione_media = 1/tau_filtro_accelerazione_media;
%%      numeratore z-trasformata filtro
num_filtro_accelerazione_media = w_filtro_accelerazione_media*DeltaT(i);
%%      secondo termine denominatore z-trasformata filtro
den_filtro_accelerazione_media = (1 - w_filtro_accelerazione_media*DeltaT(i));
%%      3 - filtro per la stima di accelerazione
%%      pulsazione di taglio
w_filtro_stima = 1/tau_filtro_stima;
%%      numeratore z-trasformata filtro
num_filtro_stima = w_filtro_stima*DeltaT(i);
%%      secondo termine denominatore z-trasformata filtro
den_filtro_stima = (1 - w_filtro_stima*DeltaT(i));
```

### **3 calcolo delle accelerazioni dei due assi**

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

19 di 46

%% calcola le accelerazioni dei due assi per mezzo delle differenze  
%% finite del primo ordine all'indietro

if DeltaT(i) == 0

a1d = a1d\_old;

a2d = a2d\_old;

a1df = a1df\_old;

a2df = a2df\_old;

a1\_filtrata = a1\_filtrata\_old;

a2\_filtrata = a2\_filtrata\_old;

a1\_molto\_filtrata = a1\_molto\_filtrata\_old;

a2\_molto\_filtrata = a2\_molto\_filtrata\_old;

else

a1d = (v1d - v1d\_old)/DeltaT(i);

a2d = (v2d - v2d\_old)/DeltaT(i);

%% filtro singole accelerazioni (0.5 s)

a1df = filtro(num\_filtro\_iniziale,den\_filtro\_iniziale,a1d,a1df\_old);

a2df = filtro(num\_filtro\_iniziale,den\_filtro\_iniziale,a2d,a2df\_old);

%% filtro su accelerazione media (4 s)

a1\_filtrata

filtro(num\_filtro\_accelerazione\_media,den\_filtro\_accelerazione\_media,...

a1d,a1\_filtrata\_old);

a2\_filtrata

filtro(num\_filtro\_accelerazione\_media,den\_filtro\_accelerazione\_media,...

a2d,a2\_filtrata\_old);

%% filtro su accelerazione media (8 s)

a1\_molto\_filtrata = filtro(num\_filtro\_stima,den\_filtro\_stima,...

a1d,a\_stimata\_old);

a2\_molto\_filtrata = filtro(num\_filtro\_stima,den\_filtro\_stima,...

a2d,a\_stimata\_old);

end

#### **4 variabili che dipendono dalla velocità stimata e dalla pendenza**

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
20 di 46

```

%%      4.1 - calcolo della lunghezza dell'intervallo di 'ritardo' con cui
%%      ripristinare le 2 velocità in seguito al bloccaggio di un asse
      N_asse_bloccato = round(lin_cost(vs_old,v_asse_bloccato,N_asse_bloccato_max));
      N_asse_bloccato = max(2,N_asse_bloccato);

%%      4.2 - tratto a velocità costante inizio pattinamento
      N_inizio_pattinamento
      round(lin_cost(vs_old,v_max_inizio_pattinamento,N_inizio_pattinamento_max));
      N_inizio_pattinamento = max(N_inizio_pattinamento,2);

%%      4.3 - delta v 'galleggiamento'
      maggiorazione_v
      lin_cost(vs_old,v_discriminante_galleggiamento,maggiorazione_v_iniziale);

%%      4.4 - la soglia per la commutazione trazione/frenatura dipende dalla
pendenza
      soglia_commutazione = - di + soglia_per_slittamento;

%%      4.5 - accelerazione massima da utilizzare in fase di slittamento dei due assi
      a_massima = a_massima_rif - di;

%%      4.6 - accelerazione da applicare in fase di slittamento se l'accelerazione
dell'asse
      supera il valore massimo ammissibile
      a_adeguamento_1 = a_adeguamento_rif - di;

%%      4.7
      LT_d_media_clippata = drm + di;

%%      4.8
      HT_d_media_clippata = di + kprime*dr;

```

## 5 confronto delle velocità con i valori di soglia

```

%%      5.0 confronto tra due campioni successivi dello stesso asse
Polo_p1 = 1- T_CampF / TAU_F
V1d = Polo_1*V1d_precedente + (1 - Polo_p1)*v1d

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
21 di 46

```

V2d = Polo_1*V2d_recedente + (1 - Polo_p1)*v2d
V1d_precedente = V1d
V2d_precedente = V2d
%%      5.1 confronto tra le velocità dei due assi
v1d_gt_v2d = (v1d > v2d);

%%      5.2 confronto con la soglia di velocità nulla per valutare se
%%      gli assi sono bloccati o meno
v1_at_rest = (v1d < soglia_vel_nulla);
v2_at_rest = (v2d < soglia_vel_nulla);

%%      5.3 confronto della velocità degli assi con quella stimata
%%      per valutare se deve essere effettuato o meno il tratto
%%      a velocità costante all'inizio di un pattinamento
vs_far_from_v_wheels = (vs_old > (max(v1d,v2d) + delta_v_adeguamento));

%%      5.4 confronto la differenza tra le velocità dei due assi
%%      con la soglia di pattinamento/slittamento
%%      DELTA_V_WITHIN_LIM = 0 SE IL DELTA V SUPERA LA SOGLIA
%%      DELTA_V_WITHIN_LIM = 1 SE IL DELTA V NON SUPERA LA SOGLIA
deltav_within_lim = (abs(v1d - v2d) <= soglia_var_velo);

%%      5.5 confronto la differenza tra le velocità dei due assi
%%      con il valore di 'tolleranza' utilizzato nella valutazione
%%      della stabilità di un asse
deltav_within_tol = (abs(v1d - v2d) <= tolleranza);

```

## **6 confronto delle accelerazioni dei due assi con i valori di soglia**

```

%%      6.1 confronto delle accelerazioni dei due assi con le soglie
%%      di pattinamento/ slittamento
%%      6.1.1 pattinamento
%%      acc1_within_lim = 1 se non viene superata la soglia
%%      acc1_within_lim = 0 se viene superata la soglia

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
22 di 46

```
acc1_within_sliding_lim = (a1df > -soglia_acce_pattinamento);
acc2_within_sliding_lim = (a2df > -soglia_acce_pattinamento);
%%      6.1.2 slittamento
acc1_within_slipping_lim = (a1df < soglia_acce_slittamento);
acc2_within_slipping_lim = (a2df < soglia_acce_slittamento);
```

## **7 valutazione delle variabili asse1\_bloccato ed asse2\_bloccato**

```
%%      7.1
%%      UN ASSE PUO' ESSERE CONSIDERATO LIBERO SE
%%      - LA SUA VELOCITA' SUPERA LA SOGLIA DI VELOCITA' NULLA
%%      - TALE CONDIZIONE SI MANTIENE PER ALMENO N_ASSE_BLOCCATO
CAMPIONI,
%%      DOVE N_ASSE_BLOCCATO CAMPIONI E' FUNZIONE DELLA VELOCITA'
STIMATA
asse1_libero = true_for_a_while(asse1_libero,N_asse_bloccato+1,MAX_N+2,(~
v1_at_rest));
asse2_libero = true_for_a_while(asse2_libero,N_asse_bloccato+1,MAX_N+2,(~
v2_at_rest));
%%      7.2
asse1_bloccato = 1 - asse1_libero;
asse2_bloccato = 1 - asse2_libero;
```

## **assegnazione variabile "assi\_disponibili" (assi disponibili per l'elaborazione)**

```
%%      VALUTAZIONE DEL NUMERO DELLE INFORMAZIONI DISPONIBILI

if (~(asse1_bloccato(1) == LIBERO)) & (~(asse2_bloccato(1) == LIBERO))
%%      8.1
asse_disponibili = NESSUNO;
elseif (~(asse1_bloccato(1) == LIBERO)) & (asse2_bloccato(1) == LIBERO)
%%      8.2
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
23 di 46

```

    assi_disponibili = ASSE2;
elseif (asse1_bloccato(1) == LIBERO) & ~(asse2_bloccato(1) == LIBERO))
%%      8.3
    assi_disponibili = ASSE1;
else
%%      8.4
    assi_disponibili = ENTRAMBI;
end

```

## 9 calcolo dell'accelerazione media

```

%%      il calcolo dell'accelerazione media viene effettuato
%%      con procedure diverse a seconda del numero di informazioni
%%      disponibili

switch assi_disponibili
case ASSE1
%%      9.1
%%      asse 2 bloccato - prendo solo l'accelerazione dell'asse 1
    a_media = a1df;
    a_media_filtrata = a1_filtrata;
    a_media_molto_filtrata = a1_molto_filtrata;

%%      NELLA VALUTAZIONE DELLA D MEDIA CLIPPATA CONSIDERO SOLO
%%      L'INFORMAZIONE RELATIVA ALL'ASSE 1
    d_media_clippata = condiziona(LT_d_media_clippata,HT_d_media_clippata,...
        (maggiorazione_decelerazione - a1_filtrata));

    a_stima_riferimento =
condiziona(LT_d_media_clippata,HT_d_media_clippata,...
        (maggiorazione_decelerazione - a1_molto_filtrata));

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

24 di 46

```
a_mean_gt_commutation_treshold = (a_media > soglia_commutazione);
a_filt_gt_commutation_treshold = (a_media_filtrata > soglia_commutazione);
```

```
%%      controlla accelerazione asse 1
slitta1 = ~(acc1_within_slipping_lim);
pattina1 = ~(acc1_within_sliding_lim);
%%      velocità da utilizzare in caso di slittamento
v_ref_slipping = v1d;
```

```
a_ref_slipping = a1df;
```

```
case ASSE2
```

```
%%      9.2
%%      asse 1 bloccato - prendo solo l'accelerazione dell'asse 2
a_media = a2df;
a_media_filtrata = a2_filtrata;
```

```
a_media_molto_filtrata = a2_molto_filtrata;
```

```
d_media_clippata = condiziona(LT_d_media_clippata,HT_d_media_clippata,...
(maggiorazione_decelerazione - a2_filtrata));
```

```
a_stima_riferimento = -
condiziona(LT_d_media_clippata,HT_d_media_clippata,...
(maggiorazione_decelerazione - a2_molto_filtrata));;
```

```
a_mean_gt_commutation_treshold = (a_media > soglia_commutazione);
```

```
a_filt_gt_commutation_treshold = (a_media_filtrata > soglia_commutazione);
```

```
%%      controlla accelerazione asse 2
```



**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
25 di 46

```
slitta2 = ~(acc2_within_slipping_lim);
pattina2 = ~(acc2_within_sliding_lim);
```

```
%%      velocità da utilizzare in caso di slittamento
v_ref_slipping = v2d;
```

```
a_ref_slipping = a2df;
```

case ENTRAMBI

```
%%      9.3
```

```
%%      gli assi non sono bloccati
```

```
%%      9.3.1
```

```
%%      9.3.1.1 ACCELERAZIONE (MEDIA DELLE ACCELERAZIONI DEGLI ASSI
FILTRATE) FILTRO 0.5 S
```

```
    a_media = 0.5 * (a1df + a2df);
```

```
%%      9.3.1.2 ACCELERAZIONE MEDIA FILTRO 4 S
```

```
    a_media_filtrata
```

```
    filtro(num_filtro_accelerazione_media,den_filtro_accelerazione_media,...
           0.5*(a1d+a2d),a_media_filtrata_old);
```

```
%%      ACCELERAZIONE MEDIA FILTRO 8 S
```

```
    a_media_molto_filtrata = filtro(num_filtro_stima,den_filtro_stima,...
                                     0.5*(a1d+a2d),a_media_molto_filtrata_old);
```

```
%%      9.3.1.3 D MEDIA CLIPPATA
```

```
    d_media_clippata = condiziona(LT_d_media_clippata,HT_d_media_clippata,...
                                  (maggiorazione_decelerazione - a_media_filtrata));
```

```
    a_stima_riferimento
```

```
    condiziona(LT_d_media_clippata,HT_d_media_clippata,...
               (maggiorazione_decelerazione - a_media_molto_filtrata));
```

```
%%      9.3.2
```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
26 di 46

```

%%      9.3.2
%%      L'ASSE 1 NON PATTINA SE:
%%      - 1 IL VALORE ASSOLUTO DELLA DIFFERENZA TRA LE VELOCITA' DEI
DUE ASSI
%%      E' MINORE DELLA SOGLIA FISSATA (DEF. PUNTO 5.4)
%%      - 2 L'ACCELERAZIONE DELL'ASSE SUPERA LA SOGLIA FISSATA PER
IL PATTINAMENTO(NEGATIVA)
%%      PATTINA = 0 - NON PATTINAMENTO
%%      PATTINA = 1 - PATTINAMENTO
pattina1 = ~(deltav_within_lim & acc1_within_sliding_lim);
pattina2 = ~(deltav_within_lim & acc2_within_sliding_lim);

%%      9.3.3
slitta1 = ~(deltav_within_lim & acc1_within_slipping_lim);
slitta2 = ~(deltav_within_lim & acc2_within_slipping_lim);

%%      9.3.4
%%      confronto dell'accelerazione media
%%      con la soglia per a commutazione da trazione a frenatura
%%      e viceversa
a_mean_gt_commutation_treshold = (a_media > soglia_commutazione);
a_filt_gt_commutation_treshold = (a_media_filtrata > soglia_commutazione);

%%      9.3.5
%%      se le condizioni di aderenza vengono perse a causa
%%      del superamento da parte del delta_v del limite
%%      massimo si considera il segno dell'accelerazione media
%%      per valutare se è un pattinamento o uno slittamento
if ~deltav_within_lim & (~FIP) & (~FISL)
%%      accelera_o_frena = 0 - frena
%%      accelera_o_frena = 1 - accelera
accelera_o_frena = a_mean_gt_commutation_treshold;
end % fine controllo deltatv

```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

27 di 46

%%            velocità da utilizzare in caso di slittamento

v\_ref\_slipping = min([v1d,v2d]);

if v1d\_gt\_v2d

    a\_ref\_slipping = a2df;

else

    a\_ref\_slipping = a1df;

end

case NESSUNO

%%            entrambi gli assi sono bloccati

a\_media = 0;

a\_media\_filtrata = 0;

a\_media\_molto\_filtrata = 0;

a\_stima\_riferimento = 0;

d\_media\_clippata = dr;

a\_mean\_gt\_commutation\_treshold = (a\_media > soglia\_commutazione);

a\_filt\_gt\_commutation\_treshold = (a\_media\_filtrata > soglia\_commutazione);

slitta1 = VERO;

slitta2 = VERO;

pattina1 = VERO;

pattina2 = VERO;

accelera\_o\_frena = FRENA;

%%            velocità da utilizzare in caso di slittamento

v\_ref\_slipping = min([v1d,v2d]);

if v1d\_gt\_v2d

    a\_ref\_slipping = a2df;

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

28 di 46

```
else
    a_ref_slipping = a1df;
end
```

```
otherwise
    break
end
```

## 10 confronto per slittamento

```
a_ref_slipping_gt_a_massima = (a_ref_slipping > a_massima);
```

## 11 valutazione della stabilità degli assi

```
%%      11.1 calcolo della massima variazione di accelerazione dei 2 assi
%%      nei 2 s che precedono l'istante considerato
```

```
delta_a1df = max(a1df_finestra) - min(a1df_finestra);
```

```
delta_a2df = max(a2df_finestra) - min(a2df_finestra);
```

```
%%      11.2 CONFRONTA LA MASSIMA VARIAZIONE DI ACCELERAZIONE CON LA SOGLIA
```

```
jerk1_within_lim = (delta_a1df < SOGLIA_JERK);
```

```
jerk2_within_lim = (delta_a2df < SOGLIA_JERK);
```

```
%%      11.3 VALUTAZIONE VARIABILI V1_COULD BE STABLE E V2 COULD BE STABLE
```

```
if deltav_within_tol
```

```
%%      la differenza tra le velocità dei due assi è piccola
```

```
%%      controllo la 'stabilità' di entrambi
```

```
%%      controllo su accelerazione e jerk asse 1
```

```
v1_could_be_stable =(acc1_within_sliding_lim) & (jerk1_within_lim);
```

```
v2_could_be_stable =(acc2_within_sliding_lim) & (jerk2_within_lim);
```

```
elseif v1d_gt_v2d
```

```
%asse 1 più veloce
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
29 di 46

```

v1_could_be_stable =(acc1_within_sliding_lim) & (jerk1_within_lim);
v2_could_be_stable = FALSO;
else % v2d > v1d
%%      asse 2 più veloce
v2_could_be_stable =(acc2_within_sliding_lim) & (jerk2_within_lim);
v1_could_be_stable = FALSO;
end

%%      VALUTAZIONE VARIABILI ASSE1_STABILE E ASSE2_STABILE
asse1_stabile
true_for_a_while(asse1_stabile,N_asse_stabile+2,MAX_N+2,v1_could_be_stable);
asse2_stabile
true_for_a_while(asse2_stabile,N_asse_stabile+2,MAX_N+2,v2_could_be_stable);

```

## 12 condizioni di pattinamento/slittamento degli assi

```

%%      12.1
not_slitta1_and_not_slitta2 = (~slitta1) & (~slitta2);
not_pattina1_and_not_pattina2 = (~pattina1) & (~pattina2);
%%      12.2
assi_non_slittano
true_for_a_while(assi_non_slittano,N_non_slittamento+2,MAX_N+2,not_slitta1_and_n
ot_slitta2);
assi_non_pattinano
true_for_a_while(assi_non_pattinano,N_non_pattinamento+2,MAX_N+2,...
(not_pattina1_and_not_pattina2 & ~begin_of_a_sliding));
%%      12.3
ripresa_di_aderenza = FIP | FISL;
%%      12.4
condizione_fine_slittamento = (assi_non_slittano(1)== NON_SLITTA) &
(((assi_disponibili == ENTRAMBI) | ...
(assi_disponibili == NESSUNO) |...
((assi_disponibili == ASSE1) & ripresa_di_aderenza & jerk1_within_lim) |...

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

30 di 46

```
((assi_disponibili == ASSE2) & ripresa_di_aderenza & jerk2_within_lim) |...
((assi_disponibili == ASSE1) & ~ripresa_di_aderenza )| ...
((assi_disponibili == ASSE2) & ~ripresa_di_aderenza ));
%%      12.5
condizione_fine_pattinamento = (assi_non_pattinano(1)==NON_PATTINA) &
(((assi_disponibili == ENTRAMBI) | ...
(assi_disponibili == NESSUNO) |...
((assi_disponibili == ASSE1) & ripresa_di_aderenza & jerk1_within_lim) |...
((assi_disponibili == ASSE2) & ripresa_di_aderenza & jerk2_within_lim) |...
((assi_disponibili == ASSE1) & ~ripresa_di_aderenza )| ...
((assi_disponibili == ASSE2) & ~ripresa_di_aderenza ))) ;
```

### 13 inizio pattinamento

```
%%      13.1
%%      IS SLIDING = 1 SE:
%%      -(NON SIAMO IN FASE DI NON_PATTINAMENTO (GLI ASSI STANNO
PATTINANDO)
%%      AND
%%      E' UN INIZIO PATTINAMENTO/
%%      AND
%%      LA VELOCITA' STIMATA E' 'VICINA' A QUELLA DEGLI ASSI)
%%      OPPURE
%%      (NON SIAMO IN FASE DI NON_PATTINAMENTO (GLI ASSI STANNO
PATTINANDO)
%%      AND
%%      NON E' UN INIZIO PATTINAMENTO)
%%      --> POTREI EFFETTUARE IL TRATTO A V COSTANTE
%%      IS SLIDING = 0 SE SIAMO IN FASE DI NON_PATTINAMENTO OPPURE
SE LA VELOCITA' STIMATA
%%      NON E' 'VICINA' A QUELLA DEGLI ASSI
%%      --> NON POSSO EFFETTUARE IL TRATTO A V COSTANTE
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
31 di 46

```
%is_sliding = ((~(not_pattina1_and_not_pattina2 & condizione_fine_pattinamento) &
FIP)|...
%   (~(not_pattina1_and_not_pattina2 & condizione_fine_pattinamento) & ~FIP &
(~vs_far_from_v_wheels)))&...
%   (~(assi_disponibili == NESSUNO));
is_sliding = ((cosa == NON_PATTINAMENTO | cosa == NON_SLITTAMENTO |cosa
== SLITTAMENTO |cosa == ADERENZA_NON_CERTA_SLITTAMENTO)...
&   (~vs_far_from_v_wheels)    &   ~not_pattina1_and_not_pattina2    &
accelera_o_frena == FRENA);
%%      inizializza il contatore quando riconosce l'inizio di un pattinamento
if (cosa == NON_PATTINAMENTO | cosa == NON_SLITTAMENTO |cosa ==
SLITTAMENTO |cosa == ADERENZA_NON_CERTA_SLITTAMENTO)
count_inizio_pattinamento = 0;
end

%%      13.2
%%      variabile BEGIN OF A SLIDING per valutare l'inizio di un pattinamento
%%      ovvero per decidere se effettuare il tratto a velocità costante
%% BEGIN OF A SLIDING(1) = 0  SE IS_SLIDING = 0
%% BEGIN OF A SLIDING(1) = 0.5 SE IS_SLIDING = 1, MA TALE CONDIZIONE SI
E' VERIFICATA PER 'POCO',
%%      TEMPO, OVVERO BEGIN_OF_A_SLIDING(N_INIZIO_PATTINAMENTO) =
0
%% BEGIN OF A SLIDING(1) = 1  SE IS_SLIDING = 1, E TALE CONDIZIONE SI E'
VERIFICATA PER UN
%%      TEMPO 'SUFFICIENTE',  OVVERO
BEGIN_OF_A_SLIDING(N_INIZIO_PATTINAMENTO) = 0.5
%%      OPPURE BEGIN_OF_A_SLIDING(N_INIZIO_PATTINAMENTO) = 1

%%      IL TRATTO A VELOCITA' COSTANTE VIENE EFFETTUATO
NELL'INTERVALLO DI TEMPO IN CUI SI HA
%% BEGIN OF A SLIDING = 0.5 (INIZIO PATTINAMENTO)
if is_sliding | (begin_of_a_sliding_old & (count_inizio_pattinamento <=
N_inizio_pattinamento))
%%      inizio pattinamento, tratto a velocità costante
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
32 di 46

```

count_inizio_pattinamento = count_inizio_pattinamento + 1;
begin_of_a_sliding = VERO;
else
%%      fine del tratto a velocità costante, reset del contatore
count_inizio_pattinamento = 0;
begin_of_a_sliding = FALSO;
end
%begin_of_a_sliding
true_for_a_while(begin_of_a_sliding,N_inizio_pattinamento,MAX_N,is_sliding);
=
```

#### 14 variabile COSA

```

if (accelera_o_frena == ACCELERERA)

    if not_slitta1_and_not_slitta2
        if condizione_fine_slittamento
%%      14.1
            cosa = NON_SLITTAMENTO;
        else
%%      14.2
            cosa = ADERENZA_NON_CERTA_SLITTAMENTO;
        end
    else
%%      14.3
%%      ALMENO UNO DEI DUE ASSI E' IN SLITTAMENTO
        cosa = SLITTAMENTO;
    end
    elseif not_pattina1_and_not_pattina2
%%      FRENATURA
%%      NESSUN ASSE STA PATTINANDO
        if condizione_fine_pattinamento
%%      14.4
```



**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

33 di 46

```

        cosa = NON_PATTINAMENTO;
    elseif (begin_of_a_sliding)
%%      14.5
%%      - ACCELERA O FRENA = FRENA
%%      - GLI ASSI NON PATTINANO MA LA RIPRESA DI ADERENZA NON SI
%%      E' MANTENUTA PER UN TRATTO 'SUFFICIENTE'
%%      - IL PATTINAMENTO E' INIZIATO DA 'POCO' TEMPO
%%      ---> EFFETTUA IL TRATTO A VELOCITA' COSTANTE
        cosa = TRATTO_A_V_COSTANTE;
    else
%%      14.6
%%      - ACCELERA O FRENA = FRENA
%%      - GLI ASSI NON PATTINANO MA LA RIPRESA DI ADERENZA
%%      NON SI E' MANTENUTA PER UN TRATTO 'SUFFICIENTE'
%%      - IL PATTINAMENTO E' INIZIATO DA UN TEMPO SUFFICIENTE
%%      ---> ADERENZA NON CERTA IN PATTINAMENTO
        cosa = ADERENZA_NON_CERTA_PATTINAMENTO;
    end
    elseif begin_of_a_sliding
%%      14.7
%%      - ACCELERA O FRENA = FRENA
%%      - GLI ASSI STANNO PATTINANDO
%%      - IL PATTINAMENTO E' INIZIATO DA 'POCO' TEMPO
%%      ---> EFFETTUA IL TRATTO A VELOCITA' COSTANTE
        cosa = TRATTO_A_V_COSTANTE;
    else
%%      14.8
        cosa = PATTINAMENTO;
    end

```

**15 Switch su COSA**

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
34 di 46

```

switch cosa
case NON_SLITTAMENTO
%%      15.1 - LA VELOCITA' STIMATA E' PARI AL MIN(V1D,V2D)
        could_commute = ~(a_filt_gt_commutation_treshold);
        commutation_control
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commute);
%%      condizione sull'asse singolo...
%%      se l'asse 1 è bloccato e la massima variazione di accelerazione
%%      dell'asse 2 è limitata
%%      è passato un intervallo di tempo sufficiente da quando si
%%      è verificata la fine dello slittamento, elaboro con la
%%      procedura prevista in caso di aderenza (non slittamento)
        FIP = FALSO;
        FISL = FALSO;
%%      in fase di aderenza posso commutare dalla fase di trazione a
%%      quella di frenatura sulla base del segno dell'accelerazione media
%%      filtrata con il filtro da 0.5 s
        if a_mean_gt_commutation_treshold
%%      mantieni
            accelera_o_frena = ACCELERA;
        else
%%      commuta
            accelera_o_frena = FRENA;
        end
%%      reset del contatore x controllo commutazione
        commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);
%%      la velocità stimata è pari al minimo tra v1d e v2d (a meno che
%%      questa non sia inferiore alla soglia di velocità nulla

        vs = max(v1d,v2d);

```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
35 di 46

```

%%      stima dell'accelerazione
%%      in aderenza prendo l'accelerazione media filtrata a 0.5 s
a_stimata = a_media_filtrata;

case ADERENZA_NON_CERTA_SLITTAMENTO

%%      15.2
%%      gli assi sono stabili ma la condizione non si è ancora mantenuta
%%      per un tempo sufficiente, elaboro con la procedura prevista in condizioni
%%      di aderenza non certa in slittamento

      could_commute = ~(a_filt_gt_commutation_treshold);
      commutation_control
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commute) =
te);
%%      la commutazione trazione/frenatura viene effettuata se l'accelerazione
%%      media, filtrata con il filtro da 4s, cambia segno e se tale variazione
%%      di segno si mantiene per almeno 4 s
%%      controllo se è possibile effettuare la commutazione,
%%      in caso affermativo:
%%      - ridefinisco la variabile accelera_o_frena
%%      - effettuo il reset del vettore commutation_control
%%      - ridefinisco la variabile could_commute
if commutation_control(1) == CAN_COMMUTATE
      accelera_o_frena = FRENA;
      commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);
      could_commute = FALSO;
end

if a_ref_slipping_gt_a_massima
      vs = adegua_sl(vs_old,v_ref_slipping,a_adeguamento_1,T);

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

36 di 46

```

else
    vs = v_ref_slipping;
end

%%      stima dell'accelerazione
%%      in slittamento prendo l'accelerazione media filtrata a 4 s
a_stimata = a_media_molto_filtrata;

case SLITTAMENTO
%%      15.3
%%      LA VELOCITA' STIMATA E' PARI AL MAX(V1D,V2D)
could_commute = ~(a_filt_gt_commutation_treshold);
commutation_control =
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commute);
%%      almeno uno dei due assi sta slittando, elaboro con la procedura
%%      prevista in caso di slittamento
FISL = VERO;
%%      la commutazione trazione/frenatura viene effettuata se l'accelerazione
%%      media, filtrata con il filtro da 4s, cambia segno e se tale variazione
%%      di segno si mantiene per almeno 4 s
%%      controllo se è possibile effettuare la commutazione,
%%      in caso affermativo:
%%      - ridefinisco la variabile accelera_o_frena
%%      - effettuo il reset del vettore commutation_control
%%      - ridefinisco la variabile could_commute
if commutation_control(1) == CAN_COMMUTATE
    accelera_o_frena = FRENA;
    commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);
    could_commute = FALSO;
end
%%      ELABORAZIONE V STIMATA

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

37 di 46

```

%%      SELEZIONE DELL'ASSE + LENTO
      if a_ref_slipping_gt_a_massima
%%      SE L'ACCELERAZIONE SUPERA IL VALORE MASSIMO
%%      INTEGRA CON L'ACC. A_ADEGUAMENTO_1
      vs = adegua_sl(vs_old,v_ref_slipping,a_adeguamento_1,T);
    else
%%      LA VELOCITA' STIMATA E' UGUALE A QUELLA
%%      DELL'ASSE + LENTO
      vs = v_ref_slipping;
    end

%%      stima dell'accelerazione
%%      in slittamento prendo l'accelerazione media filtrata a 4 s
      a_stimata = a_media_molto_filtrata;

    case NON_PATTINAMENTO
%%      15.4
%%      valutazione della variabile could_commutate:
%%      could_commutate = VERO se l'accelerazione media filtrata supera la soglia
di commutazione
%%      could_commutate = FALSO se l'accelerazione media filtrata è minore della
soglia di commutazione
      could_commutate = (a_filt_gt_commutation_treshold);
%%      valutazione della variabile commutation_control

      commutation_control
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commuta
te);

      FIP = FALSO;
      FISL = FALSO;
      if a_mean_gt_commutation_treshold

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

38 di 46

```

%%      commuta
      accelera_o_frena = ACCELERERA;
    else
%%      mantieni
      accelera_o_frena = FRENA;
    end

%%      reset del contatore x controllo commutazione
      commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);

      if v1d_gt_v2d
        vs = adeguamento(vs_old,v1d,d_adequamento,T);
        sto_adequando_asse1 = controlla_adequamento(vs,v1d);
      else
%%      v2d > v1d
        vs = adeguamento(vs_old,v2d,d_adequamento,T);
        sto_adequando_asse2 = controlla_adequamento(vs,v2d);
      end

%%      paracadute...
      vs = max([v1d,v2d,vs]);

%%      stima dell'accelerazione
%%      in aderenza prendo l'accelerazione media filtrata a 0.5 s
      a_stimata = a_media_filtrata;

      case ADERENZA_NON_CERTA_PATTINAMENTO
%%      15.5
        could_commute = (a_filt_gt_commutation_treshold);
        commutation_control =
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commute);

%%      la commutazione trazione/frenatura viene effettuata se l'accelerazione
%%      media, filtrata con il filtro da 4s, cambia segno e se tale variazione
%%      di segno si mantiene per almeno 4 s

```

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO

39 di 46

```

%%      controllo se è possibile effettuare la commutazione,
%%      in caso affermativo:
%%      - ridefinisco la variabile accelera_o_frena
%%      - effettuo il reset del vettore commutation_control
%%      - ridefinisco la variabile could_commutate
if commutation_control(1) == CAN_COMMUTATE
    accelera_o_frena = ACCELERERA;
    commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);
    could_commutate = FALSO;
end

if asse1_stabile(1) == STABILE
%%      asse 1 stabile
    if ~v1_at_rest
        vs = adeguamento(vs_old,v1d,d_adeguamento,T);
    else
        vs = adeguamento(vs_old,v1d,dr,T);
    end
    sto_adequando_asse1 = controlla_adeguamento(vs,v1d);
%%      paracadute...
    vs = max([v1d,v2d,vs]);
elseif asse2_stabile(1) == STABILE
%%      asse 2 stabile
    if ~ v2_at_rest
        vs = adeguamento(vs_old,v2d,d_adeguamento,T);
    else
        vs = adeguamento(vs_old,v2d,dr,T);
    end
    sto_adequando_asse2 = controlla_adeguamento(vs,v2d);
%%      paracadute...
    vs = max([v1d,v2d,vs]);

```

**SCMT**

 Codifica: **RFI TC.PATC SR CM 03 M99 B**

 FOGLIO  
 40 di 46

```

    else
%%      non ci sono assi stabili a cui adeguare la v stimata,
%%      procedo per integrazione usando la d media clippata
      vs = integra_in_frenata(vs_old,d_media_clippata,T);
%%      galleggiamento
      vs = max([v1d+maggiorazione_v,v2d+maggiorazione_v,vs]);
    end

%%      stima dell'accelerazione
%%      in pattinamento prendo -d_media_clippata
      a_stimata = a_stima_riferimento;

  case PATTINAMENTO
%%      15.6
%%      almeno uno dei due assi sta pattinando, seguo la procedura
%%      prevista in caso di pattinamento
%%      15.6.1 COMMUTAZIONE

      could_commute = (a_filt_gt_commutation_treshold);
      commutation_control =
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commute);
%%      la commutazione trazione/frenatura viene effettuata se l'accelerazione
%%      media, filtrata con il filtro da 4s, cambia segno e se tale variazione
%%      di segno si mantiene per almeno 4 s
%%      controllo se è possibile effettuare la commutazione,
%%      in caso affermativo:
%%      - ridefinisco la variabile accelera_o_frena
%%      - effettuo il reset del vettore commutation_control
%%      - ridefinisco la variabile could_commute
      if (commutation_control(1) == CAN_COMMUTATE)
        accelera_o_frena = ACCELERA;
        commutation_control = CANNOT_COMMUTATE*ones(MAX_N+2,1);

```



**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
41 di 46

```

        could_commutate = FALSO;
    end

%%      15.6.2 FLAG INIZIO PATTINAMENTO
%%      FLAG INIZIO PATTINAMENTO, DURANTE IL PATTINAMENTO IL SUO
%%      VALORE E' "VERO"
        FIP = VERO;

%%      15.6.3 ELABORAZIONE VELOCITA'
%%      INIZIO ELABORAZIONE DELLA VELOCITA' STIMATA
    if asse1_stabile(1) == STABILE
%%      15.6.3.1 ASSE 1 STABILE
%%      asse 1 stabile
        if ~v1_at_rest
            vs = adeguamento(vs_old,v1d,d_adequamento,T);
        else
            vs = adeguamento(vs_old,v1d,dr,T);
        end
        sto_adequando_asse1 = controlla_adequamento(vs,v1d);
%%      paracadute...
        vs = max([v1d,v2d,vs]);
    elseif asse2_stabile(1) == STABILE
%%      15.6.3.2 asse 2 stabile
        if ~v2_at_rest
            vs = adeguamento(vs_old,v2d,d_adequamento,T);
        else
            vs = adeguamento(vs_old,v2d,dr,T);
        end
        sto_adequando_asse2 = controlla_adequamento(vs,v2d);
%%      paracadute...
        vs = max([v1d,v2d,vs]);

    else

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

42 di 46

```

%%      15.6.3.3 NESSUNO DEI DUE ASSI E' STABILE
%%      non ci sono assi stabili a cui adeguare la v stimata,
%%      procedo per integrazione usando la d media clippata
      vs = integra_in_frenata(vs_old,d_media_clippata,T);
%%      SE NESSUNO DEI DUE ASSI PUO' ESSERE CONSIDERATO STABILE
%%      LA VELOCITA' STIMATA DEVE ESSERE MAGGIORE DI QUELLA DEI
DUE ASSI
%%      INCREMENTATA galleggiamento
      vs = max([v1d+maggiorazione_v,v2d+maggiorazione_v,vs]);
    end
%%      stima dell'accelerazione
%%      in pattinamento prendo -d_media_clippata
      a_stimata = a_stima_riferimento;

    case TRATTO_A_V_COSTANTE
%%      15.7
%%      LA COMMUTAZIONE FRENATURA/ACCELERAZIONE VIENE
EFFETTUATA SE
%%      L'ACCELERAZIONE MEDIA FILTRATA (4 S) SUPERA LA SOGLIA DI
COMMUTAZIONE
%%      PER UN TEMPO 'SUFFICIENTE'

%%      CONFRONTO L'ACCELERAZIONE MEDIA FILTRATA CON LA SOGLIA
DI COMMUTAZIONE
      could_commute = (a_filt_gt_commutation_treshold);

%%      VERIFICO CHE LA CONDIZIONE SI SIA MANTENUTA PER ALMENO 4 S
      commutation_control
true_for_a_while(commutation_control,N_commutazione+1,MAX_N+2,could_commuta
te);

%%      CALCOLO VELOCITA' STIMATA
      vs = vs_old;

```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO

43 di 46

%% IN OGNI CASO LA V STIMATA DEVE ESSERE MAGGIORE O UGUALE  
ALLE VELOCITA'

%% DEGLI ASSI

$vs = \max([v1d, v2d, vs]);$

%% AGGIORNAMENTO FLAG INIZIO PATTINAMENTO

FIP = VERO;

%% la commutazione trazione/frenatura viene effettuata se l'accelerazione

%% media, filtrata con il filtro da 4s, cambia segno e se tale variazione

%% di segno si mantiene per almeno 4 s

%% controllo se è possibile effettuare la commutazione,

%% in caso affermativo:

%% - ridefinisco la variabile accelera\_o\_frena

%% - effettuo il reset del vettore commutation\_control

%% - ridefinisco la variabile could\_commutate

if (commutation\_control(1) == CAN\_COMMUTATE)

%% POSSO EFFETTUARE LA COMMUTAZIONE

accelera\_o\_frena = ACCELERA;

%% RESET DEL VETTORE PER IL CONTROLLO SULLA COMMUTAZIONE

commutation\_control = CANNOT\_COMMUTATE\*ones(MAX\_N+2,1);

could\_commutate = FALSO;

end

%% stima dell'accelerazione

%% all'inizio del pattinamento prendo una percentuale della dr

$a\_stimata = \min(-dr*0.3, a\_stima\_riferimento);$

otherwise

break

end

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
44 di 46

**velocita' in km/h**

```
vs_km = (vs * 3.6);
v1_km = v1d * 3.6;
v2_km = v2d * 3.6;
```

**Calcolo dello spazio percorso**

```
spazio = spazio_old + vs * T;
```

**Memorizzazione dei valori delle variabili**

```
store_new_20032003
```

**Linee di ritardo**

```
%%    vettore per il controllo della commutazione dalla fase di
%%    accelerazione a quella di frenatura o viceversa nelle fasi
%%    di pattinamento, slittamento, aderenza non certa e tratto
%%    a velocità costante
```

```
    commutation_control
delay_line(commutation_control,MAX_N+2,could_commutate);
```

```
%%    vettori per il controllo del bloccaggio degli assi
    asse1_libero = delay_line(asse1_libero,MAX_N+2,LIBERO);
    asse2_libero = delay_line(asse2_libero,MAX_N+2,LIBERO);
```

```
%%    vettore per il controllo dell'inizio di una fase di pattinamento,
%%    in cui la velocità stimata deve essere costante
```

```
%%
    begin_of_a_sliding
delay_line(begin_of_a_sliding,MAX_N,IS_NOT_SLIDING);
```

```
%%    vettori per il controllo della stabilità degli assi
    asse1_stabile = delay_line(asse1_stabile,MAX_N+2,NON_STABILE);
    asse2_stabile = delay_line(asse2_stabile,MAX_N+2,NON_STABILE);
```

**SCMT**

Codifica: **RFI TC.PATC SR CM 03 M99 B**

FOGLIO  
45 di 46

%% vettori in cui si memorizza l'accelerazione dei due assi

a1df\_finestra = delay\_line(a1df\_finestra,N\_asse\_stabile,a1df);

a2df\_finestra = delay\_line(a2df\_finestra,N\_asse\_stabile,a2df);

%% vettori per il controllo della ripresa di aderenza degli assi

assi\_non\_slittano = delay\_line(assi\_non\_slittano, MAX\_N+2,SLITTA);

assi\_non\_pattinano = delay\_line(assi\_non\_pattinano, MAX\_N+2,PATTINA);

---

**Memorizza le variabili dello step precedente che servono nell'elaborazione**

---

%% count\_asse1\_old = count\_asse1;

%% count\_asse2\_old = count\_asse2;

v1d\_old = v1d;

v2d\_old = v2d;

a1d\_old = a1d;

a2d\_old = a1d;

a1df\_old = a1df;

a2df\_old = a2df;

a1\_filtrata\_old = a1\_filtrata;

a2\_filtrata\_old = a2\_filtrata;

a1\_molto\_filtrata\_old = a1\_molto\_filtrata;

a2\_molto\_filtrata\_old = a2\_molto\_filtrata;

a\_media\_molto\_filtrata\_old = a\_media\_molto\_filtrata;

a\_stimata\_old = a\_stimata;

**SCMT**

Codifica: **RFI** **TC.PATC** **SR** **CM** **03** **M99** **B**

FOGLIO  
46 di 46

```
if (assi_disponibili == NESSUNO)
    a_media_filtrata_old = 0;
else
    a_media_filtrata_old = a_media_filtrata;
end
is_sliding_old = is_sliding;
vs_old = vs;
spazio_old = spazio;

begin_of_a_sliding_old = begin_of_a_sliding;
```

---

**Fine calcolo velocità e spazio percorso**

---

end % ciclo for