Enhanced Locomotion Control for a Planetary Rover

Thierry Peynot & Simon Lacroix
LAAS/CNRS
7, av. du Colonel Roche
31077 Toulouse Cedex 4 - France
{Thierry.Peynot,Simon.Lacroix}@laas.fr

***Abstract*— This article presents an approach to improve and monitor the behavior of a skid-steering rover on rough terrains. An adaptive locomotion control generates speeds references to avoid slipping situations. An enhanced odometry provides a better estimation of the distance travelled. A probabilistic classification procedure provides an evaluation of the locomotion efficiency on-line, with a detection of *Locomotion Faults*. Results obtained with a Marsokhod rover are presented throughout the paper.**

I. Introduction

An efficient motion control is the basis on which any mobile robot relies to fulfill its missions. On rough terrains, this control problem is particularly difficult to achieve, significant differences between the reference motion and the actual motion occur, mainly because of strong wheel slippages.

The problem is that the easiest and cheapest way to measure these differences is odometry, which by essence can not detect slippages. In the absence of other means to measure the motions at high rates, one must develop enhanced motion control strategies to prevent slippages as much as possible. Moreover, an autonomous rover should be able to detect excessive slippages when they occur, to know that in some situations the robot is actually no longer moving forward. That could give the rover an opportunity to find a better moving strategy.

Most recent papers dealing with slipping control rely on physics based motion control, with models of the wheel-soil interaction [1][6], the contact forces [4], the frictions [8][2] or using force sensors. Other methods assume the shape of the terrain [7]. These achievements usually aim at correcting slippages by chosing the appropriate traction according to the forces estimated.

Instead, this paper presents a speed command for the wheels of a skid-steering rover with a real-time adaptation to the shape of the terrain, that *avoids creating any slipping situation* due to non-relevant wheel speed references (section III). It also presents an improvement of odometry on rough terrains and a *Locomotion Monitoring* that evaluates on line the efficiency of locomotion and enables *Locomotion Faults* detection (section IV. These achievements are made on the sole basis of speeds and chassis configuration/attitude angles measurements, whithout any additional sensors.

II. The Robot Lama

The development of the algorithms described in this paper and their validations were based on many experimentations with the Marsokhod rover Lama (figure 1). The chassis of Lama is composed of three pairs of independently driven *non-directional* wheels, mounted on 3 axles that can roll relatively to one another (angles α figure 1). The angle β between the arms that support the front and read axles is also free to move, thus giving high obstacle climbing skills. The rover mast is mounted so that it is the bissectrix of the two arms.

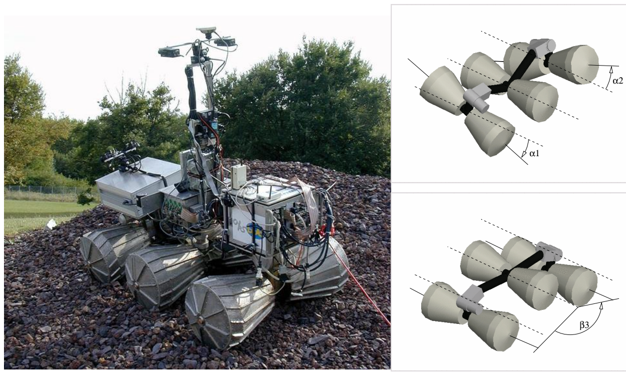

Fig. 1. Lama's Chassis

Lama is endowed with the following sensors:

- 6 high resolution *optical encoders* to measure each wheel speed (ω_{wheel})
- A 2 axes *inclinometer* providing the robot attitude by measuring the roll ϕ and the pitch ψ
- *potentiometers* measure the angles α_1, α_2, β_3 that define the robot chassis configuration (fig. 1)
- a precise fiber-optics that provides ω_{Gyro}
- a centimeter-accuracy RTK differential GPS was also used for test and validation purposes.

The robot motions are defined by a couple of translational and rotational speeds (V_{ref}, ω_{ref}), which is converted into 6 wheel reference angular speeds. The wheels being non-orientable, the robot turns in a skid-steering way: two different speeds (V_r and V_l) are applied to the right and left wheels of the center axle (which causes its rotation) with a similar and compatible operation on the 2 other axles (figure 2).

The basic motion control consists in applying *the same reference speed on the 3 wheels* that are *on the same side* of the robot (ω_r on the right, ω_l on the left), considering $V_r = R\,\omega_r$ and $V_l = R\,\omega_l$.

Odometry gives an estimation of Δs, the length of the arc of circle travelled during Δt, and $\Delta\theta$, estimated with the center axle wheels speeds ω_l and ω_r (figure 2):

$$\begin{aligned} \Delta s &= \Delta t\, R\,(\omega_r + \omega_l)/2 \\ \Delta\theta &= \Delta t\, R\,(\omega_r - \omega_l)/e \end{aligned} \qquad (1)$$

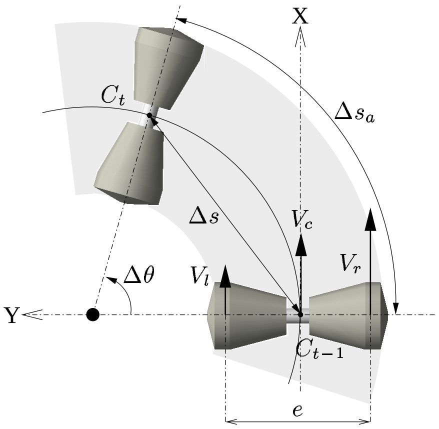

Fig. 2. The position estimation with odometry uses a measure of Δs_a, a good approximation of Δs when the sampling frequency is high in comparison with the robot's velocity.

Elementary 3D displacements $(\Delta x, \Delta y, \Delta z)$ can be calculated using the curvilinear abscissa Δs and the variations of the heading $\Delta\theta$ and the pitch $\Delta\phi$:

$$\begin{aligned} \Delta x &= \Delta s\, cos(\Delta\theta/2)\, cos(\Delta\phi) \\ \Delta y &= \Delta s\, sin(\Delta\theta/2)\, cos(\Delta\phi) \\ \Delta z &= -\Delta s\, sin(\Delta\phi) \end{aligned}$$

Because of unpredictable ground/wheel frictions, the information returned by the wheel encoders do not allow a faithfull estimation of the rotation speed: for that purpose, the gyrometer is used to provide the measure of $\Delta\theta$ [3].

III. Terrain Adaptive Locomotion Control

A. Principle

The basic command strategy presented above does not take into account the influence of the shape of the terrain. Actually, the direct application of ω_r and ω_l *forces* some wheels to slip on uneven terrains.

This phenomenon is illustrated in 2D and 3D on figures 3 & 4: the norms of the 3 linear speeds vectors are the

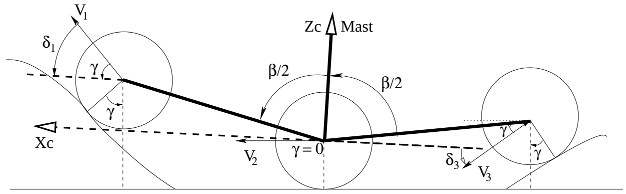

Fig. 3. Illustration of the linear speeds of the 3 wheels on the left side while Lama travels on an uneven terrain

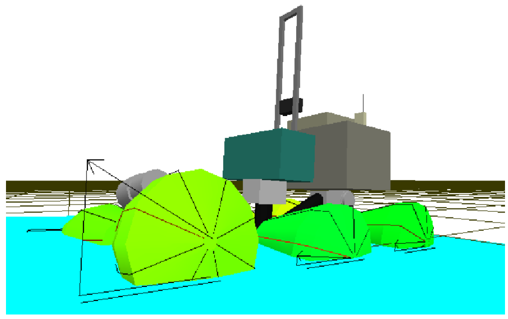

Fig. 4. 3D representation of Lama with the linear speeds of each wheel and their projections on (X_cY_c). The front-left wheel is climbing onto a rock

same but their projections on the plane of the center axle ((X_cY_c), considered as the plane of the robot's trajectory) are quite different. As the 3 wheels belong to the same solid (the robot), some are bound to slip. *For each wheel, the angle between its linear speed and the ground has to be taken into account to prevent slippages.*

Thus, the purpose of the ”Terrain Adaptive Locomotion Control” (TALC) is to have the *effective* linear speeds of the 6 wheels respect the kinematics relations between them, given that they belong to they are mechanically part of the same robot. If that condition is respected at any time, the TALC should not *induce* any slipping situation. Obviously, this does not mean that no slippage will ever actually occur, but the application of the TALC will reduce them significantly.

For instance, if one front or back wheel has to climb over a rock whereas all the others stay on a flat terrain (as in figure 4), a higher speed will be applied to the climbing wheel to compensate for the influence of the new angle between the wheel's linear speed and the trajectory plane.

B. Implementation

1) Principle: Thanks to the various proprioceptive sensors available on-board Lama, the configuration and attitude of the robot is known at any time. Using this data and the result of its differentiation, and given the reference speeds to achieve, the direction and the norm of the linear speed of each wheel are estimated. These norms V_{wheel} give the appropriate wheel reference speeds to apply: $\omega_{wheel} = V_{wheel}/R$.

2) Frames: The linear speeds of each wheel with respect to the ground ($\Re_G$ referential) have to be computed. Another referential will be used: the *center-axle referential* ($\Re_c$, see figure 5). Its center (named C) is the middle of the robot center axle, the Y axis is the axle oriented from the right side to the left side and the Z axis is along the robot's "mast", pointed up. The robot always travels along the X axis of $\Re_c$ (X_c) and its rotation $\omega = \omega_{Gyro}$ is made and measured in the (X_cY_c) plane.

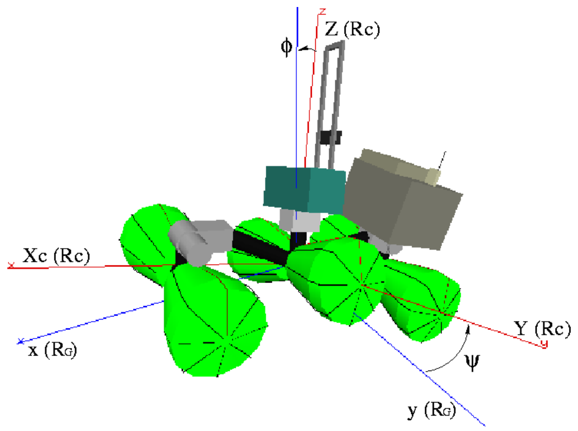

Fig. 5. The $\Re_c$ frame w.r.t. the Ground

Two other frames are introduced: $\Re_l$ and $\Re_r$. $\Re_l$ is the result of a translation of $\Re_c$ along its Y axis so that the center C_l of $\Re_l$ is the center of the middle-axle left wheel. Simmetrically, $\Re_r$ the frame corresponding to the middle-axle right wheel.

As said above, the robot motions are defined by two reference speeds: Vr for the right side and Vl for the left side. Thus, *the goal of the TALC is to have the X_c component of the velocity of $\Re_l$ (respectively $\Re_r$) equal to V_l (resp. V_r)*

3) New Reference Speeds Calculation: Let consider a wheel center denoted B, on the side s (r or l). First, $\vec{V}_{B/\Re_c}$, the speed of B w.r.t. the frame $\Re_c$, is only due to the robot's configuration variations, known thanks to the sensors measuring the configuration and attitude angles (α_1, α_2, β_3, ϕ and ψ).
Second, the speed of B with respect to the ground frame ($\Re_G$) can be expressed as:

$$\vec{V}_{B/\Re_G} = \vec{V}_{B/\Re_s} + \vec{V}_{C_s/\Re_G} + \overrightarrow{BC_s} \wedge \vec{\Omega}_{Rs/\Re_G}$$

As a consequence of the link between $\Re_s$ and $\Re_c$, $\vec{\Omega}_{Rs/\Re_G} = \vec{\Omega}_{Rc/\Re_G}$ and $\vec{V}_{B/\Re_s} = \vec{V}_{B/\Re_c}$. Thus, $\vec{V}_{B/\Re_G} = \vec{V}_{B/\Re_c} + \vec{V}_{C_s/\Re_G} + \overrightarrow{BC_s} \wedge \vec{\Omega}_{Rc/\Re_G}$.

The TALC sets $(\vec{V}_{C_s/\Re_G})_{X_c}$ to V_s. $(\vec{V}_{C_s/\Re_G})_{Y_c} = 0$, as Y_c contains the robot instantaneous rotation center. Finally, $(\vec{V}_{C_s/\Re_G})_{Z_c} = \dot{z}_{C_s}$ can be calculated using the relation between C_s and C: $\vec{V}_{C_S/\Re_G} = \vec{V}_{C/\Re_G} + \overrightarrow{C_sC} \wedge \vec{\Omega}_{Rc/\Re_G}$, and the estimation of $(V_{C/\Re_G})_{Z_c}$ made thanks to the chassis and on-board sensors.
Consequently, and considering the expression of $\vec{\Omega}_{Rc/\Re_G}$, the velocity of B can finally be expressed in $\Re_c$ as:

$$\vec{V}_{B/\Re_G} = \vec{V}_{B/\Re_c} + \begin{pmatrix} V_s \\ 0 \\ \dot{z}_{C_s} \end{pmatrix} + \overrightarrow{BC_s} \wedge \begin{pmatrix} \dot{\psi} \\ \dot{\phi} \\ \omega \end{pmatrix}$$

Thus, if the global reference speeds of the robot are respected, the velocity of B should be $\vec{V}_B = (\dot{x}\,\dot{y}\,\dot{z})^T$, calculated above. As $\dot{y}$ is neglictible with respect to $\dot{x}$ and $\dot{z}$, especially for slow robots such as Lama, the norm $V_B = \sqrt{\dot{x}^2 + \dot{z}^2}$ gives the speed that should be applied to the wheel: $w_{wheel} = V_B/R$ (rolling without slipping).

Finally, all the relevant speed references are calculated so that the projection of the middle axle speed on (X_cY_c) is V_l on the left side and V_r on the right side, and so that the kinematics relations between the wheels are respected.

4) Contact Angles Estimation: If δ_B is the angle of $\vec{V}_B$ with respect to the (X_cY_c) plan (in direction of travel), $cos(\delta_B) = \dot{x}/V_B$. As the orientation of $\Re_c$ w.r.t. the ground is known at any time (thanks to the attitude sensors), the angle γ_B of the speed w.r.t. *horizontal* can be calculated. This angle also is the wheel-ground contact angle.
Thus, the TALC also provides an estimation of all the wheel-ground contact angles: γ_{wheel}. This information enables to make coherent comparisons of different speeds and will be used to monitor the locomotion (section IV). The estimation is based on the measurements of the configuration and attitude angles and the desired speeds ω_{ref} and V_{ref} only, contrary to the method in [4][5], which uses all the wheel speeds measurements. Thus, as long as the global motion of the robot respects the references, the contact-angles estimation will remain accurate *for all wheels* even if *some* wheels are slipping.

C. Results

Figure 6 illustrates the elementary wheel speed corrections generated by the TALC during a simple test: the robot makes a *5 cm/s* straight motion. The terrain is flat, except for one rock on the trajectory of the rover's left side. Consequently, each left wheel climbs over this rock, one after the other. To compensate for its influence on the shape of the terrain, each wheel elementary speed is augmented as it goes up and down the rock. Thanks to that correction, the speed references respect the kinematics relations between the wheels.

D. Enhancing the odometric position estimation

The estimation of Δs presented in equation 1 relies on the following hypothesis: *the projection of the wheel linear speed on the instantaneous evolution plane defined by (X_C, Y_C) in which Δs is computed can be expressed as $V_{wheel} = R\omega_{wheel}$, as long as that wheel is not slipping*. This is a quite good approximation on flat terrains, not on

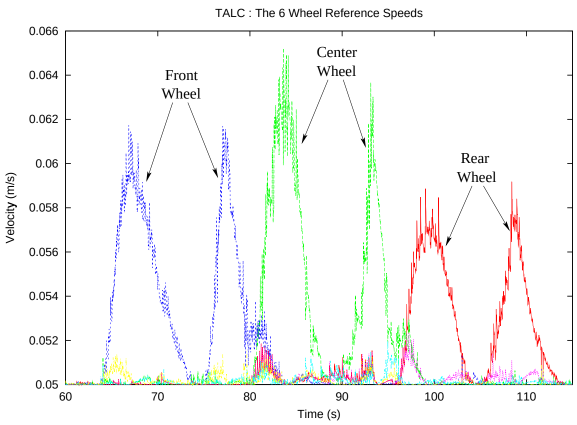

Fig. 6. The 6 linear reference speeds generated by the TALC during a simple test

rough terrains where the angle δ of a wheel's linear speed from the robot's evolution plane (X_cY_c) must be taken into account.[1] Thus, a better estimation of Δs is:

$$\Delta s \;=\; \Delta t\, R\,(\omega_r\, cos(\delta_r) + \omega_l\, cos(\delta_l))/2$$

However, only the speeds of the two center-axle wheels are used, whereas the 6 wheel speeds are available: the odometry position estimation should benefit from this additional information. The purpose is to select on each side the wheel that slips the less.

The selection relies on the hypothesis that slipping wheels turn faster than they should, which is reasonable given the weight of Lama and the kind of terrain explored (special cases such as ice are excluded). In other words, the corresponding distance actually covered on the ground is shorter than the one indicated by the wheel speed measurement: this was confirmed by experimentations with the GPS as reference. The results of the *Linear Speeds Coherence Watching* (section IV-A) are used to make the decision by testing the signs of the ΔV_x computed.

This enhanced odometry method provides a much better estimation of Δs, as illustrated in figure 7.

IV. Locomotion Monitoring

In some situations on very rough terrains, because of excessive slippages, the rover might have great difficulty in moving, although its 6 wheels keep on turning. Such situations are *Locomotion Faults*. On board autonomous rovers, it is crucial to be able to detect such faults, in order to correct them or to trigger another moving strategy (for example, with Lama, the peristaltic mode [1]). This is performed through the computation of *speeds coherence indicators*, which give an evaluation of the actual efficiency of the rover locomotion. These indicators are used on-line to give information to a Markov Decision

[1]This angle is different from γ but it is also estimated by the TALC module, see III-B.4

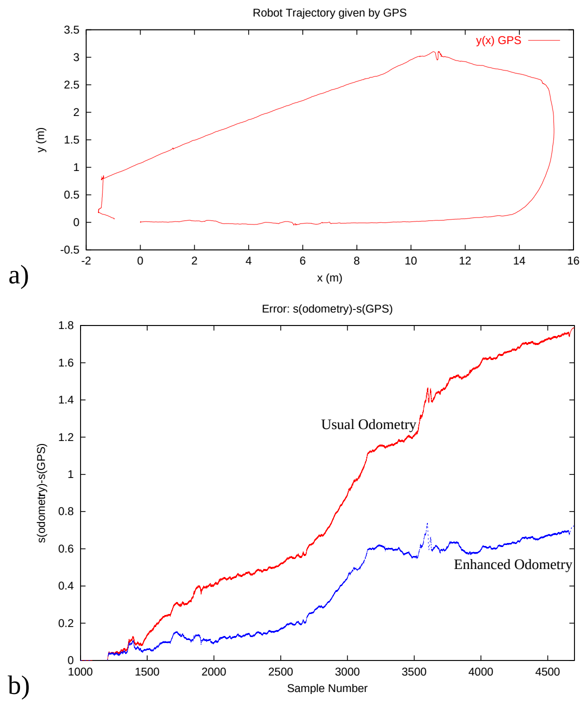

Fig. 7. Comparison of the curvilinear abscissa (s) estimation results with the *Usual Odometry* and the *Enhanced Odometry*. (a): Robot Trajectory given by the GPS, (b): The errors between s(odometry) and s(GPS)

Process which provides probabilities that the robot is in a slipping situation or in a *Locomotion Fault* situation.

A. Speeds Coherence Indicators

1) Linear Speeds Coherence: As elements of the same solid, the expressions of the speeds of two robot's wheels (A & B) on the same side should satisfy certain relations at any time. Indeed, if K is the wheel considered (A or B):

$$\vec{V}_{K/\Re_G} = \vec{V}_{K/\Re_c} + \vec{V}_{C_s/\Re_G} + \overrightarrow{KC_s} \wedge \vec{\Omega}_{Rc/\Re_G}$$

Consequently, if $\vec{W}(K) = \vec{V}_{K/\Re_G} - \vec{V}_{K/\Re_c}$, the following relation should be true at any time:

$$\vec{W}(B) - \vec{W}(A) \;=\; \overrightarrow{BA} \wedge \vec{\Omega}_{Rc/\Re_G} \qquad (2)$$

To detect slippages, we check the X_c component of (2), corresponding to the direction of travel. $W(A)_{X_c}$ and $W(B)_{X_c}$ are permanently computed on the basis of the measured wheels speeds ($V_{wheel} = R\,\omega_{wheel}$) and the angles estimations produced by the TALC. The speeds will be considered as *coherent* if relation (2) is true. Otherwise, the difference $\Delta V_{x(AB)} = (\vec{W}(B) - \vec{W}(A) - \overrightarrow{BA} \wedge \vec{\Omega}_{Rc/\Re_G})_{X_c}$ gives an estimation of the *incoherence* between the speeds measurements. The average of the three ΔV_x computed for each side will be an indicator of incoherence calculated at each time sample. This monitoring provides 2 *linear speeds coherence indicators*.

2) Rotation Speeds Coherence: On board Lama, a measure of the robot rotation speed is available (ω_{Gyro}). This precise speed measure can also be estimated using the wheels speeds measurements by:

$$\omega_{Odo} = (V_r - V_l)/e$$

where V_r and V_l are the linear speeds of a right and left wheel (on the same axle), and e the axle length. The expression of $V_s = V_r \, or \, V_l$ is: $V_s = R\omega_s \, cos(\delta_{wheel})$,

The average of all the $(\omega_{Odo} - \omega_{Gyro})$ computed is the third indicator, which gives an estimation of the *rotation speeds coherence*.

Figure 8 illustrates the behavior of these 3 indicators, especially in *Locomotion Fault* situations.

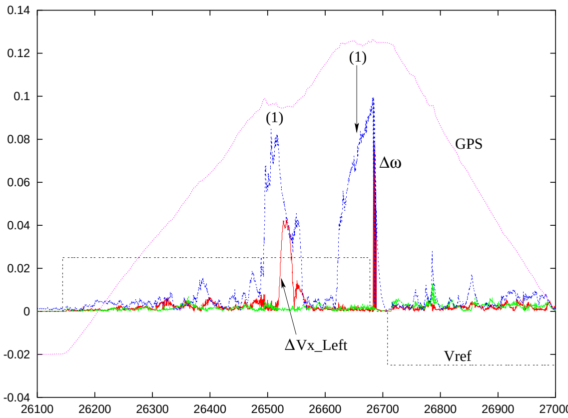

Fig. 8. The 3 "Speeds Coherence Indicators" in Locomotion Faults Situations (1). The indicators make significant increases during the two Locomotion Faults revealed by the GPS (which shows the actual evolution of the rover's position on the axis of travel).

B. Locomotion monitoring

The 3 speeds coherence indicators are used as attributes (or features) in a probabilistic classification procedure: a Markov Decision Process evaluates the robot situation on-line according to the previous probabilities calculated, the current values of the features, and their comparison with prototypes recorded during a *Supervised Learning* stage.

1) Supervised Learning: This operation aims at collecting information while the actual behavior of the robot is known at any time. Many experiments were made with Lama, with situations among 3 main categories, corresponding to 3 *states*:

- **State 0**: "Easy" situation for the rover. It seems that there is no *abnormal* slippages.
- **State 1**: Difficult situation for the rover. Some abnormal slippages happen, but the robot is still moving forward.
- **State 2**: *Locomotion Fault*. Because of excessive slippages the rover does no longer move forward, even though its wheels keep on turning.

During the learning experiments, the current state is determined by an operator. This decision and the other interesting data (V_{ref}, ω_{ref} and the 3 speeds coherence indicators) are recorded to be used later as prototypes.

Because the rover is a skid-steering platform, its behavior highly depends on the values of the couple (V_{ref}, ω_{ref}). Thus, several series of experiments were made with different usual values for this couple, each value corresponding to one prototypes base, which contains the features recorded.

Several other Supervised Learning operations could be made with more different kinds of terrain or conditions (e.g. wet soil) to constitute as much prototype bases. Then, additional exteroceptive sensors would be needed to determine the type of the current terrain on-line and choose the appropriate prototype base consequently. All this data could also be recorded in common prototype bases but the detector would be far less discriminating.

2) On-line Situation Evaluation: Because of uncertainties and some unknown elements the decision will be based on a probabilistic method. Moreover, the evolution in time of the robot's situation has to be taken into account, especially before a Locomotion Fault situation. Thus, a Markov Decision Process ([9][7]) is used to calculate at any time the probability that the robot is in state 0, 1 or 2, and make the appropriate decision.

3) Markov Decision Process: If we make the observation obs_t at time t, the probability that the rover is in state s_k will be:

$$p(s_k|obs^t) = \eta_t \, p(obs_t|s_k) \sum_{i=0}^{K-1} p_{ik} \, p(s_i|obs^{t-1}) \quad (3)$$

where: obs^t are all the observations made until time t, p_{ik} is the probability of the transition from state i to state k, K is the number of states, $p(obs_t|s_k)$ is probability that observation obs_t is made knowing that the rover is in state s_k, and η_t is a normalization coefficient (so that $\sum p(s_k|obs^t) = 1$).

4) Transition Probabilities: The transition probabilities were set empirically to the values shown in figure 9 based on the observations made during the supervised learning. They can be adapted according to the detector sensibility wanted or the kind of terrain, if the robot is able to detect it.

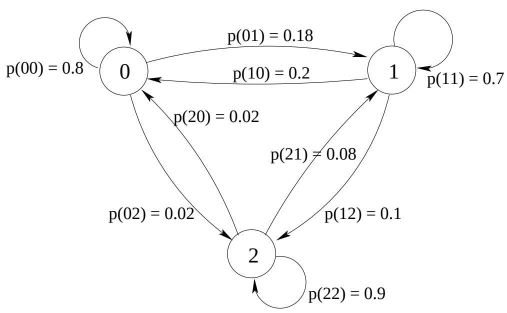

Fig. 9. The States and Transition Probabilities

5) Probability Density: To calculate $p(X|s_k)$, we use a ***Nearest Neighbors***. The search on-line for the nearest neighbors of each sample would be too much time consuming: thus, the probabilities $p(X|s_k)$ are computed off-line for each learning sample. Then, on-line, for each new sample, the system will only search for the nearest prototype and the probability densities for the current sample will be the ones associated to that prototype.

6) On-line Calculation of the State Probabilities: To each sample, at time t, is associated the interesting data: $V_{ref}(t)$, $\omega_{ref}(t)$ and the 3 *attributes* values at t (the 3 speeds coherence indicators). The first operation is to select the prototypes base the most adapted to the current situation (with the closest (V_{ref}, ω_{ref}) couple). Then, we look for the nearest prototype in this base and use the associated state probabilities as an estimation of the probability densities for this sample. Now, as all the elements needed are available, the state probabilities for the current sample can be calculated thanks to equation (3). The complete computation cycle for each sample is presented in figure 10.

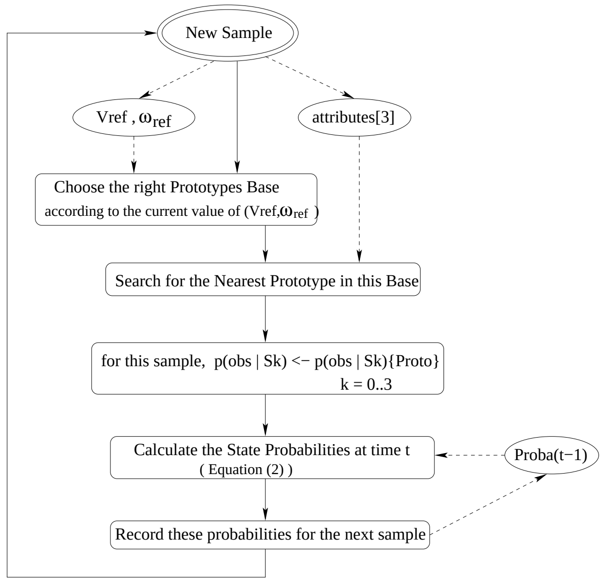

Fig. 10. On-line calculation of the State Probabilities

The 3 state probabilities obtained will be transmitted to a supervisor that will decide which is the current state of the rover.

Fig. 11 shows an example of the results obtained in comparison with the opinion of a human operator.

This procedure provides an evaluation of the efficiency of locomotion on-line, enabling the rover to detect *Locomotion Faults*.

V. SUMMARY

We have presented an enhanced locomotion control (the TALC) that makes a real-time adaptation to the shape of

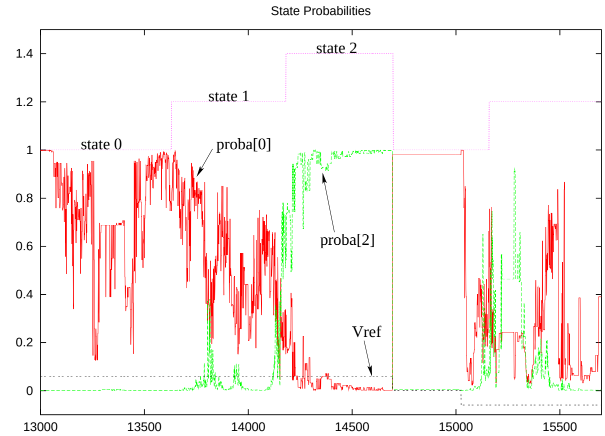

Fig. 11. Comparison between the state probabilities calculated and the current state according to the operator

the terrain to avoid creating slipping situations. It has been tested on a skid-steering rover with an articulated chassis but this method could be applied on any wheeled rover. Our wheel-ground contact angle estimation together with an automatic wheel selection have lead to an improvement of the position estimation by odometry. We have also proposed a method to evaluate on-line the efficiency of the robot's locomotion with a detection of Locomotion Faults. This method seems to be quite efficient using only speed information, but it may benefit from the use of additional data, such as the wheels' motors current.

VI. REFERENCES

[1] P. Bidaud R. Chatila G. Andrade-Barroso, F. Ben Amar. Modeling robot-soil interaction for planetary rover motion control. In *IROS*, 1998.

[2] G. Martucci P. Rocco V. Stampacchia G. Ferretti, G. Magnani. Friction model validation in sliding and presliding regimes with high resolution encoders. In *8th ISER*, 2002.

[3] L. Feng J. Borenstein. Gyrodometry: A new method for combining data from gyros and odometry in mobile robots. In *ICRA*, 1996.

[4] S. Dubowsky K. Iagnemma. Mobile robot rough-terrain control (rtc) for planetary exploration. In *Proceedings of ASME IDETC/CIE : 26th Biennial Mechanisms and Robotics Conference*, 2000.

[5] S. Dubowsky K. Iagnemma. Vehicle wheel-ground contact angle estimation: with application to mobile robot traction control. In *7th International Symposium on Advances in Robot Kinematics*, 2000.

[6] T. Watanabe K. Yoshida, H. Hamano. Slip-based traction control of a planetary rover. In *8th ISER*, 2002.

[7] R. Washington O. Aycard. State identification for planetary rovers: Learning and recognition. In *ICRA*, 2000.

[8] S. Dubowsky S. Farritor, H. Hacot. Physics-based planning for planetary exploration. In *IROS*, 1998.

[9] J. Fernandez V. Verma, R. Simmons. Probabilistic models for monitoring and fault diagnosis. In *2nd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, 2002.