Control of a wheeled stair-climbing robot using linear programming

Luc Jaulin
Laboratoire E^3I^2, ENSIETA,
2 rue François Verny
29806 Brest

E-mail : luc.jaulin@ensieta.fr

Abstract : This paper introduces a new formalism to deal with constrained dynamic systems. It shows that, when the constraints are linear, the controller can take advantage of linear programming algorithms (such as the simplex) to guarantee that the constraints will be satisfied. To illustrate the contribution of the approach, the control problem of a wheeled stair-climbing robot is considered. Masses supported by the robot have to be moved in order to avoid any sliding of the wheels. For such control problems where strong nonlinearities occur, conventional control methods fail to provide any reliable controller.

Keywords : Hybrid systems, linear programming, mass transfer, mobile robots, nonlinear control, set approach, simplex algorithm.

1. Introduction

In this paper, we shall consider the class of constrained dynamic systems which can be described by

$$\begin{array}{ll} \text{(i)} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \text{(ii)} & (\mathbf{x}(t), \mathbf{v}(t)) \in \mathbb{V}, \end{array} \tag{1.1}$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the *evolution input vector*, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the *state vector*, $\mathbf{f} : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_x}$ is the *evolution function*, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the *viable input vector* and $\mathbb{V}$ is the *viable set*. The methodology to be proposed is devoted to systems described by (1.1), but can easily be extended to a larger class of hybrid systems [7], where more differential equations are involved.

We have assumed here that the evolution inputs and viable inputs are completely independent, *i.e.*, $\mathbf{u}$ acts only in (i) whereas $\mathbf{v}$ affects only (ii). This is not always the case. However, for many robotics applications, it is possible to conceive the robot and to insert the actuators at the right place in order to get a decoupling between the evolution and the viable inputs. If the value for $\mathbf{v}(t)$, chosen at time t, is such that $(\mathbf{x}, \mathbf{v}) \in \mathbb{V}$ then the system is said to be *alive*. Otherwise, it is *dead*. In practice, the condition $(\mathbf{x}, \mathbf{v}) \notin \mathbb{V}$ translates into events such as "an element of the system brakes down" or "the robot is sliding", etc. In such situations the differential equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u})$ is not satisfied anymore and the behavior of the system becomes unpredictable. Consequently, all should be done to avoid such a situation.

From a control viewpoint, two problems should be treated. The first one consists in finding the evolution input vector $\mathbf{u}$ in order to get a satisfactory behavior of the system. This problem can be solved using classical techniques even when $\mathbf{f}$ is nonlinear (see, *e.g.*, [8, 3, 9]). The second problem to be considered is to find a viable input vector $\mathbf{v}$ such that the set-membership relation (ii) is satisfied for the given state vector $\mathbf{x}$. This research should be done for each t in a very short time.

Two cases can be considered :

- If $(\mathbf{x}, \mathbf{v}) \in \mathbb{V}$ corresponds to a set of affine constraints with respect to $\mathbf{v}$, such as $\mathbf{A}(\mathbf{x}).\mathbf{v} + \mathbf{b}(\mathbf{x}) \leq \mathbf{0}$ then a linear programming approach (for instance based on the simplex algorithm [5]) can be used to find one $\mathbf{v}$ which satisfies the constraints. This task may be done rather quickly if the dimension of $\mathbf{v}$ is lower than 100.
- If $(\mathbf{x}, \mathbf{v}) \in \mathbb{V}$ is composed with constraints of the form $\mathbf{f}(\mathbf{x}, \mathbf{v}) \leq \mathbf{0}$, where $\mathbf{f}$ is nonlinear with respect to $\mathbf{v}$, then interval constraint propagation methods [4, 2, 10] can be used to find one feasible $\mathbf{v}$. This could be done rather quickly if the dimension of $\mathbf{v}$ is small (typically lower than 10).

The purpose of this paper is to illustrate the methodology on a two-dimensional robot equipped with three wheels. As many existing mobile robot (see e.g. [14] for wheeled robots, [13] for legged robots or [11] for tracked robots), our robot is designed to move in an uneven environment, but its originality is that a mass transfer system is used to get over obstacles. Whereas classical control approaches cannot solve the control of the mass transfer problem, linear programming methods will be shown to be able to solve it efficiently.

Section 2 presents the robot to be considered. Section 3 provides a method based on the simplex algorithm to control it. Some simulations of the behavior of the controlled robot are given on Section 4. Section 5 concludes the paper.

2. Robot

Figure 2.1 presents a two-dimensional robot made by three driving wheels and two pendulums. The center of the ith wheel will be denoted by $\mathbf{c}_i$. The back wheel and the middle wheel are linked by a platform with weight μ_1. The front wheel and the middle wheel are linked by a platform with weight μ_2. At the top of each pendulum is located two weights μ_3 and μ_4. We shall denote by $\mathbf{m}_1$ and $\mathbf{m}_2$ the centers of the two platforms and by $\mathbf{m}_3$ and $\mathbf{m}_4$ the vertices of the pendulums. The angles v_1 and v_2 between the pendulum and the corresponding platform can be tuned as desired and will be used to prevent the robot from any sliding. The abscissa of the center of the back wheel is denoted by x. The robot is equipped with a strong electric motor which is linked to each of the three wheels. It can be

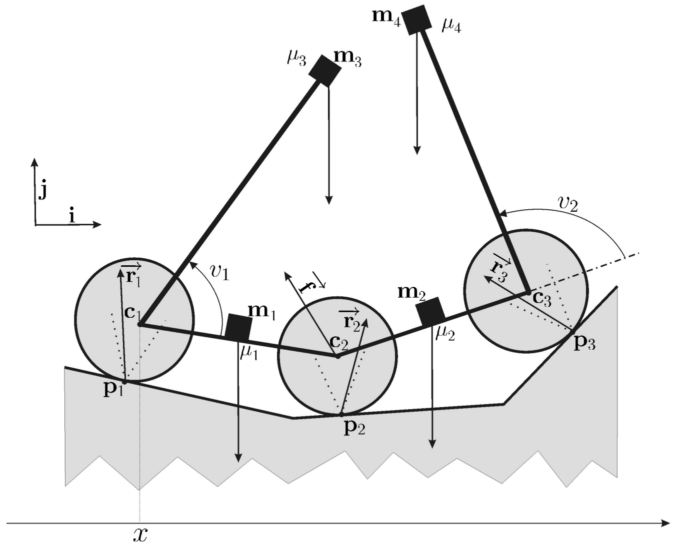

Figure 2.1: Two dimensional three-wheeled climbing robot to be moved in an uneven urban environment

assumed that when the robot is not sliding, x satisfies the following differential equation

$$\dot{x} = u, \tag{2.1}$$

where the evolution input u can be chosen arbitrarily. Here, we shall choose u small enough to be allowed to assume that the robot has a quasi-static motion. We shall assume that all wheels keep a contact with the ground. Thus, our robot has three degrees of freedom: x, v_1 and v_2. The contact point $\mathbf{p}_i$ between the ith wheel and the ground will be assumed to be unique.

2.1. Geometry of the robot

For our simulations, the ground will be assumed to be made with a collection of connected segments $[\mathbf{a}_j, \mathbf{b}_j]$, $1 \leq j \leq j_{\max}$ such that $\mathbf{b}_j = \mathbf{a}_{j+1}$ and $\mathbf{b}_{jx} > \mathbf{a}_{jx}$. From the knowledge of x, it is possible to compute $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{m}_1$ and $\mathbf{m}_2$ (the computation of $\mathbf{m}_3$ and $\mathbf{m}_4$ requires the knowledge of $\mathbf{v}$). The calculus requires only the resolution of first and second degree polynomial equations and can thus be performed quickly and exactly. The generation of these polynomial equations is rather laborious and will not be detailed here.

For an actual robot and for practical situations, points $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{m}_1$ and $\mathbf{m}_2$ are not computed, but should be measured. Localization methods (see e.g. [12]) combined with fusion methods based on different types of sensors (cameras, sonars, accelerometers, ...) should then be developed to achieve this goal.

2.2. Fundamental principle of static

Assume that the robot is immobile and denote by $\overrightarrow{\mathbf{r}}_i$ the reaction force generated by the ground onto the ith wheel. The Coulomb cone $\mathcal{C}_i$ associated with the ith wheel and represented with dotted lines on Figure 2.1 is the cone with vertex $\mathbf{p}_i$ whose symmetric axis in orthogonal to the ground and with an half aperture angle $\phi = 0.54$ (corresponding to a tire/concrete contact). If there exist $\overrightarrow{\mathbf{r}}_1^0 \in \mathcal{C}_1, \overrightarrow{\mathbf{r}}_2^0 \in \mathcal{C}_2, \overrightarrow{\mathbf{r}}_3^0 \in \mathcal{C}_3$ such that the fundamental static conditions are fulfilled then the robot is known to be at a steady position and will not slide [6]. Note that the actual values for $\overrightarrow{\mathbf{r}}_1, \overrightarrow{\mathbf{r}}_2, \overrightarrow{\mathbf{r}}_3$ cannot be computed without an in-depth knowledge of the internal tensions inside the robot (which is unrealistic here) and may be different from $\overrightarrow{\mathbf{r}}_1^0, \overrightarrow{\mathbf{r}}_2^0, \overrightarrow{\mathbf{r}}_3^0$. Let us now write the fundamental static conditions of the robot in order to translate the non-sliding condition into the form $(x, v_1, v_2) \in \mathbb{V}$, (since a quasi-static motion is assumed, the dependency with respect to $\dot{x}$ does not exist). When the robot does not move, we have

$$\left\{\begin{array}{lcl} -\overrightarrow{\mathbf{p}_1\mathbf{m}_1} \wedge \mu_1\mathbf{j} + \overrightarrow{\mathbf{p}_1\mathbf{c}_2} \wedge \overrightarrow{\mathbf{f}} - \overrightarrow{\mathbf{p}_1\mathbf{m}_3} \wedge \mu_3\mathbf{j} & = & 0 \\ -\overrightarrow{\mathbf{p}_2\mathbf{m}_2} \wedge \mu_2\mathbf{j} - \overrightarrow{\mathbf{p}_2\mathbf{c}_2} \wedge \overrightarrow{\mathbf{f}} + \overrightarrow{\mathbf{p}_2\mathbf{p}_3} \wedge \overrightarrow{\mathbf{r}}_3 - \overrightarrow{\mathbf{p}_2\mathbf{m}_4} \wedge \mu_4\mathbf{j} & = & 0 \\ \overrightarrow{\mathbf{r}}_1 - (\mu_1 + \mu_3)\mathbf{j} + \overrightarrow{\mathbf{f}} & = & 0 \\ \overrightarrow{\mathbf{r}}_2 - \overrightarrow{\mathbf{f}} - (\mu_2 + \mu_4)\mathbf{j} + \overrightarrow{\mathbf{r}}_3 & = & 0, \end{array}\right. \tag{2.2}$$

where $\tilde{\mathbf{f}}$ denotes the reaction force generated by the second platform onto the first platform. A scalar decomposition yields

$$\left\{\begin{array}{lcl} -\mu_1 (m_{1x} - p_{1x}) + (c_{2x} - p_{1x}) f_y - (c_{2y} - p_{1y}) f_x - \mu_3 (m_{3x} - p_{1x}) & = & 0 \\ -\mu_2 (m_{2x} - p_{2x}) - (c_{2x} - p_{2x}) f_y + (c_{2y} - p_{2y}) f_x & & \\ + (p_{3x} - p_{2x}) r_{3y} - (p_{3y} - p_{2y}) r_{3x} - \mu_4 (m_{4x} - p_{2x}) & = & 0 \\ r_{1x} + f_x & = & 0 \\ r_{1y} - \mu_1 - \mu_3 + f_y & = & 0 \\ r_{2x} - f_x + r_{3x} & = & 0 \\ r_{2y} - f_y - \mu_2 - \mu_4 + r_{3y} & = & 0. \end{array}\right. \tag{2.3}$$

This system can be written into a matrix form as

$$\mathbf{A}_1(x).\mathbf{y} = \mathbf{b}_1(x), \tag{2.4}$$

where

$$\mathbf{A}_1(x) = \begin{pmatrix} 0&0&0&0&0&0&p_{1y}-c_{2y}&c_{2x}-p_{1x}&-\mu_3&0\\ 0&0&0&0&p_{2y}-p_{3y}&p_{3x}-p_{2x}&c_{2y}-p_{2y}&p_{2x}-c_{2x}&0&-\mu_4\\ 1&0&0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&1&0&0\\ 0&0&1&0&1&0&-1&0&0&0\\ 0&0&0&1&0&1&0&-1&0&0 \end{pmatrix}$$

$$\mathbf{b}_1(x) = \begin{pmatrix} \mu_1(m_{1x}-p_{1x})-\mu_3 p_{1x}\\ \mu_2(m_{2x}-p_{2x})-\mu_4 p_{2x}\\ 0\\ \mu_1+\mu_3\\ 0\\ \mu_2+\mu_4 \end{pmatrix}$$

$$\mathbf{y} = (r_{1x}, r_{1y}, r_{2x}, r_{2y}, r_{3x}, r_{3y}, f_x, f_y, m_{3x}, m_{4x})^{\mathrm{T}}. \quad (2.5)$$

The system of equations (2.4) has ten unknowns for six equations. If $\mathbf{v} = (v_1, v_2)$ is known, then it remains two more unknowns than equations. This is due to the fact that our robot is a statically undetermined structure with a second order hyperstatic equilibrium [6].

2.3. Non-sliding conditions

None of the wheels will slide if all $\overrightarrow{\mathbf{r}}_i$ belong to their corresponding Coulomb cones. Thus, we should have

$$\det(\overrightarrow{\mathbf{r}}_i, \mathbf{u}_i^-) \leq 0 \text{ and } \det(\mathbf{u}_i^+, \overrightarrow{\mathbf{r}}_i) \leq 0, \quad (2.6)$$

where $\mathbf{u}_i^-$ and $\mathbf{u}_i^+$ denote the two vectors supporting the ith Coulomb cone $\mathcal{C}_i$. These two inequalities can be rewritten into a matrix form as

$$\begin{pmatrix} u_{iy}^- & -u_{ix}^- \\ -u_{iy}^+ & u_{ix}^+ \end{pmatrix} \overrightarrow{\mathbf{r}}_i \leq \mathbf{0}. \quad (2.7)$$

Thus, we should have,

$$\mathbf{A}_2(x).\mathbf{y} \leq \mathbf{0}, \quad (2.8)$$

where

$$\mathbf{A}_2(x) = \begin{pmatrix} u_{1y}^- & -u_{1x}^- & 0&0&0&0&0&0&0&0\\ -u_{1y}^+ & u_{1x}^+ & 0&0&0&0&0&0&0&0\\ 0&0&u_{2y}^-&-u_{2x}^-&0&0&0&0&0&0\\ 0&0&-u_{2y}^+&u_{2x}^+&0&0&0&0&0&0\\ 0&0&0&0&u_{3y}^-&-u_{3x}^-&0&0&0&0\\ 0&0&0&0&-u_{3y}^+&u_{3x}^+&0&0&0&0 \end{pmatrix}. \quad (2.9)$$

Figure 2.2 represents a computed situation where the robot is immobile and not sliding. Vectors $\mathbf{u}_i^-, \mathbf{u}_i^+$ delimiting the Coulomb cones are represented with dotted lines. Since all ground reaction forces $\overrightarrow{\mathbf{r}}_i$ are inside their corresponding cones, the robot cannot slide.

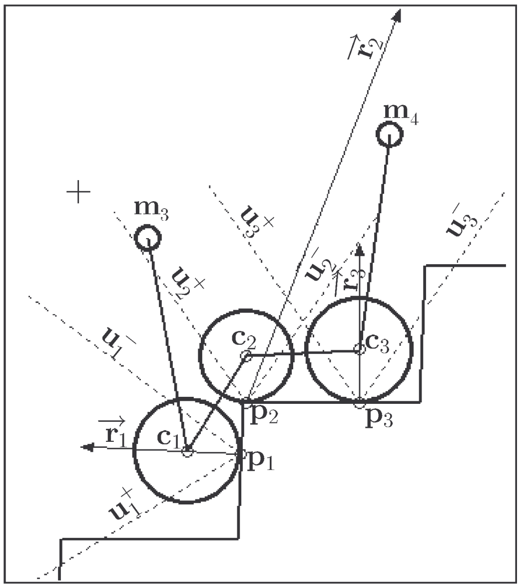

Figure 2.2: A configuration where the middle wheel is almost slipping. This picture has been obtained after some computations in order to be certain that the fundamental principle of static is satisfied

2.4. Feasible conditions for the pendulums

First, we have

$$\begin{array}{rcl} m_{3x} & = & c_{1x} + \ell_1 \cos\left(v_1 + \arctan\left(\frac{c_{2y}-c_{1y}}{c_{2x}-c_{1x}}\right)\right), \\ m_{4x} & = & c_{3x} + \ell_2 \cos\left(v_2 + \arctan\left(\frac{c_{3y}-c_{2y}}{c_{3x}-c_{2x}}\right)\right). \end{array} \tag{2.10}$$

Moreover the variables m_{3x} and m_{4x} should be such that the pendulums be at feasible locations : the pendulums should not intersect the ground or the robot itself. This condition can be translated into

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .\mathbf{y} \in \begin{pmatrix} [m_{3x}^{\min}, m_{3x}^{\max}] \\ [m_{4x}^{\min}, m_{4x}^{\max}] \end{pmatrix}, \tag{2.11}$$

where the bounds $m_{3x}^{\min}, m_{3x}^{\max}, m_{4x}^{\min}, m_{4x}^{\max}$ can be computed from x.

2.5. Recapitulation of the constraints

Our robot can be described by

$$\begin{array}{lccc} \text{(i)} & \dot{x} & = & u \\ \text{(ii)} & (x, v_1, v_2) & \in & \mathbb{V} \end{array} \tag{2.12}$$

where the condition $(x, v_1, v_2) \in \mathbb{V}$ is equivalent to

$\exists\,(r_{1x}, r_{1y}, r_{2x}, r_{2y}, r_{3x}, r_{3y}, f_x, f_y, m_{3x}, m_{4x})$ such that (2.4), (2.8), (2.11) and (2.10) are satisfied

3. Control of the robot

The control problem we wish to solve now is the following: find the angles v_1 and v_2 such that the fundamental principle of static, the non-sliding conditions and the feasible conditions for the pendulums are satisfied. To achieve this task, we shall act as follows:

1. From x, and by using elementary geometry, compute the vectors $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{m}_1, \mathbf{m}_2$ and the two intervals $[m_{3x}^{\min}, m_{3x}^{\max}]$, $[m_{4x}^{\min}, m_{4x}^{\max}]$.
2. Solve the following linear program

$$\begin{array}{ll} & \min e, \\ s.t. & \text{(i) } \mathbf{A}_1(x).\mathbf{y} = \mathbf{b}_1(x) \\ & \text{(ii) } \mathbf{A}_2(x).\mathbf{y} \leq e.\mathbf{1}, \\ & \text{(iii) } \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.\mathbf{y} \in \begin{pmatrix} [m_{3x}^{\min}, m_{3x}^{\max}] \\ [m_{4x}^{\min}, m_{4x}^{\max}] \end{pmatrix} \end{array} \tag{3.1}$$

   where $\mathbf{1} = (1\ 1\ 1\ 1\ 1\ 1)^{\mathrm{T}}$. Since the rank of $\mathbf{A}_1$ is full, the linear program always has a solution. If the minimum $\hat{e}$ is negative, the non-sliding condition (2.8) can be satisfied and the robot will not slide. If $\hat{e}$ is positive, the robot will slide.

3. From $\mathbf{y}$, compute the viable inputs v_1 and v_2 such that (2.10) is satisfied.
4. Compute independently the control $\mathbf{u}$ using a classical control method.

4. Results

To illustrate the control developed in the previous sections, we shall move the robot slowly in a urban irregular ground. As illustrated by Figure 4.1 the pendulums react in order to avoid any sliding from the robot. The simulation of the control has been performed with SCILAB[1] and the source code is available at `www.ensieta.fr/e3i2/Jaulin/etas.sce`. For each sample, the computation of the control $\mathbf{v}$ required less than 0.01 sec on a Pentium 1.5 GHz. The shape of the ground has been chosen so as to be as chaotic as possible while still preserving the possibility to move the pendulums in order to avoid any sliding. For the simulation, the parameters that have been chosen for the robot are $\phi = 0.54$ for the angle friction coefficient, $\rho_1 = 85$mm, $\rho_2 = 75$mm, $\rho_3 = 85$mm for the radius of the wheels and $\ell_1 = \ell_2 = 350$mm for the lengths of the pendulums. The weights of the platforms and the pendulum are given by $\mu_1 = \mu_2 = 70$N, $\mu_3 = \mu_4 = 20$N. The height and the width of the stairs have been chosen equal to 220mm and 280mm, respectively. On Figure 4.1, dotted lines represent the Coulomb cones. Since each cone contains the corresponding reaction force created by the ground onto the wheel, the robot never slides.

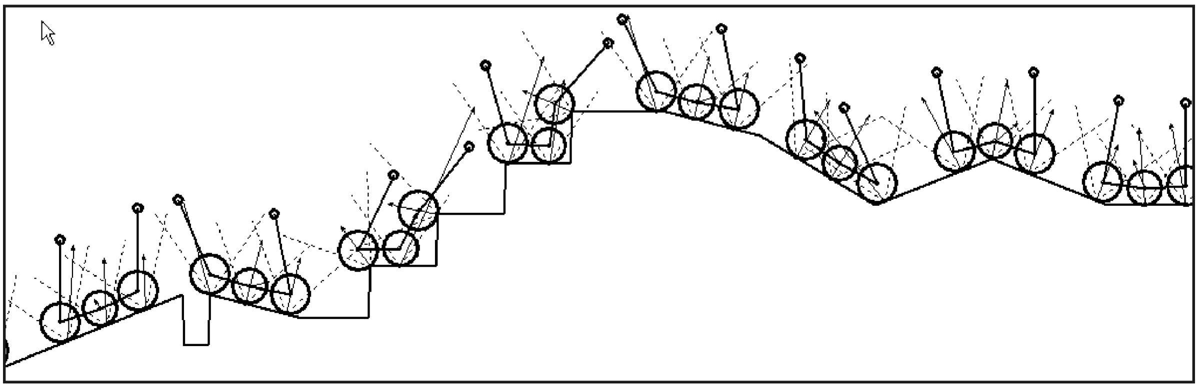

Figure 4.1: Different configurations for the robot; Since the Coulomb conditions are always satisfied, the robot never slides

5. Conclusion and perspectives

Mobile robots usually use wheels on smooth terrains and legs on rough terrains. When wheeled robots try to overcome obstacles which are bigger than the wheels, the robot slides. The first contribution of the paper is to show that a mass transfer system combined with an appropriate controller makes it possible to overcome obstacles with a wheeled robot, even when the obstacles are larger than the wheels.

The second contribution of the paper is to introduce linear programming methods (such as the simplex algorithm) for control. As an illustration, a two-dimensional three-wheeled robot, where pendulums had to be moved in order to avoid any sliding, has been considered.

The control approach that has been proposed can be used for systems involving linear viable constraints. For some problems, such has those involving mechanical robots, constraints are often linear. However, for most application problems, the constraints are nonlinear and more costly methods such as constraints propagation methods should then be used to find the viable control input $\mathbf{v}$.

Figure 5.1 represents the robot built by the robotics team of the ENSIETA engineering school (`www.ensieta.fr`) that has won the 2005 robot cup organized by ETAS, France. The robot can be seen as a three-dimensional version of the robot treated in this paper. It has been proven to be very competitive on irregular grounds but failed to cross over some compulsory obstacles (such as stairs). For the 2006 competition, we are planning to implement the mass transfer system on the robot. Carrying out the mass transfer as well as the localization of the robot with respect to the ground remain to be developed.

Acknowledgement: The author wants to thank A. Neme for helpful discussions on statically undetermined structures. He also wants to thank and all the members of the ENSIETA

Figure 5.1: Robot on which the mass transfer system will be implemented

robotics team for their substantial participation to the achievement of the robot. This team is composed with

- P. Gourmelen, J.F. Guillemette, M. Jaffres, P. Martinat who have designed and construct the mechanics of the robot;
- A. Arnold, P. Cambon, P. Dhaussy, R. Heas, F. Leroy, G. Le Maillot and their students T. Abdoul, L. Bachir, J. Despeaux, V. Jouault, R. Rivière, P. Yvon, N. Zharas who have developed the sensors, the electronics and the programs implemented in the robot.

References

[1] Scilab, `www-rocq.inria.fr/scilab/scilab.html`.

[2] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, pages 230–244, Las Cruces, NM, 1999.

[3] R. H. Bishop and R. Dorf. *Modern Control Systems*. Pearson Higher Education, 9th ed, 1978.

[4] J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.

[5] G. Dantzig. Programming in linear structure. *USAF, Washington D.C.*, 1948.

[6] J. Gere and S. Timoshenko. *Mechanics of materials*. CBS, second edition, 2002.

[7] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (Eds.). In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, NY, 1993.

[8] A. Isidory. *Nonlinear Control Systems: An Introduction, 3rd Ed.* Springer-Verlag, New-York, 1995.

[9] L. Jaulin. *Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base).* Hermès, London, 2005.

[10] L. Jaulin, M. Kieffer, O. Didrit, E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics.* Springer-Verlag, London, 2001.

[11] L. Matthies, Y. Xiong, R. Hogg, D. Zhu, A. Rankin, B. Kennedy, M. Hebert, R. Maclachlan, C. Won, T. Frost, G. Sukhatme, M. McHenry, and S. Gold-berg. A portable, autonomous urban reconnaissance robot. 2000.

[12] D. Meizel, A. Preciado-Ruiz, and E. Halbwachs. Estimation of mobile robot localization: geometric approaches. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 463–489. Plenum Press, New York, NY, 1996.

[13] U. Saranli, M. Buehler, and D. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20:616–631, 2001.

[14] R. Volp, J. Balaram, T. Ohm, and R. Ivlev. The rocky 7 march rover prototype. 1996.