Robudog's Design, Modeling and Control

Philippe Bidaud, Sébastien Barthélémy*, Pierre Jarrault*,*
*Damien Salle**, Eric Lucet**,*

**Institut des Systèmes Intelligents et de Robotique*

Université Pierre et Marie / CNRS FRE 2507

Pyramide T55

4, Place Jussieu 75252 Paris Cedex 05 - France

***Robosoft*

Tecnhopole d'Izarbel

64300 Bidard - France

1 Introduction

The growing interest in interactive robotics or in cognitive robotics has motivated the development of numerous animal-like robots among which the Sony/AIBO is probably the most popular one. AIBO has been used extensively in robot-dog soccer competitions or in human robot interactions for developing socially interactive robots. If we except AIBO, most of the animal-like robots can be assimilated to toys and the need for an open and evolutionary robotic platform is still present. Robudog has been designed especially to provide to researchers, teachers and hobbyists a powerful platform with advanced locomotion performances, evolutionary sensing capabilities and computing power capacities adapted to the development of complex behaviors.
This paper gives an overview of the mechanical design of Robudog, a dog-like quadruped which has been developed jointly by Robosoft and the Intelligent Systems and Robotics Institute. The mechanical design of the RobuDog is described as well as its forward kinematics model. Solutions for the leg inverse kinematics problem are given in a closed-form integrating the kinematic redundancy. Then, the input/output velocity model is developed. Its implementation in simulation shows how the kinematic redundancy can be used to improve its manoeuvrability and its stability. We will also show how it is possible to

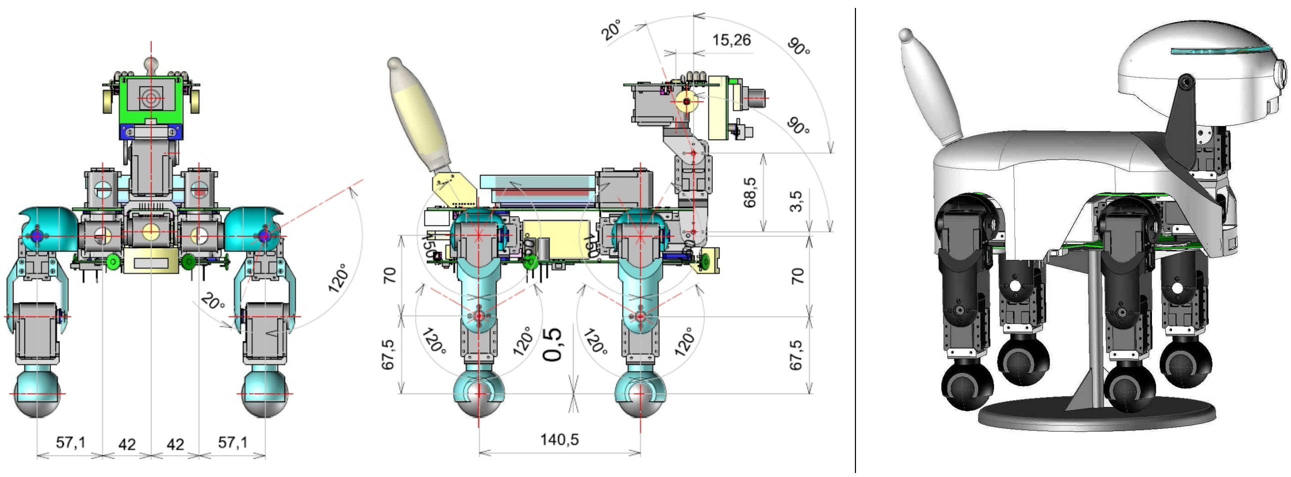

Fig. 1: Vues of the Robudog

formulate the control problem for taking into account non-slippage constraints and stability margin optimization during whole body-motions. Several basic behaviors will be used to demonstrate the applicability of the proposed method.
The principal RobuDOG characteristics are:

- Great moving freedom: thanks to the several degrees of freedom the quadruped robots have a large variety of possible movements.
- Direction cognition: a colour camera is positioning on the robot head and allows the detection of a ball. So the robot can track the ball and move to follow it.
- Completely autonomy: the robot dispose of several sensors, contact sensor (bumpers) to detect the contact with the ground, gyroscope and accelerometer, used also to the inclination computation.
- Wi-Fi communication that enable the dog to communicate with the user and with another dog too (for example to coordinate different dogs in a playground).

2 Mechanical Design and kinematic models for the quadruped

RubuDog is a 4-legged robot of which the front legs have four degrees of freedom (dof) and the rear legs have three dof. Figure 2 shows a shematic view of the Robudog kinematic structure. Front and rear legs have similar kinematic. The difference is that an additional revolute joint has been introduced to get a shoulder complex at the forelegs. This lead to a redundant kinematic making it more manoeuvrable than other dogs such as the AIBO's one for example. Each limb can be view as a separate "'manipulator"', defining an open-chain from a common base (the dog body) to the foot . All of the Robudog's joints are

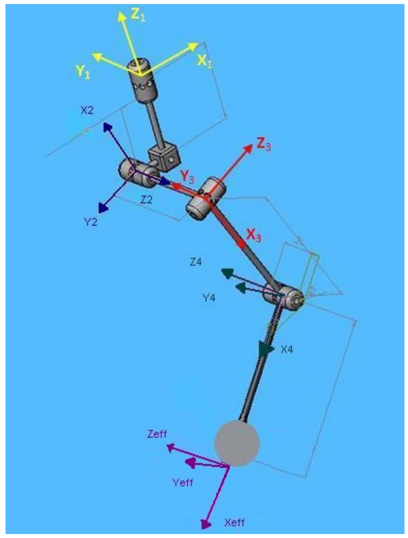

Fig. 2: Kinematic scheme of the Robudog

revolute. A local coordinate frame $(O_i^j, x_i^j, y_i^j, z_i, j)$ is attached to each link i of the leg j by following the modified Denavit-Hartenberg convention []. The reference frame $\mathcal{R}_0^j$ is attached to the leg j. The parameters defining the different euclidien transformations between 2 consecutive links are group togther in the table 1 and 2.

	T_{01}	T_{12}	T_{23}	T_{34}
α_i	0	0	0	0
d_i	0	0	0	0
θ_i	0	0	0	0
r_i	0	0	0	0

Tab. 1: table title

Based on these parameters, the forward kinematic model giving the foot tip coordinates in $\mathcal{R}_0$ can be established from the structure equation :

$$T_{0,0^j} \begin{pmatrix} P_x^j \\ P_y^j \\ P_z^j \\ 1 \end{pmatrix} = T_{03} \begin{pmatrix} D\,C_4 + d_4 \\ 0 \\ -D\,S_4 \\ 1 \end{pmatrix} \quad (1)$$

which leads to the vector equation under the general form $\mathbf{X}^j = \mathbf{f}(q)$:

$$T_{0,0^j} \begin{pmatrix} P_x^j \\ P_y^j \\ P_z^j \\ 1 \end{pmatrix} = \begin{pmatrix} (C_1C_2C_3 - S_1S_3)(D\,C_4 + d_4) - D\,C_1S_2S_4 + d_2C_1 - r_2S_1 \\ (S_1C_2C_3 + C_1S_3)(D\,C_4 + d_4) - D\,S_1S_2S_4 + d_2S_1 + r_2C_1 \\ -S_2C_3(D\,C_4 + d_4) - D\,C_2S_4 \\ \\ 1 \end{pmatrix} \tag{2}$$

the closed-form solution can be obtained only for the the rear legs which are not cinematically redundant. Analytical solutions for θ_2^j θ_3^j and θ_2^j are derived from the previous leg kinematic model in appendix 1.
The closed-form for the redundant forelegs inverse kinematic can be obtained by using the general method - derived from the Lagrangian multiplier method - previously proposed by **??**. The main advantages of this method compare to the others resolved motion methods is that it gives the exact solution and it is much more computationaly efficient. Following this method for solving the redundancy, a set of addditional equations derived from a continuous first-order criteria function are used. The crieria fonction used here the one which reflects the manipulability of the mechanism. If the manipulability is defined as :

$$H = det(JJ^t) \tag{3}$$

where J is the Jacobian which is in the frame $(O_3^j, x_0^j, y_0^j, z_0^j)$:

$$J_{0^j} = \begin{pmatrix} 0 & -S_1 & C_1S_2 & -C_1C_2S_3 - S_1C_3 \\ 0 & C_1 & S_1S_2 & -S_1C_2S_3 + C_1C_3 \\ 1 & 0 & C_2 & S_2S_3 \\ -(d_2S_1 + r_2C_1) & 0 & 0 & -d_4C_1S_2 \\ (d_2C_1 - r_2S_1) & 0 & 0 & -d_4S_1S_2 \\ 0 & 0 & 0 & -d_4C_2 \end{pmatrix} \tag{4}$$

By multiplying the gradient vector of $h = \nabla H$ by the matrix $Z = [J_{n-m}J^{-1} : I_{n-m}]$ which satisfy :

$$Zh = 0 \tag{5}$$

the $n = 4$ kinematic unknowns can be solved from the 2 5 scalar equations.

3 Hardware and Software Control Architecture

Figure 3 provides an overview of RobuDog's mechatronics architecture.

3.1 Hardware

- **Actuation** : The fourteen RobuDOg's active joints are driven by Robotis Dynamixel. Robot Actuators are high-performance actuators controlled by digital

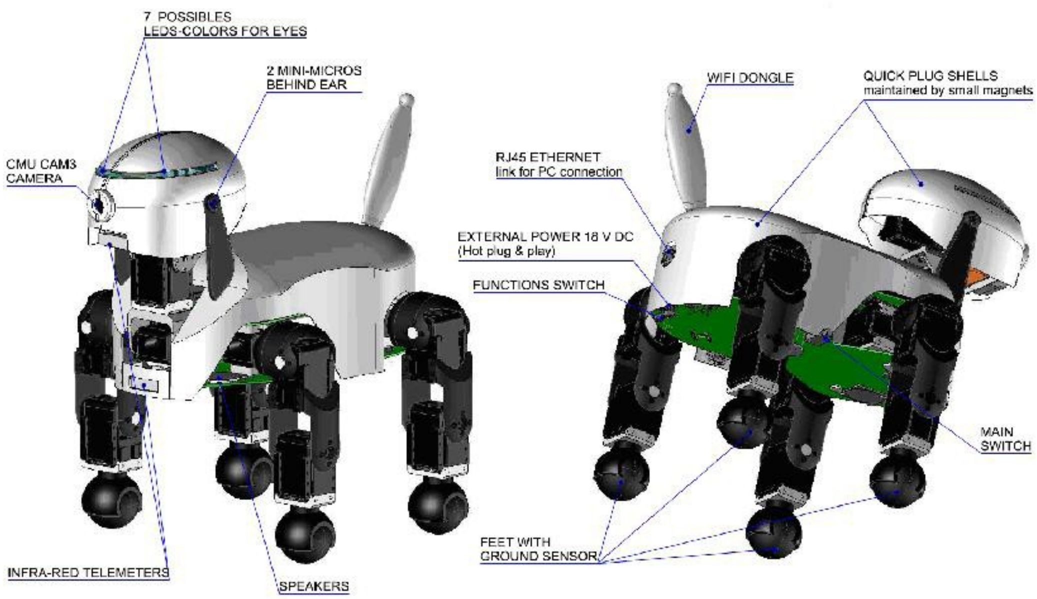

Fig. 3: RobuDog's description

packet communication. Position and speed can be controlled with a resolution of 1024 steps. It can provide current position or speed as well.

- **Postural sensors** : The posture change of the central body can be measured by a IDG-300 gyro that is a dual-axis gyroscope and the value of this posture can be updated relatively to the Earth's gravity by a three-axis accelerometer. It can also be used for feet slippage and fall detection.

- **Other sensors** : Each one of the four feet of the robot are equiped with a bumper to detect the foot-soil interaction, and two infra-red Sharp GP2D120 sensors permit to detect obstacles.

- **Camera** : RobuDog is equipped with a CMUcam3 (352x288 pixels RGB color sensor) which is an ARM7TDMI based fully programmable embedded computer vision sensor

- **On-board computer** It is run by an embedded computer AMD Geode LX 800 whose processor operates at 500MHz (512 Mb and 2 Gb SS, Wifi and Ethernet RJ-45) and which supports the Microsoft Windows XP and XP Embedded operating systems.

- **Autonomy and communications** A Lithium-Polymer battery 12VDC/4Ah ensures an autonomy of approximately 4 hours.

3.2 Programmation

The programming environment is based on the Robosoft robuBOX® and Microsoft® Robotics Studio. The robuBOX® allows optimal and riskless design of robot controllers by use of provided reference designs of architectures. The same softwares run in simulation and on the real robot. A collection of Services are done for:

- Drivers for robotics actuators, sensors and communication bus.
- Robotics algorithms, functions, behaviors, GUIs: from obstacle detection to path following, fleet management and autonomous navigation.

Generic interfaces definitions allowing easy re-use of services and easy integration of new algorithms in the existing architecture are established for data definition and exchange. This software architecture of the Robudog is presented here in figure 3.2: Based on the desired configurations of walk of the robots, an Excel macro generates two xml files : one for the configurations of each joint of one leg, and one for global positions with the four legs.

The environment also integrates a simulator based on the PhysX engine from AGEIA. For the simulation, the RobuDOG robot is provided with 3D models including the graphic 3D meshes and the physics and dynamics properties. The real or simulated robuDOG can be operated by 3 ways:

- Manually (default mode), using the Xbox360 wireless Gamepad. GamePadRobudog-Control service.
- Remotely, using a Graphical User Interface.
- Autonomously, developing customized software using robuBOX and MSRS.

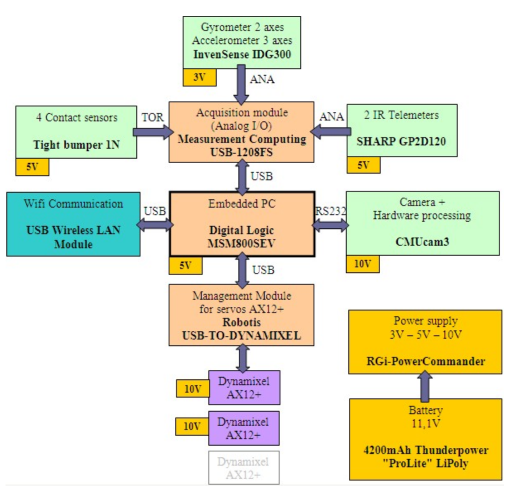

Fig. 4: RobuDog components

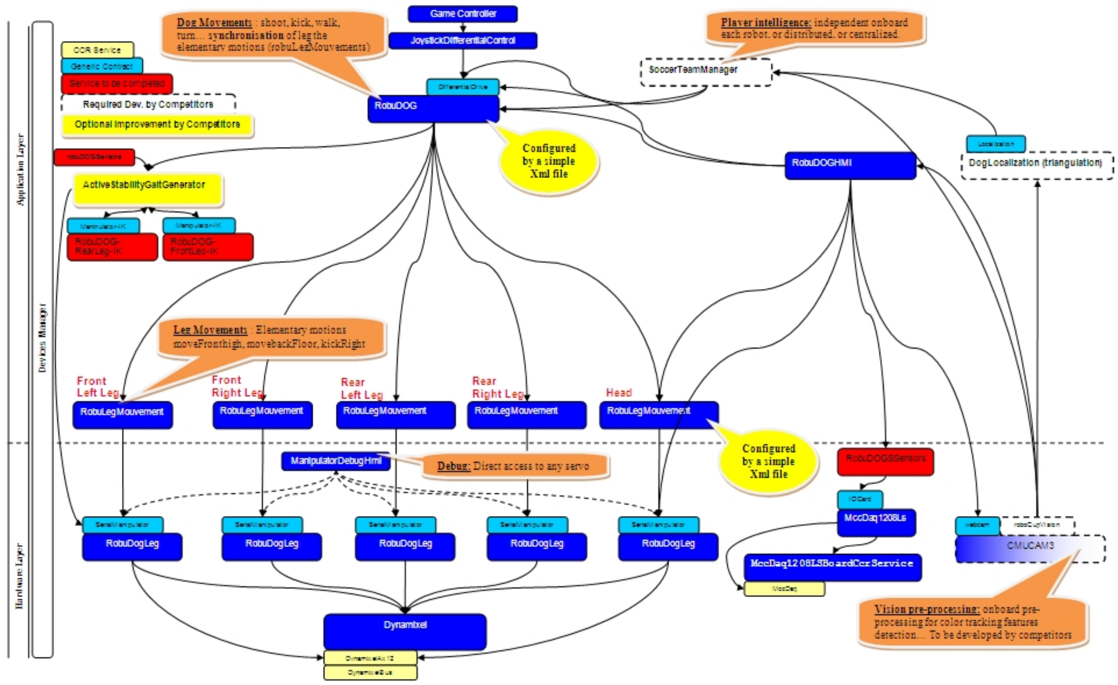

Fig. 5: Software architecture

4 Velocity control

The development of a kinematic controller leads to determine the set of active joint trajectories for a given trunk trajectory in position and orientation with respect to the ground frame. For this it is necessary to have the input/output velocity model. A conventional process for deriving this model for such mechanim consists in considering it as a free-flying multi-limb system constrainted by non-sliding conditions at the foot/floor interaction. When considering each limb separatly, we can define the foot tip velocity vector V_{P_j} with respect to the fixe frame $\mathcal{R}_f$ as :

$$V_{P_j} = V_0 + \hat{a}_j \omega + Jv_j \dot{q}_j = A_j \dot{X} + Jv_j \dot{q}_j \quad j = 1, 4$$

where : V_0, ω are respectively the linear and angular body velocity vectors wrt the fixed frame, $\hat{a}_j$ the matrix-form of $P_j O$ vector cross-product. Jv_j is the Jacobian of the limb j restricted to the linear motion. By concataining the previous relationships into a matrix form we get :

$$S\, V_P = \bar{S} A\, \dot{X} + Jv\, \dot{q} \tag{6}$$

where :

- $S \in R^{3n \times 3n}$ is a diagonal matrix defining the constraints on the foot tip velocities (0 if no sliding or defined by the trajectory imposed to the swing foot with respect to the body frame)

- $V_P = (V_{P_1}, V_{P_2}, V_{P_3}, V_{P_4})^t \in R^{3 \times n}$
- $Jv_B \in R^{3n \times n}$ is the block matrix formed by the individual limb Jacobians whose screws are expressed at point O.

¿From (6) $\dot{q}$ can be solved by using the Moore-Penrose pseudo-inverse of Jv_B :

$$\dot{q} = Jv_B^+[S\,V_P - \bar{S}A\,\dot{X}] + \alpha(I - Jv_B^+ Jv_B)\dot{q}_o \tag{7}$$

5 Dynamic Behavior Synthesis using Arboris

This section introduces a formulation of the problem for whole-body motion synthesis for satisfying the contact stability in a dynamic evolution of the system. Contact stability measures have been introduced through the position of some reference points with respect to the contact support area such as ZMP, FRI, CMP ground reference points provide a quantitative information regarding the static or dynamic equilibrium of the posture. A limitation in these measures is that do not integer the limits on frictioal forces as well as joint torque capabilities. To overcome this restriction the force closure measures which have been developed by the past for grasp and fixture analysis can be reconsidered. A formulation of the frictional constraints as a feasable contact wrench domain has lead to a more universal measure to estimate if the foot contact is sufficiently weakly. Based on this latter representation, multiple motion objectives subject to constraints imposed by actuation limits and non-sliding frictional contact constraints can be formulated as a QP problem. We will use the sit to stand to illustrate the implementation in the Arboris software of this formulation for the synthesis of elementary behaviors for the RobuDog.

6 References

References

P.H. Chang, *A Closed form solution for inverse kinematics of robot manipulators with redundancy*, **I**EEE Journal of Robotics and Automation, Vol 3, No 5, Oct 1987

7 Appendix

- Appendix 1 : Homogeneous transformation matrices

$$T_{01} = \begin{pmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \; T_{12} = \begin{pmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & 1 & r_2 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \; T_{02} = \begin{pmatrix} C_1C_2 & -C_1S_2 & -S_1 & -r_2S_1 \\ S_1C_2 & -S_1S_2 & C_1 & r_2C_1 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{23} = \begin{pmatrix} C_3 & -S_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \; T_{34} = \begin{pmatrix} C_4 & -S_4 & 0 & d_4 \\ 0 & 0 & 1 & 0 \\ -S_4 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{13} = \begin{pmatrix} C_2C_3 & -C_2S_3 & S_2 & 0 \\ S_3 & C_3 & 0 & r_2 \\ -S_2C_3 & S_2S_3 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{03} = \begin{pmatrix} C_1C_2C_3 - S_1S_3 & -C_1C_2S_3 - S_1C_3 & C_1S_2 & -r_2S_1 \\ S_1C_2C_3 + C_1S_3 & -S_1C_2S_3 + C_1C_3 & S_1S_2 & r_2C_1 \\ -S_2C_3 & S_2S_3 & C_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Appendix 2 : Inverse Kinematic Solutions for the back legs

¿From the equation :

$$T_{12}^{-1} \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix} = T_{23} \begin{pmatrix} D\,C_4 + d_4 \\ 0 \\ -D\,S_4 \\ 1 \end{pmatrix}$$

with : $T_{12}^{-1} = \begin{pmatrix} C_2 & 0 & -S_2 & -d_2C_2 \\ -S_2 & 0 & -C_2 & d_2S_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

we get the following relationships :

$$x_PC_2 - z_PS_2 = (D\,C_4 + d_4)C_3$$

$$x_PS_2 + z_PC_2 = D\,S_4$$

$$y_P = (D\,C_4 + d_4)S_3$$

which can be rewriten as :

$$x_PC_2 - z_PS_2 = AC_3$$

$$x_PS_2 + z_PC_2 = B$$

$$y_P = AS_3$$

θ_4 can be solved from these equations

$$x_P^2 + z_P^2 + y_P^2 = A^2 + B^2$$

$$OP_j^2 = (D\,C_4 + d_4)^2 + (D\,S_4)^2 = D^2 + 2Dd_4C_4 + d_4^2$$

$$C_4 = (1/2Dd_4)(x_P^2 + z_P^2 + y_P^2 - D^2 - d_4^2)$$

$$S_4 = (1 - C_4^2)^{1/2}$$

$$\theta_4 = atan2(S_4, C_4)$$

Then θ_3 can be solved, since :

$$S_3 = y_P/(D\,C_4 + d_4)$$

and $C_3 = (1 - C_3^2)^{1/2}$

$$\theta_3 = atan2(S_3, C_3)$$

Finally θ_2 can be obtained from : $x_P S_2 = D\,S_4 - z_P C_2$

$$x_P^2(1 - C_2^2) - z_P^2 C_2^2 - 2 z_P B C_2 = 0$$

$$(-x_P^2 - z_P^2) C_2^2 + 2 z_P B C_2 - B^2 = 0$$

$$C_2 = \frac{z_P B \pm \sqrt{(z_P B)^2 - (x_P^2 + z_P^2) B}}{(x_P^2 + z_P^2)}$$

Identically we get :

$$S_2 = \frac{2 x_P B \pm \sqrt{(x_P B)^2 - (x_P^2 + z_P^2) B}}{(x_P^2 + z_P^2)}$$

- Appendix 3 :