Feedforward control of vibrations in flexible and lightweight robots

Arthur Lismonde[1], Valentin Sonneville[2] and Olivier Brüls[1]

[1]*Department of Aerospace and Mechanical Engineering, University of Liège, Belgium, {alismonde,o.bruls}@ulg.ac.be*
[2]*Aerospace Engineering, University of Maryland, USA, vspsonn@umd.edu*

Abstract

Lightweight and flexible robots have a high potential in today tendency to use collaborative automation. Thanks to their reduced weight and increased compliance, such systems benefit from an intrinsic safety that reduce the risk of injury in case of unexpected collision. However, the controller of such system has to be carefully designed to deal properly with the flexible behavior of the links and joints.

This work focuses on the control of the flexible behavior in general 3D robotic manipulators. In particular, an innovative feedforward control command is developed to reduce vibrations in the robot during its motion. First, a finite element model of the robot is built using rigid bodies, flexible beam elements and kinematic joints elements [1]. Based on it, the inverse dynamics is solved using so-called stable inversion techniques [2]. These methods can deal with the non-minimum phase nature, i.e. unstable nature, of such flexible and nonlinear systems. Here, a constrained optimization formulation, introduced in [3] for 2D systems and extended here to 3D systems, is used to solve the inverse dynamics problem. In future work, this method could be implemented on a robotic testbed with a flexible end-link to test its performances.

1 Introduction

Robot manipulators are evolving in order to meet current needs regarding accuracy and safety. To improve the latter, lighweight and collaborative robot can be a good alternative. However, controlling lightweight and flexible structures is not an easy task as they can be subjected to vibration and elastic deformation issues. Such flexible manipulators are said to be underactuated since they potentially have an infinite number of degrees of freedom (dof) and a finite number of actuators. To reduce such flexibility issues, the controller of such manipulator has to be designed carefully. Feedback action can be implemented to compensate for vibrations see, e.g., [4]. A second possibility is to model such flexible multibody system (MBS) in order to compute an input feedforward control signal that results in a vibration-free motion of the robot. Both the feedforward and the feedback control methods can be combined to achieve robust performances as presented in [5, 6].

To perform an end-effector trajectory tracking task, an example of feedforward commands for the manipulator would be the torques of each of its joints. To find those inputs, the inverse dynamics of the MBS needs to be solved. In the case of a flexible system, some internal dynamics remains when the output trajectory is prescribed. The system is said to be non-minimum phase when this internal dynamics is unstable. If the inverse dynamics of a non-minimum phase system is simply solved using time integration algorithms or computed torques methods, the resulting input control can be unbounded. In order to obtain a bounded solution, a non-causal solution can be considered. A time domain inverse dynamics method is presented and tested for a linear system in [7]. For flexible nonlinear systems, a stable inversion method is presented in [2] and is applied in [8, 9]. An optimal control approach is proposed in [3] for 2D multibody systems. The present work extends this last method to solve the inverse dynamics of flexible 3D systems. The flexible MBS is modeled using nonlinear beam finite elements [10], rigid bodies and kinematic joints [1] formulated on the special Euclidean group $SE(3)$. The inverse dynamics is then stated as an optimization problem where the amplitude of the internal dynamics has to be minimized. The prescribed end-effector trajectory is defined as an additional servo constraint of the optimization problem.

Please note that the present paper is a summary of references [11, 12], please see the latter for more details.

2 Dynamic model of the flexible multibody system

The finite element formalism can be used to model the dynamics of flexible MBS including rigid and flexible bodies interconnected by kinematic joints [1, 10]. With the special Euclidean group $SE(3)$ formalism, the position and orientation of each element composing the finite element mesh is represented as a 4×4 homogeneous transformation matrix $\mathbf{H}_I$ with a rotation $\mathbf{R}_I \in SO(3)$ and a position $\mathbf{p}_I \in \mathbb{R}^3$ component.

$$\mathbf{H}_I = \begin{pmatrix} \mathbf{R}_I & \mathbf{p}_I \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3)$$

This representation leads to a local frame representation that reduces the non-linearity of the equations of motion and allows a representation of the rotations without singularity issues.
The configuration $\mathbf{H}$ can then be represented as a bloc diagonal matrix that gathers each so-called nodal variable described above.

$$\mathbf{H} = \text{diag}(\mathbf{H}_1, ..., \mathbf{H}_N)$$

Thanks to r control inputs $\mathbf{u} = [u_1, ..., u_r]$, the end-effector position $\mathbf{y}_{eff}$ of the MBS follows a prescribed trajectory $\mathbf{y}_{presc}(t)$. The latter is evolving in time and is defined by r scalar components. If such MBS is defined using N nodal variables $\mathbf{H}_I$, the equations that govern the dynamics of such MBS are

$$\dot{\mathbf{H}}_I = \mathbf{H}_I \tilde{\mathbf{v}}_I \quad \text{with } I = 1, ..., N \quad (1)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{g}(\mathbf{H}, \mathbf{v}) + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{A}\mathbf{u} \quad (2)$$

$$\boldsymbol{\Phi}(\mathbf{H}) = \mathbf{0} \quad (3)$$

$$\mathbf{y}_{eff}(\mathbf{H}) - \mathbf{y}_{presc}(t) = \mathbf{0} \quad (4)$$

where $\mathbf{M}$ is the system symmetric mass matrix, $\mathbf{v} = (\mathbf{v}_1^T, ..., \mathbf{v}_N^T)^T$ is the vector of nodal velocities, $\mathbf{g}$ is the vector of internal and complementary inertia forces, $\mathbf{B}$ is the gradient of the kinematic constraints $\boldsymbol{\Phi}$, which are used to represent the connections imposed by the kinematic joints. The matrix $\mathbf{A}$ is a boolean matrix that applies the controls $\mathbf{u}$ on the system. The m dimensional vector $\boldsymbol{\lambda}$ is Lagrange multipliers related to the m kinematic constraints $\boldsymbol{\Phi}$. The last equation is called the *servo* constraint [13] and fixes a part of the motion. It assures that the end-effector position $\mathbf{y}_{eff}$ follows the prescribed trajectory $\mathbf{y}_{presc}(t)$.
In the case of an underactuated 3D system, $6N - m - r > 0$ is the dimension of the internal dynamics i.e., the flexible dynamics, which is represented by Eqs. (1)-(4). The trajectory would be completely specified if some initial conditions were provided for the internal dynamics. However, if the internal dynamics is unstable, the forward propagation of the initial condition would lead to an unbounded solution requiring very large control efforts $\mathbf{u}$, as represented in Fig. 1. A bounded solution can be defined using an optimization formulation, in which the initial conditions on the dynamics are left free.

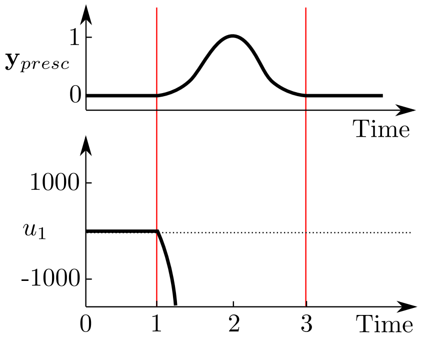

Fig. 1: Unbounded solution of the inverse dynamics when the internal dynamics is unstable.

3 Optimization problem

Based on the model of the flexible manipulator, the inverse dynamics problem is formulated as a constrained optimization problem where the internal dynamics is to be minimized. Considering that the internal dynamics can be represented using a function $\phi(\mathbf{H})$ depending on the nodal configurations, the optimization problem is the minimization of the objective function J on the time lap $T = t_f - t_i$.
Mathematically,

$$\min_{\mathbf{H}} J = \min_{\mathbf{H}} \frac{1}{2T} \int_{t_i}^{t_f} \|\phi(\mathbf{H})\|^2 dt \quad (5)$$

subjected to the equality constraints defined by the equation of motion of the flexible MBS Eqs. (1)-(4) for $t \in [t_i, t_f]$. One may observe that, in this fomulation, no initial and final values of $\mathbf{H}$ and $\mathbf{v}$ are defined. They are determined by the optimization algorithm itself.

3.1 Optimization process

To start the optimization process, an initial guess of the trajectory $\mathbf{H}(t)$ is required. To compute it, we can solve the inverse dynamics of an equivalent rigid manipulator which is a purely algebraic problem, since there is no internal dynamics in this case. Let this initial trajectory have a hat $\hat{\bullet}$ e.g., $\hat{\mathbf{H}}(t)$. The optimization is then carried out using a direct transcription method, i.e. the time interval is first discretized in s time steps t^k $(k = 1, ..., s)$ so that the optimization problem is reformulated as a discrete Nonlinear Programming (NLP) problem. Eventually, after a few iterations, the optimized trajectory is found as $(\mathbf{H}^1, ..., \mathbf{H}^s)$. Fig. 2 illustrates the process.
In the discrete settings, the minimization of the objective function, previously given by Eq. (5), can therefore be written in its discrete form

$$J = \frac{1}{2T} \sum_{k=1}^{s} \left[\|\phi(\mathbf{H}^k)\|^2 \right] h \quad (6)$$

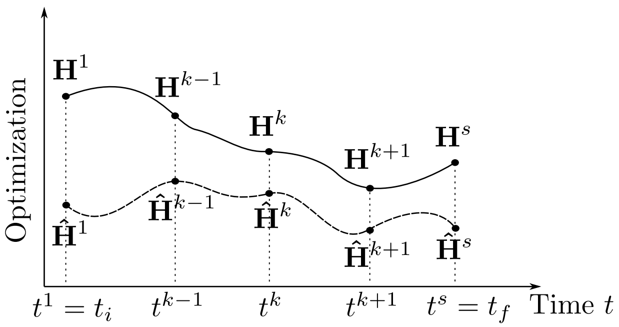

Fig. 2: Direct transcription method: optimization starting from the initial guess $(\hat{\mathbf{H}}^1, ..., \hat{\mathbf{H}}^s)$ that leads to the optimal trajectory $(\mathbf{H}^1, ..., \mathbf{H}^s)$.

subjected to the discrete constraints at each time step t^k

$$\dot{\mathbf{H}}_I^k - \mathbf{H}_I^k \tilde{\mathbf{v}}_I^k = \mathbf{0} \tag{7}$$

$$\mathbf{M}^k \dot{\mathbf{v}}^k + \mathbf{g}(\mathbf{H}^k, \mathbf{v}^k) + \mathbf{B}^{k,T} \boldsymbol{\lambda}^k - \mathbf{A}\mathbf{u}^k = \mathbf{0} \tag{8}$$

$$\boldsymbol{\Phi}(\mathbf{H}^k) = \mathbf{0} \tag{9}$$

$$\mathbf{y}_{eff}(\mathbf{H}^k) - \mathbf{y}_{presc}(t^k) = \mathbf{0} \tag{10}$$

where h is the time step size, $I = 1, ..., N$ and $k = 1, ..., s$. Additional time integration constraints are required to connect the discrete nodal configurations, velocities and accelerations.

3.2 Optimization variables

After discretization, the unknown variables of the optimization problem are

$$(\mathbf{H}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \mathbf{a}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, ..., \mathbf{H}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \mathbf{a}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

with $\mathbf{H}^k = \text{diag}(\mathbf{H}_1^k, ..., \mathbf{H}_N^k)$ and each $\mathbf{H}_I^k \in SE(3)$. Obviously, some components of this set of variables belong to the $SE(3)$ group and are represented as matrices, for $k = 1, ..., s$. Classical optimization methods are not able to solve problems formulated with such matrix representation and dedicated methods are needed. Alternatively, in order to solve this problem using classical techniques, a reformulation based on a vectorial incremental variables is thus proposed.

We introduce the vector of incremental variables $\Delta\mathbf{q}^T = (\Delta\mathbf{q}_1^T, ..., \Delta\mathbf{q}_N^T)$, which determines the change between the initial guess $\hat{\mathbf{H}}$ and their current value $\mathbf{H}$.

At time step k, the relation between the configuration $\mathbf{H}^k = \text{diag}(\mathbf{H}_1^k, ..., \mathbf{H}_N^k)$ and the incremental variables $\Delta\mathbf{q}^{k,T} = (\Delta\mathbf{q}_1^{k,T}, ..., \Delta\mathbf{q}_N^{k,T})$ is

$$\mathbf{H}_I^k \quad = \quad \hat{\mathbf{H}}_I^k \exp_{SE(3)}(\widetilde{\Delta\mathbf{q}_I^k}) \tag{11}$$

where $\hat{\mathbf{H}}_I^k$ represents the position and orientation of node I at time step k for the initial guess.

The actual design variables $\mathbf{x}$ are thus

$$\mathbf{x} = (\Delta\mathbf{q}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \mathbf{a}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, ..., \Delta\mathbf{q}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \mathbf{a}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

The optimization problem has now vectorial design variables and can be solved using a classical NLP algorithm. For consistency, the configurations $\mathbf{H}^k$ and $\mathbf{H}^{k+1}$ at two consecutive time steps are also related through another exponantial mapping and a time related incremental variable $\Delta\mathbf{Q}^k$. The relation between the relevant variables is illustrated in Fig. 3. Each arrow represents an exponential mapping $\exp_{SE(3)}(\tilde{\bullet})$ with either time incremental or configuration incremental arguments, i.e. $\Delta\mathbf{Q}^k$ and $\Delta\mathbf{q}^k$ respectively.

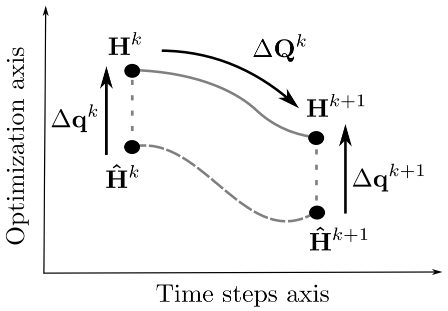

Fig. 3: Relation between variables $\Delta\mathbf{q}^k$, $\Delta\mathbf{Q}^k$ and $\mathbf{H}^k$.

4 3D serial example

The inverse dynamics of a flexible 3D system is now solved using the proposed approach. A serial 3 dof manipulator, as shown in Fig. 4, is considered. It is composed of two links: an upper arm and a forearm. The end-effector is modeled as a point mass m_{end} at the tip of the forearm. The upper arm and the forearm both have length l and a tubular square cross section. The upper arm has a side length a_1 and an edge thickness e_1. The forearm has a side length a_2 and an edge thickness e_2. While the former has a greater cross section and is considered as a rigid body element, the forearm is considered flexible and is modeled using 4 beam elements. The description of the beam formulation on $SE(3)$ can be found in [10]. The upper arm connects the first two hinge joints, controlled using inputs u_1 and u_2, to the third one, controlled using input u_3. The outputs of the system are the x, y and z components of the end-effector position $\mathbf{y}_{eff}$. The first hinge joint has its axis along axis z. The second and the third hinge joints initially have their axis along axis y. In the initial position, each link makes a 45° angle with respect to the x axis. By analysing the poles of the serial system, we find that the first unstable pole is located at the frequency of 13 Hz.

The trajectory the end-effector has to follow is a planar circular arc in the yz plane. The motion profile is built using a seventh order polynomial in order to insure continuity of the position, velocities, accelerations and jerks over time. The end-effector starts from position $[2l\cos(45°)\ 0\ 0]$ and goes to position $[2l\cos(45°)\ l\ 0]$. The radius of the circular arc is thus $l/2$ m. The trajectory is covered in 1.1 s and the pre- and postactuation phases both last 0.2 s: the total simulation time is 1.5 s. The material parameters and dimensions of the 3D flexible arm can be found in Table 1.

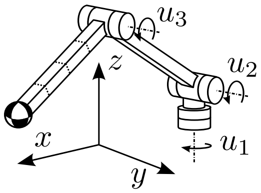

Fig. 4: Serial 3D arm system with one rigid body and 4 beam elements.

Tab. 1: Parameters of the serial arm system.

Upper arm	$l = 1$ m	$a_1 = 0.05$ m	$e_1 = 0.01$ m $\rho = 2700$ kg/m^3
Forearm	$l = 1$ m $E = 70$ GPa	$a_2 = 0.0075$ m $\nu = 0.3$	$e_2 = 0.0015$ m $\rho = 2700$ kg/m^3
End-eff.		$m_{end} = 0.1$ kg	

The convergence of the optimization process is quite sensitive to the initial guess of the problem. To compute it, a complete rigid system is considered. The beams of the forearm are replaced with a rigid body with the same geometrical and material properties. When gravity is acting on the robotic arm, it is important to correct the initial guess with the static deflection of the actual flexible arm. Regarding the numerical parameters of the generalized-α method, a spectral radius of $\rho_\infty = 0.01$ is considered ($\beta = 0.98$, $\gamma = 1.48$, $\alpha_m = -0.97$ and $\alpha_f = 0.01$). In order to best capture the system's dynamics, the system is discretized into $s = 150$ steps. This means that the time step size h is 0.01 s, which is about a tenth of the first unstable frequency.

Using the default tolerances of the FMINCON solver in Matlab®, the optimization is completed after 5 iterations and lasts 4 minutes (using a x64 bits i7-4600u CPU with 16 Gb RAM memory). The command inputs $\mathbf{u}$ and $\mathbf{u}_{rigid}$, with and without flexibility considerations respectively, are compared in Fig. 5. Some visible differences can be observed but the pre- and post-actuation in the input commands are hardly visible. These pre- and post-actuation phases can be observed by looking at the velocity profile of the three joints in Fig. 6. One can see that after 1.3 s, the velocity of the third joint is still varying. Although the torque u_3 is nearly zero in the post-actuation phase, it still results in some internal motion in the arm. It is important to point out that although some motion is present in the joints, the end-effector does not actually move during the post-actuation.

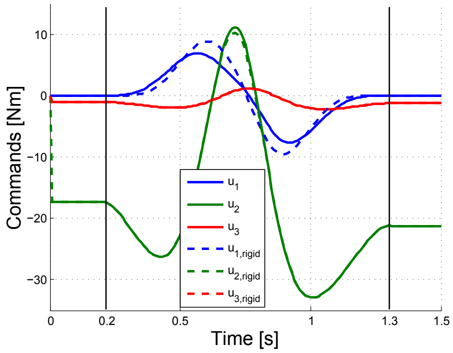

Fig. 5: Resulting joint velocity of a flexible 3D arm.

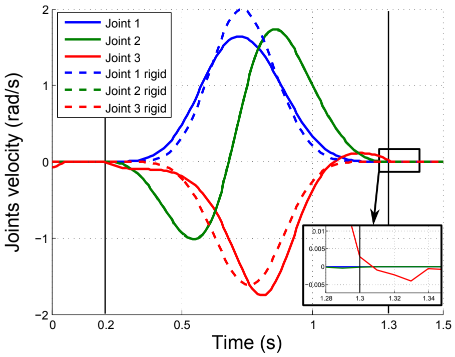

Fig. 6: Velocity of the three joint of a flexible 3D arm.

In order to verify the computed optimal inputs $\mathbf{u}$, both inputs $\mathbf{u}$ and $\mathbf{u}_{rigid}$ are applied to the flexible system and a direct dynamic analysis is performed. These inputs lead to end-effector trajectories shown in Fig. 7. The relative tracking error resulting from both direct simulations can be calculated at each time step t^k using (12).

$$e^k = \frac{\|\mathbf{y}_{presc} - \mathbf{y}_{eff}\|}{\|\mathbf{y}_{presc}\|} \tag{12}$$

where $\|\bullet\|$ is the classical Euclidean norm or L_2 norm. The relative rms error is then calculated as

$$e_{rms} = \sqrt{\frac{1}{s}\sum_{k=1}^{s}(e^k)^2} \tag{13}$$

The relative rms error e_{rms} is equal to 1.1% when $\mathbf{u}_{rigid}$ is used as input and drops down to $e_{rms} = 0.3\%$ when $\mathbf{u}$ is used.

5 Conclusion

In this work, the inverse dynamics of 3D flexible robotic arm is successfully solved using an constrained optimization formulation. The MBS is first modeled using finite elements formulated on the special euclidean

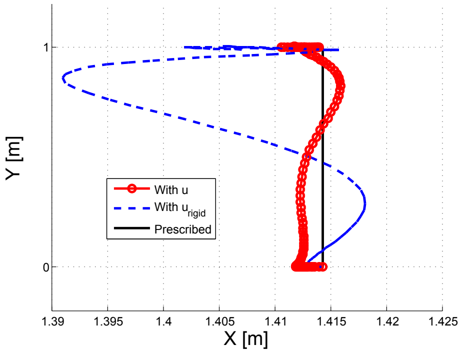

Fig. 7: End-effector trajectory of a flexible 3D arm using $\mathbf{u}$ and $\mathbf{u}_{rigid}$.

group $SE(3)$. Based on it, the inverse dynamics problem is defined as an optimization problem where the internal dynamics of the MBS is minimized. A direct transcription method is used to discretize the continuous optimization problem into a discrete optimization problem. In order to use classical optimization solvers, vectorial incremental variable are introduced to avoid optimization variables defined on $SE(3)$. The input commands computed using such methods manage to improve the tracking precision of the 3D flexible manipulator, as shown by the serial example. In further work, with the design of a suitable feedback loop, this method will be applied for off-line computation of the feedforward command of an experimental testbed.

Acknowledgements

The first author would like to acknowledge the Belgian Fund for Research training in Industry and Agriculture for its financial support (FRIA grant).

References

[1] V. Sonneville and O. Brüls, “A formulation on the special Euclidean group for dynamic analysis of multibody systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 9, 2014.

[2] S. Devasia, D. Chen, and B. Paden, “Nonlinear inversion-based output tracking,” *IEEE Transactions on Automatic Control*, vol. 41, no. 7, 1996.

[3] G. J. Bastos, R. Seifried, and O. Brüls, “Inverse dynamics of serial and parallel underactuated multibody systems using a DAE optimal control approach,” *Multibody System Dynamics*, vol. 30, pp. 359–376, 2013.

[4] R. Franke, J. Malzahn, T. Nierobisch, F. Hoffmann, and T. Bertram, “Vibration control of a multi-link flexible robot arm with fiber-bragg-grating sensors,” *In Proceedings of IEEE International Conference on Robotics and Automation*, 2009.

[5] J. Malzahn, M. Ruderman, A. S. Phung, F. Hoffmann., and T. Bertram, “Input shaping and strain gauge feedback vibration control of an elastic robotic arm,” *In Proceedings of IEEE Conference on Control and Fault Tolerant Systems*, 2010.

[6] A. De Luca, “Feedforward/feedback laws for the control of flexible robots,” *In Proceedings of the IEEE International Conference on Robotics & Automation*, 2000.

[7] D.-S. Kwon and W. J. Book, “A time-domain inverse dynamic tracking control of a single link flexible manipulator,” *Journal of Dynamic Systems, Measurement and Control*, vol. 116, pp. 193–200, 1994.

[8] R. Seifried, *Dynamics of Underactuated Multibody Systems: Modeling, Control and Optimal Design*. Solid Mechanics and its Applications, Springer, 2014.

[9] R. Seifried and P. Eberhard, “Design of feed-forward control for underactuated multibody systems with kinematic redundancy,” *Motion and Vibration Control: Selected papers from MOVIC 2008*, 2009.

[10] V. Sonneville, A. Cardona, and O. Brüls, “Geometrically exact beam finite element formulated on the special euclidean group SE(3),” *Computer Methods in Applied Mechanics & Engineering*, vol. 268, pp. 451–474, 2014.

[11] A. Lismonde, V. Sonneville, and O. Brüls, “Solving the inverse dynamics of a flexible 3d robot for a trajectory tracking task,” *In Proceedings of 21st International Conference on Methods and Models in Automation and Robotics*, 2016.

[12] A. Lismonde, V. Sonneville, and O. Brüls, “Trajectory planning of soft link robots with improved intrinsic safety,” *In Proceedings of the 20th World Congress of the International Federation of Automatic Control*, 2017.

[13] W. Blajer and K. Kolodziejczyk, “A geometric approach to solving problems of control constraints: Theory and a dae framework,” *Multibody System Dynamics*, vol. 11, pp. 343 – 364, 2004.