Control of an Aerial Blimp Based on Visual Input

Sjoerd van der Zwaan, Matteo Perrone, Alexandre Bernardino, José Santos-Victor*

Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais,
1049-001 Lisboa,
Portugal.

Abstract. This paper describes a method for controlling a lighter-than-air robotic vehicle based on visual input. Due to the motion disturbances in the environment (currents), these tasks are important to keep the vehicle stabilized relative to an external reference frame. The main difficulties are related to the non-holonomic constraints of the blimp moving in 3D, having a limited number of controllable degrees of freedom. The relative position of the vehicle with respect to a docking station is tracked using vision. A planar surface is chosen as a reference plane which allows visual tracking of an environmental region, based on planar projective transformations. The observed image deformations of the tracked surface reveal the camera motion parameters, from which the vehicles pose can be reconstructed. This information is used to realize non-linear feedback control. Experiments and results are described and discussed.

1 Introduction

Recently, research on the utilization of unmanned aerial vehicles has grown with an increasing interest on robotic airships, also known as blimps or lighter-than-air vehicles. The motivation behind it is that airships outperform airplanes and helicopters in low-speed, low altitude applications, having an enormous potential for tasks like environmental and traffic monitoring, climate research, transportation, etc. Several references can be found in literature about airship modeling for autonomous navigation [2, 3].

This work is integrated in the NARVAL[2] project, where one of the goals consists in using visual feedback for controlling an underwater vehicle. Since the blimp kinematics and dynamics are a reasonable approximation of those of the submarine, we use the blimp as a test-bed for experiments in a laboratory environment. As water is more viscous than air, the control will be simpler in that case, when compared to the blimp.

The problem addressed is that of controlling a small-sized, indoor blimp based on visual input, in order to attain some desired position and orientation face to an external reference frame. Posture stabilization can be seen as a generalization of the docking and station keeping tasks. Station keeping consists in stabilizing the vehicle against some external reference frame or environmental region, thus rejecting external disturbances like currents. The docking problem consists of controlling the vehicle in order to attain a desired position and orientation relative to a chosen coordinate frame.

Assuming short range regions in the environment, one can use vision in order to extract information about the vehicle's pose and use this information for visual servoing. Assuming that an image patch is initially identified by some user, the temporal changes of this image patch induced by the vehicle's motion, need to be tracked. As the vehicle moves in 3D, such an image can be distorted according to a fairly general set of deformations when compared to those usually considered in robotics and computer vision (translational or affine motions). In this paper, the tracking system is based on full planar projective transformations in real-time, thus accommodating a wide set of possible image deformations.

Section 2 describes the method used for visual tracking of the identified image patch. Section 3 describes the airborne blimp together with its dynamic model. The control aspects are further detailed in Section 4 and Section 5 describes the robot pose estimation from the visual input. Experiments and simulation results are described in Section 6. Finally, in Section 7, conclusions are drawn and future work is indicated.

* Email:{sjoerd,mperrone,alex,jasv}@isr.ist.utl.pt

[2] ESPRIT-LTR Project 30185, NARVAL - Navigation of Autonomous Robots via Active Environmental Perception

2 Tracking of Image Regions

Given a reference image I_0 and a target image I_1, the registration problem is that of computing an image transformation $(u', v') = \omega(u, v)$, that maps a region R of the reference image onto a specified region of the target image, such that $I_1(u, v) = I_0\big(\omega(u, v)\big)$. Usually, these transformations, are parameterized as a function of a vector $\mathbf{q}$: $\omega(u, v) = \omega(u, v; \mathbf{q})$. In this work we are mostly interested in tracking planar regions. Planar motions cannot be adequately modeled by simple image transforms, like affine or translational. A projective planar transformation is the exact motion model when a camera rotates about its eyepoint or if the image surface is planar [4]. The 2D projective transformation depends on 8 parameters:

$$u' = \frac{q_1 u + q_2 v + q_3}{q_7 u + q_8 v + 1}, \quad v' = \frac{q_4 u + q_5 v + q_6}{q_7 u + q_8 v + 1} \tag{1}$$

To register the two image regions, the best possible match can be obtained through the minimization of an error function, using an appropriate norm, such as the sum-of-squared-differences ($L2$-error criterion). We assume that at each time instant we have a prediction of the current transformation parameters, $\mathbf{q_0}$. The most simple prediction may be the parameters of the last time step. More elaborate prediction methods can rely on Kalman filtering. In our case we compute the optic flow in the region and adjust an affine model to the computed flow. This provides a first step towards the solution and only small adjustments remain to be made. Assuming that I_0 and I_1 have been approximately registered and considering images as column vectors, the $L2$ error function is given by:

$$e(\Delta\mathbf{q}) = \frac{1}{2} \parallel I_1^{'} - \bar{I}_0(\Delta\mathbf{q}) \parallel^2 \tag{2}$$

, where $I_1^{'} = I_1(\omega^{-1}(x, y; \mathbf{q_0}))$ is the image obtained from mapping I_1 approximately onto I_0 using the initial estimate of the transform parameters ($\mathbf{q_0}$) and $\bar{I}_0(\Delta\mathbf{q}) = I_0(\omega(x, y; \Delta\mathbf{q}))$ suggests small adjustments in order to further optimize the registration between I_0 and I_1. To minimize this error function, usual gradient descent methods use the partial derivatives of $\bar{I}_0(\Delta\mathbf{q})$ whose discrete approximations can be computed as $\bar{I}_0(\delta\mathbf{e}_i) - I_0$, where $\mathbf{e}_i$ is the i'th basis vector and δ is an adequate value for discretization, depending on the shape of the error function. However, for an 8 dimensional space this method has a weak interpolation capability, i.e. it only searches the space effectively in the coordinate directions, and within a range determined by the parameter δ. Furthermore, minimizing eq. (2) using an exhaustive search on the parameter space would be impractical. Instead we assume a set of vectors $\{\Delta\mathbf{q}_i : i \in (1 \dots m)\}$ that sample the parameter space more densely for the expected image deformations and range. The parameter vector $\Delta\mathbf{q}$ can be expressed as a linear combination of the various $\Delta\mathbf{q}_i$:

$$\Delta\mathbf{q} = \sum_{i=1}^{m} k_i \Delta\mathbf{q}_i \tag{3}$$

Now the image space can be considered as a function of the parameter vector $\mathbf{k} = [k_1 \dots k_m]^T$. The new parameterization is given by: $\hat{I}_0(\mathbf{k}) = \bar{I}_0(\sum_{i=1}^{m} k_i \Delta\mathbf{q}_i)$. For small deviations about $\mathbf{k} = \mathbf{0}$ we have the first order approximation: $\hat{I}_0(\mathbf{k}) \approx I_0 + \sum_{i=1}^{m} \frac{\partial \hat{I}_0}{\partial k_i} k_i$. A suitable discrete approximation of each partial derivative in the previous equation is given by:

$$\frac{\partial \hat{I}_0}{\partial k_i} = \hat{I}_0(\mathbf{e}_i) - I_0 = \bar{I}_0(\Delta\mathbf{q}_i) - I_0 = B_i \tag{4}$$

In [1], the set of vectors B_i are denoted "Difference Templates" and are also used for image registration, but they are justified in a different form. The error function now takes the form:

$$e(\mathbf{k}) = \frac{1}{2} \parallel D - B\mathbf{k} \parallel^2 \tag{5}$$

where D is the difference image $I_1 - I_0$ and B is the partial derivatives matrix: $B = [B_1 \dots B_m]$. Vector k can be determined minimizing:

$$min_k \parallel D - B\mathbf{k} \parallel^2 \quad \Rightarrow \quad B^T D - B^T B\mathbf{k} = 0 \tag{6}$$

After determining $\mathbf{k}$, the solution for $\Delta\mathbf{q}$ can be calculated using the linear combination in eq. (3). Using the updated estimates of the parameter vector $\Delta\mathbf{q}$, this process can proceed in subsequent iterations and at different resolutions of the image pyramid.

3 Airship Operation and Modeling

The small-size indoor blimp used for this work is illustrated in Figure 1. It is equipped with an on-board camera and video link, sending live images to a computer where image processing is realized. Here, proper control signals are derived and sent to the blimp via a radio link.

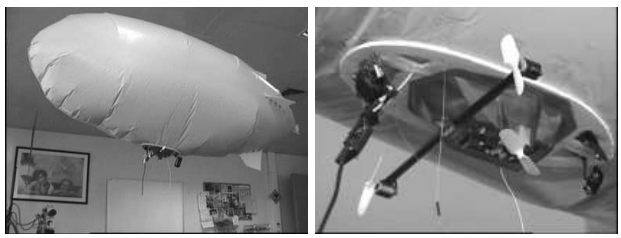


Fig. 1.: Radio controlled indoor blimp with on-board camera.

In order to derive a mathematical model, it is useful to first review some physical principles of airship operation [3]. First of all, an important characteristic is the aerostatic lift, which, unlike the lift forces generated over a wing surface, is independent of flight speed. The aerostatic lift force comes from Archimedes' Principle and is equal to the mass of the volume of air displaced by the airship's envelope. The aerostatic lift is also known as the buoyant force. An upward lift is obtained when the airship's envelope contains a gas with a density lower then air. Helium is the most commonly used lifting gas.

Second, a buoyant body in motion in a fluid displaces each fluid particle in the direction of motion of the body. The fluid therefore gains kinetic energy and the body experiences a resistance to its motion. This effect can be taken into account by considering added mass and inertia terms. As the blimp displaces a relative large volume of air during flight, these properties become significant and the body behaves as if it had a mass and moments of inertia substantially higher than those indicated by conventional physical methods.

Finally, during airship flight, aerodynamic forces are generated, which are a complex function of its shape, flight speed and control surfaces. In general, these can be divided into forces generated over control surfaces (introducing roll, pitch and yaw moments) and drag forces, from which the airship experiences a resistance to its motion. The drag forces are due to linear and quadratic skin friction (arising from laminar and turbulent boundary layer conditions) and due to vortex shedding, which depends on the streamline of the airship hull. In the particular case of a small-sized indoor blimp moving at low speed, only linear skin friction coefficients can be considered and control surfaces will have no effect. Therefore, the controllable degrees of freedom for the indoor blimp are determined by the geometrical arrangement of the propulsive units attached to the gondola.

The airship has a non-uniform mass distribution and its center-of-mass is difficult to locate. Therefore, motion has to be referenced to by a system of orthogonal body axes, $\{LTA\}$, placed at the geometric center of the envelope volume, as shown in Figure 2. The centre of volume can be assumed to coincide with the center

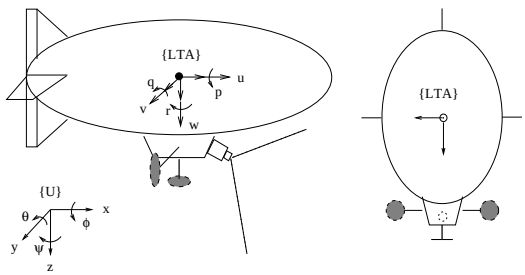


Fig. 2.: Definition of reference frames.

of buoyancy. Considering the vehicle as a rigid body (not taking into account its elasticity) the dynamic

model can be obtained by writing down the Newton-Euler equations of motion resolved into the body-fixed reference frame, resulting in [8, 3]:

$$M\dot{\nu} + C(\nu) + D(\nu)\nu + G = \tau, \tag{7}$$

where $\nu = [u \quad v \quad w \quad p \quad q \quad r]^T$, is the velocity vector containing the three components of linear velocity (u, v, w) and the three components of angular velocity (p, q, r), $M_{6\times6}$ is the inertia matrix containing all (added) masses and inertia terms, $C_{6\times1}(x)$ is the vector of Coriolis and centrifugal forces, $D_{6\times1}(x)$ is the vector of aerodynamic forces, $G_{6\times1}$ is a vector containing the gravitational and buoyancy forces and $\tau_{6\times1}$ is the vector of propulsion forces. For the computation of each of these components we refer to [8, 3, 9].

For control and navigation purposes, the velocity vector in eq.(7) must be transformed to the earth-fixed inertial frame $\{U\}$, leading to the kinematic relations. The relative position and orientation of $\{LTA\}$ with respect to $\{U\}$ will be denoted by the vector $\eta = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$, where a standard *roll, pitch* and *yaw* angles representation is assumed for the orientation. The kinematic equations can then be written as:

$$\dot{\eta} = \begin{bmatrix} J_1(\phi,\theta,\psi) & 0_{3\times3} \\ 0_{3\times3} & J_2(\phi,\theta,\psi) \end{bmatrix} \nu, \tag{8}$$

where J_1 is the corresponding rotation matrix and J_2 a Jacobian matrix relating the angular velocity vector to the time derivatives of the attitude parameters. The Jacobian J_2 can be easily calculated from the rotation matrix (see [5]).

4 Decoupled Control Design

The control problem addressed here is that of stabilizing the blimp about a desired pose in the horizontal plane defined by the xy-axes of the earth-fixed reference frame, while maintained at some fixed height. Assuming zero roll and pitch angles (which is a reasonable approximation for an indoor blimp moving at low speed), the model described by eq.(7) and (8) can be considered decoupled into two non-interacting (or lightly interacting) subsystems, describing motion in the horizontal plane and in the vertical plane of the earth-fixed reference frame.

Motion in the horizontal plane can be described by a simplified kinematic model:

$$\dot{x} = u cos\psi \quad \dot{y} = u sin\psi \quad \dot{\psi} = r \tag{9}$$

where (x, y) are Cartesian coordinates of the origin of the vehicle reference frame in the horizontal plane, ψ is the orientation (yaw-angle) of the vehicle in the horizontal plane and (u, r) are the vehicle linear forward velocity and angular velocity around its z-axis expressed in the vehicle reference frame. These are controlled by the common-mode and differential-mode signals resulting from the thrust of the left and right propellers. This motion model captures the non-holonomic constraints of the blimp motion in the horizontal plane. In [7], a non-linear control law is proposed to stabilize a non-holonomic robot about a desired posture (x_r, y_r, ψ_r) by using smooth time-varying feedback control. Using this control strategy, the desired behavior for the closed loop system is to stabilize the blimp in the horizontal plane towards any final position and orientation with feedback of the type:

$$\begin{bmatrix} T_{cmn} \\ T_{diff} \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -g(t) & -k_3 \end{bmatrix} \begin{bmatrix} R^T_{2\times2}(\psi) & 0 \\ & 0 \\ 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \psi - \psi_r \end{bmatrix} \tag{10}$$

where T_{cmn} is the common-mode signal and T_{diff} the differential mode signal of the resulting propeller thrust, k_1 and k_3 are positive numbers, $R(\psi)$ is the 2×2 rotation matrix describing the robot orientation in the horizontal plane and $g(t)$ is a bounded function at least once differentiable such that $\delta g(t)/\delta t$ does not tend to zero when t tends to infinity; for instance, $g(t) = sint$.

The vertical model can be obtained from eq.(7) and (8) by considering the subsystem given by the state vector: $[w \quad q \quad \theta \quad z]^T$. Linearizing the subsystem about some equilibrium, it is possible to obtain a linear system, from which the transfer function can be obtained relating the blimp altitude, z, to the control input,

T_v, which is the thrust of the vertical propeller. Using this transfer function, a PID-controller can be tuned for dynamically controlling the altitude:

$$T_v = K_p e_z(t) + K_d \dot{e}_z(t) + K_i \int_0^t e(\tau) d\tau \tag{11}$$

where $e_z = z_r - z$ is the error in altitude and T_v is the thrust of the vertical propeller. The closed loop system aims at stabilizing the blimp at some fixed height, rejecting any disturbances coming from variations in the buoyancy force or dynamic coupling effects between the blimp forward velocity, u, (generating small pitching moments) and altitude, z.

5 Robot Pose Estimation from Image Registration

The observed deformations in the image plane of a static planar surface viewed from a camera traveling in 3D, reveals information about the camera motion. In section 2, we introduced the problem of image registration and proposed a method to track the projection of a planar surface in the image-plane over time, based on planar projective transformations. In this section we consider the problem of robot pose estimation from this image registration.

Consider the camera attached to the blimp and traveling in 3D space while forming the image of a plane. The position and orientation of the camera retina at successive instances of time are related to the initial retina position by a rotation matrix R and a translation vector $\mathbf{t}$ applied to the canonical coordinate system attached to the camera. The plane is defined by its normal $\mathbf{n}$ and distance d to the origin of the camera coordinate system at the initial time instance. Also consider the coordinate system to be attached to the camera such that the projection of a point $M(X, Y, Z)$ is the point $m(x, y)$ determined by the relations: $X/x = Y/y = Z$. Note that (x, y) are metric image coordinates which can be related to the real image coordinates in pixels, (u, v), by an affine transformation determined by the camera intrinsic parameters:

$$\begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} = \begin{bmatrix} fK_x & 0 & u_0 \\ 0 & fK_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{12}$$

where f is the camera focal length, K_x and K_y determine the pixel dimensions and u_0 and v_0 give the offset in pixels of the center of the image plane.

In [6] it is shown that the projections m_1 and m_2 into the camera retina of a $3D$ point $M(X, Y, Z)$ belonging to a plane P are related by a homography: $m_2 = Am_1$. The homography depends only on the relative position and orientation of the camera retina at the different time instances and on the relative position of the plane in relation to the camera retina at the initial time t_1, as illustrated in Figure 3.

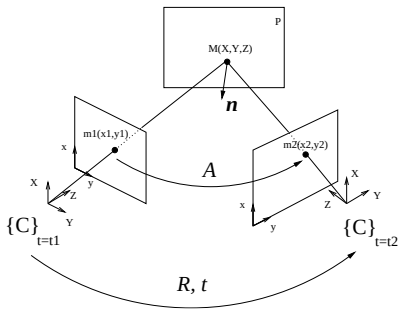


Fig. 3.: Geometry of the camera motion problem; the camera canonical coordinate system is given by $\{C\}$.

Moreover, the matrix A is related to the geometric parameters of the camera motion problem by the relation:

$$A = dR + \mathbf{t}\mathbf{n}^t \tag{13}$$

Determining the homography thus allows to explicitly compute the normal $\mathbf{n}$ to the plane and the rotation R of the camera retina together with the translation $\mathbf{t}$ during the time interval $[t_1, t_2]$, where the translation vector is scaled by the distance d to the plane. The solution adopted for this decomposition is described in [6] and is based on single value decomposition of A.

With the image registration technique described in section 2, the tracker at each time instance outputs an update of the current image transform parameters. This transform relates the image points (u', v') in the current observed image to the points (u, v) in the initial identified patch. Writing the parameters in eq.(1) into matrix form, the homography A is related to the transform parameters by the camera intrinsic parameters:

$$A = P^{-1} \begin{bmatrix} q_1 & q_2 & q_3 \\ q_4 & q_5 & q_6 \\ q_7 & q_8 & 1 \end{bmatrix} P \tag{14}$$

Decomposing this homography at each time instance and knowing the distance d to the plane then reveals the camera rotation and translation with respect to its initial position. Taking the camera initial coordinate frame as the earth-fixed reference frame and knowing the camera position and orientation with respect to the robot reference frame then allows to reconstruct the robot trajectory over time and estimate its current pose.

6 Results

A set of real experiments were done using the blimp with the on-board camera. Figure 4 shows the results of tracking an image window. In spite of the distortion introduced by the blimp's camera, the implemented tracker successfully follows the specified image patch while the blimp is moving in 3D. Feature tracking is realized at a sampling rate of 12Hz.

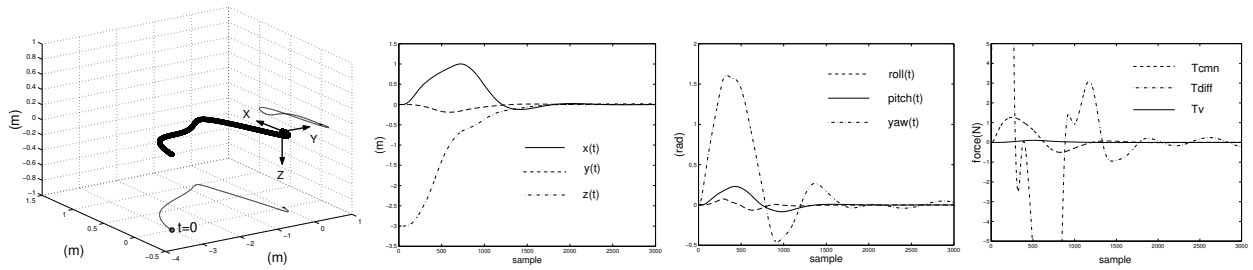
Fig. 4.: Tracking an image window from the flying blimp.

Early result were obtained from real experiments, concerning the problem of automatically controlling the blimp based on measurements in the image plane. These results were encouraging but are not described in this paper however. Here we present results obtained with the decoupled control design when applied to an implemented simulation model of the blimp, including its kinematics and dynamics.

Figure 5 shows the simulation results of a station keeping experiment with the blimp. At the left side the trajectory in 3D is illustrated under closed-loop control. The initial lateral deviation in the horizontal plane ($x_0 = 0, y_0 = -3, \psi_0 = 0$) is reduced and the control stabilizes the origin ($x_r = 0, y_r = 0, \psi_r = 0$). The non-linear controller, designed for a kinematic model, performs well when applied to the non-linear dynamic model of the blimp, generating smooth control signals. Note that at sample $t \approx 600$ the vehicle starts moving backward. The blimp altitude is dynamically controlled to remain fixed at $z = 0$. The non-zero linear velocity (u) generated by the non-linear controller introduces a small pitch angle, resulting in small variations of the blimp height. These variations are dynamically compensated by the altitude controller.

7 Conclusions

In this paper we have explained how the problems of docking and station keeping of an airship can be addressed using visual feedback, selecting an image patch on the docking station. The presented method tracks the selected region, thus following its temporal evolution in $3D$-space. This information is then used



a b c d

Fig. 5.: Simulation results of a station keeping test with the blimp; (a) blimp under closed-loop control while recovering an initial lateral deviation; projections of the trajectory onto the horizontal and vertical plane are shown together with the earth-fixed reference frame ; (b) blimp position over time; (c) blimp orientation over time; (d) generated control signals.

to reconstruct the blimp trajectory and estimate its current pose. A decoupled control strategy is explored, using time-varying non-linear feedback for pose stabilization in the horizontal plane. Although this controller was designed based on a kinematic model, simulation shows encouraging results when applied to the blimp dynamic model. The main difficulties arise in those situations for which positioning errors occur along the blimp y-axis. In this case, since the blimp is a non-holonomic vehicle and does not have lateral degrees of freedom, the only solution is to compensate this error by rotating the blimp.

Work is being carried out to develop more robust and adequate control strategies, taking into account the blimp non-linear dynamic and kinematic relations. However, even with the decoupled control system it was possible to obtain encouraging results.

Acknowledgements

This work has been funded by ESPRIT-LTR Project 30185, NARVAL.

References

1. M. Gleicher. Projective Registration with Difference Decomposition. *IEEE Conf. of Computer Vision and Pattern Recognition*, pages 331–337, jun 1997.
2. A. Elfes and S. Sigueira Bueno and M. Bergerman and J.Jr.G Ramos. A Semi-Autonomous Robotic Airship for Environmental Monitoring Missions. *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.
3. S.B. Varella Gomes and J.Jr.G. Ramos. Airship Dynamic Modeling for Autonomous Operation. *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.
4. R. Szelinski. Image Mosaicing for Tele-Reality Applications. *IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, 1994.
5. John J. Craig. *Introduction to Robotics.* Addison Wesley, 2nd edition, 1986.
6. O. Faugeras *Three-dimensional Computer Vision.* MIT press 1993.
7. C. Canudas de Wit and B. Siciliano and G. Bastin (Eds). *Theory of Robot Control.* Springer-verlag, 1996.
8. Thor I. Fossen. *Guidance and Control of Ocean Vehicles.* John-Wiley & Sons, 1995.
9. G. A. Khoury and J. D. Gillett. *Airship Technology.* Cambrigde University Press, 1999.

This article was processed using the TEX macro package with SIRS2000 style