A NOVEL SIGNAL FLOW GRAPH BASED SOLUTION FOR CALCULATION OF FIRST AND SECOND DERIVATIVES OF DYNAMICAL NONLINEAR SYSTEMS

Andrea Arcangeli, Stefano Squartini, Francesco Piazza

DEIT, Università Politecnica delle Marche
Via Brecce Bianche 60131 Ancona, Italy (Europe)
Phone: +39 0712204453, fax: +390712204835, email: sts@deit.univpm.it

ABSTRACT

In this paper, the problem of calculation of first and second derivatives in general non-linear dynamical systems is addressed and an attempt of solution by means of signal flow graph (SFG) techniques is proposed. First and full second derivatives of an output of the initial system respect with the node variables of the starting SFG are delivered through an adjoint graph derived without using Lee's theorem. Mixed second derivatives are deduced by quantities attained in adjoint graphs of the original graph or graphs related to it. A detailed theoretical demonstration of these formulations is given. Even though no adjoint graph has been derived in case of mixed derivatives, the ability of the proposed method to determine all Hessian matrix entries in a complete automatic way is highlighted.

1. INTRODUCTION

The problem of sensitivity calculation of complex discrete time domain systems has been effectively dealt with in literature through Signal-Flow-Graph (SFG) techniques [1]. New algorithms for gradient computation in gradient-based linear and non-linear adaptive networks [2]-[4] have been developed. A more recent work [5] represents a generalization of previous approaches, as it delivers derivatives of an output at a given time with respect to a variation of an internal parameter not necessarily at the same instant, allowing to derive the gradient information for dynamical non-linear system learning. This is achieved by a suitable adjoint network, whose definition is completely based on Lee's theorem [2]. This method allowed to work out an automatic procedure involving Jacobean based information for system adapting, but it has left open the problem of Hessian matrix calculation. The present paper is claimed to give a solution to such a task. On purpose, a new adjoint graph (defined without using Lee's theorem) is derived to yield first and full second derivatives, from which the mixed ones can be deduced, as it will be explained later on.

As in [5], the class of systems here addressed is that of non-linear dynamical ones, like dynamical neural networks. They can be represented by SFGs. A SFG is a set of nodes and oriented branches. A branch is oriented from the initial node i and the final node j. There are three different types of nodes: input nodes, which have only one outgoing branch, output nodes, which have only one incoming branch, and general n-nodes, which sum of incoming branches and distribute the result to outcoming branches. All other branches beside input and output ones are called f-branches. Each of them have two variable associated: the initial x one at the tail of branch and the final one v at the head of the branch. We can distinguish among two different types of f-branches in dependence of the relation between x and v. Taking into account we address the case of discrete time systems and that in our notation only branches are indexed, we can list them as:

1) Static branch: $v_j = f_j\left(x_j(t), \alpha_j(t), t\right)$

2) Single delay branch : $v_j = z^{-1}x_j(t) = x_j(t-1)$

Function f of the static branch is requested to be differentiable and it can change by time, since it depends on the time-varying parameter $\alpha_j(t)$. This let it cover a large class of nonlinearities, the most common functional relations being:

$$\begin{cases} v_j = f_j\left(x_j(t)\right) & \textit{non-linear branch} \\ v_j = w_j(t)x_j(t) & \textit{linear branch} \end{cases}$$

2. FIRST AND FULL SECOND DERIVATIVES

Here we derive an adjoint graph for automatic computation of first and second derivatives of an output of the original graph respect with its nodes. The word adjoint is not here used as rigorously as its definition requires: in fact the derived graph for derivatives calculation is a reverse graph but does not preserve the same topology of the original one. Such a graph is characterized by two level, one exclusively for first derivatives and both for determination of full second derivatives. This fact immediately lets us introduce the most relevant difference between this approach and that one suitable for general systems in literature [1]: the Lee's theorem has not been employed to define the functional relations of the adjoint graph branches. Manipulation of derivative operation has revealed to suffice for our purposes. However, we got a generalization of previous approach: indeed, if we consider only the part of the adjoint graph relative to first derivatives we notice that it coincides with the adjoint graph referred in [1]. Concerning second derivatives, it must be noted that only the full ones are here calculated: the mixed ones will be considered in the following.

Now we can proceed to describe which are the transformations to do to get the adjoint graph from the topology of the original one. They are listed in Table 1. The first step consists on showing the validity of such transformations in case of topologically linear graph, i.e. a SFG where each node has just one incoming and one outgoing branches, and where there are one input and one output branches. The starting branch to analyze is the output one, that should represent the two-level input branch for the adjoint SFG. As output branch is a zero-memory branch and different outputs are independent each other (also for non-linear topology SFGs), the following holds:

$$\begin{cases} \dfrac{\partial y_k(t)}{\partial y_j(t)} = \delta_{kj} \\ \dfrac{\partial y_k(t)}{\partial y_k(t')} = \dfrac{\partial y_k(t)}{\partial y_k(t-\tau)} = \delta(\tau) = \begin{cases} 1 & \text{if } \tau = 0 \\ 0 & \text{if } \tau > 0 \end{cases} \end{cases}$$

In case of second derivative, it suffices to derivate further:

$$y_j'' = \frac{\partial^2 y_k(t)}{\partial y_j(t')^2} = \frac{\partial^2 y_k(t)}{\partial y_j(t-\tau)^2} = 0$$

Let us continue showing the formulas derived in case of static branches. We move from the supposition that derivatives respect with node v_j are correct, that means:

$$\begin{cases} v_j(t-\tau) = f_j\left(x_j(t-\tau), \alpha_j(t-\tau), t-\tau\right) \\ v_j'(\tau) = \dfrac{\partial y_k(t)}{\partial v_j(t-\tau)}, \quad v_j''(\tau) = \dfrac{\partial^2 y_k(t)}{\partial v_j(t-\tau)^2} \end{cases}$$

Hence, we can calculate derivatives of the output respect with the node $x_j = v_{j-1}$. Concerning the first derivative we have:

$$x_j'(\tau) = \frac{\partial y_k(t)}{\partial x_j(t-\tau)} = \frac{\partial y_k(t)}{\partial v_j(t-\tau)} \frac{\partial v_k(t-\tau)}{\partial x_j(t-\tau)} = v_j'(\tau) \cdot f_j'(t-\tau)$$

while for second derivative it can be deduced:

$$\begin{aligned} x_j''(\tau) &= \frac{\partial^2 y_k(t)}{\partial x_j^2(t-\tau)} = \frac{\partial}{\partial x_j(t-\tau)}\left(\frac{\partial y_k(t)}{\partial x_j(t-\tau)}\right) = \\ &= \frac{\partial}{\partial x_j(t-\tau)}\left(\frac{\partial y_k(t)}{\partial v_j(t-\tau)} \frac{\partial v_k(t-\tau)}{\partial x_j(t-\tau)}\right) = \\ &= \frac{\partial^2 y_k(t)}{\partial x_j \partial v_j(t-\tau)} \frac{\partial v_k(t-\tau)}{\partial x_j(t-\tau)} + \frac{\partial y_k(t)}{\partial v_j(t-\tau)} \frac{\partial^2 v_k(t-\tau)}{\partial x_j^2(t-\tau)} = \\ &= v_j''(\tau)\left(f_j'(t-\tau)\right)^2 + v_j'(\tau) \cdot f_j''(t-\tau) \end{aligned}$$

where we named:

$$\begin{cases} f_j'(t-\tau) = f_j'\left(x_j(t-\tau), \alpha_j(t-\tau), t-\tau\right) \\ f_j''(t-\tau) = f_j''\left(x_j(t-\tau), \alpha_j(t-\tau), t-\tau\right) \end{cases}$$

Regarding to unit delay branch we can write, under the same afore mentioned hypotheses:

$$\begin{cases} x_j'(\tau) = \dfrac{\partial y_k(t)}{\partial x_j(t-\tau)} = \dfrac{\partial y_k(t)}{\partial v_j(t-\tau+1)} = v_j'(\tau-1) \\ x_j''(\tau) = \dfrac{\partial^2 y_k(t)}{\partial x_j(t-\tau)^2} = \dfrac{\partial^2 y_k(t)}{\partial v_j(t-\tau+1)^2} = v_j''(\tau-1) \end{cases}$$

In case of input branch we do not have to derive anything since the corresponding branch in adjoint SFG fulfils only the role to deliver the derivatives of the node to which it is connected. So, proof of validity of transformation set can conclude here if we limit our analysis to topologically linear SFG. However, we can generalize such a result, observing that a general graph can be seen as several topologically linear graphs connected through addition and distribution nodes, (Figure 1). This means to arrange previous formulas to fit then to this new scenario. For example, we can not employ input branches in adjoint SFGs (for each topologically linear graph involved) as:

$$y_j' = \delta_{kj}\delta(\tau), \quad y_j'' = 0$$

since these are not the right derivatives respect with those input nodes in corresponding original SFGs. However, we shall be able to show that an addition node is a distribution node in the adjoint graph and vice versa. Indeed, let us suppose, without loss of generality that the system output depends on the addition node through a generic function with memory as follows:

$$y_k = h(\tilde{u}_\Sigma), \quad \tilde{u}_\Sigma = \tilde{u}_1 + \ldots + \tilde{u}_n = \sum_{j=1}^{n} \tilde{u}_j$$

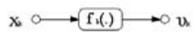
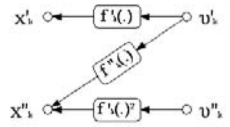
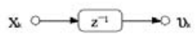
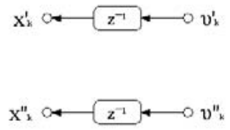
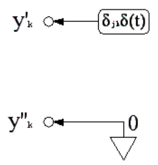
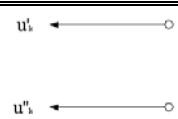
f	**Original SFG**	**Adjoint SFG**
Static	 $v_k(t) = $ $= f_k\left(x_k(t), \alpha_k(t), t\right)$	 $x_k'(\tau) = f_k'(\tau) \cdot v_k'(\tau)$ $x_k''(\tau) = f_k''(\tau) \cdot v_k'(\tau) +$ $+ f_k'(\tau)^2 \cdot v_k''(\tau)$ $f_k'(\tau) = \left.\dfrac{\partial f_k}{\partial x_k}\right\rvert_{x_k(t-\tau), \alpha_k(t-\tau), (t-\tau)}$ $f_k''(\tau) = \left.\dfrac{\partial^2 f_k}{\partial x_k^2}\right\rvert_{x_k(t-\tau), \alpha_k(t-\tau), (t-\tau)}$
Unit Delay	 $v_k(t) =$ $= \begin{cases} x_k(t-1) & t > 0 \\ 0 & t = 0 \end{cases}$	 $x_k'(\tau) = \begin{cases} v_k'(\tau-1) & \tau > 0 \\ 0 & \tau = 0 \end{cases}$ $x_k''(\tau) = \begin{cases} v_k''(\tau-1) & \tau > 0 \\ 0 & \tau = 0 \end{cases}$
Output	 $y_k(t) = x_k(t)$	 $y_k' = \delta_{jk}\delta(\tau)$ $y_k'' = 0$
Input	 $v_k(t) = u_k(t)$	 $u_k'(\tau) = v_k'(\tau)$ $u_k''(\tau) = v_k''(\tau)$

Table 1. Transformations of functional relations in *f*-branches from original graph to adjoint one.

where *n* is the number of incoming branches. We have then:

$$\frac{\partial y_k}{\partial \tilde{u}_j} = \frac{\partial y_k}{\partial \tilde{u}_\Sigma} \frac{\partial \tilde{u}_\Sigma}{\partial \tilde{u}_j}, \quad \frac{\partial \tilde{u}_\Sigma}{\partial \tilde{u}_j} = 1$$

$$\Rightarrow v_j' = \frac{\partial y_k}{\partial \tilde{u}_j} = \frac{\partial y_k}{\partial \tilde{u}_\Sigma} = v_\Sigma' \qquad \forall j = 1..n$$

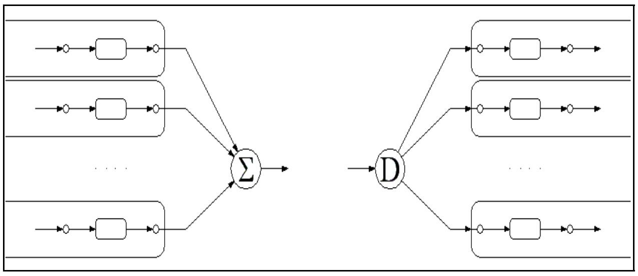


Figure 1. Addition and distribution nodes connecting topologically linear SFGs.

Similar considerations can be made for second derivatives:

$$v_j" = \frac{\partial^2 y_k}{\partial \tilde{u}_j^{\,2}} = \frac{\partial}{\partial \tilde{u}_j}\left(\frac{\partial y_k}{\partial \tilde{u}_j}\right) = \frac{\partial}{\partial \tilde{u}_j}\left(\frac{\partial y_k}{\partial \tilde{u}_\Sigma}\right) =$$

$$= \frac{\partial^2 y_k}{\partial \tilde{u}_\Sigma^{\,2}} \frac{\partial \tilde{u}_\Sigma}{\partial \tilde{u}_j} = \frac{\partial^2 y_k}{\partial \tilde{u}_\Sigma^{\,2}} = v_\Sigma", \qquad \forall j = 1..n$$

Concerning the distribution node, we can underlie, without loss of generality, that there is a general function with memory describing the dependence of desired output on the n outgoing branches: $y = h(u_1, u_2, ..., u_n)$. Then, $\tilde{u}_1 = \tilde{u}_2 = ... = \tilde{u}_n = \tilde{u}$ holds, where $\tilde{u}$ is the value of the incoming branch. This let us state:

$$\tilde{u}'(\tau) = \frac{\partial y_k(t)}{\partial \tilde{u}(t-\tau)} = \sum_{j=1}^{n} \frac{\partial y_k(t)}{\partial \tilde{u}_j(t-\tau)} \frac{\partial \tilde{u}_j(t-\tau)}{\partial \tilde{u}(t-\tau)} =$$

$$= \sum_{j=1}^{n} \frac{\partial y_k(t)}{\partial \tilde{u}_j(t-\tau)} = \sum_{j=1}^{n} \tilde{u}_j'(\tau)$$

$$\tilde{u}"(\tau) = \frac{\partial^2 y_k(t)}{\partial \tilde{u}(t-\tau)^2} = \sum_{j=1}^{n} \tilde{u}_j"(\tau)$$

In conclusion, it can be affirmed that the set of transformations in Table 1 are valid for generic topology SFGs.

3. MIXED SECOND DERIVATIVES

We are not interested for the moment in deriving an extended adjoint graph to consider also the case of mixed second derivatives, but just in showing how these quantities can be calculated directly from those ones obtained by means of the method described in the previous section. That is why we shall omit the time variable from the following expressions, assuming that the output $y_k(t)$ at time t depends on node variables $x_i(t-\tau_1)$ $x_j(t-\tau_2)$ at different times $t-\tau_1$, $t-\tau_1$. The term we are interested on will be thus restricted to its topological meaning:

$$\frac{\partial^2 y_k(t)}{\partial x_i(t-\tau_1)\partial x_j(t-\tau_2)} = \frac{\partial^2 y_k}{\partial x_i \partial x_j} \qquad (1)$$

We shall distinguish three cases, in accordance with the relationship between node variables x_i, x_j, as it follows.

3.1 Case 1: only dependent variables $\mathbf{x_i, x_j}$

The dependence between x_i, x_j is given by $x_j = f(x_i)$ and the output y depends on x_i only through x_j; we shall assume that f is invertible, consequently yielding $x_i = F(x_j) = f^{-1}(x_j)$. In compliance with formulas of previous sections, the following notation holds: $x_i'' = \frac{\partial^2 y}{\partial x_i^2}$, $x_i'' = \frac{\partial y}{\partial x_i}$, $x_{ij}'' = \frac{\partial^2 y}{\partial x_i \partial x_j}$. From that, equation (1) and inverse function theorem, it derives that:

$$\begin{aligned} \frac{\partial^2 y}{\partial x_i \partial x_j} &= \frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial x_i}\frac{\partial x_i}{\partial x_j}\right) = \frac{\partial^2 y}{\partial x_i^2}\frac{\partial x_i}{\partial x_j} + \frac{\partial y}{\partial x_i}\frac{\partial^2 x_i}{\partial x_i \partial x_j} = \\ &= x_i'' \cdot \left(1\!\!\diagup\!\! f'\right) + x_i' \cdot \left(-f''\!\!\diagup\!\! f'\right) = x_{ij}'' \end{aligned} \qquad (2)$$

$$\begin{aligned} \frac{\partial^2 y}{\partial x_j \partial x_i} &= \frac{\partial}{\partial x_j}\left(\frac{\partial y}{\partial x_j}\frac{\partial x_j}{\partial x_i}\right) = \frac{\partial^2 y}{\partial x_j^2}\frac{\partial x_j}{\partial x_i} + \frac{\partial y}{\partial x_j}\frac{\partial^2 x_j}{\partial x_j \partial x_i} = \\ &= x_j'' \cdot f' + x_j' \cdot \left(f''\!\!\diagup\!\! f'\right) = x_{ji}'' \end{aligned} \qquad (3)$$

It can be underlined that:

- $x_{ij}'' \neq x_{ji}''$: in fact x_i, x_j are dependent and Schwarz's theorem can not be applied.
- x_{ij}'' and x_{ji}'' can be calculated by means of first and full second derivatives relative to the original graph (x_j'', x_j) or to sub-graphs of it (f', f'').

3.2 Case 2: only independent variables $\mathbf{x_i, x_j}$

Conditions of Schwarz's theorem are assumed to be valid: this is supported by the independence between x_i, x_j. Therefore x_{ij}'' and x_{ji}'' are expected to be identical. Anyway, we need further topological information for our purposes: that is the knowledge of a "common" node between x_i, x_j, namely x_k. This node is dependent on both x_i and x_j, through two different functions, i.e. $x_k = f_i(x_i)$, $x_k = f_j(x_j)$. In particular, we can state that f_j' does not depend on x_i, and f_j' on x_j as well, if we choose x_k to be the first common node, i.e. the first addition node combining x_i, x_j. This means that $\frac{\partial^2 x_k}{\partial x_i \partial x_j} = 0$ and thus:

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial x_k}\frac{\partial x_k}{\partial x_j}\right) = \frac{\partial^2 y}{\partial x_i \partial x_k}\frac{\partial x_k}{\partial x_j} + \frac{\partial y}{\partial x_k}\frac{\partial^2 x_k}{\partial x_i \partial x_j} = x_{ik}'' \cdot f_j'$$

$$\frac{\partial^2 y}{\partial x_j \partial x_i} = x_{jk}'' \cdot f_i'$$

We can observe that calculation of mixed derivatives in such a case involves quantities attainable from Case 1 and first derivatives of suitable sub-graphs.

3.3 Case 3: both dependent and independent variables $\mathbf{x_i, x_j}$

In contrast to case 1, output y depends on x_i not only through x_j but also through different nodes (Figure 1), i.e. we can write $y = g(x_j, x_i) = g(f(x_i), x_i)$. We can think to split x_i into two new variables, $\tilde{x}_i$ and $\overline{x}_i$, the latter being linked to x_j through

$\tilde{x}_i = F(x_j) = f^{-1}(x_j)$, and the former being completely independent from x_j. This let us deduce the formulation of mixed derivatives we are interested to:

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = x''_{ij} = \frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial x_j}\right) = \left[\frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial \tilde{x}_i}\right)\right]\frac{\partial \tilde{x}_i}{\partial x_j} + \frac{\partial y}{\partial \tilde{x}_i}\left[\frac{\partial}{\partial x_i}\left(\frac{\partial \tilde{x}_i}{\partial x_j}\right)\right]$$

Observing that $\frac{\partial y}{\partial \tilde{x}_i} = \frac{\partial y}{\partial x_j} \cdot f' = \frac{\partial y}{\partial x_i} - \frac{\partial y}{\partial \overline{x}_i}$, we shall have:

$$x''_{ij} = x''_i \cdot \left(1/f'\right) - x'_j \cdot \left(f''/f'\right) - \left(1/f'\right)\left[\frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial \overline{x}_i}\right)\right] \qquad (4)$$

that immediately reduces to (2) as soon as $\overline{x}_i = 0$. Let us focus on the third term in (4), that is given by:

$$\frac{\partial}{\partial x_i}\left(\frac{\partial y}{\partial \overline{x}_i}\right) = \frac{\partial^2 y}{\partial \tilde{x}_i \partial \overline{x}_i} + \overline{x}''_i$$

where $\overline{x}''_i$ is the full second derivatives of the graph deduced from the original one by splitting x_i into $\tilde{x}_i$ and $\overline{x}_i$. We can now afford to implement what done in case 2, and write:

$$\frac{\partial^2 y}{\partial \tilde{x}_i \partial \overline{x}_i} = \frac{\partial^2 y}{\partial \tilde{x}_i \partial x_k} \cdot \frac{\partial x_k}{\partial \overline{x}_i} = \left(1/\tilde{g}'\right)\left[\tilde{x}''_i - \tilde{x}'_i \cdot \left(\tilde{g}''/\tilde{g}'\right)\right] \cdot \frac{\partial x_k}{\partial \overline{x}_i}$$

where x_k is the first common node between $\tilde{x}_i$ and $\overline{x}_i$, and $x_k = \tilde{g}(\tilde{x}_i)$, $x_k = \overline{g}(\overline{x}_i)$. Summarising, we get the final formula:

$$\begin{cases} x''_{ij} = x''_i \cdot \left(1/f'\right) + x'_j \cdot \left(-f''/f'\right) - \left(1/f'\right)\left[\dfrac{\partial^2 y}{\partial \tilde{x}_i \partial \overline{x}_i} + \overline{x}''_i\right] \\ \dfrac{\partial^2 y}{\partial \tilde{x}_i \partial \overline{x}_i} = \dfrac{\partial^2 y}{\partial \tilde{x}_i \partial x_k} \cdot \dfrac{\partial x_k}{\partial \overline{x}_i} = \dfrac{\partial^2 y}{\partial \tilde{x}_i \partial x_k} \cdot \overline{g}' \\ \dfrac{\partial^2 y}{\partial \tilde{x}_i \partial x_k} = \left(1/\tilde{g}'\right)\left[\tilde{x}''_i - \tilde{x}'_i \cdot \left(\tilde{g}''/\tilde{g}'\right)\right] \end{cases} \qquad (5)$$

where $x_j = f(\tilde{x}_i)$, $x_k = \tilde{g}(\tilde{x}_i)$, $x_k = \overline{g}(\tilde{x}_i)$ hold. It can be easily noticed that only first and full second derivatives relative to the original graph, the graph derived by splitting the dependent variable into his two components, and particular sub-graphs of them occur in (5). Similarly, we can derive the following for the reverse mixed derivative:

$$\begin{cases} x''_{ji} = \left(1/f'\right)\left[x''_j \cdot (f')^2 + x'_j \cdot f''\right] + \dfrac{\partial^2 y}{\partial x_j \partial \overline{x}_i} \\ \dfrac{\partial^2 y}{\partial x_j \partial \overline{x}_i} = \dfrac{\partial^2 y}{\partial x_j \partial x_k} \cdot \dfrac{\partial x_k}{\partial \overline{x}_i} = \dfrac{\partial^2 y}{\partial x_j \partial x_k} \cdot \overline{g}' \\ \dfrac{\partial^2 y}{\partial x_j \partial x_k} = \left(1/\tilde{g}'\right)\left[\tilde{x}''_i - \tilde{x}'_i \cdot \left(\tilde{g}''/\tilde{g}'\right)\right] \end{cases} \qquad (6)$$

where $x_j = f(\tilde{x}_i)$, $x_k = h(x_j)$, $x_k = \overline{g}(\tilde{x}_i)$ hold. Same considerations as above can be made. In particular, (6) reduces to (3) when $\overline{x}_i = 0$. Then, we can rewrite (5) and (6) as in the following compact way:

$$\begin{cases} x''_{ij} = \tilde{x}''_i \cdot \left[\left(1/f'\right) \cdot \left(1 + \left(\overline{g}'/\tilde{g}'\right)\right)\right] - x'_i \cdot \left[\left(f''/(f')^2\right) + \left(\tilde{g}''/(\tilde{g}')^2\right) \cdot \overline{g}'\right] \\ x''_{ji} = x''_j \cdot \left[f' + \left(\overline{g}'/h'\right)\right] + x'_j \cdot \left[\left(f''/f'\right) - \left(h''/h'\right)\overline{g}'\right] \end{cases}$$

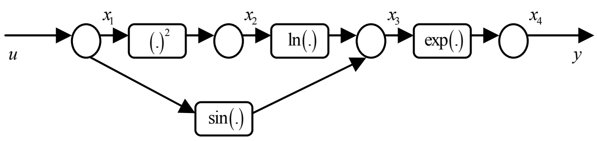


Figure 2. A simple SFG example in case 3: x_1, x_2 are the both dependent and independent node variables addressed.

4. CONCLUSIONS

SFG techniques have been successfully employed here to calculate first and second derivatives in dynamical non-linear systems. Firstly, a suitable adjoint graph has been derived without using Lee's theorem to yield first and full second derivatives of an output of the original graph respect with one of its nodes automatically. This let previous SFG based approach for sensitivity calculation [2]-[5] be considered as particular case thereof. Secondly an helpful formulation for mixed second derivatives has been deduced in order to make them dependent on quantities attainable by means of suitable adjoint graphs for first and full second derivatives, even though a proper graph description of such dependencies actually lacks. Derivation of that is the first issue for future works. Then, application of proposed approach to adapting problems where calculation of Jacobean and Hessian based information is involved, with special care to dynamical neural systems, is also being studied. Finally, the authors are investigating how to extend the present approach to the case of multirate systems, as recently done [6] for calculation of first derivatives.

REFERENCES

[1] A.Y. Lee, "Signal Flow Graphs—Computer Aided System Analysis and Sensitivity Calculations" *IEEE Transactions on Circuits and Systems*, vol. 21, no. 2, pp. 209-216, 1974.

[2] E.A. Wan, and F. Beaufays, "Diagrammatic Derivation of Gradient Algorithms for Neural Networks", *Neural Computation*, vol. 8, pp. 182-201, 1996.

[3] S. Osowski, "Signal flow graphs and neural networks", *Biological Cybernetics*, 70, pp. 387-395, 1994.

[4] G. Martinelli, and R. Perfetti, "Circuit theoretic approach to the backpropagation learning algorithm", *Proc. of the IEEE Int. Symposium on Circuits and Systems*, 1991.

[5] P. Campolucci, A. Uncini, and F. Piazza, "A signal Flow-Graph Approach to On-Line Gradient Calculation", *Neural Computation,* vol. 12, no. 8, August 2000.

[6] F. Rosati, P. Campolucci, and F. Piazza, " Learning in Linear and Non-Linear Multirate Digital Systems by Signal Flow Graphs", *IEEE ISCAS*, vol.2, pp. 6-9, May 2001.