Intermediate desired value approach for continuous transition among multiple tasks of robots
Jaemin Lee, Nicolas Mansard, Jaeheung Park

To cite this version:

HAL Id: hal-00667212
https://hal.archives-ouvertes.fr/hal-00667212
Submitted on 7 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Intermediate Desired Value Approach for Continuous Transition among Multiple Tasks of Robots

Jaemin Lee*, Nicolas Mansard**, and Jaeheung Park*†

*Department of Intelligent Convergence Systems, Seoul National University, Republic of Korea.
{ljm1918, park73}@snu.ac.kr
**Laboratoire d’Architecture et d’ Analyse des Systems, Universite de Toulouse, France.
nicolas.mansard@laas.fr

Abstract—As the capability of robots is getting improved, more various tasks are expected to be performed by the robots. Complex operation of the robots can be composed of many different tasks. These tasks are executed sequentially, simultaneously, or in a combined way of both. This paper discusses the transition issue among multiple tasks on how the transition can be effectively and smoothly achieved. The proposed approach is to compose intermediate desired values to smooth the transitions rather than to modify control laws. The new approach can be directly applied to hierarchically constructed tasks, where the continuous inverse solution is limited. Also, the new approach is easily applicable to many existing platforms since it does not require to change control or inverse process.

In addition, it is briefly discussed the case when there is not enough DOF for two tasks with different priorities. In such case the lower-priority task cannot be fully performed. Its control result is discussed to provide an intuition of the effectiveness of the proposed approach for prioritized multiple tasks.

The experimental validations of the proposed approach were conducted in two scenarios: multi-point control using a 3 DOF planar robot in simulation and joint limit avoidance using a 7 DOF physical robot. The smooth and effective task transitions are demonstrated in both cases.

I. INTRODUCTION

The robots are mostly programmed to execute a specific pre-defined set of tasks in a well defined environment. However, as more complex robots are designed to operate in dynamic environments, various tasks or set of tasks should be executed sequentially or simultaneously [1], [2]. Dealing with these situations, it is important to design strategies for transition among various tasks [3], [4], [5].

Considering general task transitions, this paper is focused on the situation when a task is added or deleted. Once established, this approach can be used for many task transition situations. Such cases can occur also in traditional applications such as dealing with joint limit where joint limit avoidance can be defined as a task in specific regions [6]. Feature tracking in visual servoing is another application where some of the features are lost or added during operation [6], [7], [8]. Also, Singularity avoidance in manipulation [9] and switching behaviors of humanoid robots would be also one of the cases [11], [12], [13].

Without proper consideration of transition, there will be discrete and abrupt changes at the input of the robot, which is typically either desired joint velocity or joint torque. The goal of the paper is to provide an intelligent transition method that enables the robot to perform smooth task transition.

In this paper, the concept and validation are presented in a kinematic control approach, where joint velocities are the inputs to the robot given desired task space velocities. In the following section, the source of discontinuity is explained. One approach dealing with this discontinuity is then proposed by introducing intermediate desired values as a new task specification.

The proposed approach is compared with the continuous inverse solution in [14], which modifies the inversion process. The continuous inverse approach is to compose a control law by modifying the inverse operation with an activation matrix. It is revealed that the two approaches are equivalent in the case of equal priority tasks. However, the new approach can be directly applied to hierarchically constructed tasks, where the continuous inverse solution is limited. Also, the new approach is easily applicable to many existing platforms since it does not require to change control or inverse process.

In addition, it is briefly discussed the case when there is not enough DOF for two tasks with different priorities. In such case the lower-priority task cannot be fully performed. Its control result is discussed to provide an intuition of the priority-based control and the effectiveness of the proposed approach for prioritized multiple tasks.

When there are two tasks or features, $x_1$ and $x_2$, the corresponding Jacobians are defined as $J_1$ and $J_2$.

$$
\dot{x}_1 = J_1 \dot{q}, \quad \dot{x}_2 = J_2 \dot{q}
$$

When only one task, $x_1$, is to be controlled, the associated joint velocity, $\dot{q}_1$, can be computed as

$$
\dot{q}_1 = J_1^+ \dot{x}_1
$$

where $(\cdot)^+$ denotes a pseudo-inverse of the quantity. The inverse could be replaced by any of the generalized inverses.

Now, if another task, $x_2$, is to be controlled together with $x_1$, the joint velocity can be computed as

$$
\dot{q} = J^+ \dot{x}
$$

where

$$
J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix}, \quad \dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}.
$$

†Jaeheung Park is the corresponding author. park73@snu.ac.kr
The joint velocity for both tasks (Equation (3)) might be 
discontinuous from the joint velocity for only the first task
(Equation (2)) at the instance when the second feature is
inserted.

One explanation for this discontinuous behavior is that
it comes from the property of pseudo-inverse. The pseudo-
inverse computes a solution to meet the feature specification
and minimize the 2-norm of \( \dot{q} \). Obviously, when we achieve
one feature or two features the solution would not be
continuous.

More physically meaningful explanation is as follows. When we control the first task or feature, the computed joint
velocity, \( q_1 \), indirectly affect the task, \( x_2 \), although this task
is not interested during the execution. Then, the problem is
that we later attempt to control or specify this feature with
different values all of sudden. Therefore the joint velocity
will show a discrete change.

Fundamentally, the problem is that the solution, \( \dot{q}_1 \), from
Equation (2) generates a value for the second task as \( J_2 \dot{q}_1 \)
and minimize the 2-norm of \( \dot{q} \). Obviously, when we achieve
the second feature the quantity. Then the joint velocity is
\( \dot{x}_2 \neq J_2 \dot{q}_1 \)

Therefore, when we add this task, \( x_2 \), the specification
for \( x_2 \) in Equation (4) should be continuous from the value,
\( J_2 \dot{q}_1 \). Otherwise, the pseudo-inverse will give a solution with
a discrete change. The discrete change would occur during
task removal due to the same reason.

III. INTERMEDIATE DESIRED VALUE APPROACH

A. Concept of Continuous Transition Strategy

From the observation in the previous section, a proposed
approach is to design a task specification of \( \dot{x}_2 \) smoothly from
\( J_2 \dot{q}_1 \), where \( \dot{q}_1 \) is from Equation (2), such that the solution
would not show any abrupt change.

\[
\dot{x}_i = \begin{pmatrix} \dot{x}_i^1 \\ \dot{x}_i^2 \end{pmatrix}
\]

(6)

and

\[
\dot{x}_i^1 = \dot{x}_1 \\
\dot{x}_i^2 = h_2 \dot{x}_2 + (1 - h_2) J_2 J_1^+ \dot{x}_1
\]

(7)

where \( h_2 \) is an activation variable from 0 to 1 when the task,
\( x_2 \), is inserted. The superscript, \( i \), denotes the intermediate
value of the quantity. Then the joint velocity is

\[
\dot{q} = J^+ \dot{x}_i
\]

(8)

This solution is continuous if we change \( h_2 \) from 0 to 1
continuously.

B. Generalization

An activation parameter for the first task, \( x_1 \), can also
be defined as \( h_1 \). Then,

\[
\dot{q} = J^+ \dot{x}_i
\]

(9)

\[
\dot{x}_i^1 = h_1 \dot{x}_1 + (1 - h_1) J_1 J_2^+ \dot{x}_2 \\
\dot{x}_i^2 = h_2 \dot{x}_2 + (1 - h_2) J_2 J_1^+ \dot{x}_1
\]

(10)

This solution is in fact the same as the one for one task with
activation matrix, \( H \), in [16]. Refer to Appendix for the
proof of equivalence.

C. More than Two tasks

When there are tasks with more than two activation
parameters, Equation (10) can be extended. For example, the
solution for the tasks with three activation parameters is the
following:

\[
\dot{q} = J^+ \dot{x}_i
\]

(11)

And

\[
\begin{align*}
\dot{x}_1^1 &= h_1 \dot{x}_1 + (1 - h_1) J_1 \dot{q}_{(h_2, h_3)} \\
\dot{x}_2^1 &= h_2 \dot{x}_2 + (1 - h_2) J_2 \dot{q}_{(h_1, h_3)} \\
\dot{x}_3^1 &= h_3 \dot{x}_3 + (1 - h_3) J_3 \dot{q}_{(h_1, h_2)}
\end{align*}
\]

(12)

where \( \dot{q}_{(h_m, h_n)} \) denotes the solution for the tasks \( m \) and \( n \)
with activation parameters of \( h_m \) and \( h_n \). The derivation of
Equations (11) and (12) is straight-forward from Equations
(9) and (10).

IV. INTERMEDIATE DESIRED VALUE APPROACH FOR
PRIORITIZED TASKS

A new task may be added with a different priority from
the existing tasks. The solution of the previous section can
be re-arranged in this case. For example, when we had a
task set as \( x_2 \) and the associated Jacobian, \( J_2 \), another higher
prioritized task set, \( x_1 \), can be inserted, such as joint limits
or collision avoidance. The controller for \( x_2 \) only was

\[
\dot{q}_2 = J_2^+ \dot{x}_2.
\]

(13)

Now with the higher prioritized task, \( x_1 \),

\[
\dot{q} = J_1^+ \dot{x}_1 + N_1 \dot{q}_0
\]

(14)

where

\[
N_1 = I - J_1 J_1^+
\]

(15)

\[
\dot{q}_0 = (J_2 N_1) J_1^+ (\dot{x}_2 - J_2 J_1^+ \dot{x}_1).
\]

(16)

However, transition from Equation (13) to Equation (14) will
have discontinuity. Therefore, a similar approach to the one
introduced in the previous section can be applied.

\[
\dot{q} = J_1^+ \dot{x}_1 + N_1 \dot{q}_0
\]

(17)

where

\[
\dot{x}_1^1 = h_1 \dot{x}_1 + (1 - h_1) J_1 \dot{q}_2 \\
\dot{x}_2^1 = J_2^+ \dot{x}_2 \\
\dot{q}_0^1 = (J_2 N_1) J_1^+ (\dot{x}_2 - J_2 J_1^+ \dot{x}_1)
\]

(18)

The above approach provides a continuous solution for
\( 0 \leq h_1 \leq 1 \) because \( \dot{q}_2 \) from Equation (13) is the same as \( \dot{q} \)
from Equation (17) when \( h_1 = 0 \) (See Appendix II).

However, the above solution does not have an activation
parameter for \( x_2 \). This could be implemented by introducing
\( h_2 \).

\[
\dot{q} = J_1^+ \dot{x}_1 + N_1 \dot{q}_0
\]

(19)
where
\[\begin{align*}
\dot{x}_1^i &= h_1\dot{x}_1 + (1 - h_1)J_1\dot{q}_2^i \\
\dot{q}_2^i &= J_2^+ h_2\dot{x}_2 \\
\dot{q}_0 &= (J_2N_1)^+(h_2\dot{x}_2 + (1 - h_2)J_2J_1^+ h_1\dot{x}_1 - J_2J_1^+ \dot{x}_1^i).
\end{align*}\] (20)

The equations (19) and (20) can be written as
\[\dot{q} = J_1^+ \dot{x}_1 + N_1(J_2N_1)^+ (\dot{x}_2^i - J_2J_1^+ \dot{x}_1^i)\] (21)
where
\[\begin{align*}
\dot{x}_1^i &= h_1\dot{x}_1 + (1 - h_1)J_1J_2^+ h_2\dot{x}_2 \\
\dot{x}_2^i &= h_2\dot{x}_2 + (1 - h_2)J_2J_1^+ h_1\dot{x}_1
\end{align*}\] (22)

A. What is the difference from the two approaches with or without priorities?

As we can see from Equation (21) and (22), the only difference in formulations for the approaches with or without priorities is the computational structure. The intermediate goal values, \(\dot{x}'\), are computed in the same procedure.

If the two tasks are independent, the two approaches with or without priorities should give the same result. The difference comes only if they are conflicting.

B. More than Two Tasks

The same extension approach as the previous solution applies. For the three task sets,
\[\dot{q} = \dot{q}_1 + \dot{q}_2 + \dot{q}_3\] (23)
\[\begin{align*}
\dot{q}_1 &= J_1^+ \dot{x}_1^1 \\
\dot{q}_2 &= N_1(J_2N_1)^+ (\dot{x}_2^1 - J_2\dot{q}_1^1) \\
\dot{q}_3 &= N_1N_2(1)(J_3N_1N_2(1))^+ (\dot{x}_3 - J_3(\dot{q}_1 + \dot{q}_2))
\end{align*}\] (24)
and
\[N_2(1) = I - (J_2N_1)^+J_2N_1\] (25)
\[\begin{align*}
\dot{x}_1^1 &= h_1\dot{x}_1 + (1 - h_1)J_1\dot{q}_{(h_2,h_3,\text{prio})} \\
\dot{x}_2^1 &= h_2\dot{x}_2 + (1 - h_2)J_2\dot{q}_{(h_1,h_3,\text{prio})} \\
\dot{x}_3^1 &= h_3\dot{x}_3 + (1 - h_3)J_3\dot{q}_{(h_1,h_2,\text{prio})},
\end{align*}\]
where \(\dot{q}_{(h_m,h_n,\text{prio})}\) denotes the solution for the tasks \(m\) and \(n\) with activation parameters of \(h_m\), \(h_n\), and given priorities.

V. Simulation and Experimental Results

The proposed approach for task transition has been verified in ROBOTICS LAB [17] simulation environment and physical robot. ROBOTICS LAB provides not only physics-based simulation environment but also real time control module and programmable interface for the user.

In the following sub-sections, the multiple point control with different priorities is demonstrated in simulation with a 3-DOF planar robot. Then, the end-effector control with joint limit avoidance is executed on a 7-DOF manipulator. During the experiments, the kinematic control law is implemented as
\[\ddot{x} = -\lambda (x - x_d)\] (27)
where \(x\) and \(x_d\) are the current and desired task, and \(\lambda\) is a positive gain for decreasing task error.

A. Task transition in a 3 DOF planar robot: Simulation

In this section, the proposed task transition approach is demonstrated in a 3 DOF planar robot when a new task inserted with a lower priority than that of an existing task. The purpose of the section is to demonstrate the important concepts in a relatively simpler case. Also, the effect of the priorities is explained and shown when the tasks are partially conflicting due to the lack of degrees of freedom of the robot.

The two positioning tasks are defined as positioning of the end-effector and the 2nd link (Figure 1).

\[\dot{x}_1 = \begin{pmatrix} \dot{x}_{e,x} \\ \dot{x}_{e,y} \end{pmatrix}, \quad \dot{x}_2 = \begin{pmatrix} x_{\text{link2},x} \\ x_{\text{link2},y} \end{pmatrix}\] (28)
where \((.,.)\) and \((.,.)\) denote the tasks related to the end-effector and the end point of the 2nd link, respectively. The robot begins to control only the end-effector position, \(x_1\). Later the 2nd link position control is inserted after 2 seconds (\(t = 4\) sec) (Figure 2 and 3).

In Figure 2, the control results are plotted when the second task (positioning task of the 2nd link) is abruptly inserted as a low priority task. The results of the proposed continuous approach are plotted in Figure 3. The activation parameter \(h_2\) increased from 0.0 to 1.0 for 0.5 seconds. The plot of
Although the inserted task cannot be fully performed due to the lack of degrees of freedom for both tasks. This phenomenon is explained from the global and instantaneous point of view in Figure 4 and 5, respectively. At the end of the control in Figure 4, the position of the 2nd link comes to the closest point to the desired position within the workspace, which is a circle centered at the end-effector position.

In the instantaneous view, the second task execution can be decomposed into the effect of higher-priority task and its control in the task null space (Figure 5).

\[
\frac{\dot{x}_2}{\dot{q}} = J_2^T \frac{\dot{x}_1}{\dot{q}} = \dot{x}_{2,prio} + \dot{x}_{2,\text{null}}
\]

(29)

where

\[
\dot{x}_{prio} = J_2^T \frac{\dot{x}_1}{\dot{q}}
\]

\[
\dot{x}_{null} = J_2^T \frac{\dot{q}}{\dot{q}}
\]

(30)

The term, \(\dot{x}_{2,prio}\), is the effect from the control of the previous task with a higher priority. Then, the control in the null space, \(\dot{x}_{2,\text{null}}\), set \(\dot{x}_2\) to be closest to its desired value.

Although the inserted task cannot be fully performed because of not enough degrees of freedom for both tasks, it is demonstrated that the task is executed optimally under priority-based control. Using recursive control structure with the proposed approach, many tasks can be controlled partially or completely with various levels of priorities.

**B. 7-DOF manipulator with elbow joint limit: Experiment**

Joint limit task for a 7 DOF manipulator is controlled through the proposed approach in this section. It must be considered as a higher prioritized task than other given tasks. When the joint limit task is inserted, continuity of task transition can be ensured by the proposed approach.

1) **Definition of Prioritized Tasks:** In this section, two task sets are considered with priorities. One is a joint limit task and the other is consisted of position and orientation of end-effector. The priority of joint limit task is higher than that of task related to the end-effector.

\[
\frac{\dot{x}_1}{\dot{q}} = \left[ \begin{array}{c} \dot{q}_{\text{limit}} \\ \dot{v}_e \\ \omega_e \end{array} \right], \quad \frac{\dot{x}_2}{\dot{q}} = \left[ \begin{array}{c} v_e \\ \omega_e \end{array} \right]
\]

(31)

where \(v_e\) and \(\omega_e\) are the linear and angular velocities of the end-effector.

When the joint position is not in joint limit region, the end-effector positioning task is controlled by

\[
\frac{\dot{q}}{\dot{q}} = J_2^T \frac{\dot{x}_2}{\dot{q}}
\]

(32)
Now when the joint limit task with high priority is inserted without intermediate desired value method, the kinematic control with the priorities provides the solution of

$$
\dot{\mathbf{q}} = J_1^+ \dot{x}_1 + N_1(J_2N_1)^+ (\dot{x}_2 - J_2J_1^+ \dot{x}_1)
$$

(33)

The result of this kinematic control through an abrupt change produces oscillatory behavior. The oscillation comes from rapid task transition at the activation border (Figure 8). It means that pushing toward and backward at the joint limit are iterated in sequence.

In the experiments, the end-effector of the robot is controlled to follow a line to reach a desired goal position. While the end-effector task is being executed, the Elbow joint reaches a joint limit buffer at right before 5 seconds (Figure 7, 8, and 9). Figure 8 shows instability due to the iterative abrupt changes in control due to the conflicting joint limit and end-effector control. The results of the proposed approach are plotted in Figure 9, which is explained in detail in the following sub-sections.

2) Activation Buffer: For the joint limit task execution, the activation parameter is defined as a function of the corresponding joint angle. In the activation buffer, the activation parameter increases from 0.0 at the beginning of the buffer to 1.0 at the limit. For inserting high-priority task, the activation parameter, $h_1$, is defined as

$$
h_1 = \begin{cases} 
1.0 & \text{if } q_{\text{elbow}} \geq q_{\text{upperLimit}} \\
0.0 & \text{if } q_{\text{lowerLimit}} + \beta \leq q_{\text{elbow}} \leq q_{\text{upperLimit}} - \beta \\
f(q_{\text{elbow}}) & \text{if } q_{\text{upperLimit}} - \beta < q_{\text{elbow}} < q_{\text{upperLimit}} \\
g(q_{\text{elbow}}) & \text{if } q_{\text{lowerLimit}} < q_{\text{elbow}} < q_{\text{lowerLimit}} + \beta \\
1.0 & \text{if } q_{\text{elbow}} \leq q_{\text{lowerLimit}}
\end{cases}
$$

(34)

where $\beta$ is the buffer length ($\beta = 0.2(rad)$), $q_{\text{upperLimit}} = 1.6(rad)$, and $q_{\text{lowerLimit}} = -1.6(rad)$ in this example. The activation parameter is plotted in Figure 6. Equation (35) are activation functions varying in activation buffers.

$$
f(q_{\text{elbow}}) = 0.5 + 0.5\sin\left(\frac{\pi}{\beta}(q_{\text{elbow}} - (q_{\text{upperLimit}} - \beta)) - \frac{\pi}{2}\right)
$$

$$
g(q_{\text{elbow}}) = 0.5 + 0.5\sin\left(\frac{\pi}{\beta}(q_{\text{elbow}} - q_{\text{lowerLimit}}) + \frac{\pi}{2}\right)
$$

(35)

3) Intermediate Desired Value: Discontinuity and instability of task execution occur when tasks are transitioned abruptly. For continuous task transition, the intermediate desired value for the joint limit task for the elbow is set to be

$$
\dot{x}_e = h_1 \dot{x}_e + (1 - h_1)J_1J_2^+ \dot{x}_2
$$

(36)

where $h_1$ is the activation function defined in Equation (34). When the elbow joint is in the activation buffer zone, the joint
VI. Conclusion

The intermediate desired value approach for task transition is presented in this paper. The proposed approach is to provide continuity during transition using existing control frameworks with minimal performance compromise. The approach without priorities are shown to be equivalent to continuous inverse method [14]. Then, it is further extended to the cases of tasks with priorities.

When task transition is executed by the proposed approach, the transitioned tasks can be executed smoothly with various levels of priorities including high priority and low priority. When there is not enough DOF of the manipulator,
the task with a lower priority may not be fully performed but partially. The simulation result of such a case is shown to demonstrate that the proposed approach can be used successfully with the priority-based control framework.

We are currently working on extending the proposed approach for the dynamic control framework. And the issue of computation cost of the proposed approach will be investigated in the future.

VII. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (No.2010-0005799) and Seoul R&B Program (No.JP100106).

REFERENCES


APPENDIX I

EQUIVALENT WITH CONTINUOUS INVERSE METHOD

The proposed approach is to modify the desired values for $x_2$ to smooth the solution out. This solution can be rearranged as follows:

$$\dot{q} = J^+ \left( h_2 \dot{x}_2 + (1-h_2) J_2 J_2^+ \dot{x}_1 \right)$$  \hspace{1cm} (I.1)

$$\dot{q} = J^+ \left( \begin{array}{c} I \\ \frac{1}{h_2} J_2 J_2^+ \end{array} \right) \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}$$  \hspace{1cm} (I.2)

where

$$J^+ = \begin{pmatrix} I \\ (1-h_2) J_2 J_2^+ \end{pmatrix}$$  \hspace{1cm} (I.3)

During the above derivation the following property has been used:

$$J^+ J J_j^+ = J^+ \begin{pmatrix} I \\ J_2 J_2^+ \end{pmatrix}$$  \hspace{1cm} (I.4)

$$J_j^+ = J^+ \begin{pmatrix} I \\ J_2 J_2^+ \end{pmatrix}$$  \hspace{1cm} (I.5)

The operator from $\dot{x}$ to $\dot{q}$ in Equation (1.2) is equivalent to the continuous inverse $J^+H$ in [14] when $h_1 = 1$. In a more general case when both $h_1$ and $h_2$ are not equal to zero or one, Equation (1.2) becomes

$$\dot{q} = J^+ \begin{pmatrix} h_1 I \\ (1-h_2) h_1 J_2 J_2^+ \end{pmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}$$  \hspace{1cm} (I.6)

The equivalence with the continuous pseudo inverse can be easily derived.

$$J^+H = J^+ \begin{pmatrix} h_1 I \\ J_2 J_2^+ \end{pmatrix} \begin{pmatrix} 0 \\ I \end{pmatrix} + h_2 \begin{pmatrix} 0 \\ J_2 J_2^+ \end{pmatrix} - h_1 h_2 \begin{pmatrix} 0 \\ J_2 J_2^+ \end{pmatrix}$$  \hspace{1cm} (I.7)

APPENDIX II

CONTINUITY IN PRIORITY BASED APPROACH

Equation (17), when $h_1 = 0$, is

$$\dot{q} = J_2^+ J_2 J_2^+ \dot{x}_2 + N_1 (J_2 N_1)^+ (\dot{x}_2 - J_2 J_2^+ J_2 J_2^+ \dot{x}_2) = \{J_2^+ J_2 J_2^+ \} J_2^+ \dot{x}_2$$  \hspace{1cm} (II.1)

$$\dot{q} = J_2^+ J_2 J_2^+ \dot{x}_2 + N_1 (J_2 N_1)^+ (J_2 J_2^+ J_2 J_2^+ \dot{x}_2) = \{J_2^+ J_2 J_2^+ \} J_2^+ \dot{x}_2$$  \hspace{1cm} (II.1)

When $h_1 \neq 0$,

$$\dot{q} = J_2^+ \dot{x}_2 + h_1 \{J_2^+ - N_1 (J_2 N_1)^+ J_2 J_2^+ \} (\dot{x}_1 - J_2 J_2^+ \dot{x}_2)$$  \hspace{1cm} (II.2)