RECURSIVE INVERSE DYNAMICS

RECURSIVE INVERSE DYNAMICS

We assume at the outset that the manipulator under study is of the serial type with n+1 links including the base link and n joints of either the revolute or the prismatic type.

The underlying algorithm consists of two steps:

- Kinematic Computations: required to determine the twists of all the links and their time derivatives in terms of $\theta, \dot{\theta}, \ddot{\theta}$
- Dynamic Computations: required to determine both the constraint and the external wrenches.

Henceforth, revolute joints are referred to as R, prismatic joints as *P*.

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

We will use the Denavit-Hartenberg (DH) notation. Moreover, every 3-D vector-component transfer from the $\mathcal{F}_i$ frame to the frame $\mathcal{F}_{i+1}$ requires a multiplication by Q_i^T. Likewise, every component transfer from the frame $\mathcal{F}_{i+1}$ to the $\mathcal{F}_i$ frame requires a multiplication by Q_i.

If we have: $[r_i] = [r_1, r_2, r_3]$ and we need: $[r_i]_{i+1}$

then we proceed as follows:

$$[r]_{i+1} = Q_i^T [r]_i$$

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

If we recall the form of Q_i we then have:

$$[r]_{i+1} = \begin{bmatrix} cos\theta_i & sen\theta_i & 0 \\ -\lambda_i sin\theta_i & \lambda_i cos\theta_i & \mu_i \\ \mu_i sin\theta_i & -\mu_i cos\theta_i & \lambda_i \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_1 cos\theta_i + r_2 cos\theta_i \\ -\lambda_i r + \mu_i r_3 \\ \mu_i r + \lambda_i r_3 \end{bmatrix}$$

Where: $\lambda_i = cos\alpha_i$ and $\mu_i = sen\alpha_i$

While: $r = r_1 sin\theta_i - r_2 cos\theta_i$

Likewise, if we have $[v]_{i+1} = [v_1, v_2, v_3]^T$ and we need $[v]_i$, we use the component transformation given below:

$$[v]_i = \begin{bmatrix} \cos\theta_i & -\lambda_i sin\theta_i & \mu_i \sin\theta_i \\ \sin\theta_i & \lambda_i cos\theta_i & -\mu_i cos\theta_i \\ 0 & -\mu_i & \lambda_i \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 cos\theta_i + v sin\theta_i \\ v_1 sin\theta_i + v cos\theta_i \\ v_2 \mu_i + v_3 \lambda_i \end{bmatrix}$$

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

It is now apparent that every coordinate transformation between successive frames, whether forward or backward, requires eight multiplications and four additions.

We indicate the units of multiplications and additions with M and A, respectively.

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

The angular velocity and acceleration of the *i*th link are computed recursively as follows:

$$\omega_i = \begin{cases} \omega_{i-1} + \dot{\theta_i} e_i\,, & ith - joint\ R \\ \omega_i & ith - joint\ P \end{cases}$$

$$\dot{\omega}_i = \begin{cases} \dot{\omega_{i-1}} + \omega_{i-1} \times \dot{\theta}_i e_i + \ddot{\theta}_i e_i, & ith - joint\ R \\ \dot{\omega}_{i-1} & ith - joint\ P \end{cases}$$

for i = 1, 2, . . . , n, where ω_0 and $\dot{\omega}_0$ are the angular velocity and angular acceleration of the base link.

This equations are valid in any coordinate frame.

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

In view of outwards recursive nature of the kinematic relations it is apparent that a transfer from $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$ coordinate is needed, which can be accomplished by multiplying either ei or any other vector withe (i-1) subscript by matrix Q_i^T.

Hence, the angular velocities and accelerations are computed recursively, as indicated below:

$$\omega_i = \begin{cases} Q_i^T(\omega_{i-1} + \dot{\theta}_i e_i), & ith - joint\ R \\ Q_i^T & ith - joint\ P \end{cases}$$

$$\dot{\omega}_i = \begin{cases} Q_i^T(\dot{\omega_{i-1}} + \omega_{i-1} \times \dot{\theta}_i e_i + \ddot{\theta}_i e_i & ith\ - joint\ R \\ Q_i^T \dot{\omega}_{i-1} & ith\ - joint\ R \end{cases}$$

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

If the base link is an inertial frame, then:

$$\omega_0 = 0 \qquad\qquad \dot{\omega}_0 = 0$$

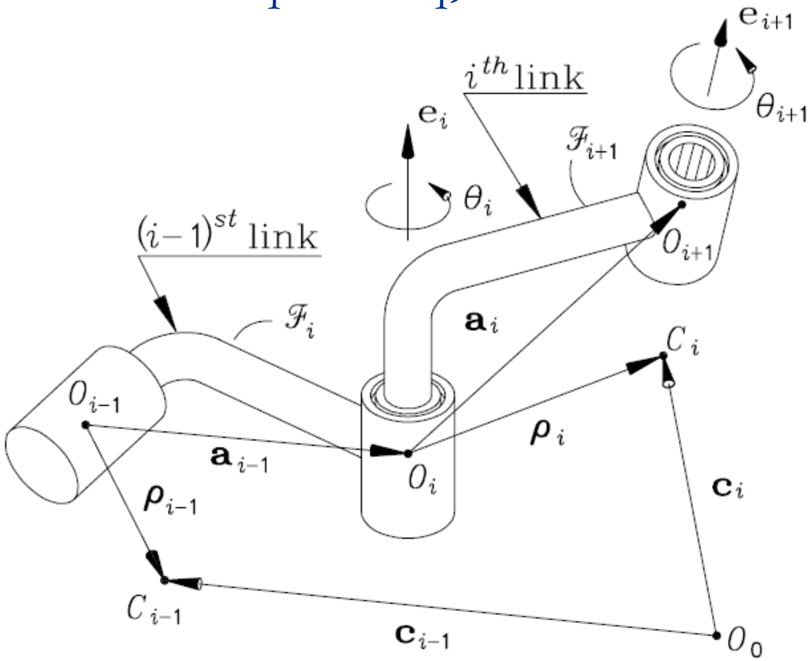
Furthermore, in order to determine the number of operations required to calculate $\dot{\omega}_i$ in $\mathcal{F}_{\mathrm{i+1}}$ when $\dot{\omega}_{i-1}$ is available in $\mathcal{F}_{\mathrm{i}}$, we note that:

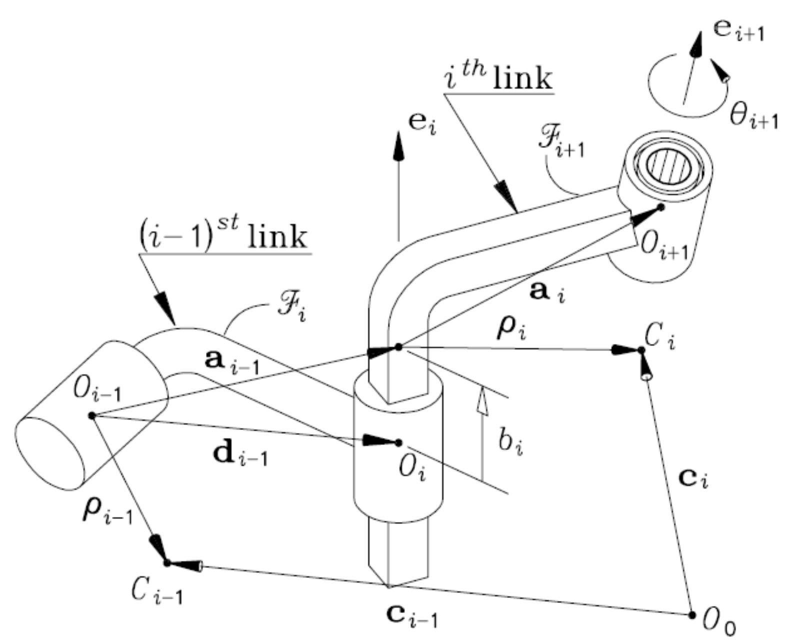
$$[\omega_{i-1} \times \dot{\theta}_i e_i]_{\mathrm{i}} = \begin{bmatrix} \dot{\theta}_i \omega_y \\ -\dot{\theta}_i \omega_x \\ 0 \end{bmatrix}$$

Where ω_x, ω_y e ω_z are the $\boldsymbol{\omega}_{i\text{-}1}$ components in $\mathcal{F}_{\mathrm{i-1}}$

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

Furthermore, let $\mathbf{c}_i$ be the position vector of C_i, the mass center of the ith link, ρ_i being the vector directed from O_i to C_i, as shown in Figures





KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

The position vectors of two successive mass centers thus observe the relationships:

- if the ith joint is R:

$\delta_{i-1} = a_{i-1} - \rho_{i-1}$ $c_i = c_{i-1} + \delta_{i-1} + \rho_i$

- if the ith joint is P:

$\delta_{i-1} = d_{i-1} - \rho_{i-1}$ $c_i = c_{i-1} + \delta_{i-1} + b_i e_i + \rho_i$

Note that in the presence of a revolute pair at the ith join, the difference $a_{i-1} - \rho_{i-1}$ is constant in $\mathcal{F}_i$. Likewise, in the presence of a prismatic pair at the same joint, the difference $d_{i-1} - \rho_{i-1}$ is constant in $\mathcal{F}_i$.

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

We derive the corresponding relations between the velocities and accelerations of the mass centers of links i - 1 and i, namely,

- if the ith joint is R:

$$\dot{c}_i = \dot{c}_{i-1} + \omega_{i-1} \times \delta_{i-1} + \omega_i \times \rho_i$$

$$\ddot{c}_i = \ddot{c}_{i-1} + \dot{\omega}_{i-1} \times \delta_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times \delta_{i-1}) + \dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i)$$

- if the ith joint is R:

$$u_i = \delta_{i-1} + \rho_i + b_i e_i \qquad v_i = \omega_i \times u_i$$

$$\dot{c}_i = \dot{c}_{i-1} + v_i + \dot{b}_i e_i$$

$$\ddot{c}_i = \ddot{c}_{i-1} + \dot{\omega}_i \times u_i + \omega_i \times (v_i + 2\dot{b}_i e_i + \ddot{b}_i e_i)$$

KINEMATICS COMPUTATIONS: OUTWARD RECURSIONS

For i = 1, 2, . . . , n, where $\dot{c}_0$ and $\ddot{c}_0$ are the velocity and acceleration of the mass center of the base link.

If the latter is an inertial frame, then

$$\omega_0 = 0 \qquad \dot{\omega}_0 = 0 \qquad \dot{c}_0 = 0 \; \ddot{c}_0 = 0$$

The expressions above are invariant. They hold in any coordinate frame, as long as all vectors involved are expressed $\mathcal{F}_i$ in that frame.

However, we have vectors in the frame, and hence a coordinate trasformation is needed. This coordinate trasformation is taken into account in the following algorithm whereby the logical variable R is true if the ith joint is R; otherwise is false

ALGORITHM – OUTWARD RECURSION

Read $\{Q_i\}_0^{n-1}, c_0, \omega_0, \dot{c_0}, \dot{\omega}_0, \ddot{c_0}, \{\rho_i\}_1^n, \{\delta_1\}_0^{n-1}$

For i=1 till n step 1 do:

Update Q_i

if R then:

$c_i \leftarrow Q_i^T(c_{i-1} + \delta_{i-1}) + \rho_i$

$\omega_i \leftarrow Q_i^T\left(\omega_{i-1} + \dot{\theta}_i e_i\right)$

$u_{i-1} \leftarrow \omega_{i-1} \times \delta_{i-1}$

$v_i \leftarrow \omega_i \times \rho_i$

$\dot{c}_i \leftarrow Q_i^T(\dot{c_{i-1}} + u_{i-1})v_i$

$\dot{\omega}_i \leftarrow Q_i^T\left(\dot{\omega_{i-1}} + \omega_{i-1} + \dot{\theta}_i e_i + \ddot{\theta}_i e_i\right)$

$\ddot{c}_i \leftarrow Q_i^T(\ddot{c_{i-1}} + \dot{\omega_{i-1}} \times \delta_{i-1} + \omega_{i-1} \times u_{i-1}) + \dot{\omega_i} \times \rho_i + \omega_i \times v_i$

ALGORITHM – OUTWARD RECURSION

Else

$u_i \leftarrow Q_i^T \delta_{i-1} + \rho_i + b_i e_i$

$c_i \leftarrow Q_i^T c_{i-1} + u_i$

$\omega_i \leftarrow Q_i^T \omega_{i-1}$

$v_i \leftarrow \omega_i \times u_i$

$w_i \leftarrow \dot{b}_i e_i$

$\dot{c}_i \leftarrow Q_i^T c_{i-1} + v_i + w_i$

$\dot{\omega}_i \leftarrow Q_i^T \dot{\omega_{i-1}}$

$\ddot{c}_i \leftarrow Q_i^T \ddot{c_{i-1}} + \dot{\omega}_i \times u_i + \omega_i \times (v_i + w_i + w_i) + \ddot{b}_i e_i$

Endif

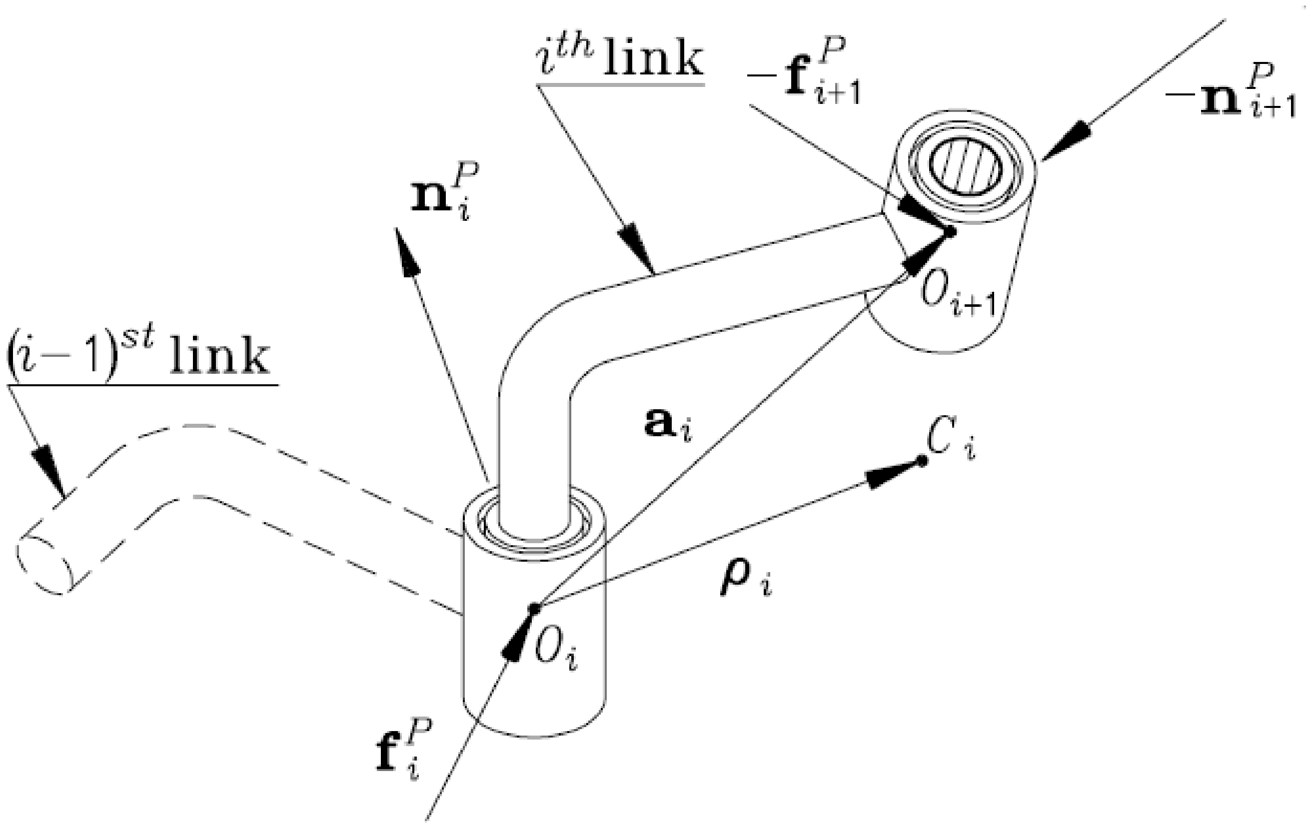
Enddo

ALGORITHM – OUTWARD RECURSION

If, moreover, we take into account that the cross product of two arbitrary vectors requires 6M and 3A, we then have the operation counts given below:

	R joint		P joint	
	M	A	M	A
Q_i	4	0	4	0
c_i	8	10	16	15
ω_i	8	5	8	4
$\dot{c}_i$	20	16	14	11
$\dot{\omega}_i$	10	7	8	4
$\ddot{c}_i$	32	28	20	19

DYNAMICS COMPUTATIONS: INWARD RECURSIONS



Free-body diagram of the ith link

DYNAMICS COMPUTATIONS: INWARD RECURSIONS

A free-body diagram of the EE appears in figure. Note that the link is acted upon by a nonworking constraint wrench, exerted though the nth pair, and a working wrench; the latter involves both active and dissipative forces and moments. Although dissipative forces and moment are difficult to model.

Since this forces and moment depend only on joint variable and joint rates, the can be calculated one the cinematic variable are known.

DYNAMICS COMPUTATIONS: INWARD RECURSIONS

Hence, the force and the moment that (i-1)st link excert on th ith link throught the ith joint produce non working constraint and active wrences.

That is for a revolute pair:

$$n_i^P = \begin{bmatrix} n_i^x \\ n_i^y \\ \tau_i \end{bmatrix} \qquad f_i^P = \begin{bmatrix} f_i^x \\ f_i^y \\ f_i^z \end{bmatrix}$$

in which n_i^P and f_i^P are the nonzero $\mathcal{F}$i-components of the nonworking constraint moment exerted by the (i −1)st link on the ith link; obviously, this moment lies in a plane perpendicular to Z_i, whereas τ_i is the active torque applied by the motor at the said joint.

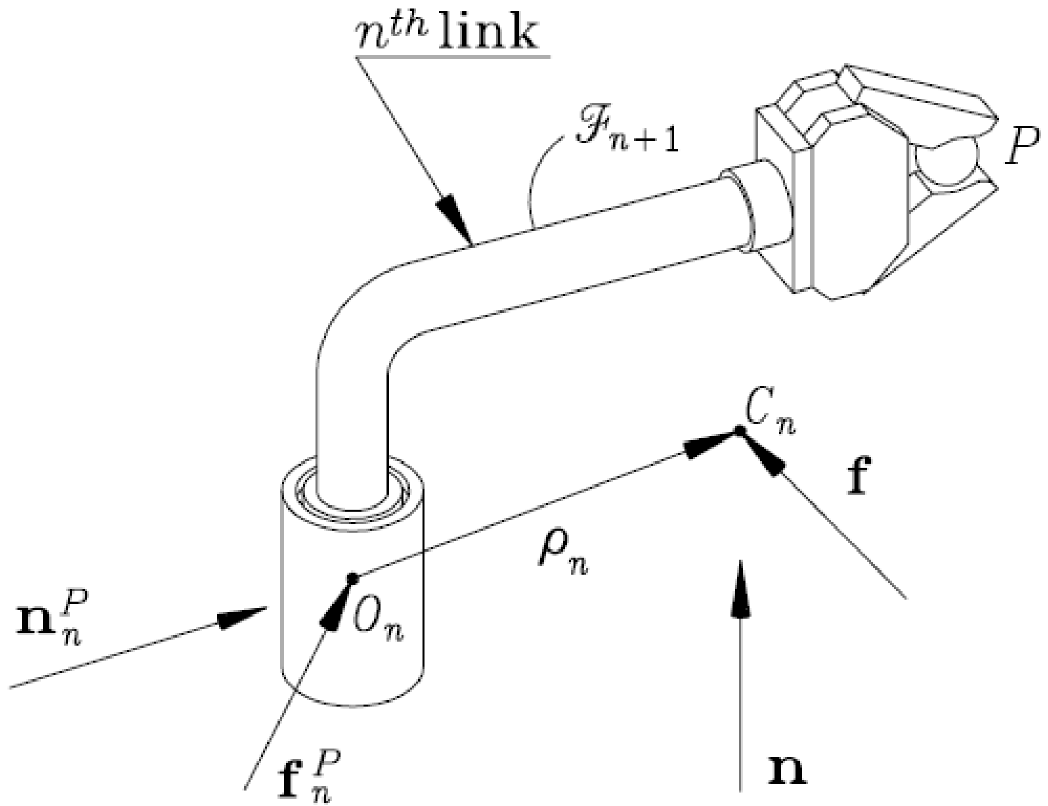
DYNAMICS COMPUTATIONS: INWARD RECURSIONS

For a prismatic pair, one has:

$$n_i^P = \begin{bmatrix} n_i^x \\ n_i^y \\ n_i^z \end{bmatrix} \qquad f_i^P = \begin{bmatrix} f_i^x \\ f_i^y \\ \tau_i \end{bmatrix}$$

Whre vector n_i^P contains only nonworking constraint torques, while τ_i is now the active force exerted by ith motor in the Z^i direction, f_i^x and f_i^y being the nonzero $\mathcal{F}$i-components of the nonworking constraint force excerted by the ith joint on the ith link, which is perpendicular to the Zi axis.

DYNAMICS COMPUTATIONS: INWARD RECURSIONS



Free-body diagram of the end-effector

DYNAMICS COMPUTATIONS: INWARD RECURSIONS

From the figure above , the Newton-Euler equations of the end-effector are:

$$f_n^P = m_n \ddot{c}_n - f$$
$$n_i^P = I_n \dot{\omega}_n + \omega_n \times I_n \omega_n - n + \rho_n \times f_n^P$$

where f and n are the external force and moment, the former being applied at the mass center of the EE. The Newton-Euler equations for the remaining links are derived based on the free-body diagram namely,

$$f_i^P = m_i \ddot{c}_i - f_{i+1}^P$$
$$n_i^P = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i + n_{i+1}^P + (a_i - \rho_i) \times f_{i+1}^P + \rho_i \times f_i^P$$

DYNAMICS COMPUTATIONS: INWARD RECURSIONS

The vectors n^P_i and f^P_i indicate the couples and active forces, denoted by t_i. In fact, if the i-th joint is rotationally has:

$$\tau_i = e_i^T n_i^P$$

if the i-th joint is prismatic then the actuator force reduces to:

$$\tau_i = e_i^T f_i^P$$

The foregoing relations are written in invariant form. In order to perform the computations involved, transformations that transfer coordinates between two successive frame are required. In taking these coordinate transformations into account, we derive the Newton-Euler algorithm from the above equation.

ALGORITHM – INWARD RECURSIONS

$f_n^P \leftarrow m_n \ddot{c}_n - f$

$n_n^P \leftarrow I_n \dot{\omega}_n + \omega_n \times I_n \omega_n - n + \rho_n \times f_n^P$

If R then

$\tau_n \leftarrow (n_n^P)_z$

Else

$\tau_n \leftarrow (f_n^P)_z$

For i=n-1 till 1 step-1 do

$\phi_{i+1} \leftarrow Q_i f_{i+1}^P$

$f_i^P \leftarrow m_i \ddot{c}_i - \phi_i$

$n_i^P \leftarrow I_i \dot{\omega}_i + \omega_i \times I_i \omega_i + \rho_i \times f_i^P + Q_i n_{i-1}^P + (a_i - \rho_i) \times Q_i f_{i+1}^P$

If R then

$\tau_i \leftarrow \left(n_i^P\right)_z$

else

$\tau_i \leftarrow \left(f_i^P\right)_z$

enddo

COMPLEXITY OF DYNAMICS COMPUTATIONS

A summary of all the calculations is shown in Table:

Row #	M	A
2	30	27
6	3(n-1)	3(n-1)
Total	55n-22	44n-14

The total number of additions and moltiplications For M_d can be calculated with the following formulas:

$$M_d = 55n - 22 \qquad A_d = 44n - 14$$