Here is the G6 function:

$$G6a(\omega) := \left(\frac{1}{j\cdot\omega}\right)^{-3} \qquad G6a(1) = -j \qquad \arg(G6a(1)) = -90\,\text{deg}$$

or, the same thing:

$$G6(\omega) := (j\cdot\omega)^3 \qquad G6(1) = -j \qquad \arg(G6(1)) = -90\,\text{deg}$$

and then look at

$$G7(\omega) := \frac{1}{(j\cdot\omega)} \qquad G7(1) = -j \qquad \arg(G7(1)) = -90\,\text{deg}$$

Since the values G6(1) and G7(1) are identical, *there is no way any function can distinguish the two based on the end numeric evaluation.* I believe the quest to define an argument function based solely on the function value is doomed to fail. This holds true for any phase unwrapping function as well, since the starting points will be the same, although the desire is to have them differ.

The desired result requires "looking under the hood" so to speak - it must depend on the structure of the function and not solely on the evaluated result.

If the transfer function is factored into linear terms, and the phase calculated as the sum of phases of each term:

$$G6mag(\omega) := \prod_{k=0}^{2} |j\cdot\omega| \qquad G6mag(1) = 1$$

$$G6arg(\omega) := \sum_{k=0}^{2} \arg(j\cdot\omega) \qquad G6arg(1) = 270\,\text{deg}$$

(compare with previous result, above)

and for G7:

$$G7arg(\omega) := -\sum_{k=0}^{0} \arg(j\cdot\omega) \qquad G7arg(1) = -90\,\text{deg}$$

One possible path is to factor the transfer function into linear terms, and evaluating the phase term by term. In this case, any factor has the form $s + a$, and its phase will be in the range (0,+90) deg. summing the phase contributions of each term (+ for numerator, - for denominator) should give the desired result.

If the transfer function has the factored form

$$H(s) = C\frac{\prod_{k=0}^{N}\left(s + z_k\right)}{\prod_{k=0}^{M}\left(s + p_k\right)}$$

then the phase will be given by

$$\arg H(s) = \arg(C) + \sum_{k=0}^{N} \arg\left(s + z_k\right) - \sum_{k=0}^{M} \arg\left(s + p_k\right)$$

Since none of the terms can have a phase outside the normal (-180,+180) deg range, there will be no wrapping within any term, and consequently no wrapping of the sum. I think this is what Andrew is looking for. Implementation is left as an exercise (for somebody other than me).

CAVEAT: none of this can be done based on the value of the function alone - it requires knowledge of how the function is constructed.